

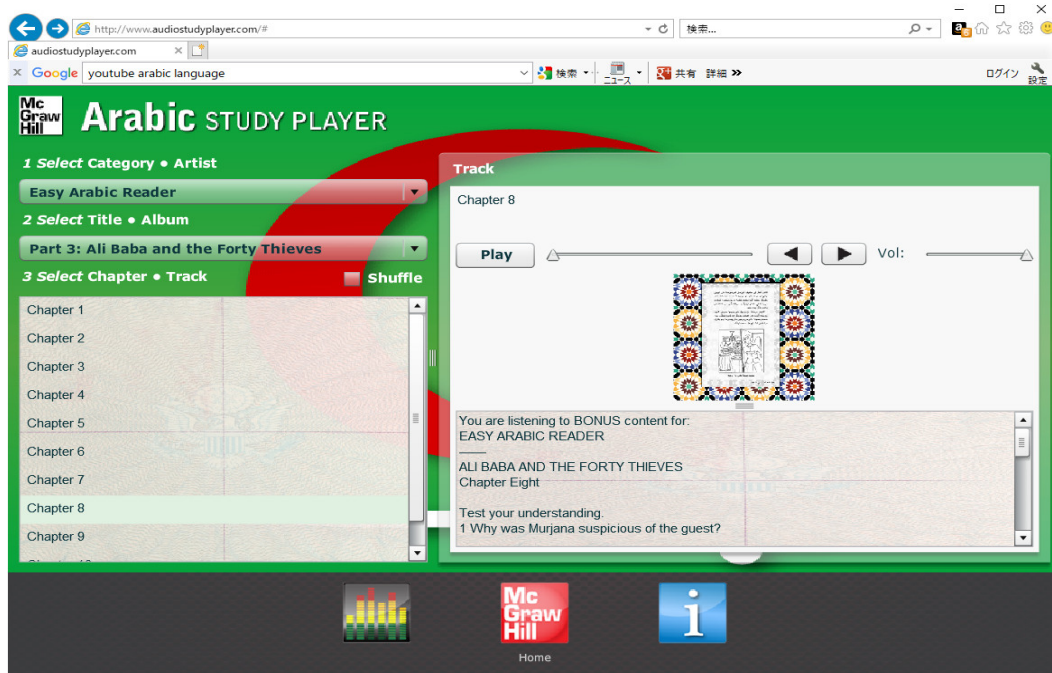
音声データの有効利用と $\text{\LaTeX}$ を使って囲碁・将棋の局 面のカードを効率的に作るためのソフト開発及びトラン プのカード画像作成の試み

文責 : 高知大学名誉教授 中村 治

プログラミングは、データ構造とプログラミング言語を使って、アルゴリズムをコンピュータに理解させることですが、実はコンピュータは道具を作る道具であり、道具を作る道具を作る道具でもあり、道具を作る道具を作る道具を作る道具でもあり、…、で、プログラミングはコンピュータで自分の必要な道具を作ることもできます。プログラミングができるようになると「自分の欲しい道具」を自分で自由に作ることが出来ます。その時、すべてを自分で作るのではなく、他人の作った道具を部品として道具を作っても良いし、補完する部分だけを作っても良いです。こうすれば楽ができます。コンピュータはこのようにしてすごいスピードでその能力を伸ばしています。この pdf では、素晴らしいフリーソフトである `ffmpeg` と $\text{\TeX}$ を最小限度の労力で有効活用する例を示します。 $\text{\TeX}$ はプログラミング可能な文書作成ソフトであり、その能力は素晴らしく、最初はとっつき難いかもしれませんが、数学者は全員使っていて、使いだすと手放せなくなります。私は Word は十年以上使っていません。この文章も $\text{\TeX}$ で作っています。

この pdf ファイルには、音声データを加工して再生する Python のプログラム、棋譜編集ソフト Kifu for Windows V7 のデータから局面を取り出し、 $\text{\LaTeX}$ で局面を表示する C# のプログラム、チェスの局面を作り、 $\text{\LaTeX}$ で局面を表示する C# のプログラム、将棋の局面を作り、 $\text{\LaTeX}$ で局面を表示する C# のプログラム、囲碁の局面を作り、 $\text{\LaTeX}$ で局面を表示する C# のプログラム、カードゲームの画像を `metapost`、 $\text{\LaTeX}$ 、`GIMP` で作る方法について解説しています。それぞれ独立していますから、興味ある部分だけ、お読みください。

例えば、ロシア語の勉強をしようと思うとロシア人の吹き込んだ音声の CD 付きのテキストを普通購入します。この CD を、例えば、Windows Media Player で再生して何回も聞きます。あるいは、NHK のラジオロシア語講座をヴォイスレコーダやラジオサーバーで録音して、ヘッドホンで何度も聞きます。最近は、CD 付属の書籍だけでなく、mp3 データをダウンロードする方式やオンラインで音声を提供する出版社もあります。



これで自分は満足しているという人は何もする必要がありません。Windows Media Player にトラック 5 を 10 回、トラック 8 を 20 回、トラック 14 を 10 回、そしてこれらを 5 回繰り返して聞くような機能があればいいなとか、ヴォイスレコーダやラジオサーバーで録音した音声データをパソコンのフォルダーで整理して、パソコンのスピーカーで何度も聞きたい（ヘッドホンで聞くと難聴になります）、或いは、よく聞き取れない文章を何度も繰り返し聞きたい（聞き取れない文を何回も繰り返して聞きたい時には、例えば、OLYMPUS のヴォイスレコーダー Voice-Trex DS-750 で録音すれば、一部分を繰り返し再生することが出来ますが、このような機能を持った器機は高価ですし、毎回操作しなければいけないのでめんどくさいです。）と思ったとき、プログラミングが役に立ちます。更に、このような色々な音源を統一的に扱えるようにすれば便利です。、Python でプログラミングすれば、これらの要求をかなえてくれる「欲しい道具」を自分で作れます。

インターネットで検索すると .wav ファイルと .mp3 ファイルが Python を使えば、pydub をインストールする必要（更に、pyaudio と ffmpeg もインストールする必要があります。）がありますが、Python で、mp3 ファイルを再生するには、

```
from pydub import AudioSegment
from pydub.playback import play

sound = AudioSegment.from_mp3("chap1_1.mp3")
play(sound)
```

と言う Python のプログラムと同じフォルダーに、chap1_1.mp3 と ffmpeg.exe,

ffplay.exe, ffprobe.exe を置いて実行すると chap1_1.mp3 を再生してくれ、wav ファイルを再生するには、

```
from pydub import AudioSegment
from pydub.playback import play

sound = AudioSegment.from_file("07_01_01.wav", format="wav")
play(sound)
```

とすれば、07_01_01.wav を再生してくれることが分かります。

注意:これは Python3.6 では可能ですが、Python3.7 では、何らかの原因で現時点で不可能です。play(sound) だけが実行できません。Anaconda や Miniconda は Python3.7 になっていますが、丸一日試行錯誤してみましたが、上手くいきませんでした。仕方なく削除していた古い Anaconda3 のフォルダーを復元することで、復旧しました。語学の勉強をしたいので、大量に購入している CD を有効活用するには、このプログラミングができることが、最新の Python3.7 を使うことよりも優先されます。所が、強化学習の教科書を読んでいる時、tensorflow が Python3.7 には対応していないとのことで、Anaconda や Miniconda で Python3.7 を導入しても、仮想環境で、Python3.6 を導入すれば上手くと書いてありました。

コマンドプロンプトで

```
$ conda create --name tensorflow python=3.6

$ activate tensorflow

(tensorflow)$ python -m pip install --upgrade pip

(tensorflow)$ pip install --upgrade tensorflow==1.5.0

(tensorflow)$ python -c 'import tensorflow as tf; print(tf.__version__)'

1.5.0

(tensorflow)$ deactivate

$
```

のようにして、tensorflow を導入しました。パソコンが最新、最強のものではないので、古い version 1.5.0 しか導入できませんでしたが、このような方法もあります。mp3 を再生するには、pydub と pyaudio をインストールする必要

がありますが、pyaudio のインストールが Python3.7 では上手くいきません。Anaconda や Miniconda の仮想環境で、Python3.6 を導入すれば、上手くいきます。コマンドプロンプトで

```
$ conda create --name cd_sound python=3.6
```

```
$ activate cd_sound
```

```
(cd_sound)$ pip install pydub
```

```
(cd_sound)$ pip install pyaudio
```

```
(cd_sound)$ deactivate
```

```
$
```

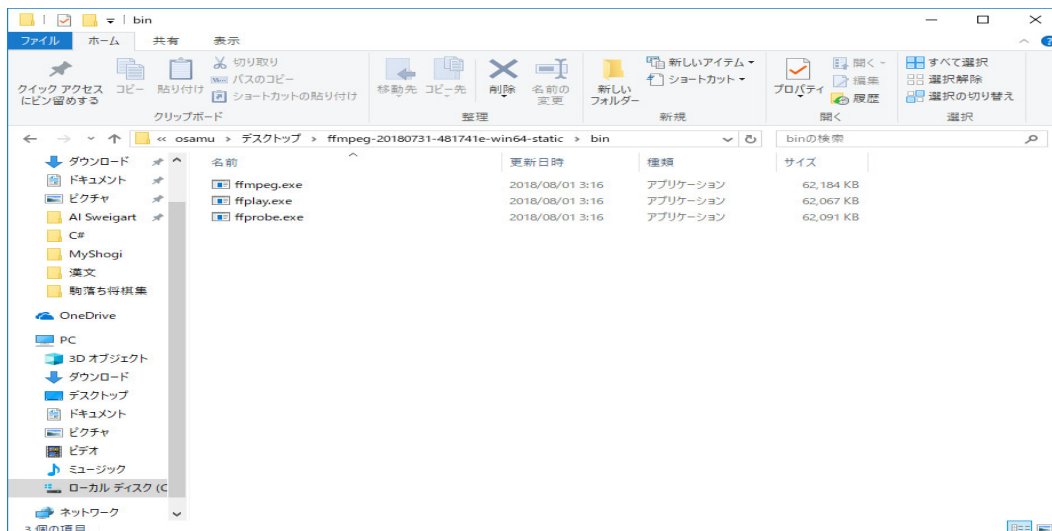
でいいです。後は

```
$ activate cd_sound
```

```
(cd_sound)$ idle
```

として、プログラミングすれば良いです。私のような素人は、プログラミング以前のこのような環境設定でつまずき、数日間悩むことになります。何がまずいのか、エラーの表示を注意してよく読むことが必要です。そして、誰か同じことでつまずいて、インターネットで質問していないか、誰かが解決方法を書き込んでいないか調べます。

ffmpeg.exe, ffplay.exe, ffprobe.exe をゲットする為には ffmpeg をインターネットからダウンロードします。ffmpeg-20180731-481741e-win64-static.zip のような名前のファイルをダウンロードし、解凍すると



ffmpeg.exe, ffplay.exe, ffprobe.exe というファイルを手に入れることができます。pip でインストールする方法 (?) もあったと記憶しています。インターネットで検索してください。

このソフトを使うと色々なことができます。

```
ffmpeg -i exapmle.wma exapmle.mp3
```

で、.wma ファイルを .mp3 ファイルに変換できます。

```
ffmpeg -i input_file.mp3
```

このようにコマンドを打つと、ファイルの情報が分かる。

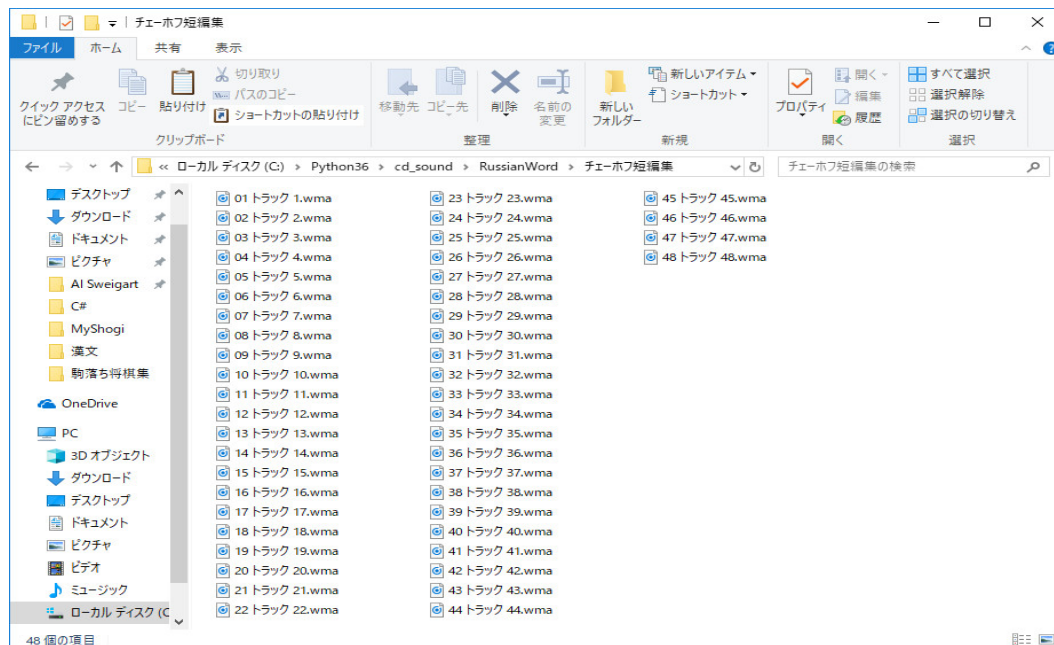
```
ffmpeg -i input_file.mp3 -t 3600 output_file.mp3
```

input_file.mp3 の、0 秒～3600 秒までを抜き出して output_file.mp3 というファイルに出力する。

```
ffmpeg -i input_file.mp3 -ss 60 -t 3540 output_file.mp3
```

input_file.mp3 の、60 秒～3600 秒までを抜き出して output_file.mp3 というファイルに出力する。他の使いかたはインターネットで調べてください。

ここで興味があるのは、.wma ファイルを .mp3 ファイルに変換できることです。と言うのは、Windows Media Player で取り込んだ CD のデータは「ミュージック」のフォルダーに



のような形で、トラックごとの.wma ファイルとして取り込まれています。Windows Media Player で再生できなくなるといかないので、この音声データを適当なフォルダーにコピーします。プログラミングが出来るとこのデータを有効活用できます。

上で述べたように、Python で、mp3 ファイルを再生するには、pydub をインストールする必要がありますが、

```
from pydub import AudioSegment
from pydub.playback import play

for i in range(3):
    sound = AudioSegment.from_mp3("chap1_1.mp3")
    play(sound)
```

と言う Python のプログラムと同じフォルダーに、chap1_1.mp3 と ffmpeg.exe, ffplay.exe, ffprobe.exe を置いて実行すると chap1_1.mp3 を 3 回連続して再生してくれます。ffmpeg.exe, ffplay.exe, ffprobe.exe のある場所に Path が通っていればいいです。

.wav ファイルを再生するには、

```
from pydub import AudioSegment
from pydub.playback import play

sound = AudioSegment.from_file("07_01_01.wav", format="wav")
play(sound)
```

とすれば良いです。更に、.wav ファイルを再生するには、

```

import pyaudio
import wave

CHUNK = 1024
filename="07_01_01.wav"

wf = wave.open(filename, 'rb')

p = pyaudio.PyAudio()

stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
                channels=wf.getnchannels(),
                rate=wf.getframerate(),
                output=True)

"""
    format   : ストリームを読み書きするときのデータ型
    channels: ステレオかモノラルかの選択 1でモノラル 2でステレオ
    rate     : サンプル周波数
    output   : 出力モード
"""

# 1024 個読み取り
data = wf.readframes(CHUNK)

while data != '':
    stream.write(data)          # ストリームへの書き込み (バイナリ)
    data = wf.readframes(CHUNK) # ファイルから 1024 個*2 個の

stream.stop_stream()
stream.close()

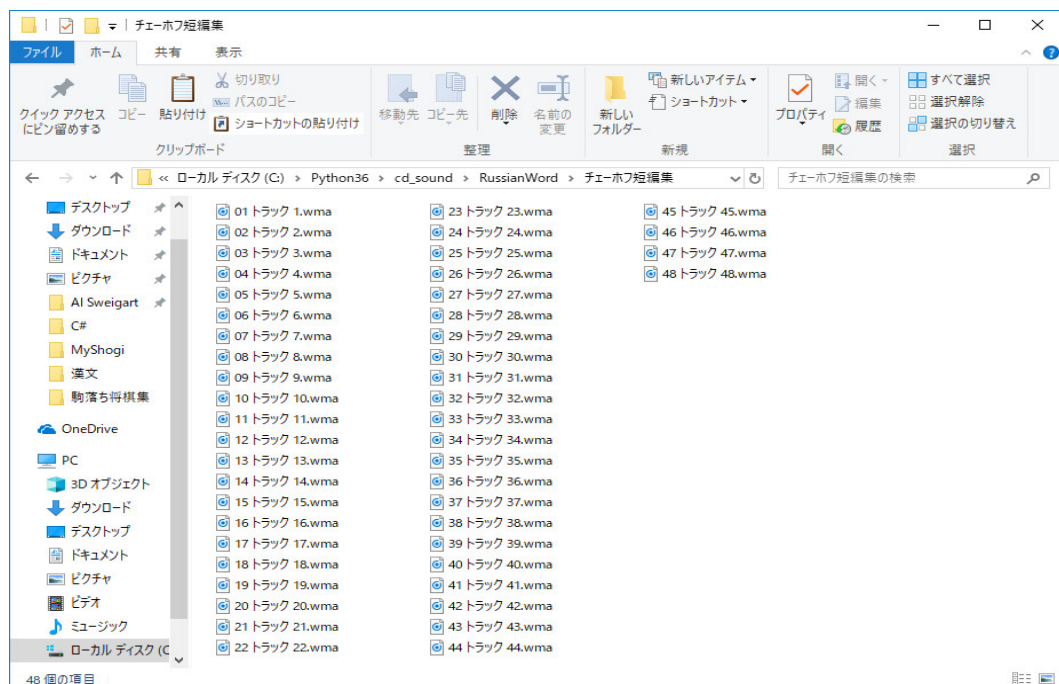
p.terminate()

```

のような Python のプログラムでも可能ですが、上のプログラムに比べ複雑です。
 .wav ファイルは.mp3 ファイルに比べ、10 倍ぐらいメモリーを必要とするので、
 .mp3 ファイルに変換して、再生する方が良いです。

```
ffmpeg -i ''01 トラック 1.wma'' 01.mp3
```

で、ファイル 01 トラック 1.wma をファイル 01.mp3 に変換できます。ファイル名に空白があると”” で囲む必要があります。これを繰り返せば、すべてのファイルを .mp3 ファイルに変換できます。しかし、手でこれをするのは大変です。Python のプログラムでこれを実行する方法をインターネットで調べます。Python で、コマンドプロンプトの命令を実行する方法です。これは簡単に見つかります。一遍に、.mp3 ファイルに変換しても良いですが、まずは練習のために、ファイル名を簡単なものに変更してみます。エクスプローラーで名前の変更が出来ますが、コマンドプロンプトの命令 ren を使ってもできます。これを Python で実行するプログラムをまず考えます。



のようなファイルは名前が長いので短くします。手動で名前を変えることが出来ますが、今の場合、ファイル名にパターンがあるので、Python のプログラムを作って、プログラムに自動的に名前を変えてもらうのが楽です。この例の場合は

```
import subprocess

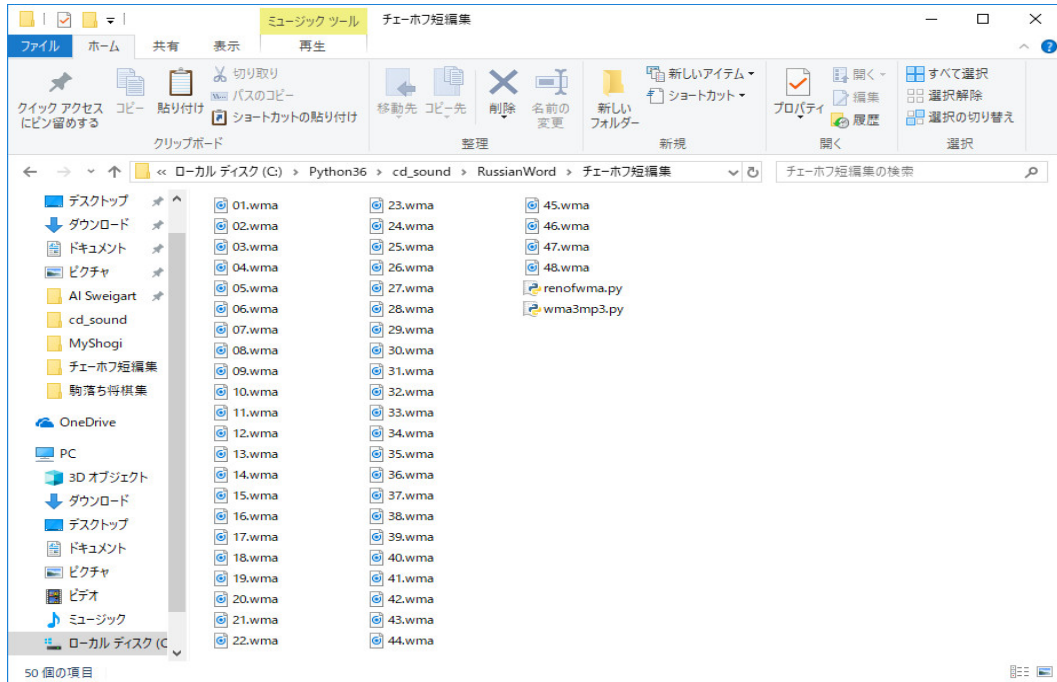
for k in range(1, 10):
    cmd = 'ren "0{} トラック {}".wma 0{}.wma'.format(k, k, k)
    returncode = subprocess.call(cmd, shell=True)
```

と

```
import subprocess
```

```
for k in range(10, 49):
    cmd = 'ren "{} トラック {}".wma {}.wma'.format(k, k, k)
    returncode = subprocess.call(cmd, shell=True)
```

とを実行すれば、



となります。すべてのファイルの名前が変更されました。勿論、

```
import subprocess

for k in range(1, 49):
    if k<10:
        cmd = 'ren "0{} トラック {}".wma 0{}.wma'.format(k, k, k)
    else:
        cmd = 'ren "{} トラック {}".wma {}.wma'.format(k, k, k)
    returncode = subprocess.call(cmd, shell=True)
```

でも良いです。次に、これらの .wma ファイルを .mp3 ファイルに変えます。

```
import subprocess

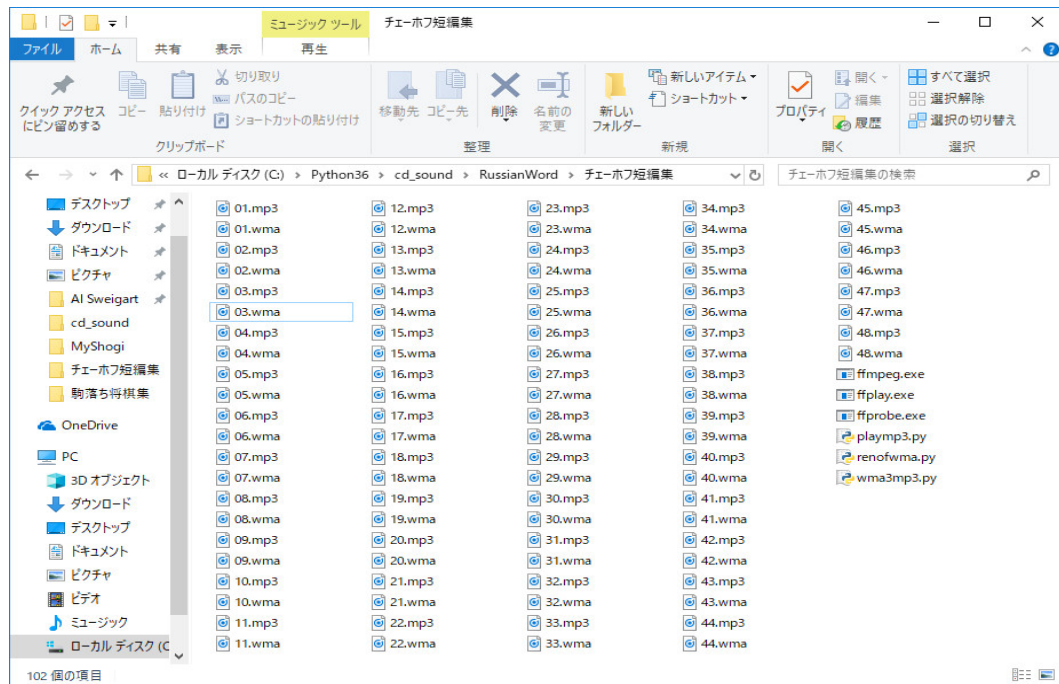
for k in range(1, 10):
    cmd = 'ffmpeg -i 0{}.wma 0{}.mp3'.format(k, k)
    returncode = subprocess.call(cmd)
```

と

```
import subprocess
```

```
for k in range(10, 49):  
    cmd = 'ffmpeg -i {}.wma {}.mp3'.format(k, k)  
    returncode = subprocess.call(cmd)
```

を実行すると



となります。すべてのファイルが .mp3 ファイルに変更されました。勿論

```
import subprocess
```

```
for k in range(1, 49):  
    if k<10:  
        cmd = 'ffmpeg -i 0{}.wma 0{}.mp3'.format(k, k)  
    else:  
        cmd = 'ffmpeg -i {}.wma {}.mp3'.format(k, k)  
    returncode = subprocess.call(cmd)
```

でも良いです。

更に、途中で、短いファイル名にするというステップを置かず、直接 .mp3 ファイルを作るプログラムをすることも可能ですが、初めてプログラミングする場合は、順番に確かめながら、段階を踏んで、プログラミングするのがいいと思います。

このフォルダには ffmpeg.exe, ffplay.exe, ffprobe.exe もコピーしています。もう .wav ファイルは必要ありませんので、同様に Python のプログラムで消すことも出来ます。例えば、

```
import subprocess

for k in range(1, 36):
    if k<10:
        cmd = 'del 0{}.wma'.format(k)
    else:
        cmd = 'del {}.wma'.format(k)

    returncode = subprocess.call(cmd, shell=True)
```

で良いです。このように、プログラミングと言うのは、自分が手で実行できるが、実際にやるのはめんどくさいことを代わりに計算機にやってもらうことです。やり方が分からないことをプログラミングすることは出来ません。計算機を使っても、本質的に、プログラミングしている人を超えることは出来ません。

例えば

```
from pydub import AudioSegment
from pydub.playback import play

n=3
for k in [3, 6, 12, 10, 5]:
    if k < 10:
        fname = '0{}.mp3'.format(k)
    else:
        fname = '{}.mp3'.format(k)
    sound = AudioSegment.from_mp3(fname)
    for i in range(n):
        play(sound)
```

このような Python のプログラムを書くことで、mp3 ファイルを好きな順序で、好きな回数再生することが出来ます。


```
*first_program2.py - C:\Python36\cd_sound\ArabicWord\アラビア語の入門1 本...
File Edit Format Run Options Window Help
from pydub import AudioSegment
from pydub.playback import play

n=3
for i in range(10):
    print("No.", i+1)
    for k in range(n):
        print("01.mp3")
        sound = AudioSegment.from_mp3("01.mp3")
        play(sound)
    ##
    for k in range(n):
        print("02.mp3")
        sound = AudioSegment.from_mp3("02.mp3")
        play(sound)
    ##
    for k in range(n):
        print("03.mp3")
        sound = AudioSegment.from_mp3("03.mp3")
        play(sound)
    ##
    for k in range(n):
        print("04.mp3")
        sound = AudioSegment.from_mp3("04.mp3")
        play(sound)
    ##
    for k in range(n):
        print("05.mp3")
        sound = AudioSegment.from_mp3("05.mp3")
        play(sound)
    |
    ##### Unit 2
    ##
    for k in range(n):
        print("06.mp3")
        sound = AudioSegment.from_mp3("06.mp3")
        play(sound)
    ##
    ##### 数字
    for k in range(n*2):
        print("07.mp3")
        sound = AudioSegment.from_mp3("07.mp3")
        play(sound)
```

Ln: 25 Col: 0

のようなプログラムにすれば、必要ないトラックをコメントアウトすることが出来ます。

更に、

```
second_russianA.py - C:\Python36\cd_sound\RussianWord\NHKロシア語単語...
File Edit Format Run Options Window Help
from pydub import AudioSegment
from pydub.playback import play

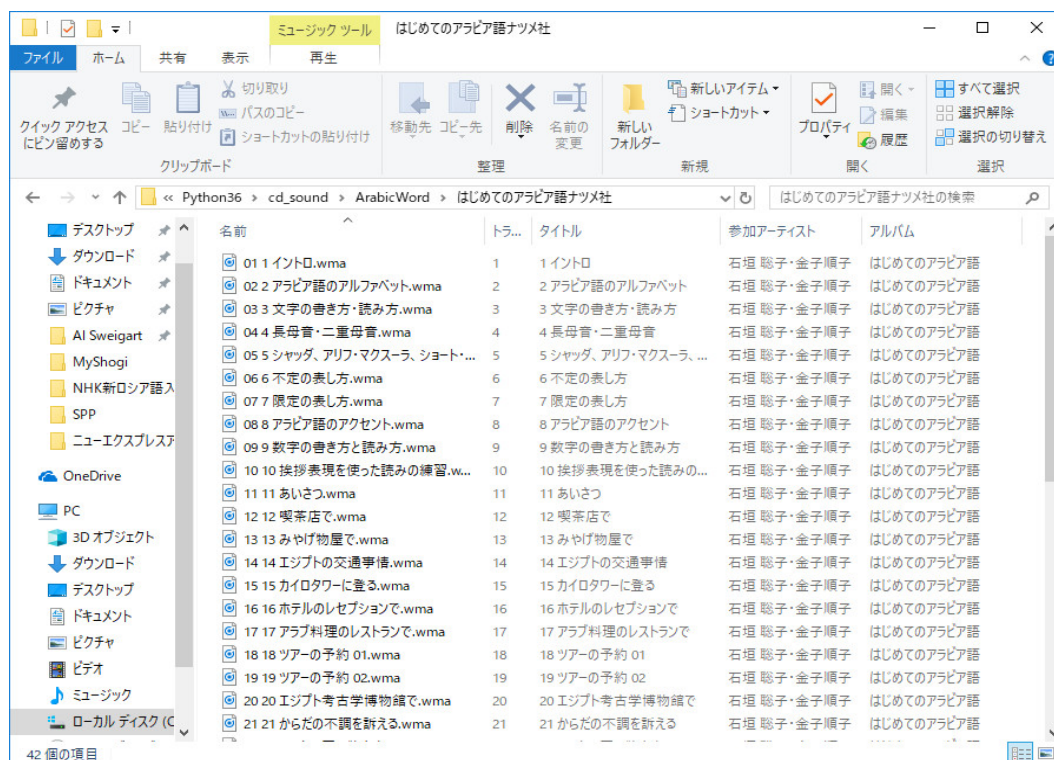
n = 3
for i in range(3):
    print("01.mp3")
    sound = AudioSegment.from_mp3("01.mp3")
    play(sound)
    print("02.mp3")
    print("Э д р а в с т в у й т ы ! : こんにちは!")
    print("Д о б р о е у т р о ! : おはよう!")
    print("Д о б р ы й д е н ь ! : こんにちは!")
    print("Д о б р ы й в е ч е р ! : こんにちは!")
    print("П р и в е т ! : やあ!")
    print("О ч е н ь п р и я т н о . : 始めまして")
    print("К а к у в а с д е л а ? : 調子はどうですか?")
    print("Х о р о ш о , с п а с и б о . А у в а с ? : 良いです、有難う。あなた")
    print("Н о р м а л ь н о . : 順調です")
    print("Д о с в и д а н и я ! : さようなら!")
    print("П о к а ! : じゃあね!")
    print("П о к о й н о й н о ч и ! : お休みなさい!")
    print("Д а . : はい。")
    print("Н е т . : いいえ。")
    print("К о н е ч н о . : 勿論。")
    for k in range(n):
        print("02.mp3")
        sound = AudioSegment.from_mp3("02.mp3")
        play(sound)
    print("03.mp3")
    print("М о ж е т б ы т ь . : そうかもしれません。")
    print("Б о л ь ш о е С п а с и б о ! : 本当に有難う!")
    print("Н е з а ч т о ! : どういたしまして!")
    print("П о ж а л у й с т а ! : どういたしまして!")
    print("И з в и н и т е . : ごめんなさい。")
    print("И ч е г о ! : 何でもありませんよ!")
    print("П р о с т и т е . : (話しかけて)すみません。")
    print("Б у д ь т е д о б р ы ! : (話しかけて)お願いします!")
    print("А л л о ! : (電話で)もしもし!")
    print("П о ж а л у й с т а . : (何かを勧めて)どうぞ。")
```

Ln: 1 Col: 0

のようなプログラムにすれば



のように、関連する情報を音声と同時に表示することが出来ます。
 ファイル名にご丁寧にファイルの内容が表示されていて、単純な名前でない場合

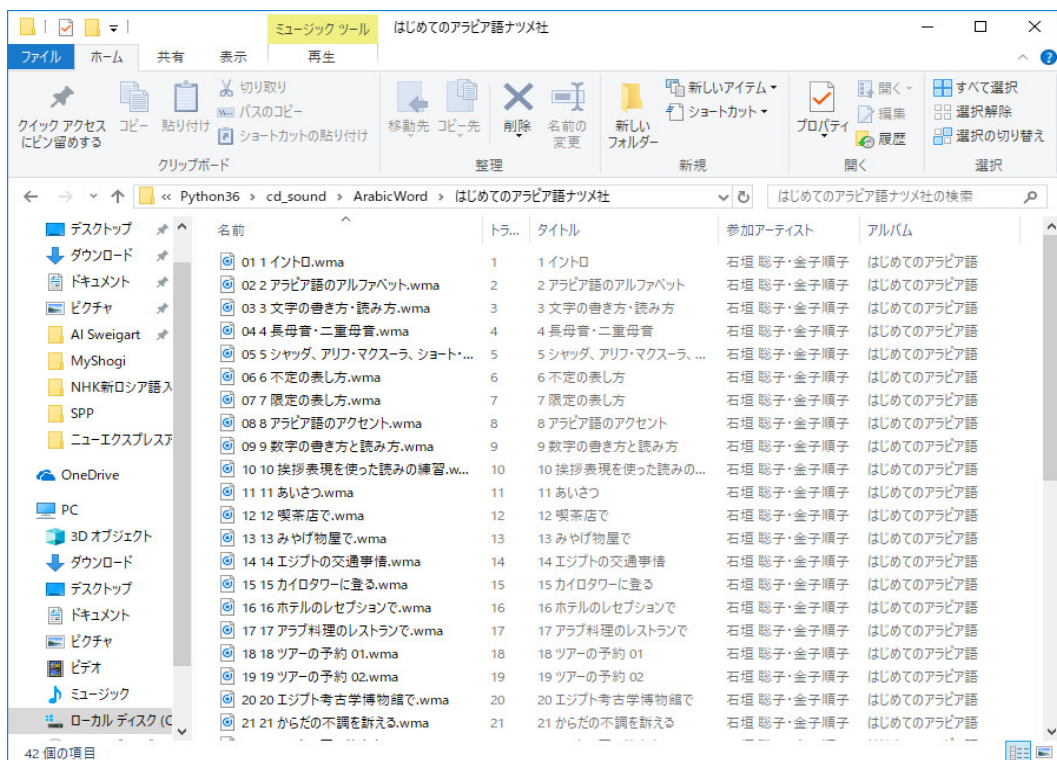


は、泣く泣く手作業で名前の変換をします。何回かやってみて、こんなことやっていられないと思いだせば、手作業でやっていることを解析し、Python のプロ

グラミングでやる方法を探ります。まず、エクスプローラーでファイル名を表示し、.wma ファイルを順番に、名前を変えていきます。従って、コマンドプロンプトの命令で、これをするには、ファイルのあるフォルダーに cd 命令で移動し、dir 命令で、ファイルを表示し、ren 命令で一つ一つ名前を変えれば良いです。これを Python で一気にやる方法をインターネットで検索すれば良いです。インターネットで気長に検索しても、全部を一気にやってくれる欲しいプログラムはまず見つかることはないので、各ステップを実行する方法をインターネットで検索します。そして、それらを組み合わせて、欲しいプログラムをでっちあげます。

カレントディレクトリにあるすべてのファイル名を得る方法やファイル名から拡張子を取り出す方法や Python における正規表現の使いかたをインターネットで調べれば、

例えば、



の場合は、

```
import os
import re
import subprocess

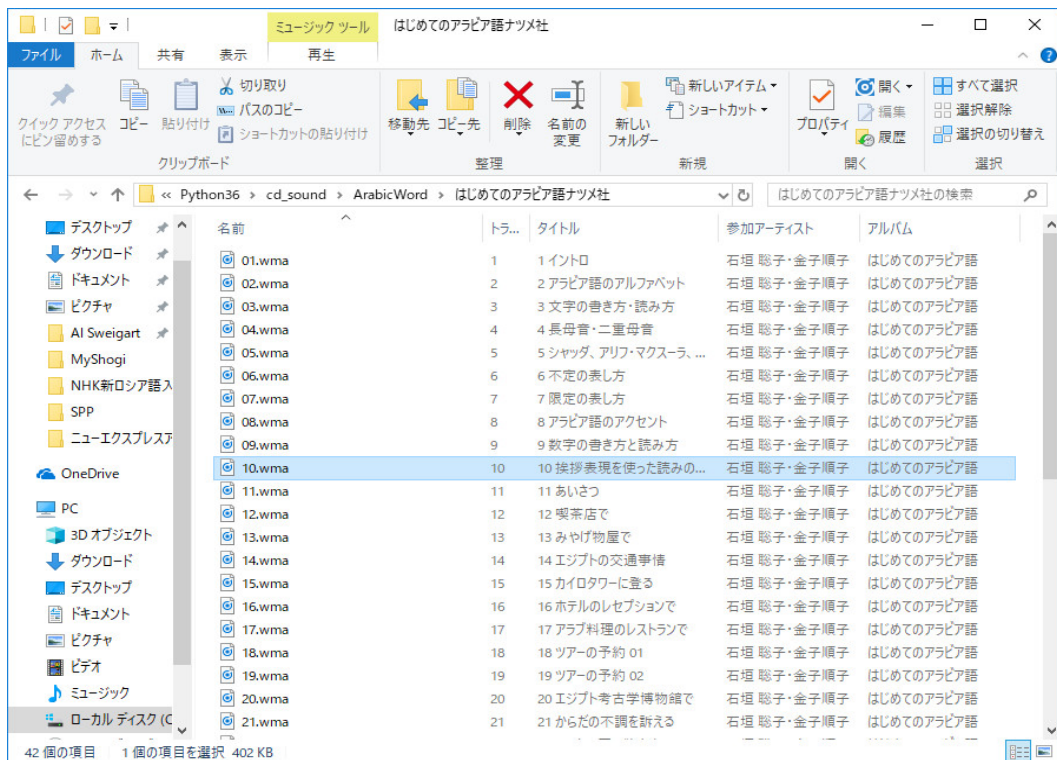
for file in os.listdir():
    base, ext = os.path.splitext(file)
    if ext == '.wma':
```

```

pattern1 = r'^(\d)+(\s)'
result1 = re.match(pattern1, base)
if result1 != None:
    words = base.split(" ")
    cmd = 'ren "{}" {}.wma'.format(file, words[0])
    returncode = subprocess.call(cmd, shell=True)

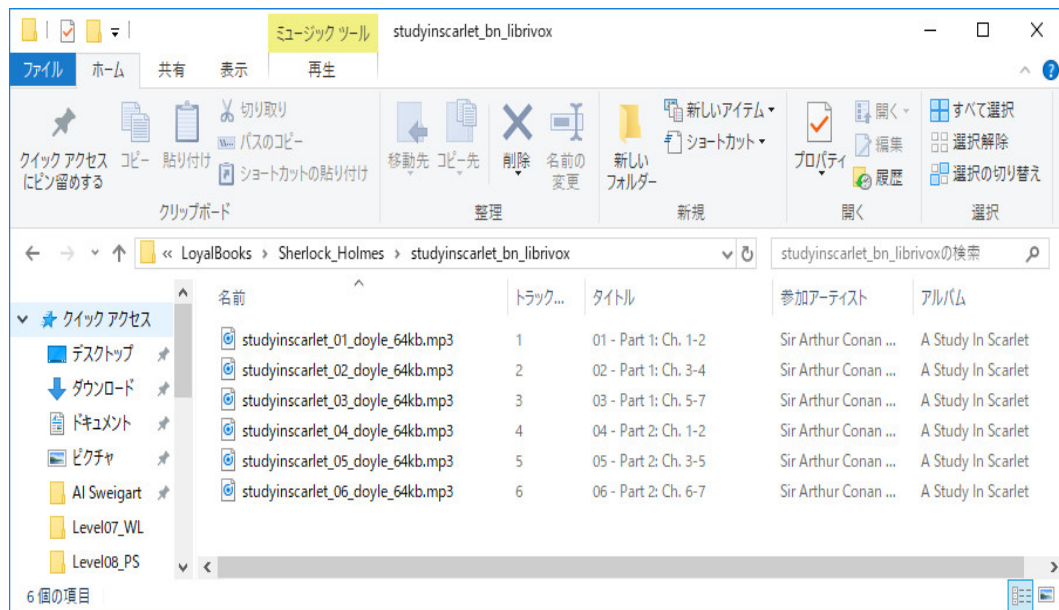
```

というプログラムを作ればいいことが分かります。これを実行すると



と変換してくれます。Python で何をしたいか、明確にしたいことが分かっているれば、必要な情報をインターネットで調べて、このようにプログラムを作ることが出来ます。有難いことです。何度も書きますが、自分が手作業で実行する時、何をやっているか分析し、その通りのことを計算機にやらしているだけです。計算機は同じことの繰り返しを苦にせず、プログラミング言語には繰り返しを表現する色々な仕組みが準備されているので、計算機を使えば楽ができます。あくまでも、自分が時間を掛ければ実行できることを計算機に肩代わりさせているだけで、本質的な意味で、計算機を使っても、自分の能力を超えることは出来ません。勿論、自分ができないことでも、ffmpeg のように、誰かがプログラミングしてくれていれば、その中身を理解できなくとも、それをブラックボックスとして、そのまま使わせてもらって、自分の欲しいプログラムを作ることが出来ます。プログラミングには、このような特性があり、急速にその利用範囲を拡大しています。

LoyalBooks から音声データをゲットした時は

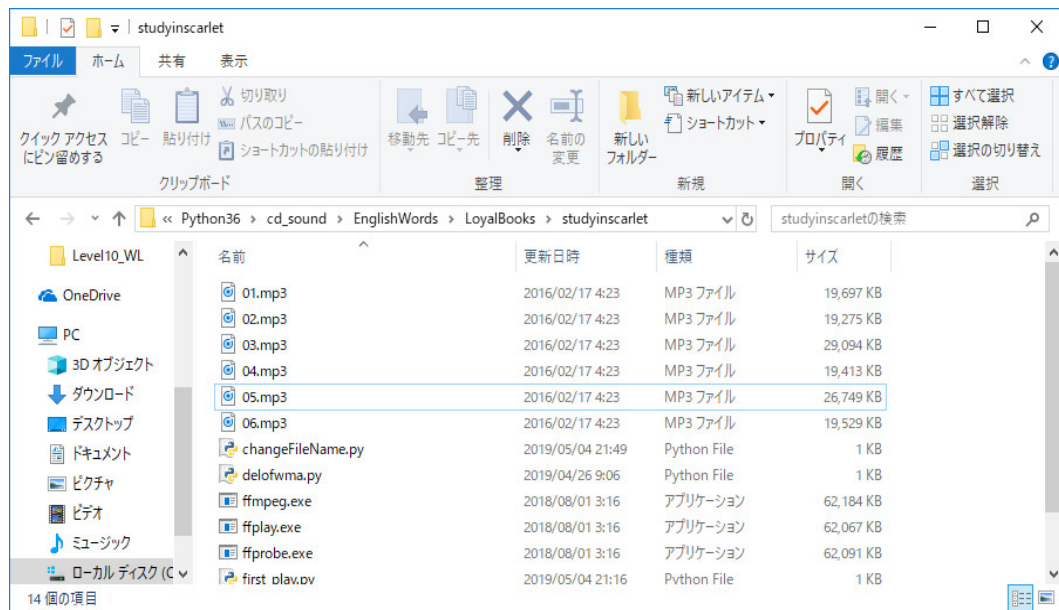


のようなファイル名になっています。既に、.mp3 ファイルになっています。この
ようなときには

```
import os
import subprocess

for file in os.listdir():
    base, ext = os.path.splitext(file)
    if ext == '.mp3':
        words = base.split("_")
        cmd = 'ren "{}" {}'.format(file, words[1])
        print(cmd)
        returncode = subprocess.call(cmd, shell=True)
```

で、



と変換してくれます。この場合は

```
import os
import re
import subprocess
```

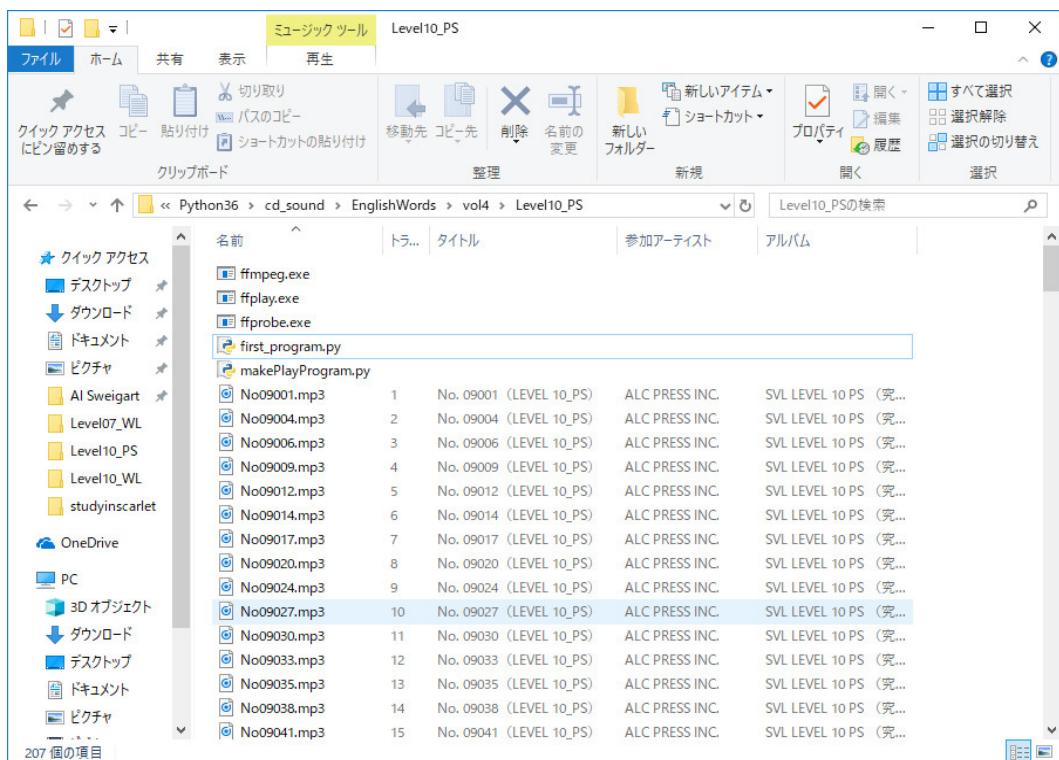
```
n=1
```

```
for file in os.listdir():
    base, ext = os.path.splitext(file)
    if ext == '.mp3':
        cmd = 'ren "{}" 0{}.mp3'.format(file, n)
        print(cmd)
        returncode = subprocess.call(cmd, shell=True)
        n += 1
```

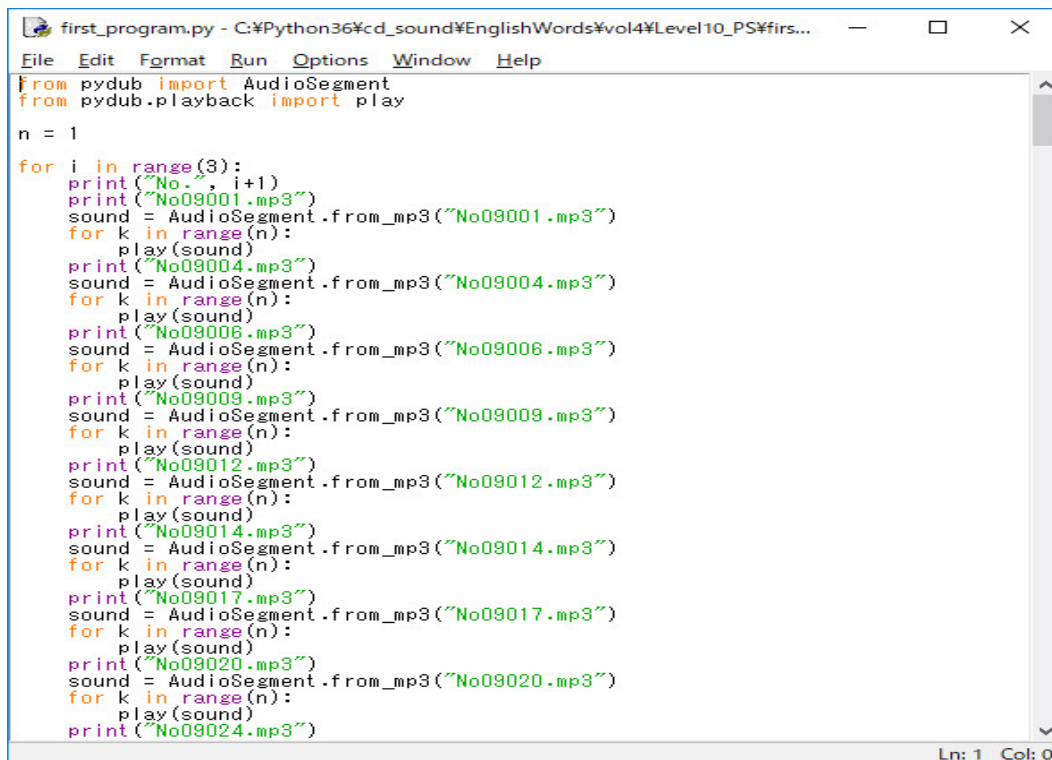
のプログラムでも良いです。

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 [Anaconda 4.4.0 (64-bit)] (default, May 11 2017, 13:25:24) [MSC v.1
900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Python36\cd_sound\EnglishWords\LoyalBooks\studyinscarlet_bn_libriov
x\changeFileName.py
ren ~studyinscarlet_01_doyler_64kb.mp3~ 01.mp3
ren ~studyinscarlet_02_doyler_64kb.mp3~ 02.mp3
ren ~studyinscarlet_03_doyler_64kb.mp3~ 03.mp3
ren ~studyinscarlet_04_doyler_64kb.mp3~ 04.mp3
ren ~studyinscarlet_05_doyler_64kb.mp3~ 05.mp3
ren ~studyinscarlet_06_doyler_64kb.mp3~ 06.mp3
>>>
```

という命令を実行しています。音声データを購入した時は、



のようなファイル名になっていることがあります。このようなときには、ファイル名の情報を表示するように



```
first_program.py - C:\Python36\cd_sound\EnglishWords\vol4\Level10_PS\first...
File Edit Format Run Options Window Help
from pydub import AudioSegment
from pydub.playback import play

n = 1

for i in range(3):
    print("No.", i+1)
    print("No09001.mp3")
    sound = AudioSegment.from_mp3("No09001.mp3")
    for k in range(n):
        play(sound)
    print("No09004.mp3")
    sound = AudioSegment.from_mp3("No09004.mp3")
    for k in range(n):
        play(sound)
    print("No09006.mp3")
    sound = AudioSegment.from_mp3("No09006.mp3")
    for k in range(n):
        play(sound)
    print("No09009.mp3")
    sound = AudioSegment.from_mp3("No09009.mp3")
    for k in range(n):
        play(sound)
    print("No09012.mp3")
    sound = AudioSegment.from_mp3("No09012.mp3")
    for k in range(n):
        play(sound)
    print("No09014.mp3")
    sound = AudioSegment.from_mp3("No09014.mp3")
    for k in range(n):
        play(sound)
    print("No09017.mp3")
    sound = AudioSegment.from_mp3("No09017.mp3")
    for k in range(n):
        play(sound)
    print("No09020.mp3")
    sound = AudioSegment.from_mp3("No09020.mp3")
    for k in range(n):
        play(sound)
    print("No09024.mp3")
```

のようなプログラムを作りたいと思うことがあります。このプログラムを手で打ち込むのは大変です。

```
import os
import subprocess

print('from pydub import AudioSegment')
print('from pydub.playback import play')
print()
print('n = 1')
print()
print('for i in range(3):')
print('    print("No.", i+1)')
for file in os.listdir():
    base, ext = os.path.splitext(file)
    if ext == '.mp3':
        print('    print("{}").format(file)')
        print('    sound = AudioSegment.from_mp3("{}").format(file)')
        print('    for k in range(n):')
        print('        play(sound)')
```

というプログラムを作り、実行して

```

Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 [Anaconda 4.4.0 (64-bit)] (default, May 11 2017, 13:25:24) [MSC v.1
900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Python36\cd_sound\EnglishWords\vol14\Level10_PS\makePlayProgram.py
from pydub import AudioSegment
from pydub.playback import play

n = 1

for i in range(3):
    print("No.", i+1)
    print("No09001.mp3")
    sound = AudioSegment.from_mp3("No09001.mp3")
    for k in range(n):
        play(sound)
    print("No09004.mp3")
    sound = AudioSegment.from_mp3("No09004.mp3")
    for k in range(n):
        play(sound)
    print("No09006.mp3")
    sound = AudioSegment.from_mp3("No09006.mp3")
    for k in range(n):
        play(sound)
    print("No09009.mp3")
    sound = AudioSegment.from_mp3("No09009.mp3")
    for k in range(n):
        play(sound)
    print("No09012.mp3")
    sound = AudioSegment.from_mp3("No09012.mp3")
    for k in range(n):
        play(sound)
    print("No09014.mp3")
    sound = AudioSegment.from_mp3("No09014.mp3")
    for k in range(n):
        play(sound)
    print("No09017.mp3")
    sound = AudioSegment.from_mp3("No09017.mp3")
    for k in range(n):
        play(sound)

```

Ln: 566 Col: 1

の表示をコピーペーストすれば良いです。

名前	トラ...	タイトル	参加アーティスト	アルバム
01 家族について話す.wma	1			
02 趣味について話す.wma	2	趣味について話す	DHC	フランス語会話とっさの...
03 好みについて話す.wma	3	好みについて話す	DHC	フランス語会話とっさの...
04 生活習慣について話す.wma	4	生活習慣について話す	DHC	フランス語会話とっさの...
05 年齢・身長・体重について話す.wma	5	年齢・身長・体重について話す	DHC	フランス語会話とっさの...
06 病状について話す.wma	6	病状について話す	DHC	フランス語会話とっさの...
07 診察を受ける.wma	7	診察を受ける	DHC	フランス語会話とっさの...
08 薬局で.wma	8	薬局で	DHC	フランス語会話とっさの...
09 好きになる.wma	9	好きになる	DHC	フランス語会話とっさの...
10 デート.wma	10	デート	DHC	フランス語会話とっさの...
11 告白する.wma	11	告白する	DHC	フランス語会話とっさの...
12 ほめる.wma	12	ほめる	DHC	フランス語会話とっさの...
13 結婚.wma	13	結婚	DHC	フランス語会話とっさの...
14 離婚.wma	14	離婚	DHC	フランス語会話とっさの...
15 オフィスで.wma	15	オフィスで	DHC	フランス語会話とっさの...
16 給料と待遇.wma	16	給料と待遇	DHC	フランス語会話とっさの...
17 お金の管理.wma	17	お金の管理	DHC	フランス語会話とっさの...
18 職場の人間関係.wma	18	職場の人間関係	DHC	フランス語会話とっさの...
19 人をほめる.wma	19	人をほめる	DHC	フランス語会話とっさの...

のようなファイル名になっている場合、ファイル名を有効利用したければ、

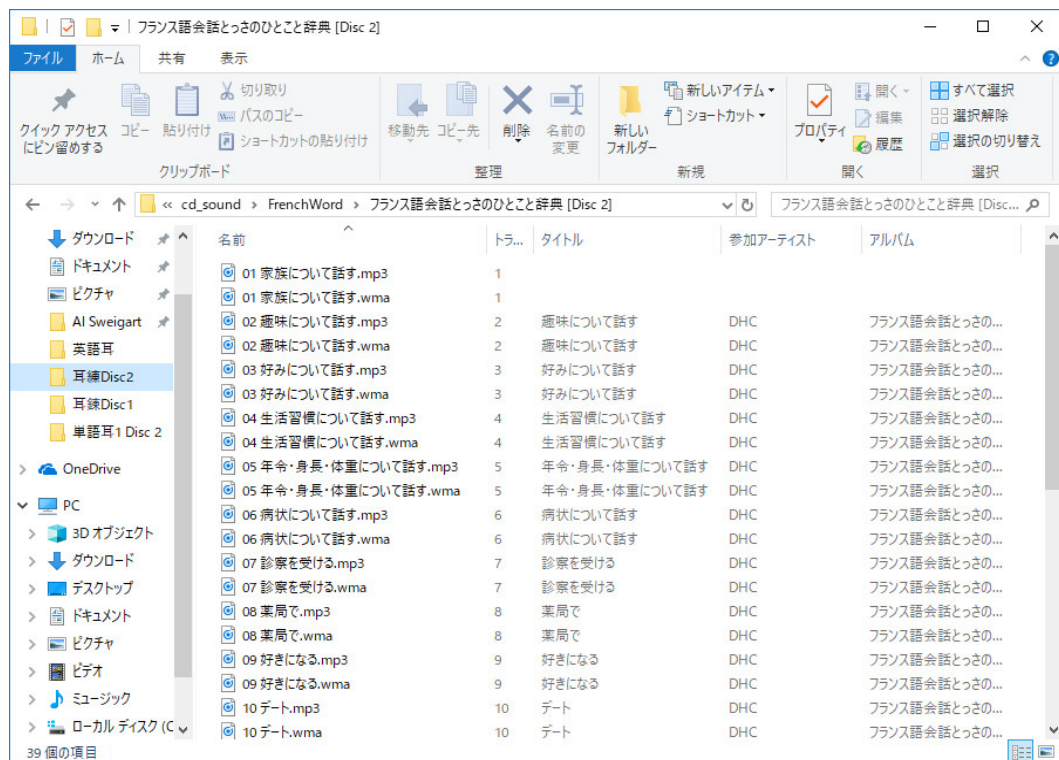

```

import os
import re
import subprocess

for file in os.listdir():
    base, ext = os.path.splitext(file)
    if ext == '.wma':
        words = base.split(" ")
        cmd = 'ffmpeg -i "{}" "{}.mp3"'.format(file, base)
        print(cmd)
        returncode = subprocess.call(cmd, shell=True)

```

というプログラムを作り、実行すれば



となるので、上のプログラム

```

import os
import subprocess

print('from pydub import AudioSegment')
print('from pydub.playback import play')
print()
print('n = 1')

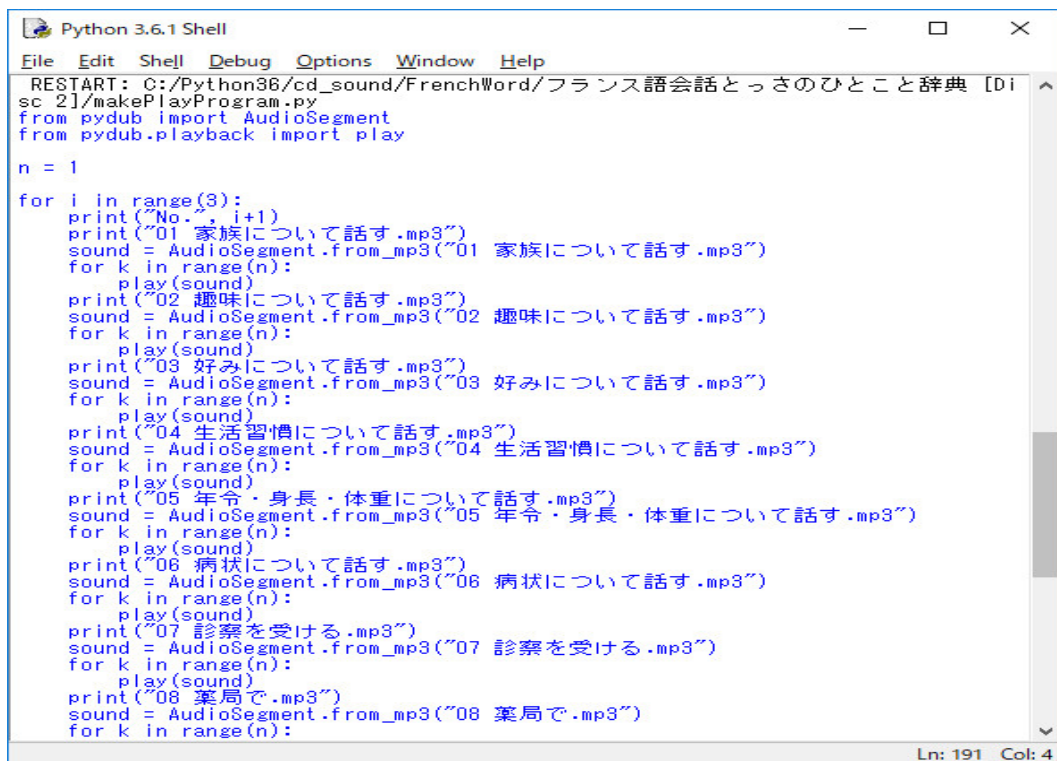
```

```

print()
print('for i in range(3):')
print('    print("No.", i+1)')
for file in os.listdir():
    base, ext = os.path.splitext(file)
    if ext == '.mp3':
        print('    print("{}").format(file)')
        print('    sound = AudioSegment.from_mp3("{}").format(file)')
        print('    for k in range(n):')
        print('        play(sound)')

```

というプログラムを作り、実行して



```

Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
RESTART: C:/Python36/cd_sound/FrenchWord/フランス語会話とっさのひとこと辞典 [Di
sc 2]/makePlayProgram.py
from pydub import AudioSegment
from pydub.playback import play

n = 1
for i in range(3):
    print("No.", i+1)
    print("01 家族について話す.mp3")
    sound = AudioSegment.from_mp3("01 家族について話す.mp3")
    for k in range(n):
        play(sound)
    print("02 趣味について話す.mp3")
    sound = AudioSegment.from_mp3("02 趣味について話す.mp3")
    for k in range(n):
        play(sound)
    print("03 好みについて話す.mp3")
    sound = AudioSegment.from_mp3("03 好みについて話す.mp3")
    for k in range(n):
        play(sound)
    print("04 生活習慣について話す.mp3")
    sound = AudioSegment.from_mp3("04 生活習慣について話す.mp3")
    for k in range(n):
        play(sound)
    print("05 年齢・身長・体重について話す.mp3")
    sound = AudioSegment.from_mp3("05 年齢・身長・体重について話す.mp3")
    for k in range(n):
        play(sound)
    print("06 病状について話す.mp3")
    sound = AudioSegment.from_mp3("06 病状について話す.mp3")
    for k in range(n):
        play(sound)
    print("07 診察を受ける.mp3")
    sound = AudioSegment.from_mp3("07 診察を受ける.mp3")
    for k in range(n):
        play(sound)
    print("08 薬局で.mp3")
    sound = AudioSegment.from_mp3("08 薬局で.mp3")
    for k in range(n):

```

の表示をコピーペーストすれば良いです。

```

from pydub import AudioSegment
from pydub.playback import play

```

音声ファイルの読み込み

```
sound = AudioSegment.from_file("07_01_01.mp3", "mp3")
```

情報の取得

```

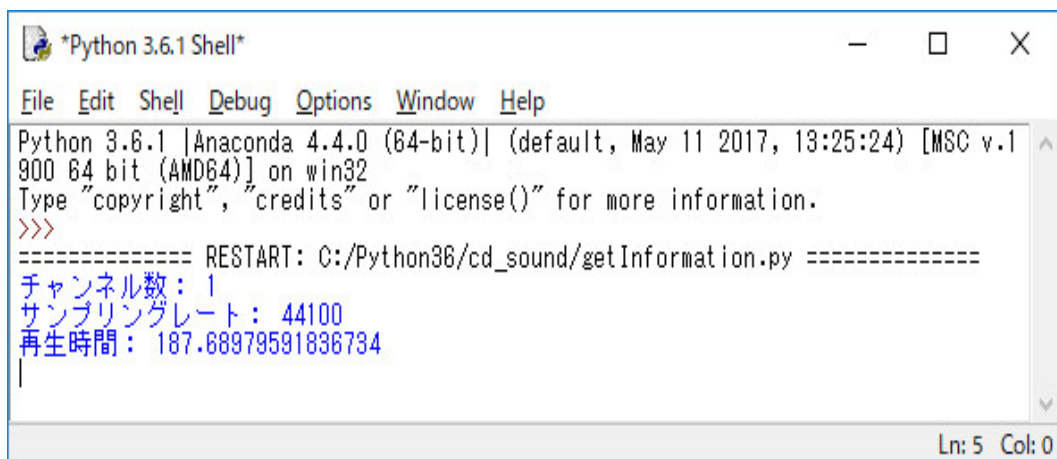
time = sound.duration_seconds # 再生時間 (秒)
rate = sound.frame_rate # サンプリングレート (Hz)
channel = sound.channels # チャンネル数 (1:mono, 2:stereo)

# 情報の表示
print('チャンネル数:', channel)
print('サンプリングレート:', rate)
print('再生時間:', time)

play(sound)

```

を実行すれば、



```

Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 [Anaconda 4.4.0 (64-bit)] (default, May 11 2017, 13:25:24) [MSC v.1
900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Python36/cd_sound/getInformation.py =====
チャンネル数: 1
サンプリングレート: 44100
再生時間: 187.68979591836734
|
Ln: 5 Col: 0

```

と.mp3 ファイルの情報を表示して、音声を再生してくれます。

例えば、OLYMPUS のヴォイスレコーダー Voice-Trex DS-750 で録音すれば、一部分を繰り返し再生することが出来ますが、このような機能を持った器機は高価です。Python のプログラム

```

from pydub import AudioSegment
from pydub.playback import play

# mp3 ファイルの読み込み
sound = AudioSegment.from_file("07_01_01.mp3", format="mp3")
# 全体を再生
play(sound)

# 最初～1分を抽出
sound0 = sound[:60000]
print("sound0")

```

```
play(sound0)
```

```
# 1分～2分を抽出
```

```
sound1 = sound[60000:120000]
```

```
print("sound1")
```

```
play(sound1)
```

```
# 最後の1分を抽出
```

```
sound2 = sound[-60000:]
```

```
print("sound2")
```

```
play(sound2)
```

で、ミリ秒で区間を指定して、一部分を再生できます。従って、



```
second_arabic1.py - C:\Python36\cd_sound\ArabicWord\アラビア語講座栄谷1...
File Edit Format Run Options Window Help
##### 第8課
print("第8課")
for k in range(n):
    print("45.mp3")
    sound = AudioSegment.from_mp3("45.mp3")
    play(sound)
    sound1 = sound[11500:23000]
    sound2 = sound1*10
    play(sound2)
    sound3 = sound[23000:34000]
    sound4 = sound3*10
    play(sound4)
for k in range(n):
    print("46.mp3")
    sound = AudioSegment.from_mp3("46.mp3")
    play(sound)
for k in range(n):
    print("47.mp3")
    sound = AudioSegment.from_mp3("47.mp3")
    play(sound)
for k in range(n):
    print("48.mp3")
    sound = AudioSegment.from_mp3("48.mp3")
    play(sound)
Ln: 282 Col: 0
```

のようなプログラムにすれば、何回も聞きたい部分を切り取って、今の場合は、10回づつ繰り返し聞くことができます。

ヴォイスレコーダーや Radio Server Pocket で録音した音声を、パソコンに接続して、パソコンに取り込むこともできます。 .wma ファイルで録音されている場合が多いです。上と同様な方法で、貴重な音声データを有効利用できます。色々やってみてください。数学やプログラミングの勉強と違って、外国語の勉強は大変です。無駄な努力を繰り返さないように、頭を使って、自分にとって有効な勉強法を探索してください。

C# などでも .mp3 ファイルの再生ができないことはないですが、Python を使った方が手軽に色々なことが出来ます。単独のファイルの再生は簡単ですが、複数のファイルの連続再生などは高度な技術が必要とされるみたいです。興味があれば、自分で調べてください。

2019 年、数学の研究や数学（特に専門外の確率・統計）やプログラミングの教育のための勉強からやっと解放されたので、家に来たがっていた本を買いためるだけで、読む時間のなかった将棋の勉強を再開しました。二、三十年間、将棋を指さなかったのも、すっかり棋力が落ちていています。元々、実力がなかったのですが、大山名人には、ペーパーテストで4段の評価をしていただいたことがあります。そのようなテスト問題は誰でも納得する解が必ずあるので、考えれば偶然分かる場合があります、実際の棋力ではありません。囲碁に比べ将棋は勝負が厳しい（防ぎ方をすぐには思いつかない嫌な手を次々指される）ので、負けるとダメージが大きいです。囲碁は自分では悪手を打ったことが分からないので、気楽ですが、全然上達しないです。本箱を探してみると駒落ちの本がたくさんありました。木村義雄著「将棋大観」、升田幸三著「将棋の必勝法」、大山康晴著「定本大山の駒落ち」、花村元司著「よくわかる駒落ち」、真部一男著「新駒落革命」、先崎学著「最強の駒落ち」、先崎学著「駒落ちのはなし」、所司和晴著「決定版駒落ち定跡」、高橋道雄著「コマ落ち新定跡」、高橋道雄著「棒銀と中飛車でコマ落ちを勝て!」、所司和晴著「次の一手」で覚える駒落ち定跡コレクション404」です。出版されている駒落ちの本は他にも幾つかありますが（多くは絶版になっていて、物によってはべらぼうな値段が付いています。）、読みもしない本をよく集めていたものです。それぞれ著者の個性あふれる素晴らしい本です。やっと趣味に時間がさけるようになったので少しずつ読み始めました。

これらの本で解説している自分で考えなければいけない重要な局面やフリーソフト（「技巧」や「屋根裏王」や「うさびょん」や「ねずみ将棋」や「K-Shogi64」）など藤井さんが使っているというすごく強いソフトからどうしようもないほど弱いものまで、フリーの将棋ソフトは色々あります。囲碁も現在は強いソフトを使うには高価な GPU が必要ですが、私の GPU なしの（安価な）パソコンでも使える Leela や AQ や Lizzie などよく知られたソフトをまとめてインストールできるようにしてくれている人もいて便利な世の中になりました。高知のような田舎にいても、将棋や囲碁の勉強が独学で効果的に無料で勉強できるようになりました。）との対局で悪手を差した局面（有難いことに「技巧」や「屋根裏王」などで解析すれば教えてくれます）を私のパソコンで印刷できるぎりぎりのサイズの市販のカード（B6 や A6 のカードが通販で 500 枚、2 千数百円で手に入ります）に印刷して自分で手筋集を作ってみようと思いました。

更に大学で第3外国語として一年間教えてもらったロシア語の文法の復習にも飽きて来たので、チェホフの小説を読もうとしましたが、まだ小説をスラスラ読むのは難しいので、ロシアの子供向けのチェスの本（子供向けだから易しいロシア語で書いていたと思いますが、普通文法書の最後で解説されている形動詞や副動詞などもバンバン出てくる高級な本でしたが、数学の本と同様、何を書いているはずかは分かりますので、辞書を引き引き何とか意味を取っています）をナウカから手に入れ読んでいたので、チェスのカードも作れるようにと考え、インターネットで $\text{\TeX}$ のスタイルファイルを探しました。囲碁のスタイルファイルは

昔、手に入れて既に使っていたので、将棋とチェスのものを探しました。将棋は2個、チェスは1個見つかりました。将棋の1個とチェスのものは、そのままではカードに印刷するには局面の図が小さすぎることが分かりました。図を大きくするために、スタイルファイルのマクロを勉強し直す為に、Knuth 著「TeX ブック」(今は翻訳は絶版ですが、原著の pdf ファイルがインターネットで無料で手に入ります。英文の方が分かりやすいです。)と「Metafont ブック」(こちらは原著も手に入らないみたいです)を本箱から見つけ、必要なところを読んでみました。この二つのスタイルファイルは TeX マクロで、初期画面から手を進めて、必要な画面を表示するというもので、すべてを TeX のマクロで書いているので、長文の TeX のマクロのプログラムを理解し、必要な修正を加えるのは大変だと分かったので断念しました。将棋のマクロを作った人は、外国人の作ったこのチェスのマクロを読んで理解してこの将棋のマクロを作ったそうです。チェスは白黒の市松模様の盤を使っているのも、罫線は外側だけで良いのも、そのスタイルファイルのフォントのプログラムを使わせてもらって、罫線は TeX のマクロで処理する TeX のプログラムをでっちあげることで、カードに局面を印刷できるようにしました。

```
\documentclass[10pt,a4paper]{jarticle}
```

```
\font\remark=skak30 scaled\magstep2
```

```
%\paperwidth 210mm%用紙の幅 (A4 サイズ縦長の用紙の幅、210mm=597pt)
```

```
%\paperheight 297mm%用紙の高さ (A4 サイズ縦長の用紙の高さ、297mm=845pt)
```

```
\voffset 0pt%ヘッダ上端までの上余白 (実際には1インチ+\voffsetの上余白となる)
```

```
\hoffset -15pt%文章領域までの左余白 (実際には1インチ+\hoffsetの左余白となる)
```

```
\topmargin -20pt%上余白からヘッダ上端までの余白
```

```
\headheight -20pt%ヘッダの高さ
```

```
\headsep 0pt%ヘッダの下端から本文領域までの高さ
```

```
\topskip 0pt%文章領域内の本文開始位置までの高さ
```

```
\oddsidemargin -15pt%奇数ページのマージン (左余白から文章領域の左端までの余白)
```

```
%\evensidemargin 0pt%偶数ページのマージン (jsbookなど左右ページでレイアウトが異なるときに指定する)
```

```
\textwidth 176mm%文章領域の幅 (A4 サイズ縦長の文章領域の幅、176mm=500pt)
```

```
\textheight 264mm%文章領域の高さ (A4 サイズ縦長の文章領域の高さ、264mm=750pt)
```

```
%\textwidth 53zw%文章領域の幅 (A4 サイズ縦長のとき) 文章領域の高さの pt × 0.107 = 全角文字数
```


%\textheight 80zw%文章領域の高さ (A4 サイズ縦長のとき) 文章領域の高さの
pt $\times$ 0.107 = 全角文字数

\footskip 20pt%文章領域の下端からフッタの下端までの高さ

\marginparsep 0pt%文章領域の右端から欄外傍注の左端の幅

\marginparwidth 0pt%傍注の幅

\marginparpush 0pt%傍注間の高さ

\begin{document}

\Huge

\centerline{

\hbox{\vrule

\vbox{\hrule

\hbox{\remark r m b l k a 0 s }

\hbox{\remark o p o p Z p o p }

\hbox{\remark 0 Z 0 Z 0 m 0 Z }

\hbox{\remark Z 0 Z 0 o 0 Z 0 }

\hbox{\remark 0 Z 0 Z P Z 0 Z }

\hbox{\remark Z 0 Z 0 Z N Z 0 }

\hbox{\remark P 0 P 0 0 0 P 0 }

\hbox{\remark S N A Q J B Z R }

\hrule }

\hskip -7pt \vrule }

}

\vskip 1cm

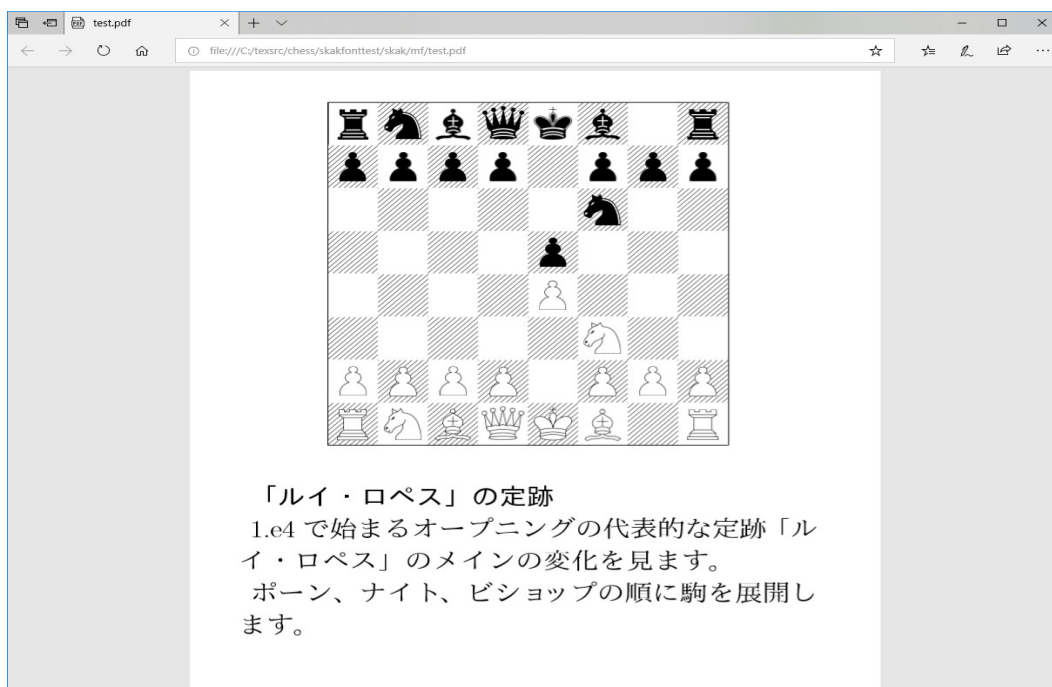
{\bf 「ルイ・ロペス」の定跡 }

1.e4 で始まるオープニングの代表的な定跡「ルイ・ロペス」のメインの変化を見
ます。

ポーン、ナイト、ビショップの順に駒を展開します。

\end{document}

で



の pdf ファイルを得ることができます。これをカードに印刷すると丁度の大きさになります。最初にごちゃごちゃ書いてあるのは、カードに印刷した時、丁度の大きさになるよう $\text{T}_{\text{E}}\text{X}$ にする指示です。

将棋はもう一人の人の作った MyShogi.sty というスタイルファイルをそのまま使って、フォントの大きさの指定を変えるだけでよいことが分かりました。次のような $\text{T}_{\text{E}}\text{X}$ ファイル

```
\documentclass[10pt,a4paper]{jarticle}
\usepackage[dvips]{graphicx}
\usepackage[dvips]{color}
\usepackage{myshogi}

%\paperwidth 210mm%用紙の幅 (A4 サイズ縦長の用紙の幅、210mm=597pt)
%\paperheight 297mm%用紙の高さ (A4 サイズ縦長の用紙の高さ、297mm=845pt)
\voffset 0pt%ヘッダ上端までの上余白 (実際には1インチ+\voffset の上余白
となる)
\hoffset -15pt%文章領域までの左余白 (実際には1インチ+\hoffset の左余白
となる)
\topmargin -20pt%上余白からヘッダ上端までの余白
\headheight -20pt%ヘッダの高さ
\headsep 0pt%ヘッダの下端から本文領域までの高さ
\topskip 0pt%文章領域内の本文開始位置までの高さ
```

`\oddsidemargin -15pt%`奇数ページのマージン (左余白から文章領域の左端までの余白)
`%\evensidemargin 0pt%`偶数ページのマージン (jsbook など左右ページでレイアウトが異なるときに指定する)
`\textwidth 176mm%`文章領域の幅 (A4 サイズ縦長の文章領域の幅、176mm=500pt)
`\textheight 264mm%`文章領域の高さ (A4 サイズ縦長の文章領域の高さ、264mm=750pt)
`%\textwidth 53zw%`文章領域の幅 (A4 サイズ縦長のとき) 文章領域の高さの pt $\times$ 0.107 = 全角文字数
`%\textheight 80zw%`文章領域の高さ (A4 サイズ縦長のとき) 文章領域の高さの pt $\times$ 0.107 = 全角文字数
`\footskip 20pt%`文章領域の下端からフッタの下端までの高さ
`\marginparsep 0pt%`文章領域の右端から欄外傍注の左端の幅
`\marginparwidth 0pt%`傍注の幅
`\marginparpush 0pt%`傍注間の高さ

`\begin{document}`

`\huge`

`\begin{center}`
`{\bf 二枚落ち基本図 (▲ 4 五歩まで)}`
`\begin{myshogi}[1.8]`
`\banmen`
`\mochigoma{\sente}{0}{0}{0}{0}{0}{0}{0}`
`\mochigoma{\gote}{0}{0}{0}{0}{0}{0}{0}`
`\koma{11}{\gote}{\kyou}`
`\koma{21}{\gote}{\kei}`
`\koma{31}{\gote}{\gin}`
`\koma{41}{\gote}{\kin}`
`\koma{51}{\gote}{\gyoku}`
`\koma{61}{\gote}{\kin}`
`\koma{81}{\gote}{\kei}`
`\koma{91}{\gote}{\kyou}`
`\koma{13}{\gote}{\fu}`
`\koma{23}{\gote}{\fu}`
`\koma{33}{\gote}{\fu}`
`\koma{43}{\gote}{\fu}`
`\koma{53}{\gote}{\gin}`
`\koma{63}{\gote}{\fu}`

```

\koma{73}{\gote}{\fu}
\koma{83}{\gote}{\fu}
\koma{93}{\gote}{\fu}
\koma{54}{\gote}{\fu}
\koma{45}{\sente}{\fu}
\koma{76}{\sente}{\fu}
\koma{17}{\sente}{\fu}
\koma{27}{\sente}{\fu}
\koma{37}{\sente}{\fu}
\koma{57}{\sente}{\fu}
\koma{67}{\sente}{\fu}
\koma{87}{\sente}{\fu}
\koma{97}{\sente}{\fu}
\koma{28}{\sente}{\hi}
\koma{88}{\sente}{\kaku}
\koma{19}{\sente}{\kyou}
\koma{29}{\sente}{\kei}
\koma{39}{\sente}{\gin}
\koma{49}{\sente}{\kin}
\koma{59}{\sente}{\gyoku}
\koma{69}{\sente}{\kin}
\koma{79}{\sente}{\gin}
\koma{89}{\sente}{\kei}
\koma{99}{\sente}{\kyou}
\end{myshogi}

```

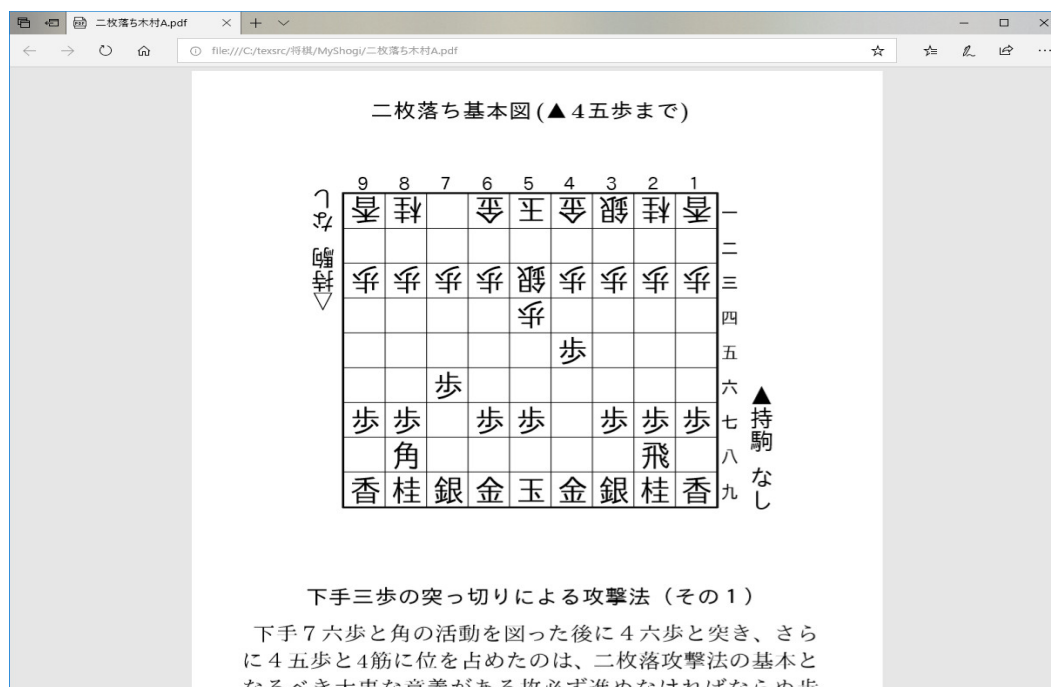
{\bf 下手三歩の突っ切りによる攻撃法（その１）}

\end{center}

下手七六歩と角の活動を図った後に四六歩と突き、さらに四五歩と四筋に位を占めたのは、二枚落攻撃法の基本となるべき大事な意義がある故必ず進めなければならぬ歩であって、この一事を知ると知らぬでは、勝敗に甚大の影響を及ぼすと同時に、又実に二枚落将棋の基礎となるべき緊要な手と言ってよい。

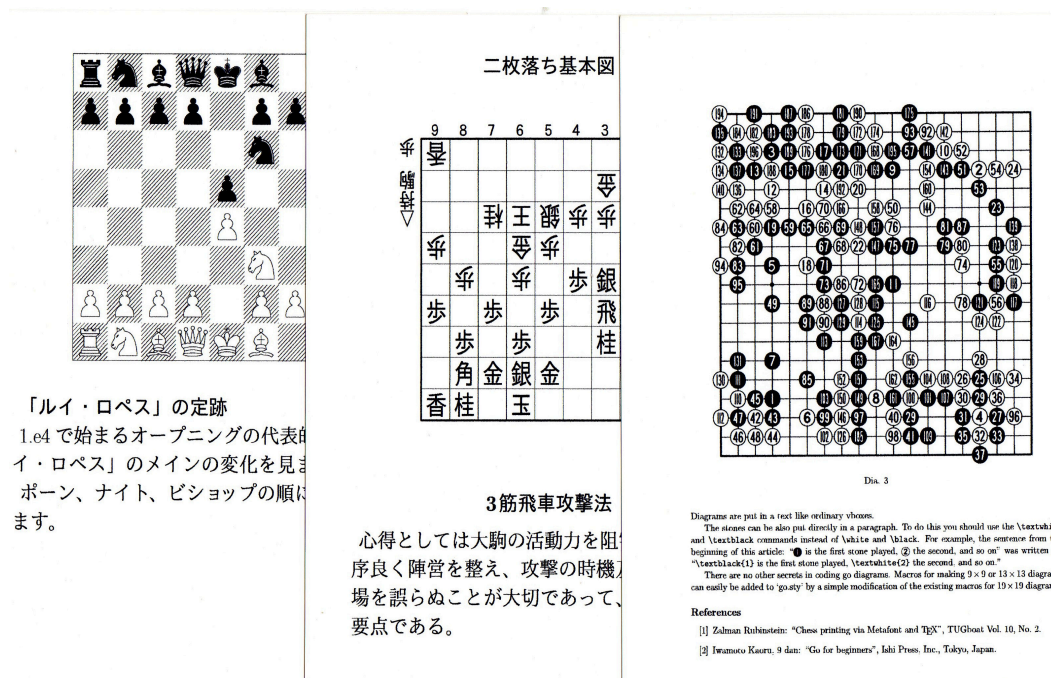
\end{document}

で、



のような pdf ファイルを作ってくれます。これをカードの大きさに印刷すれば良いです。

最終的に作るのは



のようなカードです。囲碁は囲碁用の sty ファイルをそのまま使って、昔、対局用の囲碁のプログラムを作ろうとしていた時、局面の T_EX ファイルを保存する部分も作っていました。

労力をいとわなければ、問題は解決したことになります。問題は上のファイルを作るのが厄介なことです。将棋の棋譜を編集するソフトで局面を作り、



その局面を $\text{T}_{\text{E}}\text{X}$ のファイルに変換して、保存してくれるソフトがあれば便利です。この棋譜編集ソフトは Kifu for Windows V7 ですが、ソースコードが公表されていませんので、これを作り替えて、欲しいプログラムにすることは私には出来ません。これと同じものを何らかのプログラミング言語で作り、局面を $\text{T}_{\text{E}}\text{X}$ のファイルに変換して保存してくれるソフトに拡張するのも一つの方法ですが、すでにあるソフトを有効利用して、足りない部分だけ作るのが楽な方法です。

Kifu for Windows V7 で何ができるのか、調べてみます。定跡や棋譜を打ち込み、自動的に再現させることが出来ます。棋譜は盤上で駒を動かして勉強しないと意味がないとの意見もありますが、所詮趣味で将棋を勉強しているので、自動的に棋譜を再現してくれるのを眺めるだけという楽な方法を選びます。初期画面から手を進めていくが、上の局面で同銀とする場合と6五金とする場合と分岐する場合と二通りの棋譜を記録できる場合がありますが、Kifu for Windows V7 で分岐を記録したい時に、記録できたりできなかったりします。このソフトが何を基準に分岐を記録しているか、分岐が複数ある時、このソフトの使っているアルゴリズムを理解する必要があります。内部のデータ構造はどんな複雑なものでも自由に作れますが、ファイルに保存して、復元するためには、複雑なデータ構造は使えません。

Kifu for Windows V7 が分岐を保存したファイルで、どのように処理しているか調べてみます。

Kifu for Windows V7 が正常に動いて保存した棋譜ファイル 木村1.kif は

#KIF version=2.0 encoding=Shift_JIS

---- Kifu for Windows V7 V7.31 棋譜ファイル ----

開始日時：2019/03/20 12:16

手合割：二枚落ち

下手：治

上手：木村義雄

手数----指手-----消費時間--

1	6 二銀 (71)	(0:15/00:00:15)
2	7 六歩 (77)	(0:03/00:00:03)
3	5 四歩 (53)	(0:02/00:00:17)
4	4 六歩 (47)	(0:02/00:00:05)
5	5 三銀 (62)	(0:02/00:00:19)
6	4 五歩 (46)	(0:01/00:00:06)
7	4 二玉 (51)	(0:11/00:00:30)
8	3 六歩 (37)	(0:05/00:00:11)
9	3 二玉 (42)	(0:03/00:00:33)
10	3 五歩 (36)	(0:02/00:00:13)
11	2 二銀 (31)	(0:05/00:00:38)
12	4 八銀 (39)	(0:05/00:00:18)
13	7 二金 (61)	(0:05/00:00:43)
14	4 七銀 (48)	(0:04/00:00:22)
15	4 二金 (41)	(0:04/00:00:47)
16	2 六歩 (27)	(0:04/00:00:26)
17	7 四歩 (73)	(0:03/00:00:50)
18	2 五歩 (26)	(0:04/00:00:30)
19	7 三金 (72)	(0:03/00:00:53)
20	2 四歩 (25)	(0:05/00:00:35)
21	同 歩 (23)	(0:03/00:00:56)
22	同 飛 (28)	(0:02/00:00:37)
23	2 三步打	(0:05/00:01:01)
24	2 六飛 (24)	(0:04/00:00:41)
25	6 四歩 (63)	(0:09/00:01:10)
26	7 八金 (69)	(0:05/00:00:46)
27	6 五歩 (64)	(0:04/00:01:14)
28	6 八銀 (79)	(0:03/00:00:49)
29	8 四歩 (83)	(0:04/00:01:18)
30	6 九玉 (59)	(0:04/00:00:53)
31	6 四金 (73)	(0:03/00:01:21)
32	5 八金 (49)	(0:17/00:01:10)

33	7 三桂 (81)	(0:05/00:01:26)
34	3 七桂 (29)	(0:07/00:01:17)
35	8 五歩 (84)	(0:06/00:01:32)
36	3 六飛 (26)	(0:06/00:01:23)
37	9 四歩 (93)	(0:07/00:01:39)
38	3 四歩 (35)	(0:04/00:01:27)
39	同 歩 (33)	(0:03/00:01:42)
40	同 飛 (36)	(0:04/00:01:31)
41	3 三歩打	(0:05/00:01:47)
42	3 六飛 (34)	(0:05/00:01:36)
43	9 五歩 (94)	(0:09/00:01:56)
44	7 五歩 (76)	(0:15/00:01:51)
45	同 歩 (74)	(0:05/00:02:01)
46	7 二歩打	(0:03/00:01:54)
47	6 二銀 (53)	(0:04/00:02:05)
48	4 四歩 (45)	(0:07/00:02:01)
49	5 五歩 (54)	(0:07/00:02:12)
50	4 六銀 (47)	(0:02/00:02:03)
51	6 六歩 (65)	(0:05/00:02:17)
52	同 角 (88)	(0:05/00:02:08)
53	5 三金 (42)	(0:05/00:02:22)
54	7 一歩成 (72)	(0:11/00:02:19)
55	6 五金 (64)	(2:28/00:04:50)+
56	7 二と (71)	(0:06/00:02:25)
57	6 六金 (65)	(0:02/00:04:52)
58	6 二と (72)	(0:02/00:02:27)

変化 : 55 手

55	同 銀 (62)	(0:16/00:02:38)
56	5 五銀 (46)	(0:19/00:02:38)
57	6 五金 (64)	(0:07/00:02:45)+
58	4 五桂 (37)	(0:05/00:02:43)
59	5 二金 (53)	(0:05/00:02:50)+
60	5 四銀 (55)	(0:05/00:02:48)
61	6 六金 (65)	(0:05/00:02:55)
62	同 飛 (36)	(0:02/00:02:50)
63	6 五歩打	(0:03/00:02:58)
64	5 六飛 (66)	(0:04/00:02:54)

変化 : 59 手

- 59 6 三金 (53) (1:20/00:04:05)
- 60 3 四歩打 (0:11/00:02:54)
- 61 同 歩 (33) (0:06/00:04:11)
- 62 同 飛 (36) (0:02/00:02:56)
- 63 3 三歩打 (0:07/00:04:18)
- 64 4 三歩成 (44) (0:13/00:03:09)
- 65 同 玉 (32) (0:05/00:04:23)
- 66 4 四飛 (34) (0:07/00:03:16)
- 67 5 二玉 (43) (0:07/00:04:30)
- 68 5 四銀 (55) (0:08/00:03:24)

変化 : 57 手

- 57 同 金 (64) (1:22/00:04:00)
- 58 同 角 (66) (0:08/00:02:46)
- 59 6 四銀打 (0:08/00:04:08)
- 60 4 五桂 (37) (0:06/00:02:52)
- 61 5 四金 (53) (0:09/00:04:17)
- 62 6 四角 (55) (0:05/00:02:57)
- 63 同 金 (54) (0:03/00:04:20)
- 64 5 三桂成 (45) (0:08/00:03:05)

です。これを

変化 : 57 手

- 57 同 金 (64) (1:22/00:04:00)
- 58 同 角 (66) (0:08/00:02:46)
- 59 6 四銀打 (0:08/00:04:08)
- 60 4 五桂 (37) (0:06/00:02:52)
- 61 5 四金 (53) (0:09/00:04:17)
- 62 6 四角 (55) (0:05/00:02:57)
- 63 同 金 (54) (0:03/00:04:20)
- 64 5 三桂成 (45) (0:08/00:03:05)

変化 : 59 手

59	6 三金 (53)	(1:20/00:04:05)
60	3 四歩打	(0:11/00:02:54)
61	同 歩 (33)	(0:06/00:04:11)
62	同 飛 (36)	(0:02/00:02:56)
63	3 三歩打	(0:07/00:04:18)
64	4 三歩成 (44)	(0:13/00:03:09)
65	同 玉 (32)	(0:05/00:04:23)
66	4 四飛 (34)	(0:07/00:03:16)
67	5 二玉 (43)	(0:07/00:04:30)
68	5 四銀 (55)	(0:08/00:03:24)

と変化手順を入れ替えて、木村 1_change.kif として保存し、

#KIF version=2.0 encoding=Shift_JIS

---- Kifu for Windows V7 V7.31 棋譜ファイル ----

開始日時 : 2019/03/20 12:16

手合割 : 二枚落ち

下手 : 治

上手 : 木村義雄

手数----指手-----消費時間--

1	6 二銀 (71)	(0:15/00:00:15)
2	7 六歩 (77)	(0:03/00:00:03)
3	5 四歩 (53)	(0:02/00:00:17)
4	4 六歩 (47)	(0:02/00:00:05)
5	5 三銀 (62)	(0:02/00:00:19)
6	4 五歩 (46)	(0:01/00:00:06)
7	4 二玉 (51)	(0:11/00:00:30)
8	3 六歩 (37)	(0:05/00:00:11)
9	3 二玉 (42)	(0:03/00:00:33)
10	3 五歩 (36)	(0:02/00:00:13)
11	2 二銀 (31)	(0:05/00:00:38)
12	4 八銀 (39)	(0:05/00:00:18)
13	7 二金 (61)	(0:05/00:00:43)
14	4 七銀 (48)	(0:04/00:00:22)
15	4 二金 (41)	(0:04/00:00:47)
16	2 六歩 (27)	(0:04/00:00:26)
17	7 四歩 (73)	(0:03/00:00:50)
18	2 五歩 (26)	(0:04/00:00:30)

19	7 三金 (72)	(0:03/00:00:53)
20	2 四步 (25)	(0:05/00:00:35)
21	同 步 (23)	(0:03/00:00:56)
22	同 飛 (28)	(0:02/00:00:37)
23	2 三步打	(0:05/00:01:01)
24	2 六飛 (24)	(0:04/00:00:41)
25	6 四步 (63)	(0:09/00:01:10)
26	7 八金 (69)	(0:05/00:00:46)
27	6 五步 (64)	(0:04/00:01:14)
28	6 八銀 (79)	(0:03/00:00:49)
29	8 四步 (83)	(0:04/00:01:18)
30	6 九玉 (59)	(0:04/00:00:53)
31	6 四金 (73)	(0:03/00:01:21)
32	5 八金 (49)	(0:17/00:01:10)
33	7 三桂 (81)	(0:05/00:01:26)
34	3 七桂 (29)	(0:07/00:01:17)
35	8 五步 (84)	(0:06/00:01:32)
36	3 六飛 (26)	(0:06/00:01:23)
37	9 四步 (93)	(0:07/00:01:39)
38	3 四步 (35)	(0:04/00:01:27)
39	同 步 (33)	(0:03/00:01:42)
40	同 飛 (36)	(0:04/00:01:31)
41	3 三步打	(0:05/00:01:47)
42	3 六飛 (34)	(0:05/00:01:36)
43	9 五步 (94)	(0:09/00:01:56)
44	7 五步 (76)	(0:15/00:01:51)
45	同 步 (74)	(0:05/00:02:01)
46	7 二步打	(0:03/00:01:54)
47	6 二銀 (53)	(0:04/00:02:05)
48	4 四步 (45)	(0:07/00:02:01)
49	5 五步 (54)	(0:07/00:02:12)
50	4 六銀 (47)	(0:02/00:02:03)
51	6 六步 (65)	(0:05/00:02:17)
52	同 角 (88)	(0:05/00:02:08)
53	5 三金 (42)	(0:05/00:02:22)
54	7 一步成 (72)	(0:11/00:02:19)
55	6 五金 (64)	(2:28/00:04:50)+
56	7 二と (71)	(0:06/00:02:25)
57	6 六金 (65)	(0:02/00:04:52)

58 6 二と (72) (0:02/00:02:27)

変化 : 55 手

55 同 銀 (62) (0:16/00:02:38)
56 5 五銀 (46) (0:19/00:02:38)
57 6 五金 (64) (0:07/00:02:45)+
58 4 五桂 (37) (0:05/00:02:43)
59 5 二金 (53) (0:05/00:02:50)+
60 5 四銀 (55) (0:05/00:02:48)
61 6 六金 (65) (0:05/00:02:55)
62 同 飛 (36) (0:02/00:02:50)
63 6 五歩打 (0:03/00:02:58)
64 5 六飛 (66) (0:04/00:02:54)

変化 : 57 手

57 同 金 (64) (1:22/00:04:00)
58 同 角 (66) (0:08/00:02:46)
59 6 四銀打 (0:08/00:04:08)
60 4 五桂 (37) (0:06/00:02:52)
61 5 四金 (53) (0:09/00:04:17)
62 6 四角 (55) (0:05/00:02:57)
63 同 金 (54) (0:03/00:04:20)
64 5 三桂成 (45) (0:08/00:03:05)

変化 : 59 手

59 6 三金 (53) (1:20/00:04:05)
60 3 四歩打 (0:11/00:02:54)
61 同 歩 (33) (0:06/00:04:11)
62 同 飛 (36) (0:02/00:02:56)
63 3 三歩打 (0:07/00:04:18)
64 4 三歩成 (44) (0:13/00:03:09)
65 同 玉 (32) (0:05/00:04:23)
66 4 四飛 (34) (0:07/00:03:16)
67 5 二玉 (43) (0:07/00:04:30)
68 5 四銀 (55) (0:08/00:03:24)

というファイルを Kifu for Windows V7 に読み込ますと、69 手 5 四銀 (55) でエラーと表示し、それを 木村 1_changeA.kif として保存してみると

#KIF version=2.0 encoding=Shift_JIS

---- Kifu for Windows V7 V7.31 棋譜ファイル ----

開始日時：2019/03/20 12:16

手合割：二枚落ち

下手：治

上手：木村義雄

手数----指手-----消費時間--

1	6 二銀 (71)	(0:15/00:00:15)
2	7 六歩 (77)	(0:03/00:00:03)
3	5 四歩 (53)	(0:02/00:00:17)
4	4 六歩 (47)	(0:02/00:00:05)
5	5 三銀 (62)	(0:02/00:00:19)
6	4 五歩 (46)	(0:01/00:00:06)
7	4 二玉 (51)	(0:11/00:00:30)
8	3 六歩 (37)	(0:05/00:00:11)
9	3 二玉 (42)	(0:03/00:00:33)
10	3 五歩 (36)	(0:02/00:00:13)
11	2 二銀 (31)	(0:05/00:00:38)
12	4 八銀 (39)	(0:05/00:00:18)
13	7 二金 (61)	(0:05/00:00:43)
14	4 七銀 (48)	(0:04/00:00:22)
15	4 二金 (41)	(0:04/00:00:47)
16	2 六歩 (27)	(0:04/00:00:26)
17	7 四歩 (73)	(0:03/00:00:50)
18	2 五歩 (26)	(0:04/00:00:30)
19	7 三金 (72)	(0:03/00:00:53)
20	2 四歩 (25)	(0:05/00:00:35)
21	同 歩 (23)	(0:03/00:00:56)
22	同 飛 (28)	(0:02/00:00:37)
23	2 三歩打	(0:05/00:01:01)
24	2 六飛 (24)	(0:04/00:00:41)
25	6 四歩 (63)	(0:09/00:01:10)
26	7 八金 (69)	(0:05/00:00:46)
27	6 五歩 (64)	(0:04/00:01:14)
28	6 八銀 (79)	(0:03/00:00:49)
29	8 四歩 (83)	(0:04/00:01:18)
30	6 九玉 (59)	(0:04/00:00:53)

31	6 四金 (73)	(0:03/00:01:21)
32	5 八金 (49)	(0:17/00:01:10)
33	7 三桂 (81)	(0:05/00:01:26)
34	3 七桂 (29)	(0:07/00:01:17)
35	8 五歩 (84)	(0:06/00:01:32)
36	3 六飛 (26)	(0:06/00:01:23)
37	9 四歩 (93)	(0:07/00:01:39)
38	3 四歩 (35)	(0:04/00:01:27)
39	同 歩 (33)	(0:03/00:01:42)
40	同 飛 (36)	(0:04/00:01:31)
41	3 三歩打	(0:05/00:01:47)
42	3 六飛 (34)	(0:05/00:01:36)
43	9 五歩 (94)	(0:09/00:01:56)
44	7 五歩 (76)	(0:15/00:01:51)
45	同 歩 (74)	(0:05/00:02:01)
46	7 二歩打	(0:03/00:01:54)
47	6 二銀 (53)	(0:04/00:02:05)
48	4 四歩 (45)	(0:07/00:02:01)
49	5 五歩 (54)	(0:07/00:02:12)
50	4 六銀 (47)	(0:02/00:02:03)
51	6 六歩 (65)	(0:05/00:02:17)
52	同 角 (88)	(0:05/00:02:08)
53	5 三金 (42)	(0:05/00:02:22)
54	7 一歩成 (72)	(0:11/00:02:19)
55	6 五金 (64)	(2:28/00:04:50)+
56	7 二と (71)	(0:06/00:02:25)
57	6 六金 (65)	(0:02/00:04:52)
58	6 二と (72)	(0:02/00:02:27)

変化 : 55 手

55	同 銀 (62)	(0:16/00:02:38)
56	5 五銀 (46)	(0:19/00:02:38)
57	6 五金 (64)	(0:07/00:02:45)+
58	4 五桂 (37)	(0:05/00:02:43)
59	5 二金 (53)	(0:05/00:02:50)
60	5 四銀 (55)	(0:05/00:02:48)
61	6 六金 (65)	(0:05/00:02:55)
62	同 飛 (36)	(0:02/00:02:50)

63 6 五歩打 (0:03/00:02:58)
64 5 六飛 (66) (0:04/00:02:54)

変化 : 57 手

57 同 金 (64) (1:22/00:04:00)
58 同 角 (66) (0:08/00:02:46)
59 6 四銀打 (0:08/00:04:08)+
60 4 五桂 (37) (0:06/00:02:52)
61 5 四金 (53) (0:09/00:04:17)
62 6 四角 (55) (0:05/00:02:57)
63 同 金 (54) (0:03/00:04:20)
64 5 三桂成 (45) (0:08/00:03:05)

変化 : 59 手

59 6 三金 (53) (1:20/00:05:20)
60 3 四歩打 (0:11/00:02:57)
61 同 歩 (33) (0:06/00:05:26)
62 同 飛 (36) (0:02/00:02:59)
63 3 三歩打 (0:07/00:05:33)
64 4 三歩成 (44) (0:13/00:03:12)
65 同 玉 (32) (0:05/00:05:38)
66 4 四飛 (34) (0:07/00:03:19)
67 5 二玉 (43) (0:07/00:05:45)
68 5 四角 (55) (0:08/00:03:27)

とファイルを勝手に修正します。

分岐の情報を線形モデルで保存するために、Kifu for Windows V7 はどのようなデータ構造を使っているか理解する必要があります。

木村 1.kif を

55 同 銀 (62) (0:16/00:02:38)+
56 5 五銀 (46) (0:19/00:02:38)
57 6 五金 (64) (0:07/00:02:45)+
58 4 五桂 (37) (0:05/00:02:43)
59 5 二金 (53) (0:05/00:02:50)+
60 5 四銀 (55) (0:05/00:02:48)
61 6 六金 (65) (0:05/00:02:55)

62 同 飛 (36) (0:02/00:02:50)
63 6 五歩打 (0:03/00:02:58)
64 5 六飛 (66) (0:04/00:02:54)

変化 : 55 手

55 6 五金 (64) (2:28/00:04:50)
56 7 二と (71) (0:06/00:02:25)
57 6 六金 (65) (0:02/00:04:52)
58 6 二と (72) (0:02/00:02:27)

と最初の分岐で順序を入れ替えて、それを 木村 1_changeB.kif として

#KIF version=2.0 encoding=Shift_JIS

---- Kifu for Windows V7 V7.31 棋譜ファイル ----

開始日時 : 2019/03/20 12:16

手合割 : 二枚落ち

下手 : 治

上手 : 木村義雄

手数----指手-----消費時間--

1 6 二銀 (71) (0:15/00:00:15)
2 7 六歩 (77) (0:03/00:00:03)
3 5 四歩 (53) (0:02/00:00:17)
4 4 六歩 (47) (0:02/00:00:05)
5 5 三銀 (62) (0:02/00:00:19)
6 4 五歩 (46) (0:01/00:00:06)
7 4 二玉 (51) (0:11/00:00:30)
8 3 六歩 (37) (0:05/00:00:11)
9 3 二玉 (42) (0:03/00:00:33)
10 3 五歩 (36) (0:02/00:00:13)
11 2 二銀 (31) (0:05/00:00:38)
12 4 八銀 (39) (0:05/00:00:18)
13 7 二金 (61) (0:05/00:00:43)
14 4 七銀 (48) (0:04/00:00:22)
15 4 二金 (41) (0:04/00:00:47)
16 2 六歩 (27) (0:04/00:00:26)
17 7 四歩 (73) (0:03/00:00:50)
18 2 五歩 (26) (0:04/00:00:30)
19 7 三金 (72) (0:03/00:00:53)

20	2 四步 (25)	(0:05/00:00:35)
21	同 步 (23)	(0:03/00:00:56)
22	同 飛 (28)	(0:02/00:00:37)
23	2 三步打	(0:05/00:01:01)
24	2 六飛 (24)	(0:04/00:00:41)
25	6 四步 (63)	(0:09/00:01:10)
26	7 八金 (69)	(0:05/00:00:46)
27	6 五步 (64)	(0:04/00:01:14)
28	6 八銀 (79)	(0:03/00:00:49)
29	8 四步 (83)	(0:04/00:01:18)
30	6 九玉 (59)	(0:04/00:00:53)
31	6 四金 (73)	(0:03/00:01:21)
32	5 八金 (49)	(0:17/00:01:10)
33	7 三桂 (81)	(0:05/00:01:26)
34	3 七桂 (29)	(0:07/00:01:17)
35	8 五步 (84)	(0:06/00:01:32)
36	3 六飛 (26)	(0:06/00:01:23)
37	9 四步 (93)	(0:07/00:01:39)
38	3 四步 (35)	(0:04/00:01:27)
39	同 步 (33)	(0:03/00:01:42)
40	同 飛 (36)	(0:04/00:01:31)
41	3 三步打	(0:05/00:01:47)
42	3 六飛 (34)	(0:05/00:01:36)
43	9 五步 (94)	(0:09/00:01:56)
44	7 五步 (76)	(0:15/00:01:51)
45	同 步 (74)	(0:05/00:02:01)
46	7 二步打	(0:03/00:01:54)
47	6 二銀 (53)	(0:04/00:02:05)
48	4 四步 (45)	(0:07/00:02:01)
49	5 五步 (54)	(0:07/00:02:12)
50	4 六銀 (47)	(0:02/00:02:03)
51	6 六步 (65)	(0:05/00:02:17)
52	同 角 (88)	(0:05/00:02:08)
53	5 三金 (42)	(0:05/00:02:22)
54	7 一步成 (72)	(0:11/00:02:19)
55	同 銀 (62)	(0:16/00:02:38)+
56	5 五銀 (46)	(0:19/00:02:38)
57	6 五金 (64)	(0:07/00:02:45)+
58	4 五桂 (37)	(0:05/00:02:43)

59 5 二金 (53) (0:05/00:02:50)+
 60 5 四銀 (55) (0:05/00:02:48)
 61 6 六金 (65) (0:05/00:02:55)
 62 同 飛 (36) (0:02/00:02:50)
 63 6 五歩打 (0:03/00:02:58)
 64 5 六飛 (66) (0:04/00:02:54)

変化 : 55 手

55 6 五金 (64) (2:28/00:04:50)
 56 7 二と (71) (0:06/00:02:25)
 57 6 六金 (65) (0:02/00:04:52)
 58 6 二と (72) (0:02/00:02:27)

変化 : 59 手

59 6 三金 (53) (1:20/00:04:05)
 60 3 四歩打 (0:11/00:02:54)
 61 同 歩 (33) (0:06/00:04:11)
 62 同 飛 (36) (0:02/00:02:56)
 63 3 三歩打 (0:07/00:04:18)
 64 4 三歩成 (44) (0:13/00:03:09)
 65 同 玉 (32) (0:05/00:04:23)
 66 4 四飛 (34) (0:07/00:03:16)
 67 5 二玉 (43) (0:07/00:04:30)
 68 5 四銀 (55) (0:08/00:03:24)

変化 : 57 手

57 同 金 (64) (1:22/00:04:00)
 58 同 角 (66) (0:08/00:02:46)
 59 6 四銀打 (0:08/00:04:08)
 60 4 五桂 (37) (0:06/00:02:52)
 61 5 四金 (53) (0:09/00:04:17)
 62 6 四角 (55) (0:05/00:02:57)
 63 同 金 (54) (0:03/00:04:20)
 64 5 三桂成 (45) (0:08/00:03:05)

Kifu for Windows V7 に読み込めると沢山エラーを表示して

#KIF version=2.0 encoding=Shift_JIS

---- Kifu for Windows V7 V7.31 棋譜ファイル ----

開始日時：2019/03/20 12:16

手合割：二枚落ち

下手：治

上手：木村義雄

手数----指手-----消費時間--

1	6 二銀 (71)	(0:15/00:00:15)
2	7 六歩 (77)	(0:03/00:00:03)
3	5 四歩 (53)	(0:02/00:00:17)
4	4 六歩 (47)	(0:02/00:00:05)
5	5 三銀 (62)	(0:02/00:00:19)
6	4 五歩 (46)	(0:01/00:00:06)
7	4 二玉 (51)	(0:11/00:00:30)
8	3 六歩 (37)	(0:05/00:00:11)
9	3 二玉 (42)	(0:03/00:00:33)
10	3 五歩 (36)	(0:02/00:00:13)
11	2 二銀 (31)	(0:05/00:00:38)
12	4 八銀 (39)	(0:05/00:00:18)
13	7 二金 (61)	(0:05/00:00:43)
14	4 七銀 (48)	(0:04/00:00:22)
15	4 二金 (41)	(0:04/00:00:47)
16	2 六歩 (27)	(0:04/00:00:26)
17	7 四歩 (73)	(0:03/00:00:50)
18	2 五歩 (26)	(0:04/00:00:30)
19	7 三金 (72)	(0:03/00:00:53)
20	2 四歩 (25)	(0:05/00:00:35)
21	同 歩 (23)	(0:03/00:00:56)
22	同 飛 (28)	(0:02/00:00:37)
23	2 三步打	(0:05/00:01:01)
24	2 六飛 (24)	(0:04/00:00:41)
25	6 四歩 (63)	(0:09/00:01:10)
26	7 八金 (69)	(0:05/00:00:46)
27	6 五歩 (64)	(0:04/00:01:14)
28	6 八銀 (79)	(0:03/00:00:49)
29	8 四歩 (83)	(0:04/00:01:18)
30	6 九玉 (59)	(0:04/00:00:53)
31	6 四金 (73)	(0:03/00:01:21)
32	5 八金 (49)	(0:17/00:01:10)

33	7 三桂 (81)	(0:05/00:01:26)
34	3 七桂 (29)	(0:07/00:01:17)
35	8 五歩 (84)	(0:06/00:01:32)
36	3 六飛 (26)	(0:06/00:01:23)
37	9 四歩 (93)	(0:07/00:01:39)
38	3 四歩 (35)	(0:04/00:01:27)
39	同 歩 (33)	(0:03/00:01:42)
40	同 飛 (36)	(0:04/00:01:31)
41	3 三步打	(0:05/00:01:47)
42	3 六飛 (34)	(0:05/00:01:36)
43	9 五歩 (94)	(0:09/00:01:56)
44	7 五歩 (76)	(0:15/00:01:51)
45	同 歩 (74)	(0:05/00:02:01)
46	7 二歩打	(0:03/00:01:54)
47	6 二銀 (53)	(0:04/00:02:05)
48	4 四歩 (45)	(0:07/00:02:01)
49	5 五歩 (54)	(0:07/00:02:12)
50	4 六銀 (47)	(0:02/00:02:03)
51	6 六歩 (65)	(0:05/00:02:17)
52	同 角 (88)	(0:05/00:02:08)
53	5 三金 (42)	(0:05/00:02:22)
54	7 一步成 (72)	(0:11/00:02:19)
55	同 銀 (62)	(0:16/00:02:38)+
56	5 五銀 (46)	(0:19/00:02:38)
57	6 五金 (64)	(0:07/00:02:45)
58	4 五桂 (37)	(0:05/00:02:43)
59	5 二金 (53)	(0:05/00:02:50)
60	5 四銀 (55)	(0:05/00:02:48)
61	6 六金 (65)	(0:05/00:02:55)
62	同 飛 (36)	(0:02/00:02:50)
63	6 五歩打	(0:03/00:02:58)
64	5 六飛 (66)	(0:04/00:02:54)

変化 : 55 手

55	6 五金 (64)	(2:28/00:04:50)
56	7 二と (71)	(0:06/00:02:25)
57	6 六金 (65)	(0:02/00:04:52)
58	6 二と (72)	(0:02/00:02:27)

59 6 三金 (53) (1:20/00:06:12)
 60 3 四歩打 (0:11/00:02:38)
 61 同 歩 (33) (0:06/00:06:18)
 62 同 飛 (36) (0:02/00:02:40)
 63 3 三歩打 (0:07/00:06:25)
 64 4 三歩成 (44) (0:13/00:02:53)
 65 同 玉 (32) (0:05/00:06:30)
 66 4 四飛 (34) (0:07/00:03:00)
 67 5 二玉 (43) (0:07/00:06:37)
 68 5 四歩 (55) (0:08/00:03:08)

とデータを勝手に修正します。

これらの実験で Kifu for Windows V7 がどのように動いているか分かりました。
 Kifu for Windows V7 が正常に動いて保存した棋譜ファイル 木村 1.kif

```

#KIF version=2.0 encoding=Shift_JIS
# ---- Kifu for Windows V7 V7.31 棋譜ファイル ----
開始日時：2019/03/20 12:16
手合割：二枚落ち
下手：治
上手：木村義雄
手数----指手-----消費時間--
  1 6 二銀 (71) ( 0:15/00:00:15)
  2 7 六歩 (77) ( 0:03/00:00:03)
  3 5 四歩 (53) ( 0:02/00:00:17)
  4 4 六歩 (47) ( 0:02/00:00:05)
  5 5 三銀 (62) ( 0:02/00:00:19)
  6 4 五歩 (46) ( 0:01/00:00:06)
  7 4 二玉 (51) ( 0:11/00:00:30)
  8 3 六歩 (37) ( 0:05/00:00:11)
  9 3 二玉 (42) ( 0:03/00:00:33)
 10 3 五歩 (36) ( 0:02/00:00:13)
 11 2 二銀 (31) ( 0:05/00:00:38)
 12 4 八銀 (39) ( 0:05/00:00:18)
 13 7 二金 (61) ( 0:05/00:00:43)
 14 4 七銀 (48) ( 0:04/00:00:22)
 15 4 二金 (41) ( 0:04/00:00:47)
 16 2 六歩 (27) ( 0:04/00:00:26)
 17 7 四歩 (73) ( 0:03/00:00:50)
  
```

18	2 五步 (26)	(0:04/00:00:30)
19	7 三金 (72)	(0:03/00:00:53)
20	2 四步 (25)	(0:05/00:00:35)
21	同 步 (23)	(0:03/00:00:56)
22	同 飛 (28)	(0:02/00:00:37)
23	2 三步打	(0:05/00:01:01)
24	2 六飛 (24)	(0:04/00:00:41)
25	6 四步 (63)	(0:09/00:01:10)
26	7 八金 (69)	(0:05/00:00:46)
27	6 五步 (64)	(0:04/00:01:14)
28	6 八銀 (79)	(0:03/00:00:49)
29	8 四步 (83)	(0:04/00:01:18)
30	6 九玉 (59)	(0:04/00:00:53)
31	6 四金 (73)	(0:03/00:01:21)
32	5 八金 (49)	(0:17/00:01:10)
33	7 三桂 (81)	(0:05/00:01:26)
34	3 七桂 (29)	(0:07/00:01:17)
35	8 五步 (84)	(0:06/00:01:32)
36	3 六飛 (26)	(0:06/00:01:23)
37	9 四步 (93)	(0:07/00:01:39)
38	3 四步 (35)	(0:04/00:01:27)
39	同 步 (33)	(0:03/00:01:42)
40	同 飛 (36)	(0:04/00:01:31)
41	3 三步打	(0:05/00:01:47)
42	3 六飛 (34)	(0:05/00:01:36)
43	9 五步 (94)	(0:09/00:01:56)
44	7 五步 (76)	(0:15/00:01:51)
45	同 步 (74)	(0:05/00:02:01)
46	7 二步打	(0:03/00:01:54)
47	6 二銀 (53)	(0:04/00:02:05)
48	4 四步 (45)	(0:07/00:02:01)
49	5 五步 (54)	(0:07/00:02:12)
50	4 六銀 (47)	(0:02/00:02:03)
51	6 六步 (65)	(0:05/00:02:17)
52	同 角 (88)	(0:05/00:02:08)
53	5 三金 (42)	(0:05/00:02:22)
54	7 一步成 (72)	(0:11/00:02:19)
55	6 五金 (64)	(2:28/00:04:50)+
56	7 二と (71)	(0:06/00:02:25)

57 6六金(65) (0:02/00:04:52)
58 6二と(72) (0:02/00:02:27)

変化 : 55 手

55 同 銀(62) (0:16/00:02:38)
56 5五銀(46) (0:19/00:02:38)
57 6五金(64) (0:07/00:02:45)+
58 4五桂(37) (0:05/00:02:43)
59 5二金(53) (0:05/00:02:50)+
60 5四銀(55) (0:05/00:02:48)
61 6六金(65) (0:05/00:02:55)
62 同 飛(36) (0:02/00:02:50)
63 6五歩打 (0:03/00:02:58)
64 5六飛(66) (0:04/00:02:54)

変化 : 59 手

59 6三金(53) (1:20/00:04:05)
60 3四歩打 (0:11/00:02:54)
61 同 歩(33) (0:06/00:04:11)
62 同 飛(36) (0:02/00:02:56)
63 3三歩打 (0:07/00:04:18)
64 4三歩成(44) (0:13/00:03:09)
65 同 玉(32) (0:05/00:04:23)
66 4四飛(34) (0:07/00:03:16)
67 5二玉(43) (0:07/00:04:30)
68 5四銀(55) (0:08/00:03:24)

変化 : 57 手

57 同 金(64) (1:22/00:04:00)
58 同 角(66) (0:08/00:02:46)
59 6四銀打 (0:08/00:04:08)
60 4五桂(37) (0:06/00:02:52)
61 5四金(53) (0:09/00:04:17)
62 6四角(55) (0:05/00:02:57)
63 同 金(54) (0:03/00:04:20)
64 5三桂成(45) (0:08/00:03:05)

で表現されているように、分岐があるときは

52 同 角 (88) (0:05/00:02:08)
53 5 三金 (42) (0:05/00:02:22)
54 7 一歩成 (72) (0:11/00:02:19)
55 6 五金 (64) (2:28/00:04:50)+
56 7 二と (71) (0:06/00:02:25)
57 6 六金 (65) (0:02/00:04:52)
58 6 二と (72) (0:02/00:02:27)

のように、分岐の起きるところに + の記号を付記し、分岐した変化を

変化 : 55 手

55 同 銀 (62) (0:16/00:02:38)
56 5 五銀 (46) (0:19/00:02:38)
57 6 五金 (64) (0:07/00:02:45)+
58 4 五桂 (37) (0:05/00:02:43)
59 5 二金 (53) (0:05/00:02:50)+
60 5 四銀 (55) (0:05/00:02:48)
61 6 六金 (65) (0:05/00:02:55)
62 同 飛 (36) (0:02/00:02:50)
63 6 五歩打 (0:03/00:02:58)
64 5 六飛 (66) (0:04/00:02:54)

と下に追加する。その時

52 同 角 (88) (0:05/00:02:08)
53 5 三金 (42) (0:05/00:02:22)
54 7 一歩成 (72) (0:11/00:02:19)
55 6 五金 (64) (2:28/00:04:50)+
56 7 二と (71) (0:06/00:02:25)
57 6 六金 (65) (0:02/00:04:52)
58 6 二と (72) (0:02/00:02:27)

変化 : 55 手

55	同 銀 (62)	(0:16/00:02:38)
56	5 五銀 (46)	(0:19/00:02:38)
57	6 五金 (64)	(0:07/00:02:45)+
58	4 五桂 (37)	(0:05/00:02:43)
59	5 二金 (53)	(0:05/00:02:50)+
60	5 四銀 (55)	(0:05/00:02:48)
61	6 六金 (65)	(0:05/00:02:55)
62	同 飛 (36)	(0:02/00:02:50)
63	6 五歩打	(0:03/00:02:58)
64	5 六飛 (66)	(0:04/00:02:54)

や

変化 : 55 手

55	同 銀 (62)	(0:16/00:02:38)
56	5 五銀 (46)	(0:19/00:02:38)
57	6 五金 (64)	(0:07/00:02:45)+
58	4 五桂 (37)	(0:05/00:02:43)
59	5 二金 (53)	(0:05/00:02:50)+
60	5 四銀 (55)	(0:05/00:02:48)
61	6 六金 (65)	(0:05/00:02:55)
62	同 飛 (36)	(0:02/00:02:50)
63	6 五歩打	(0:03/00:02:58)
64	5 六飛 (66)	(0:04/00:02:54)

変化 : 59 手

59	6 三金 (53)	(1:20/00:04:05)
60	3 四歩打	(0:11/00:02:54)
61	同 歩 (33)	(0:06/00:04:11)
62	同 飛 (36)	(0:02/00:02:56)
63	3 三歩打	(0:07/00:04:18)
64	4 三歩成 (44)	(0:13/00:03:09)
65	同 玉 (32)	(0:05/00:04:23)
66	4 四飛 (34)	(0:07/00:03:16)
67	5 二玉 (43)	(0:07/00:04:30)

68 5 四銀 (55) (0:08/00:03:24)

変化 : 57 手

57 同 金 (64) (1:22/00:04:00)

58 同 角 (66) (0:08/00:02:46)

59 6 四銀打 (0:08/00:04:08)

60 4 五桂 (37) (0:06/00:02:52)

61 5 四金 (53) (0:09/00:04:17)

62 6 四角 (55) (0:05/00:02:57)

63 同 金 (54) (0:03/00:04:20)

64 5 三桂成 (45) (0:08/00:03:05)

のように、「変化 : 57 手のような分岐を表現するためには、番号 5 7 をその場所から上に順にたどった時、初めて出会う番号 57 に分岐の記号 + が付記されていなければならない」と言うのが基本的なルールです。

Kifu for Windows V7 で分岐を記録したい時に、記録できたりできなかったりしていたのは、このような、「変化 : 57 手のような分岐を表現するためには、番号 5 7 をその場所から上に順にたどった時、初めて出会う番号 57 に分岐の記号 + が付記されていなければならない」と言う基本的なルールに当てはまっているかどうかで決まっていたことが理解出来ました。

Kifu for Windows V7 の使いかたが理解できたので、Kifu for Windows V7 で作ったファイルを読み込んで、

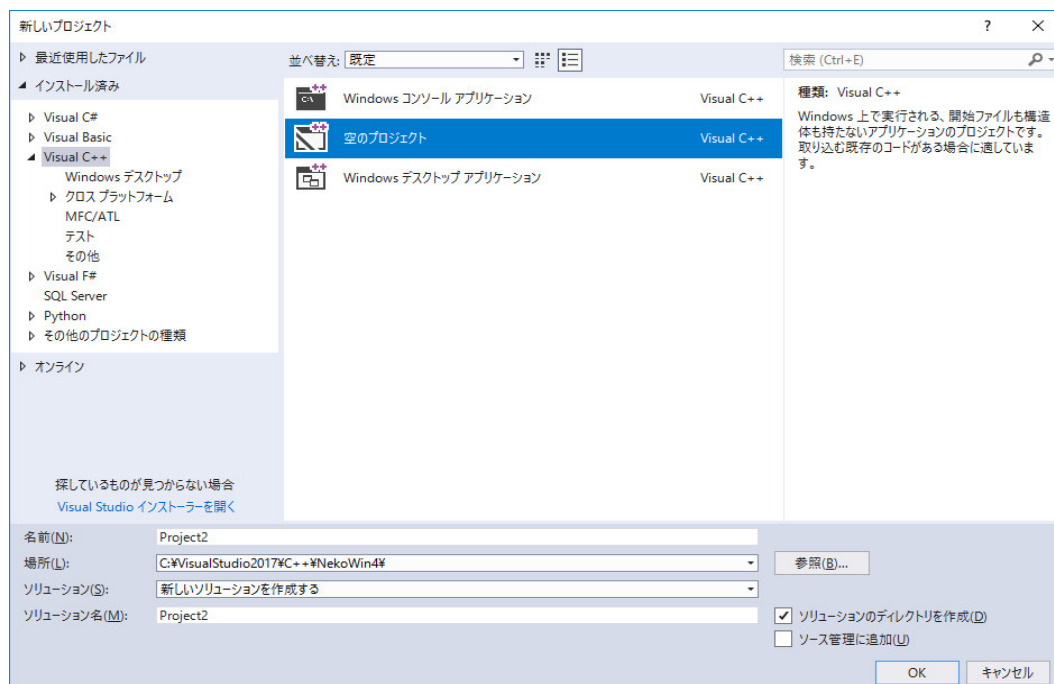
53 5 三金 (42)

のように指示すれば、その局面を描く TeX のプログラムを保存してくれるプログラムを作るのが最小の労力で出来ることです。前者のようなプログラムでは、将棋の局面をグラフィックスで表現する手間もいらないので、文字列の処理に慣れていれば多分楽です。最近の VC++ 2017 では、VC++ 2010 のように簡単にウインドウのプログラミングが出来なくなって、グラフィックスは Visual Basic か C# で作り、エンジン部分は C++ と言う MicroSoft の方針に従わざるを得なくなっている現状を考えると妥当な選択です。もっとも、昔の Windows プログラミングを思い出す意味で、VC++ 2017 で VKifu for Windows V7 と同等なものを自分で作ってみるのも勉強になるとは思いますが、まずは楽なことからやってみましょう。

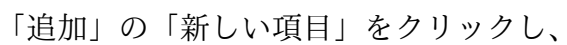
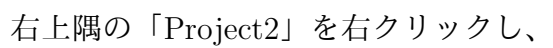
これらのプログラムはスピードが要求されるプログラムではないので、C++ 2017 でも、Python でも、Java でも、Ruby でも作れますが、ここでは、まず現在の VC++ と C# の現状を知るために、「作って覚える Visual C++ 2017 デスクトップアプリ入門」と「作って覚える Visual C# 2017 デスクトップアプリ入門」を本屋に並んでいて偶々目に付いたから購入してきて読んでみます。

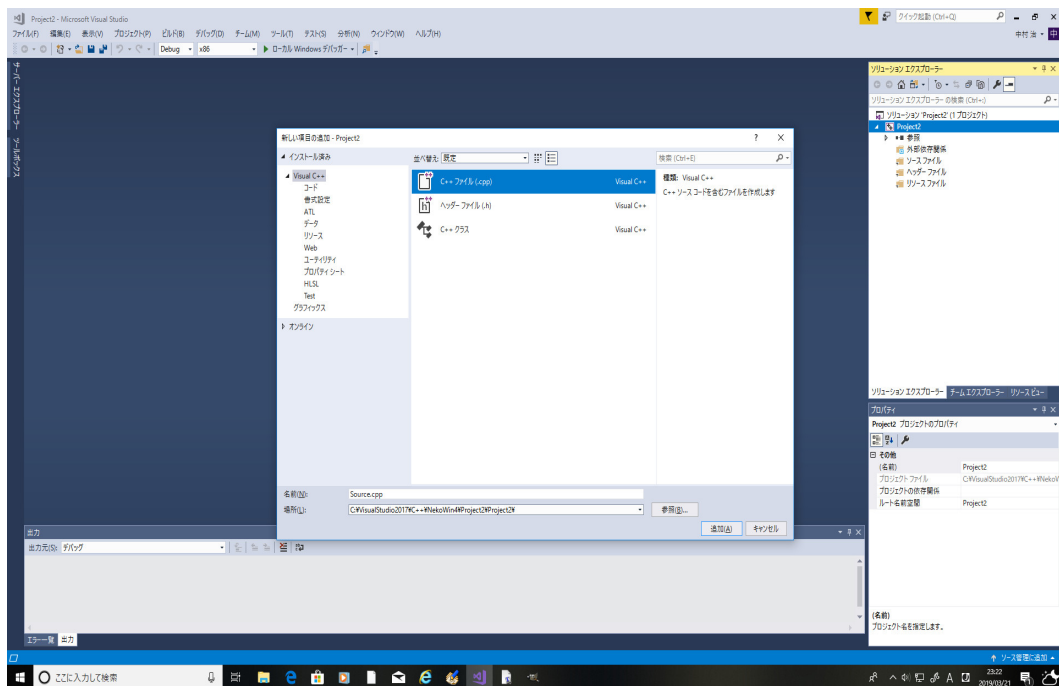
本屋でパラパラと眺めたところでは、今までのように簡単にウインドウのプログラミングができる方法が書いてあると思って、「作って覚える Visual C++ 2017 デスクトップアプリ入門」を購入したが、発売日は2018年3月10日発売だが、現在手に入る Visual Studio Community 2017 と使っているソフトのバージョンが違うのか、この本で使っている Visual C++ 2017 と私が手に入れた Visual C++ 2017 とは全然別のものであって、「作って覚える Visual C++ 2017 デスクトップアプリ入門」のサンプルプログラムがコンパイルできない。意味のない本を購入してしまった。僅かな期待を寄せていたが、今までのように簡単にウインドウのプログラミングが出来なくなって、やはりインターネットの情報通り、グラフィックスは Visual Basic か C# で作り、エンジン部分は C++ と言う Microsoft の方針に従わざるを得なくなっているみたいです。

VC++ 2017 で Windows プログラミングをするためには、本箱に眠っていた桑井康孝著「猫でもわかる Windows プログラミング」の出番です。

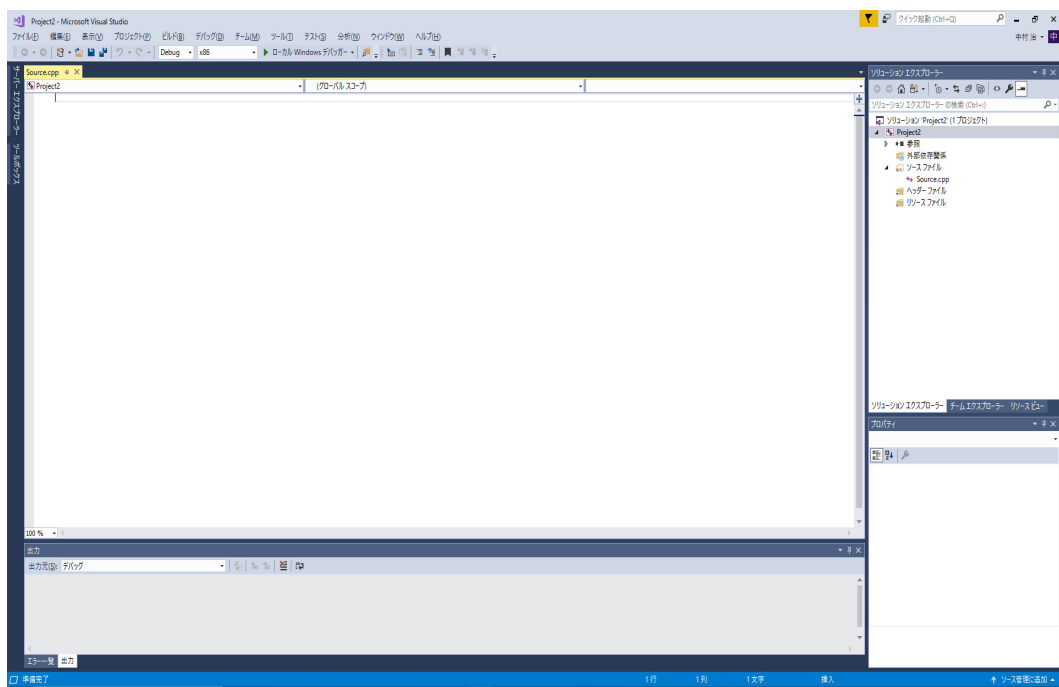


のように、「ファイル」メニューの「新規作成」の「プロジェクト」で、Visual C++ の「空のプロジェクト」を選択し、





で、ファイル名を指示して、「追加」します。



空の C++ のファイルが作られるので、

```
// graphics03.cpp
```

```
#include <windows.h>
```

```

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
ATOM InitApp(HINSTANCE);
BOOL InitInstance(HINSTANCE, int);

TCHAR szClassName[] = TEXT("graphics03"); //ウィンドウクラス

int WINAPI WinMain(HINSTANCE hCurInst, HINSTANCE hPrevInst,
    LPSTR lpsCmdLine, int nCmdShow)
{
    MSG msg;
    BOOL bRet;

    if (!InitApp(hCurInst))
        return FALSE;
    if (!InitInstance(hCurInst, nCmdShow))
        return FALSE;
    while ((bRet = GetMessage(&msg, NULL, 0, 0)) != 0) {
        if (bRet == -1) {
            break;
        } else {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }
    return (int)msg.wParam;
}

//ウィンドウ・クラスの登録

ATOM InitApp(HINSTANCE hInst)
{
    WNDCLASSEX wc;
    wc.cbSize = sizeof(WNDCLASSEX);
    wc.style = CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc = WndProc; //プロシージャ名
    wc.cbClsExtra = 0;
    wc.cbWndExtra = 0;

```

```

wc.hInstance = hInst;//インスタンス
wc.hIcon = (HICON)LoadImage(NULL,
MAKEINTRESOURCE(IDI_APPLICATION),
IMAGE_ICON,
0,
0,
LR_DEFAULTSIZE | LR_SHARED);
wc.hCursor = (HCURSOR)LoadImage(NULL,
MAKEINTRESOURCE(IDC_ARROW),
IMAGE_CURSOR,
0,
0,
LR_DEFAULTSIZE | LR_SHARED);
wc.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
wc.lpszMenuName = NULL; //メニュー名
wc.lpszClassName = szClassName;
wc.hIconSm = (HICON)LoadImage(NULL,
MAKEINTRESOURCE(IDI_APPLICATION),
IMAGE_ICON,
0,
0,
LR_DEFAULTSIZE | LR_SHARED);

return (RegisterClassEx(&wc));
}

//ウィンドウの生成

BOOL InitInstance(HINSTANCE hInst, int nCmdShow)
{
HWND hWnd;

hWnd = CreateWindow(szClassName,
TEXT("猫でもわかる四角形"), //タイトルバーにこの名前が表示されます
WS_OVERLAPPEDWINDOW, //ウィンドウの種類
CW_USEDEFAULT, //X座標
CW_USEDEFAULT, //Y座標
CW_USEDEFAULT, //幅
CW_USEDEFAULT, //高さ

```

```

NULL, //親ウィンドウのハンドル、親を作るときは NULL
NULL, //メニューハンドル、クラスメニューを使うときは NULL
hInst, //インスタンスハンドル
NULL);
if (!hWnd)
return FALSE;
ShowWindow(hWnd, nCmdShow);
UpdateWindow(hWnd);
return TRUE;
}

```

//ウィンドウプロシージャ

```

LRESULT CALLBACK WndProc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp)
{
    int id;
    HDC hdc;
    PAINTSTRUCT ps;
    HPEN hPen, hOldPen;
    HBRUSH hBrush, hOldBrush;

    switch (msg) {
    case WM_PAINT:
        hdc = BeginPaint(hWnd, &ps);
        hPen = CreatePen(PS_SOLID, 0, RGB(255, 0, 0));
        hOldPen = (HPEN)SelectObject(hdc, hPen);
        hBrush = CreateSolidBrush(RGB(0, 255, 0));
        hOldBrush = (HBRUSH)SelectObject(hdc, hBrush);
        Rectangle(hdc, 10, 10, 100, 100);

        DeleteObject(hBrush);
        hBrush = CreateHatchBrush(HS_BDIAGONAL, RGB(255, 0, 0));
        SelectObject(hdc, hBrush);
        Rectangle(hdc, 50, 50, 150, 150);
        DeleteObject(hBrush);
        hBrush = CreateHatchBrush(HS_FDIAGONAL, RGB(0, 0, 255));
        SelectObject(hdc, hBrush);
        Rectangle(hdc, 100, 100, 200, 200);
        DeleteObject(hBrush);
    }
}

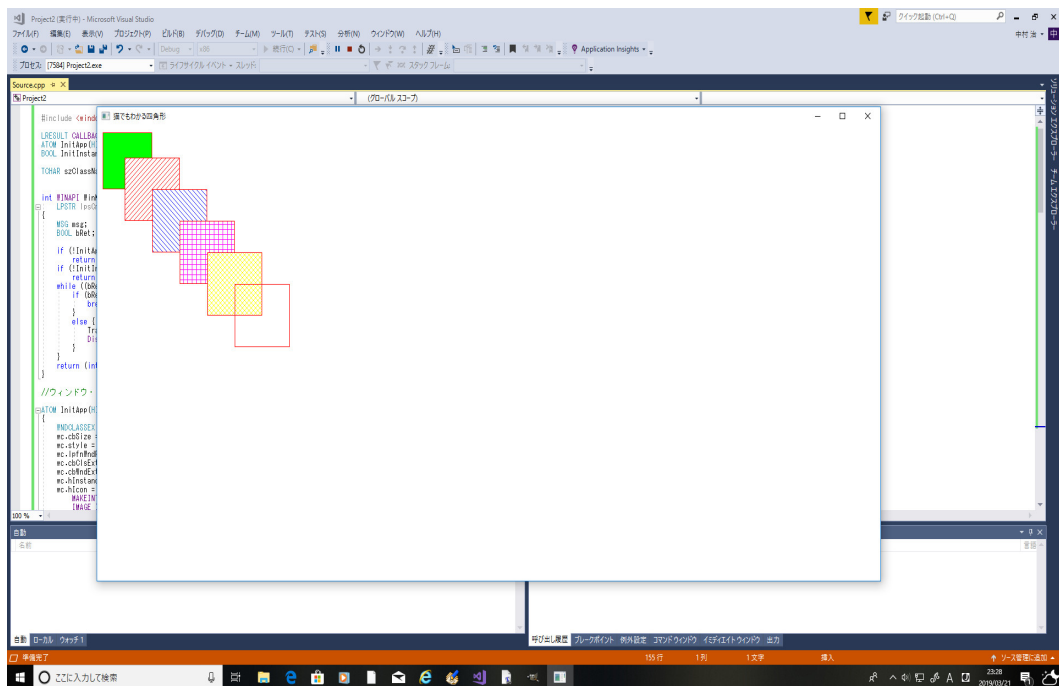
```

```

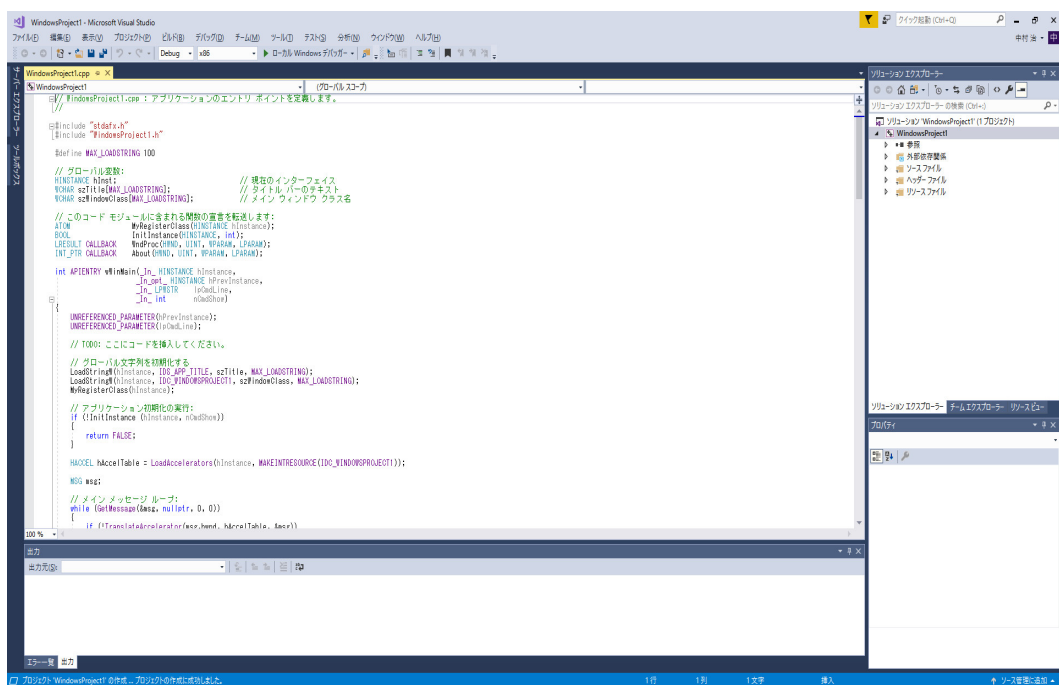
hBrush = CreateHatchBrush(HS_CROSS, RGB(255, 0, 255));
SelectObject(hdc, hBrush);
Rectangle(hdc, 150, 150, 250, 250);
DeleteObject(hBrush);
hBrush = CreateHatchBrush(HS_DIAGCROSS, RGB(255, 255, 0));
SelectObject(hdc, hBrush);
Rectangle(hdc, 200, 200, 300, 300);
DeleteObject(hBrush);
hBrush = (HBRUSH)GetStockObject(NULL_BRUSH);
SelectObject(hdc, hBrush);
Rectangle(hdc, 250, 250, 350, 350);
DeleteObject(hPen);
SelectObject(hdc, hOldPen);
SelectObject(hdc, hOldBrush);
EndPaint(hWnd, &ps);
break;
case WM_CLOSE:
id = MessageBox(hWnd,
TEXT("終了してもよろしいですか"),
TEXT("確認"),
MB_YESNO | MB_ICONQUESTION);
if (id == IDYES)
DestroyWindow(hWnd);
break;
case WM_DESTROY:
PostQuitMessage(0);
break;
default:
return (DefWindowProc(hWnd, msg, wp, lp));
}
return 0;
}

```

のような大昔の C の Windows プログラミングを打ち込めば、



となります。このような形の Windows プログラミングは今の Visual C++ 2017 でも可能です。しかし、VC++ 2017 にも、Windows デスクトップアプリケーションという選択肢があって、そのままコンパイル、実行すると空のウインドウが表示される。これにボタンなどを配置することは簡単にできないのだろうか？



これが Visual C++ 2017 の「Windows デスクトップアプリケーション」を指定した時の何もしないで作ってくれるプログラムです。

何がかかっているかと言うと

```
// WindowsProject1.cpp : アプリケーションのエントリ ポイントを定義しま
す。
//

#include "stdafx.h"
#include "WindowsProject1.h"

#define MAX_LOADSTRING 100

// グローバル変数:
HINSTANCE hInst;                                // 現在のインターフェ
イス
WCHAR szTitle[MAX_LOADSTRING];                  // タイトル バーのテキ
スト
WCHAR szWindowClass[MAX_LOADSTRING];            // メイン ウィンドウ ク
ラス名

// このコード モジュールに含まれる関数の宣言を転送します:
ATOM                MyRegisterClass(HINSTANCE hInstance);
BOOL                InitInstance(HINSTANCE, int);
LRESULT CALLBACK    WndProc(HWND, UINT, WPARAM, LPARAM);
INT_PTR CALLBACK    About(HWND, UINT, WPARAM, LPARAM);

int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
                     _In_opt_ HINSTANCE hPrevInstance,
                     _In_ LPWSTR     lpCmdLine,
                     _In_ int        nCmdShow)
{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

    // TODO: ここにコードを挿入してください。

    // グローバル文字列を初期化する
    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_WINDOWSPROJECT1, szWindowClass,
                MAX_LOADSTRING);
    MyRegisterClass(hInstance);
```



```

// アプリケーション初期化の実行:
if (!InitInstance (hInstance, nCmdShow))
{
    return FALSE;
}

HACCEL hAccelTable = LoadAccelerators(hInstance,
    MAKEINTRESOURCE(IDC_WINDOWSPROJECT1));

MSG msg;

// メイン メッセージ ループ:
while (GetMessage(&msg, nullptr, 0, 0))
{
    if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
}

return (int) msg.wParam;
}

//
// 関数: MyRegisterClass()
//
// 目的: ウィンドウ クラスを登録します。
//
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEXW wcex;

    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style          = CS_HREDRAW | CS_VREDRAW;

```

```

    wcex.lpfWndProc      = WndProc;
    wcex.cbClsExtra      = 0;
    wcex.cbWndExtra      = 0;
    wcex.hInstance       = hInstance;
    wcex.hIcon           = LoadIcon(hInstance,
        MAKEINTRESOURCE(IDI_WINDOWSPROJECT1));
    wcex.hCursor         = LoadCursor(nullptr, IDC_ARROW);
    wcex.hbrBackground   = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName     = MAKEINTRESOURCEW(IDC_WINDOWSPROJECT1);
    wcex.lpszClassName   = szWindowClass;
    wcex.hIconSm         = LoadIcon(wcex.hInstance,
        MAKEINTRESOURCE(IDI_SMALL));

    return RegisterClassExW(&wcex);
}

//
// 関数: InitInstance(HINSTANCE, int)
//
// 目的: インスタンス ハンドルを保存して、メイン ウィンドウを作成します
//
// コメント:
//
//     この関数で、グローバル変数でインスタンス ハンドルを保存し、
//     メイン プログラム ウィンドウを作成および表示します。
//
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
    hInst = hInstance; // グローバル変数にインスタンス ハンドルを格納する

    HWND hWnd = CreateWindowW(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, nullptr, nullptr,
        hInstance, nullptr);

    if (!hWnd)
    {
        return FALSE;
    }
}

```

```

    ShowWindow(hWnd, nCmdShow);
    UpdateWindow(hWnd);

    return TRUE;
}

//
// 関数: WndProc(HWND, UINT, WPARAM, LPARAM)
//
// 目的: メイン ウィンドウのメッセージを処理します。
//
// WM_COMMAND - アプリケーション メニューの処理
// WM_PAINT   - メイン ウィンドウを描画する
// WM_DESTROY - 中止メッセージを表示して戻る
//
//
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_COMMAND:
            {
                int wmId = LOWORD(wParam);
                // 選択されたメニューの解析:
                switch (wmId)
                {
                    case IDM_ABOUT:
                        DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
                        break;
                    case IDM_EXIT:
                        DestroyWindow(hWnd);
                        break;
                    default:
                        return DefWindowProc(hWnd, message, wParam, lParam);
                }
            }
        break;
        case WM_PAINT:

```

```

    {
        PAINTSTRUCT ps;
        HDC hdc = BeginPaint(hWnd, &ps);
        // TODO: HDC を使用する描画コードをここに追加してください...
        EndPaint(hWnd, &ps);
    }
    break;
case WM_DESTROY:
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

// バージョン情報ボックスのメッセージ ハンドラーです。
INT_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)
{
    UNREFERENCED_PARAMETER(lParam);
    switch (message)
    {
    case WM_INITDIALOG:
        return (INT_PTR)TRUE;

    case WM_COMMAND:
        if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)
        {
            EndDialog(hDlg, LOWORD(wParam));
            return (INT_PTR)TRUE;
        }
        break;
    }
    return (INT_PTR)FALSE;
}

```

です。これはまさに昔の C の Windows プログラミングと本質的に同じものです。糸井康孝著「猫でもわかる Windows プログラミング」や Windows プログラミングが始まった時に出版され、苦勞して学んだ「Windows プログラミング」の本を本箱から引っ張り出してプログラミングすれば、VC++ 2017 でも GUI のプログ

ラミングが出来そうです。懐かしいと言えば懐かしいですが、三十年以上も時代が遡った感じで、若い人には勧められませんが、いざとなったら VC++ 2017 でも、Windows プログラミング出来るという自信が湧いてきました。

「作って覚える Visual C# 2017 デスクトップアプリ入門」をばらばら眺めた感じでは、何も考えずに、今まで VC++ 2010 で Windows デスクトップアプリケーションを作っていた方法で、Windows デスクトップアプリケーションが作れそうです。ただ、私が興味があるプログラミングの技術をこの本から簡単に学ぶのは難しそうです。興味ないテーマも含め本全体をまじめに読んで見なければいけない。

さて、元に戻って、Kifu for Windows V7 の棋譜を元に、将棋の局面を $\text{T}_{\text{E}}\text{X}$ で表示するための $\text{T}_{\text{E}}\text{X}$ のプログラムを打ち出すプログラムですが、C# は今まで、MicroSoft 独自の普遍性のない言語だと思って勉強したことがなかったのですが、暇ですし、新しいプログラミング言語を勉強する良い機会ですから、C# で作ってみましょう。C# の「Hello world」のプログラムを見ての印象は、C++ より Java に近いように思えますが、基本部分は全部同じようなもののはずです。プログラミング言語でプログラミングした経験がないと難しいとは思いますが、私は C++ や Java で長年プログラミングした経験がありますから、本屋でたまたま見つけた、五十嵐裕貴著「スラスラわかる C#第2版」を辞書代わりに、インターネットの情報を頼りに、C++ や Java のプログラミング経験を元に、類推でプログラムを作っていきます。

まず、棋譜のファイルを読み込むことが必要です。インターネットでサンプルコードを探します。

```
using System;
using System.IO;

class TestRegex
{
    static void Main()
    {
        // ファイルにテキストを書き出し。
        using(StreamWriter w = new StreamWriter(@"test\test.txt"))
        {
            w.WriteLine("基本的に、Console クラスの文字列出力メソッドと同じ。");
            w.WriteLine("WriteLine では末尾に改行文字が加えられます。");
            int n = 5;
            double x = 3.14;
            w.Write("書式指定出力もできます → n = {0}, x = {1}", n, x);
        }
    }
}
```

```
// ファイルからテキストを読み出し。
using(StreamReader r = new StreamReader(@"test\test.txt"))
{
    string line;
    while( (line = r.ReadLine()) != null) // 1行ずつ読み出し。
    {
        Console.WriteLine(line);
    }
}
}
```

と言うのがありました。ファイル名はコマンドライン引数を使います。サンプルは

```
using System;

public class CommandLineSample
{
    public static void Main(string[] args)
    {
        for(int i=0; i<args.Length; ++i)
            Console.Write("{0}番目のコマンドライン引数は{1}です。\\n", i, args[i]);
    }
}
```

です。

従て、最初に作るプログラムは

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace TeXwriter
{
    class Program
    {
        static void Main(string[] args)
```

```

{
    // ファイルからテキストを読み出し。
    string fileName = string.Format(@"c:\駒落ち将棋集\{0}", args[0]);

    using (StreamReader r = new StreamReader(fileName))
    {
        string line;
        while ((line = r.ReadLine()) != null) // 1行ずつ読み
        {
            Console.WriteLine(line);
        }
    }
}

```

出し。

です。ここで、Kifu for Windows V7 で作ったファイルは C:\駒落ち将棋集 のフォルダーに 木村 1.kif のような名前で保存しています。コマンドラインから Texwiter 木村 1.kif のように使います。コンパイルして実行すると

```

c:\texsrc\将棋\MyShogi\TeXwriter\TeXwriter\bin\Debug>texwriter 木村1.kif
#KIF version=2.0 encoding=Shift_JIS
# ---- Kifu for Windows V7 V7.3T ?????t?@?C?? ----
?J?n?????F2019/03/20 12:16
?????F??????
?????F??
???F????`?Y
??----?w??-----?????--
1 ?U??? (71) ( 0:15/00:00:15)
2 ?V?Z?? (77) ( 0:03/00:00:03)
3 ?T?I?? (53) ( 0:02/00:00:17)
4 ?S?Z?? (47) ( 0:02/00:00:05)
5 ?T?O?? (62) ( 0:02/00:00:19)
6 ?S??? (46) ( 0:01/00:00:06)
7 ?S??? (51) ( 0:11/00:00:30)
8 ?R?Z?? (37) ( 0:05/00:00:11)
9 ?R??? (42) ( 0:03/00:00:33)
10 ?R??? (36) ( 0:02/00:00:13)
11 ?Q??? (31) ( 0:05/00:00:38)
12 ?S???? (39) ( 0:05/00:00:18)
13 ?V??? (61) ( 0:05/00:00:43)
14 ?S???? (48) ( 0:04/00:00:22)
15 ?S??? (41) ( 0:04/00:00:47)
16 ?Q?Z?? (27) ( 0:04/00:00:26)

```

です。ファイルに保存しても同じです。漢字が文字化けしています。Kifu for Windows V7 は Shift_JIS でファイルに保存しているので、Shift_JIS のファイル

を読み込むことを C# に教える必要があります。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace TeXwriter
{
    class Program
    {
        static void Main(string[] args)
        {
            // ファイルからテキストを読み出し。
            string fileName = string.Format(@"c:\駒落ち将棋集\{0}", args[0]);

            using (StreamReader r = new StreamReader(fileName,
                System.Text.Encoding.GetEncoding("shift_jis")))
            {
                string line;
                while ((line = r.ReadLine()) != null) // 1行ずつ読み
                {
                    Console.WriteLine(line);
                }
            }
        }
    }
}
```

と修正します。


```
コマンドプロンプト
c:\texsrc\将棋\MyShogi\TeXwriter\TeXwriter\bin\Debug>texwriter 木村1.kif
#KIF version=2.0 encoding=Shift_JIS
# ---- Kifu for Windows V7 V7.31 棋譜ファイル ----
開始日時: 2019/03/20 12:16
手合割: 二枚落ち
下手: 治
上手: 木村義雄
手数----指手----消費時間--
 1 6二銀(71) ( 0:15/00:00:15)
 2 7六歩(77) ( 0:03/00:00:03)
 3 5四歩(53) ( 0:02/00:00:17)
 4 4六歩(47) ( 0:02/00:00:05)
 5 5三銀(62) ( 0:02/00:00:19)
 6 4五歩(46) ( 0:01/00:00:06)
 7 4二玉(51) ( 0:11/00:00:30)
 8 3六歩(37) ( 0:05/00:00:11)
 9 3二玉(42) ( 0:03/00:00:33)
10 3五歩(36) ( 0:02/00:00:13)
11 2二銀(31) ( 0:05/00:00:38)
12 4八銀(39) ( 0:05/00:00:18)
13 7二金(61) ( 0:05/00:00:43)
14 4七銀(48) ( 0:04/00:00:22)
15 4二金(41) ( 0:04/00:00:47)
16 2六歩(27) ( 0:04/00:00:26)
```

今度は大丈夫です。

次にすべきことは、棋譜の情報をプログラム内部の情報として記憶することです。C# のデータ構造を調べます。C# には配列がありますが、前もって長さが分からないので List を使います。これは C++ の vector です。使いかたを調べます。

```
using System;
using System.Collections.Generic;

namespace Sample
{
    class Sample
    {
        static void Main()
        {
            var list = new List<string>();
            list.Add("Tokyo");
            list.Add("Osaka");
            list.Add("Nagoya");

            Console.WriteLine("[{0}]", string.Join(", ", list));
            Console.ReadKey();
        }
    }
}
```

```
}  
}  
}
```

のように使えば良いみたいです。

棋譜ファイルに保存されている情報は次のようなものです。

棋譜ファイル KIF 形式

更新 2018/9/21 柿木 義一

1. 概要

柿木将棋や「Kifu for シリーズ」の KIF 形式棋譜ファイルは、以下の形式のテキストファイルである。

2. 文字コードと改行コード

文字コードは、拡張子が ".kif" の場合は Shift-JIS、".kifu" の場合は UTF-8 である。

改行コードは、使う OS の標準的なコードとする。

Windows なら、CR,LF である。

3. 棋譜ファイルの例

```
# ---- Kifu for Windows95 V3.53 棋譜ファイル ----
```

```
開始日時：1999/07/15(木) 19:07:12
```

```
終了日時：1999/07/15(木) 19:07:17
```

```
手合割：平手
```

```
先手：先手の対局者名
```

```
後手：後手の対局者名
```

```
手数----指手-----消費時間-- # この行は、なくてもいい
```

```
1 7六歩(77) ( 0:16/00:00:16)
```

```
2 3四歩(33) ( 0:00/00:00:00)
```

```
3 中断 ( 0:03/ 0:00:19)
```

4. プログラムが読み飛ばすコメント

先頭が「#」で始まる行は、プログラムが読み飛ばすコメント行である。
どこに挿入されてもいいし、また、削除してもよい。

5. 手合割

手合割は、次の形式で、表記する。

```
手合割：平手
```

手合割は、次のいずれかでないといけない。

「平手」、「香落ち」、「右香落ち」、「角落ち」、「飛車落ち」、「飛香落ち」、「二枚落ち」、「三枚落ち」、「四枚落ち」、「五枚落ち」、「左五枚落ち」、「六枚落ち」、「八枚落ち」、「十枚落ち」、「その他」

省略したときは、平手と判断する。

6. 対局者名

「先手：」「後手：」「下手：」「上手：」以降は、対局者名を示す。

7. 対局情報

「(キーワード)：」で始まる情報を追加できる。

標準で、次の情報があり、さらにユーザが任意のものを追加できる。

「開始日時」「対局日」も可)

開始日時、終了日時は、「1999/07/15」の形式で、日付を記録しなければならない。

「終了日時」

「棋戦」

「戦型」

「表題」

「持ち時間」

「消費時間」

「場所」

「掲載」

「備考」

「先手省略名」：局面図の印刷等に使用する。

「後手省略名」：局面図の印刷等に使用する。

8. 詰将棋情報

作品番号

作品名

作者

発表誌

発表年月

出典

手数

完全性

分類

受賞

備考：対局情報としても使われる。

9. 指し手の表記方法

指し手を次の形式で表示する。

<指し手> = [<手番>]<移動先座標><駒>[<装飾子>]<移動元座標>

<手番> = "▲" | "△"

▲：先手（下手）、△：後手（上手）

※現在、手番は省略している。

<移動先座標> = <X座標><Y座標> | "同 "

<X座標> "1"～"9"：全角アラビア数字

<Y座標> "一"～"九"：全角漢数字

"同"は、直前の指し手の移動先

<駒> 駒名

玉、飛、龍、角、馬、金、銀、成銀、桂、成桂、香、成香、歩、と
龍を「竜」であらわす場合もある。

成銀を「全」、成桂を「圭」、成香を「杏」であらわす場合もある（「詰将棋パラダイス」でも使用）。

<装飾子> = ["打" | "成"]

打つ場合は、必ず「打」を付ける。

不成の場合は、"不成"を表記しない。

<移動元座標> = "(11)"~"(99)" : () で囲んだ半角 2 桁の数字

指し手には、通常のもの以外に、終局・中断時の次の表記がある

「中断」「投了」「持将棋」「千日手」「詰み」「切れ負け」「反則勝ち」「反則負け」

「切れ負け」は、時間切れで負けを示す。

「反則勝ち」は、直前の指し手が反則であることを示す。

「反則負け」は、その指し手の手番側が何らかの反則で負けたことを示す。

指し手の表記例：2 三歩成 (24)

10. 消費時間

指し手と同じ行で、指し手の後に、その指し手の消費時間を表記する。

消費時間の () 内の最初の数字は、その 1 手の消費時間（分：秒）、次の数字は、
累積の消費時間（時：分：秒）を示す。

消費時間は、省略できる。

11. 開始局面

詰将棋や任意の局面から始まる棋譜は、指し手の前に、開始局面が付く。

初期配置から始まる棋譜（駒落ちを含む）は、開始局面を省略できる。

12. 指し手のコメント

各指し手の後に「*」で始まる行があると、その指し手に対するコメントになる
（複数行可）。

初手の前のコメントは、開始局面のコメントとなる。

13. しおり

「&」で始まる行は、その前の手を指した局面に対するしおりを示す。

「&」の後は、しおり名を示す。

&<しおり名>

更に、上には記述されていないが、

55 6 五金 (64) (2:28/00:04:50)+

と

変化：55 手

と言う拡張された情報が書き込まれている。

何を List に入れるかですが、手の情報をファイルから読み込みます。棋譜ファイルから読み取るべき手の情報とは

1. 手数
2. 移動先の横の数字
3. 移動先の縦の数字
4. 駒の名称
5. 修飾子：打 or 成 or それ以外
6. 移動元の横の数字
7. 移動元の縦の数字
8. 分岐の有無（最後に + があったか？ or 分岐の初手か？ or それ以外か？）

です。どちらの手番かは手数が奇数か偶数かでわかりますから省略しています。これだけの差し手の情報があれば、ゲームが復元できます。勿論、平手か二枚落ちかといった情報もゲーム復元には必要です。これをクラスにします。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Collections.Generic;

namespace TeXwriter
{
    class Te
    {
        public int N;
        public int X;
        public int Y;
        public int koma;
        public int put;
        public int oldX;
        public int oldY;
```

```

        public int branch;
    }

    class Program
    {
        static void Main(string[] args)
        {
            var kifu = new List<Te>();

            // ファイルからテキストを読み出し。
            string fileName = string.Format(@"c:\駒落ち将棋集\{0}", args[0]);

            using (StreamReader r = new StreamReader(fileName,
                System.Text.Encoding.GetEncoding("shift_jis")))
            {
                string line;
                while ((line = r.ReadLine()) != null) // 1 行ずつ読み
                    出し。

                    {
                        Console.WriteLine(line);
                    }
            }
        }
    }
}

```

で良いみたいです。

次に、棋譜の各行から必要な情報が書いてある行を探し、その文字列を解析し、指し手の情報を List に Add していきます。そのためには string の勉強をする必要があります。第一歩として、最初に数字が書いてある行だけ打ち出すようにプログラムを修正します。これを効率的に行うには正規表現を使うのが現在では常識になっています。C# の正規表現の使いかたをインターネットの情報だけで済ますのは、情報が散発的でむづかしそうですし、五十嵐裕貴著「スラスラわかる C# 第 2 版」には基本的な C# の文法のことしか書いてなくて、正規表現については辞書代わりにはなりません。諦めて、もう一冊、出井秀行著「実践で役立つ C# プログラミングのイデオム/定石&パターン」を手に入れます。私のように色々なことを教えてもらえる友人や仲間がいないとインターネットと書籍が頼りですが、正規表現のような大きなテーマは書籍が頼りです。知りたい情報が書いている本を的確に見つけ出し手に入れなければなりません。本屋で手に取って実際に中身を確認して購入できればいいのですが、東京と違い高知のような僻地では本屋に並べ

られている本の種類・量が少なく、インターネットでの書評を頼みに、通販で購入して失敗することがよくあります。地方に住んでいると多少の失敗は仕方がないです。インターネット通販で手に入れましたが、幸い出井秀行著「実践で役立つ C# プログラミングのイデオム/定石&パターン」には差し当って欲しい情報以上の情報が記述されていました。良い本が手に入ったみたいです。

プログラミングを続けます。文字列の最初に空白文字が何個かあり、数字が続く文字列を見つけるためには、最初に

```
using System.Text.RegularExpressions;
```

を追加し、

```
string line;
var regex = new Regex(@"^\s*\d");
while ((line = r.ReadLine()) != null) // 1行ずつ読み
出し。
{
    if (regex.IsMatch(line))
        Console.WriteLine(line);
}
```

とします。これで手の情報が書かれている行だけ打ち出してくれます。分岐の情報も含まれる棋譜も対象にしたいので、

変化 : 55 手

のような行も識別できるようにします。

```
string line;
var regex = new Regex(@"^\s*\d");
var regex2 = new Regex(@"^変化");
while ((line = r.ReadLine()) != null) // 1行ずつ読み
出し。
{
    if (regex2.IsMatch(line))
        Console.WriteLine(line);
    if (regex.IsMatch(line))
        Console.WriteLine(line);
}
```

で良いです。

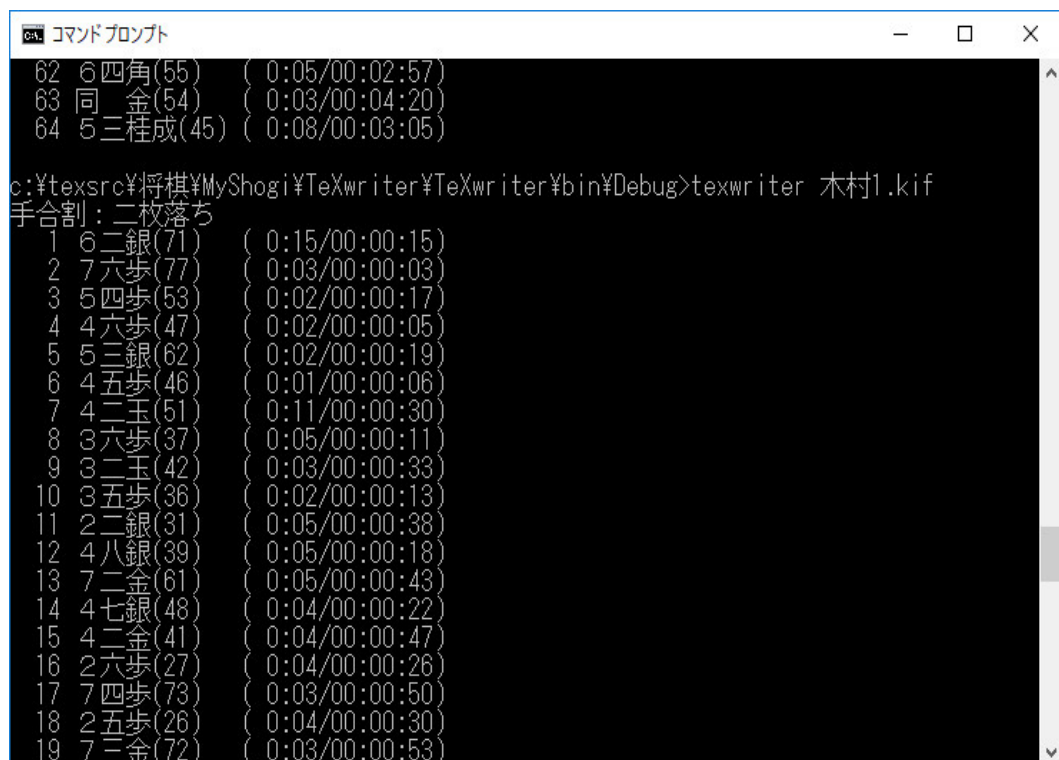
手合割 : 二枚落ち

の行も必要です。

```
string line;
var regex = new Regex(@"^\s*\d");
var regex2 = new Regex(@"^変化");
var regex3 = new Regex(@"^手合割");
while ((line = r.ReadLine()) != null) // 1行ずつ読み出し。
```

```
{
    if (regex3.IsMatch(line))
        Console.WriteLine(line);
    if (regex2.IsMatch(line))
        Console.WriteLine(line);
    if (regex.IsMatch(line))
        Console.WriteLine(line);
}
```

と修正します。



必要な行が抜き出せました。これが正規表現の凄い所です。現代的なプログラミング言語では正規表現が標準装備されています。

次に


```

20 2四歩(25)    ( 0:05/00:00:35)
21 同 歩(23)    ( 0:03/00:00:56)
22 同 飛(28)    ( 0:02/00:00:37)
23 2三歩打      ( 0:05/00:01:01)

```

のような行から手の情報を抜き出します。そのためにまず空白文字で区切られた部分を分割します。

```

if (regex.IsMatch(line))
{
    var words = line.Split(new[] { ' ', ' ', ' ', ' ' },
        StringSplitOptions.RemoveEmptyEntries);
    Console.WriteLine(words[0]+':'+words[1]
        + ':' + words[2]);
}

```

実行すると

```

コマンドプロンプト
同
5三桂成(45)
c:\texsrc\将棋\MyShogi\TeXwriter\TeXwriter\bin\Debug>texwriter 木村1.kif
手合割：二枚落ち
1:6二銀(71):
2:7六歩(77):
3:5四歩(53):
4:4六歩(47):
5:5三銀(62):
6:4五歩(46):
7:4二玉(51):
8:3六歩(37):
9:3二玉(42):
10:3五歩(36):
11:2二銀(31):
12:4八銀(39):
13:7二金(61):
14:4七銀(48):
15:4二金(41):
16:2六歩(27):
17:7四歩(73):
18:2五歩(26):
19:7三金(72):
20:2四歩(25):

```

です。上手くいっています。

手数を抜き出すには

```
int n = int.Parse(words[0]);
```

で良いです。移動先の座標を抜き出します。インターネットには正規表現を使ったアルゴリズムが掲載されていますが、ここでは単純に switch 文を使います。

22 同 飛 (28) (0:02/00:00:37)

のようなものは遡って調べないといけないので後回しにします。

```
static int getX(string word)
{
    switch (word.Substring(0,1))
    {
        case "1 ":
            return 1;
        case "2 ":
            return 2;
        case "3 ":
            return 3;
        case "4 ":
            return 4;
        case "5 ":
            return 5;
        case "6 ":
            return 6;
        case "7 ":
            return 7;
        case "8 ":
            return 8;
        case "9 ":
            return 9;
    }
    return 0;
}

static int getY(string word)
{
    Console.WriteLine("word.Substring(1, 1)="+ word.Substring(1, 1));
    switch (word.Substring(1, 1))
    {
        case "一":
            return 1;
        case "二":
```

```

        return 2;
    case "三":
        return 3;
    case "四":
        return 4;
    case "五":
        return 5;
    case "六":
        return 6;
    case "七":
        return 7;
    case "八":
        return 8;
    case "九":
        return 9;
    }
    return 0;
}

```

と定義し、

```

if (regex.IsMatch(line))
{
    var words = line.Split(new[] { ' ', ' ' },
        StringSplitOptions.RemoveEmptyEntries);
    int n = int.Parse(words[0]);
    Console.WriteLine(n);
    if (words[1] == "同")
    {
    }
    else
    {
        int x = getX(words[1]);
        int y = getY(words[1]);
        Console.WriteLine("{0} + {1}", x, y);
    }
}

```

とし、実行すれば、プログラムが正しいかどうか確かめられます。

```
cmd コマンドプロンプト
c:\texsrc\将棋\MyShogi\TeXwriter\TeXwriter\bin\Debug>texwriter 木村1.kif
手合割：二枚落ち
1
word.Substring(1, 1)=二
6 + 2
2
word.Substring(1, 1)=六
7 + 6
3
word.Substring(1, 1)=四
5 + 4
4
word.Substring(1, 1)=六
4 + 6
5
word.Substring(1, 1)=三
5 + 3
6
word.Substring(1, 1)=五
4 + 5
7
word.Substring(1, 1)=二
4 + 2
8
```

次は駒の情報の取り出しです。

41 3 三歩打 (0:05/00:01:47)

や

54 7 一歩成 (72) (0:11/00:02:19)

と言う例外的な情報もあります。

41 3 三歩打 (0:05/00:01:47)

の時には、持ち駒が変化します。駒を取った時にも持ち駒が変化しますが、それは駒を盤に配置してみないと分かりません。

54 7 一歩成 (72) (0:11/00:02:19)

の様な時には、成駒に駒名を変えることで対応します。

58 2 二成銀 (21) (0:08/00:02:08)

というのもあり、成は二通りの意味で使われています。駒名は

玉	1
飛	2
龍	3
角	4
馬	5
金	6
銀	7
成銀	8
桂	9
成桂	10
香	11
成香	12
歩	13
と	14

と数字に変換します。

```
static int getKoma(string word)
{
    int koma = 0;
    switch (word.Substring(2, 1))
    {
        case "玉":
            koma = 1; break;
        case "飛":
            koma = 2; break;
        case "龍":
            koma = 3; break;
        case "角":
            koma = 4; break;
        case "馬":
            koma = 5; break;
        case "金":
            koma = 6; break;
        case "銀":
            koma = 7; break;
        case "桂":
            koma = 9; break;
        case "香":
            koma = 11; break;
```

```

        case "歩":
            koma = 13; break;
        case "と":
            koma = 14; break;
        case "成":
            switch(word.Substring(3,1))
            {
                case "銀":
                    koma = 8; break;
                case "桂":
                    koma = 10; break;
                case "香":
                    koma = 12; break;
            }
            break;
    }
    if (word.Substring(3, 1)=="成")
    {
        koma += 1;
    }
    return koma;
}

```

と定義し、

```

if (regex.IsMatch(line))
{
    var words = line.Split(new[] { ' ', ' ', ' ' },
        StringSplitOptions.RemoveEmptyEntries);
    int n = int.Parse(words[0]);
    Console.WriteLine(n);
    if (words[1] == "同")
    {
    }
    else
    {
        int x = getX(words[1]);
        int y = getY(words[1]);
        int k = getKoma(words[1]);
        Console.WriteLine("{0}+{1}+{2}", x, y, k);
    }
}

```

```
    }
}
```

とし、実行すれば、プログラムが正しいかどうか確かめられます。

次に

50 4 五歩打 (0:09/00:02:00)

の打の情報を取り出す。

```
int x = getX(words[1]);
int y = getY(words[1]);
int k = getKoma(words[1]);
int p = 0;
if (words[1].Substring(3,1)=="打")
{
    p = 1;
}
Console.WriteLine("{0}+{1}+{2}+{3}",
    x, y, k, p);
```

で良い。

次に

57 同 金 (64) (1:22/00:04:00)

のような表現から駒名を取り出す。

```
static int getKoma(string word, int i)
{
    int koma = 0;
    switch (word.Substring(i, 1))
    {
        case "玉":
            koma = 1; break;
        case "飛":
            koma = 2; break;
        case "龍":
            koma = 3; break;
        case "角":
            koma = 4; break;
        case "馬":
```

```

        koma = 5; break;
    case "金":
        koma = 6; break;
    case "銀":
        koma = 7; break;
    case "桂":
        koma = 9; break;
    case "香":
        koma = 11; break;
    case "歩":
        koma = 13; break;
    case "と":
        koma = 14; break;
    case "成":
        switch(word.Substring(i+1,1))
        {
            case "銀":
                koma = 8; break;
            case "桂":
                koma = 10; break;
            case "香":
                koma = 12; break;
        }
        break;
    }
    if (word.Substring(i+1, 1)=="成")
    {
        koma += 1;
    }
    return koma;
}

```

と修正し、

```

if (regex.IsMatch(line))
{
    var words = line.Split(new[] { ' ', ' ', ' ', ' ' },
        StringSplitOptions.RemoveEmptyEntries);
    int n = int.Parse(words[0]);
    Console.WriteLine(n);
}

```



```

        if (words[1] == "同")
        {
            int k = getKoma(words[2], 0);
            Console.WriteLine("同 {0}", k);
        }
        else
        {
            int x = getX(words[1]);
            int y = getY(words[1]);
            int k = getKoma(words[1], 2);
            int p = 0;
            if (words[1].Substring(3,1)=="打")
            {
                p = 1;
            }
            Console.WriteLine("{0}+{1}+{2}+{3}",
                x, y, k, p);
        }
    }
}

```

とし、実行すれば、プログラムが正しいかどうか確かめられます。

次に

61 6 六金 (65) (0:05/00:02:55)

62 同 飛 (36) (0:02/00:02:50)

から、元の位置の座標を取り出します。

```

        if (words[1] == "同")
        {
            int k = getKoma(words[2], 0);
            var index = words[2].IndexOf("(");
            int ox = int.Parse(words[2].Substring(index+1, 1));
            int oy = int.Parse(words[2].Substring(index+2, 1));
            Console.WriteLine("同 {0}+{1}+{2}", k, ox, oy);
        }
        else
        {
            int x = getX(words[1]);
            int y = getY(words[1]);

```

```

int k = getKoma(words[1], 2);
int p = 0;
int ox = 0;
int oy = 0;
if (words[1].Substring(3,1)=="打")
{
    p = 1;
}
else
{
    var index = words[1].IndexOf("(");
    ox = int.Parse(words[1].Substring(index+1, 1));
    oy = int.Parse(words[1].Substring(index+2, 1));
}
Console.WriteLine("{0}+{1}+{2}+{3}+{4}+{5}",
    x, y, k, p, ox, oy);
}

```

とし、実行すれば、プログラムが正しいかどうか確かめられます。

59 5 二金 (53) (0:05/00:02:50)+

のような + があるかどうかを調べます。

```

if (regex.IsMatch(line))
{
    var words = line.Split(new[] { ' ', ' ', ' ', ' ' },
        StringSplitOptions.RemoveEmptyEntries);
    int n = int.Parse(words[0]);
    Console.WriteLine(n);
    int x = 0;
    int y = 0;
    int k = 0;
    int p = 0;
    int ox = 0;
    int oy = 0;
    if (words[1] == "同")
    {
        k = getKoma(words[2], 0);
        var index = words[2].IndexOf("(");
    }
}

```

```

        ox = int.Parse(words[2].Substring(index+1, 1));
        oy = int.Parse(words[2].Substring(index+2, 1));
    }
    else
    {
        x = getX(words[1]);
        y = getY(words[1]);
        k = getKoma(words[1], 2);
        p = 0;
        ox = 0;
        oy = 0;
        if (words[1].Substring(3,1)=="打")
        {
            p = 1;
        }
        else
        {
            var index = words[1].IndexOf("(");
            ox = int.Parse(words[1].Substring(index+1, 1));
            oy = int.Parse(words[1].Substring(index+2, 1));
        }
    }
    int b = 0;
    if (line.Contains("+"))
    {
        b = 1;
    }
    Console.WriteLine("{0}+{1}+{2}+{3}+{4}+{5}+{6}",
        x, y, k, p, ox, oy, b);
}

```

と変数の宣言を前に出すように全体を修正し、

62 同 飛 (36) (0:02/00:02:50)

のようなまだ行き先を決定できない座標は (0,0) として仮にセットするようにし、実行すれば、プログラムが正しいかどうか確かめられます。

次に

変化 : 55 手

のような行から数字を取り出し、m にセットし、次の行のデータの b を 2 とします。

ここまでで、プログラムの全体は

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Collections.Generic;
using System.Text.RegularExpressions;

namespace TeXwriter
{
    class Te
    {
        public int N;
        public int X;
        public int Y;
        public int koma;
        public int put;
        public int oldX;
        public int oldY;
        public int branch;
    }

    class Program
    {
        static void Main(string[] args)
        {
            var kifu = new List<Te>();

            // ファイルからテキストを読み出し。
            string fileName = string.Format(@"c:\駒落ち将棋集\{0}", args[0]);

            using (StreamReader r = new StreamReader(fileName,
                System.Text.Encoding.GetEncoding("shift_jis")))
            {
```

```
string line;
int m = 0;
var regex = new Regex(@"^\s*\d");
var regex2 = new Regex(@"^変化");
var regex3 = new Regex(@"^手合割");
while ((line = r.ReadLine()) != null) // 1行ずつ読み出し。

{
    int n = 0;
    int x = 0;
    int y = 0;
    int k = 0;
    int p = 0;
    int ox = 0;
    int oy = 0;
    int b = 0;
    if (regex3.IsMatch(line))
        Console.WriteLine(line);
    if (regex2.IsMatch(line))
    {
        var words = line.Split(new[] { ' ', ':', ',' },
                                StringSplitOptions.RemoveEmptyEntries);
        m = int.Parse(words[1].Substring(0, 2));
        Console.WriteLine("m={0}", m);
        continue;
    }
    if (regex.IsMatch(line))
    {
        var words = line.Split(new[] { ' ', ':' },
                                StringSplitOptions.RemoveEmptyEntries);
        n = int.Parse(words[0]);
        Console.WriteLine(n);
        if (words[1] == "同")
        {
            k = getKoma(words[2], 0);
            var index = words[2].IndexOf("(");
            ox = int.Parse(words[2].Substring(index + 1, 1));
            oy = int.Parse(words[2].Substring(index + 2, 1));
```

```

    }
    else
    {
        x = getX(words[1]);
        y = getY(words[1]);
        k = getKoma(words[1], 2);
        p = 0;
        ox = 0;
        oy = 0;
        if (words[1].Substring(3,1)=="打")
        {
            p = 1;
        }
        else
        {
            var index = words[1].IndexOf("(");
            ox = int.Parse(words[1].Substring(index + 1, 1));
            oy = int.Parse(words[1].Substring(index + 2, 1));
        }
    }
    if (line.Contains("+"))
    {
        b = 1;
    }
    if (m > 0)
    {
        b = 2;
        m = 0;
    }
    Console.WriteLine("{0}+{1}+{2}+{3}+{4}+{5}+{6}",
        x, y, k, p, ox, oy, b);
}

}

}

static int getX(string word)
{
    switch (word.Substring(0,1))
    {

```

```

        case "1 ":
            return 1;
        case "2 ":
            return 2;
        case "3 ":
            return 3;
        case "4 ":
            return 4;
        case "5 ":
            return 5;
        case "6 ":
            return 6;
        case "7 ":
            return 7;
        case "8 ":
            return 8;
        case "9 ":
            return 9;
    }
    return 0;
}

static int getY(string word)
{
    switch (word.Substring(1, 1))
    {
        case "一":
            return 1;
        case "二":
            return 2;
        case "三":
            return 3;
        case "四":
            return 4;
        case "五":
            return 5;
        case "六":
            return 6;
        case "七":
            return 7;
    }
}

```

```

        case "八":
            return 8;
        case "九":
            return 9;
    }
    return 0;
}

static int getKoma(string word, int i)
{
    int koma = 0;
    switch (word.Substring(i, 1))
    {
        case "玉":
            koma = 1; break;
        case "飛":
            koma = 2; break;
        case "龍":
            koma = 3; break;
        case "角":
            koma = 4; break;
        case "馬":
            koma = 5; break;
        case "金":
            koma = 6; break;
        case "銀":
            koma = 7; break;
        case "桂":
            koma = 9; break;
        case "香":
            koma = 11; break;
        case "歩":
            koma = 13; break;
        case "と":
            koma = 14; break;
        case "成":
            switch(word.Substring(i+1,1))
            {
                case "銀":
                    koma = 8; break;
            }
        }
    }
}

```



```

        case "桂":
            koma = 10; break;
        case "香":
            koma = 12; break;
    }
    break;
}
if (word.Substring(i+1, 1)=="成")
{
    koma += 1;
}
return koma;
}
}
}

```

となりました。

次に、行を読み込みながら棋譜の内部データ構造 Te の List kifu を作っていきます。

```

class Te
{
    public int N;
    public int X;
    public int Y;
    public int koma;
    public int put;
    public int oldX;
    public int oldY;
    public int branch;
    public Te(int n, int x, int y, int k, int p, int ox, int oy, int b)
    {
        this.N = n;
        this.X = x;
        this.Y = y;
        this.koma = k;
        this.put = p;
        this.oldX = ox;
        this.oldY = oy;
        this.branch = b;
    }
}

```

```
    }
}
```

とクラス Te のコンストラクターを追加します。

棋譜の List<Te> kifu をセットしながら、

58 同 角 (66) (0:08/00:02:46)

のような場合の行き先の座標を求めてセットします。

```
Console.WriteLine("{0}+{1}+{2}+{3}+{4}+{5}+{6}",
    x, y, k, p, ox, oy, b);
```

の部分

```
// 同?の行き先の座標を求める
if (x==0)
{
    int start = kifu.Count();
    if (b == 2) // 分岐点
    {
        int len = kifu.Count();
        for (start = len - 1; start >= 0; start--)
        {
            Te tt = kifu[start];
            if (tt.N == n) break;
        }
    }
    for (int i = start - 1; i >= 0; i--)
    {
        Te tt = kifu[i];
        if (tt.branch == 2) // 分岐点
        {
            if (tt.X > 0)
            {
                x = tt.X;
                y = tt.Y;
                break;
            }
            for (int j = i - 1; j >= 0; j--)
            {
                Te t2 = kifu[j];
```

```

        if (t2.N == tt.N)
        {
            i = j;
            break;
        }
    }
    continue;
}
else
{
    if (tt.X > 0)
    {
        x = tt.X;
        y = tt.Y;
        break;
    }
}
}
}
Te t = new Te(n, x, y, k, p, ox, oy, b);
kifu.Add(t);

```

で置き換えます。

コマンドラインで、

TeXwriter 木村1.kif 66 4 四飛

や

TeXwriter 木村1.kif 64 4 三步成

や

TeXwriter 木村1.kif 63 3 三步打

とした時、その局面の棋譜の T_EX ファイルを表示するようにします。但し、

TeXwriter 木村1.kif 61 同 歩

とはせず、行き先を表示するようにします。

そのためには、まずその手に至る経路を抜き出す必要があります。

```

string target = args[2];
int tn = int.Parse(args[1]);
int tx = getX(target);
int ty = getY(target);
int tk = getKoma(target, 2);
int tp = 0;
if (target.Length == 4)
{
    if (target.Substring(3, 1) == "打")
    {
        tp = 1;
    }
}
//Console.WriteLine("{0},{1} koma={2} p={3} : {4}",
    tx, ty, tk, tp, tn);
var moves = new List<Te>();
int sn = -1;
for (int i=0; i<kifu.Count(); i++)
{
    Te move = kifu[i];
    if (move.N != tn) continue;
    if (move.X != tx) continue;
    if (move.Y != ty) continue;
    if (move.koma != tk) continue;
    if (move.put != tp) continue;
    sn = i;
    break;
}
if (sn == -1)
{
    Console.WriteLine("引数 {0} or {1}が間違っています!",
        args[1], args[2]);
    return;
}
for (int i = sn; i >= 0; i--)
{
    Te tt = kifu[i];
    moves.Add(tt);
    if (tt.branch == 2) // 分岐点

```

```

    {
        for (int j = i - 1; j >= 0; j--)
        {
            Te t2 = kifu[j];
            if (t2.N == tt.N)
            {
                i = j;
                break;
            }
        }
    }
}
for (int i = 0; i < moves.Count(); i++)
{
    PrintTe(moves[i]);
}

```

で、逆順で `List<Te> moves` に実際の棋譜がセットされます。

将棋盤を作り、手を進めます。最後に、`TEX` のプログラムを打ち出します。長いですが、説明なしに結果だけ示します。

```

// [手合割：二枚落ち] 等を決める
int[,] ban =
{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    {0, -11, -9, -7, -6, -1, -6, -7, -9, -11},
    {0, 0, -4, 0, 0, 0, 0, 0, -2, 0},
    {0, -13, -13, -13, -13, -13, -13, -13, -13, -13 },
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 13, 13, 13, 13, 13, 13, 13, 13, 13 },
    {0, 0, 2, 0, 0, 0, 0, 0, 4, 0},
    {0, 11, 9, 7, 6, 1, 6, 7, 9, 11},
};
int[] sente_motigoma =
{
    0, 0, 0, 0, 0, 0, 0, 0,
};
int[] gote_motigoma =

```

```

{
    0, 0, 0, 0, 0, 0, 0, 0,
};
bool komaoti = false;

if (tesuwari == "二枚落ち")
{
    ban[2, 2] = 0;
    ban[2, 8] = 0;
    komaoti = true;
}
else if (tesuwari == "飛香落ち")
{
    ban[2, 8] = 0;
    ban[1, 1] = 0;
    komaoti = true;
}
else if (tesuwari == "飛落ち")
{
    ban[2, 8] = 0;
    komaoti = true;
}
else if (tesuwari == "角落ち")
{
    ban[2, 2] = 0;
    komaoti = true;
}
else if (tesuwari == "香落ち")
{
    ban[1, 1] = 0;
    komaoti = true;
}
if (komaoti == true)
{
    // moves に応じて手を進める
    for (int i = moves.Count() - 1; i >= 0; i--)
    {
        Te t = moves[i];
        if (t.N % 2 == 0) // 後手
    }
}

```

```

{
    if (t.put == 1)
    {
        ban[t.Y, t.X] = t.koma;
        switch (t.koma)
        {
            case 2:
                gote_motigoma[0] -= 1;
                break;
            case 4:
                gote_motigoma[1] -= 1;
                break;
            case 6:
                gote_motigoma[2] -= 1;
                break;
            case 7:
                gote_motigoma[3] -= 1;
                break;
            case 9:
                gote_motigoma[4] -= 1;
                break;
            case 11:
                gote_motigoma[5] -= 1;
                break;
            case 13:
                gote_motigoma[6] -= 1;
                break;
        }
    }
    else
    {
        if (ban[t.Y, t.X] != 0)
        {
            switch (ban[t.Y, t.X])
            {
                case -2:
                    gote_motigoma[0] += 1;
                    break;
                case -4:

```

```

        gote_motigoma[1] += 1;
        break;
case -6:
        gote_motigoma[2] += 1;
        break;
case -7:
        gote_motigoma[3] += 1;
        break;
case -9:
        gote_motigoma[4] += 1;
        break;
case -11:
        gote_motigoma[5] += 1;
        break;
case -13:
        gote_motigoma[6] += 1;
        break;
case -3:
        gote_motigoma[0] += 1;
        break;
case -5:
        gote_motigoma[1] += 1;
        break;
case -8:
        gote_motigoma[3] += 1;
        break;
case -10:
        gote_motigoma[4] += 1;
        break;
case -12:
        gote_motigoma[5] += 1;
        break;
case -14:
        gote_motigoma[6] += 1;
        break;
    }
}
ban[t.Y, t.X] = t.koma;
ban[t.oldY, t.oldX] = 0;

```



```

    }
}
else // 先手
{
    if (t.put == 1)
    {
        ban[t.Y, t.X] = -t.koma;
        switch (t.koma)
        {
            case 2:
                sente_motigoma[0] -= 1;
                break;
            case 4:
                sente_motigoma[1] -= 1;
                break;
            case 6:
                sente_motigoma[2] -= 1;
                break;
            case 7:
                sente_motigoma[3] -= 1;
                break;
            case 9:
                sente_motigoma[4] -= 1;
                break;
            case 11:
                sente_motigoma[5] -= 1;
                break;
            case 13:
                sente_motigoma[6] -= 1;
                break;
        }
    }
    else
    {
        if (ban[t.Y, t.X] != 0)
        {
            switch (ban[t.Y, t.X])
            {
                case 2:

```

```

        sente_motigoma[0] += 1;
        break;
case 4:
        sente_motigoma[1] += 1;
        break;
case 6:
        sente_motigoma[2] += 1;
        break;
case 7:
        sente_motigoma[3] += 1;
        break;
case 9:
        sente_motigoma[4] += 1;
        break;
case 11:
        sente_motigoma[5] += 1;
        break;
case 13:
        sente_motigoma[6] += 1;
        break;
case 3:
        sente_motigoma[0] += 1;
        break;
case 5:
        sente_motigoma[1] += 1;
        break;
case 8:
        sente_motigoma[3] += 1;
        break;
case 10:
        sente_motigoma[4] += 1;
        break;
case 12:
        sente_motigoma[5] += 1;
        break;
case 14:
        sente_motigoma[6] += 1;
        break;
}

```

```

        }
        ban[t.Y, t.X] = -t.koma;
        ban[t.oldY, t.oldX] = 0;
    }
}

// \TeX の棋譜を出力
Console.WriteLine("\\begin{center}");
Console.WriteLine("\\begin{myshogi}[1.8]");
Console.WriteLine("  \\banmen");
var motigoma = "  \\mochigoma{\\sente}";
for (int k=0; k<7; k++)
{
    motigoma += " {";
    motigoma += gote_motigoma[k].ToString();
    motigoma += "}";
}
Console.WriteLine(motigoma);
motigoma = "  \\mochigoma{\\gote}";
for (int k = 0; k < 7; k++)
{
    motigoma += " {";
    motigoma += sente_motigoma[k].ToString();
    motigoma += "}";
}
Console.WriteLine(motigoma);
for (int i=1; i<10; i++)
{
    for (int k=1; k<10; k++)
    {
        if (ban[i, k] != 0)
        {
            var te = "  \\koma{";
            te += k.ToString();
            te += i.ToString();
            te += "} ";
            te += ChangeKoma(ban[i, k]);
            Console.WriteLine(te);
        }
    }
}

```

```

    }
}
Console.WriteLine("\\end{myshogi}");
Console.WriteLine("\\end{center}");
}
else // 平手
{
    // moves に応じて手を進める
    for (int i = moves.Count() - 1; i >= 0; i--)
    {
        Te t = moves[i];
        if (t.N % 2 == 0) // 後手
        {
            if (t.put == 1)
            {
                ban[t.Y, t.X] = -t.koma;
                switch (t.koma)
                {
                    case 2:
                        gote_motigoma[0] -= 1;
                        break;
                    case 4:
                        gote_motigoma[1] -= 1;
                        break;
                    case 6:
                        gote_motigoma[2] -= 1;
                        break;
                    case 7:
                        gote_motigoma[3] -= 1;
                        break;
                    case 9:
                        gote_motigoma[4] -= 1;
                        break;
                    case 11:
                        gote_motigoma[5] -= 1;
                        break;
                    case 13:
                        gote_motigoma[6] -= 1;
                        break;
                }
            }
        }
    }
}

```

```

    }
}
else
{
    if (ban[t.Y, t.X] != 0)
    {
        switch (ban[t.Y, t.X])
        {
            case 2:
                gote_motigoma[0] += 1;
                break;
            case 4:
                gote_motigoma[1] += 1;
                break;
            case 6:
                gote_motigoma[2] += 1;
                break;
            case 7:
                gote_motigoma[3] += 1;
                break;
            case 9:
                gote_motigoma[4] += 1;
                break;
            case 11:
                gote_motigoma[5] += 1;
                break;
            case 13:
                gote_motigoma[6] += 1;
                break;
            case 3:
                gote_motigoma[0] += 1;
                break;
            case 5:
                gote_motigoma[1] += 1;
                break;
            case 8:
                gote_motigoma[3] += 1;
                break;
            case 10:

```

```

        gote_motigoma[4] += 1;
        break;
    case 12:
        gote_motigoma[5] += 1;
        break;
    case 14:
        gote_motigoma[6] += 1;
        break;
    }
}
ban[t.Y, t.X] = -t.koma;
ban[t.oldY, t.oldX] = 0;
}
}
else // 先手
{
    if (t.put == 1)
    {
        ban[t.Y, t.X] = t.koma;
        switch (t.koma)
        {
            case 2:
                sente_motigoma[0] -= 1;
                break;
            case 4:
                sente_motigoma[1] -= 1;
                break;
            case 6:
                sente_motigoma[2] -= 1;
                break;
            case 7:
                sente_motigoma[3] -= 1;
                break;
            case 9:
                sente_motigoma[4] -= 1;
                break;
            case 11:
                sente_motigoma[5] -= 1;
                break;

```

```

        case 13:
            sente_motigoma[6] -= 1;
            break;
    }
}
else
{
    if (ban[t.Y, t.X] != 0)
    {
        switch (ban[t.Y, t.X])
        {
            case -2:
                sente_motigoma[0] += 1;
                break;
            case -4:
                sente_motigoma[1] += 1;
                break;
            case -6:
                sente_motigoma[2] += 1;
                break;
            case -7:
                sente_motigoma[3] += 1;
                break;
            case -9:
                sente_motigoma[4] += 1;
                break;
            case -11:
                sente_motigoma[5] += 1;
                break;
            case -13:
                sente_motigoma[6] += 1;
                break;
            case -3:
                sente_motigoma[0] += 1;
                break;
            case -5:
                sente_motigoma[1] += 1;
                break;
            case -8:

```

```

        sente_motigoma[3] += 1;
        break;
    case -10:
        sente_motigoma[4] += 1;
        break;
    case -12:
        sente_motigoma[5] += 1;
        break;
    case -14:
        sente_motigoma[6] += 1;
        break;
    }
}
ban[t.Y, t.X] = t.koma;
ban[t.oldY, t.oldX] = 0;
}
}
}
/*
for (int i=1; i<10; i++)
{
    string temp = "";
    for (int k=1; k<10; k++)
    {
        temp += ban[i, k].ToString();
        temp += " ";
    }
    Console.WriteLine(temp);
}
*/
// \TeX の棋譜を出力
Console.WriteLine("\\begin{center}");
Console.WriteLine("\\begin{myshogi}[1.8]");
Console.WriteLine("  \\banmen");
var motigoma = "  \\mochigoma{\\sente}";
for (int k = 0; k < 7; k++)
{
    motigoma += " {";
    motigoma += sente_motigoma[k].ToString();

```



```

        motigoma += "}";
    }
    Console.WriteLine(motigoma);
    motigoma = "  \\mochigoma{\\gote}";
    for (int k = 0; k < 7; k++)
    {
        motigoma += " {";
        motigoma += gote_motigoma[k].ToString();
        motigoma += "}";
    }
    Console.WriteLine(motigoma);
    for (int i = 1; i < 10; i++)
    {
        for (int k = 1; k < 10; k++)
        {
            if (ban[i, k] != 0)
            {
                var te = "  \\koma{";
                te += k.ToString();
                te += i.ToString();
                te += "} ";
                te += ChangeKoma(ban[i, k]);
                Console.WriteLine(te);
            }
        }
    }
    Console.WriteLine("\\end{myshogi}");
    Console.WriteLine("\\end{center}");

}

static string ChangeKoma(int k)
{
    string result = "";
    if (k > 0)
    {
        result = "{\\sente} ";
    }
    else

```

```

{
    result = "{\\gote} ";
}
switch (Math.Abs(k))
{
    case 1:
        result += "{\\gyoku}";
        break;
    case 2:
        result += "{\\hi}";
        break;
    case 3:
        result += "{\\ryu}";
        break;
    case 4:
        result += "{\\kaku}";
        break;
    case 5:
        result += "{\\uma}";
        break;
    case 6:
        result += "{\\kin}";
        break;
    case 7:
        result += "{\\gin}";
        break;
    case 8:
        result += "{\\narigin}";
        break;
    case 9:
        result += "{\\kei}";
        break;
    case 10:
        result += "{\\narikei}";
        break;
    case 11:
        result += "{\\kyou}";
        break;
}

```

```

        case 12:
            result += "{\\narikyou}";
            break;
        case 13:
            result += "{\\fu}";
            break;
        case 14:
            result += "{\\tokin}";
            break;
    }
    return result;
}

static void PrintTe(Te k)
{
    Console.WriteLine("{0}:({1},{2}) k={3} p={4} ({5},{6}) b={7}",
        k.N, k.X, k.Y, k.koma, k.put, k.oldX, k.oldY, k.branch);
}

```

です。出来上がりました。プログラムの全体は

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Collections.Generic;
using System.Text.RegularExpressions;

namespace TeXwriter
{
    class Te
    {
        public int N;
        public int X;
        public int Y;
        public int koma;
        public int put;
        public int oldX;
        public int oldY;
    }
}

```

```

public int branch;
public Te(int n, int x, int y, int k, int p, int ox, int oy, int b)
{
    this.N = n;
    this.X = x;
    this.Y = y;
    this.koma = k;
    this.put = p;
    this.oldX = ox;
    this.oldY = oy;
    this.branch = b;
}
}

class Program
{
    static void Main(string[] args)
    {
        var kifu = new List<Te>();
        // ファイルからテキストを読み出し。
        string fileName = string.Format(@"c:\駒落ち将棋集\{0}", args[0]);
        string tesuwari = "二枚落ち";

        using (StreamReader r = new StreamReader(fileName,
            System.Text.Encoding.GetEncoding("shift_jis")))
        {
            string line;
            int m = 0;
            var regex = new Regex(@"^\s*\d");
            var regex2 = new Regex(@"^変化");
            var regex3 = new Regex(@"^手合割");
            while ((line = r.ReadLine()) != null) // 1行ずつ読み
                出し。

            {
                int n = 0;
                int x = 0;
                int y = 0;
                int k = 0;
                int p = 0;

```

```
int ox = 0;
int oy = 0;
int b = 0;
if (regex3.IsMatch(line))
{
    var words = line.Split(new[] { ' ', ' ', ':', ':' },
        StringSplitOptions.RemoveEmptyEntries);
    tesuwari = words[1];
    //Console.WriteLine(words[1]);
}
if (regex2.IsMatch(line))
{
    var words = line.Split(new[] { ' ', ' ', ':' },
        StringSplitOptions.RemoveEmptyEntries);
    m = int.Parse(words[1].Substring(0, 2));
    // Console.WriteLine("m={0}", m);
    continue;
}
if (regex.IsMatch(line))
{
    var words = line.Split(new[] { ' ', ' ' },
        StringSplitOptions.RemoveEmptyEntries);
    n = int.Parse(words[0]);
    //Console.WriteLine(n);
    if (words[1] == "同")
    {
        k = getKoma(words[2], 0);
        var index = words[2].IndexOf("(");
        ox = int.Parse(words[2].Substring(index + 1, 1));
        oy = int.Parse(words[2].Substring(index + 2, 1));
    }
    else
    {
        x = getX(words[1]);
        y = getY(words[1]);
        k = getKoma(words[1], 2);
        p = 0;
```

```

    ox = 0;
    oy = 0;
    if (words[1].Substring(3,1)=="打")
    {
        p = 1;
    }
    else
    {
        var index = words[1].IndexOf("(");
        ox = int.Parse(words[1].Substring(index + 1, 1));
        oy = int.Parse(words[1].Substring(index + 2, 1));
    }
}
if (line.Contains("+"))
{
    b = 1;
}
if (m > 0)
{
    b = 2;
    m = 0;
}
// 同?の行き先の座標を求める
if (x==0)
{
    int start = kifu.Count();
    if (b == 2) // 分岐点
    {
        int len = kifu.Count();
        for (start = len - 1; start >= 0; start--)
        {
            Te tt = kifu[start];
            if (tt.N == n) break;
        }
    }
    for (int i = start - 1; i >= 0; i--)
    {
        Te tt = kifu[i];
        if (tt.branch == 2) // 分岐点

```

```

        {
            if (tt.X > 0)
            {
                x = tt.X;
                y = tt.Y;
                break;
            }
            for (int j = i - 1; j >= 0; j--)
            {
                Te t2 = kifu[j];
                if (t2.N == tt.N)
                {
                    i = j;
                    break;
                }
            }
            continue;
        }
        else
        {
            if (tt.X > 0)
            {
                x = tt.X;
                y = tt.Y;
                break;
            }
        }
    }
    Te t = new Te(n, x, y, k, p, ox, oy, b);
    kifu.Add(t);
    // Console.WriteLine("{0}+{1}+{2}+{3}+{4}+{5}+{6}",
        x, y, k, p, ox, oy, b);
}
}
/*
for (int i=0; i<10; i++)
{

```

```

        PrintTe(kifu[i]);
    }
    PrintTe(kifu[20]);
    PrintTe(kifu[21]);
    PrintTe(kifu[58]);
    PrintTe(kifu[59]);
    PrintTe(kifu[78]);
    PrintTe(kifu[79]);
    PrintTe(kifu[80]);
    */
    string target = args[2];
    int tn = int.Parse(args[1]);
    int tx = getX(target);
    int ty = getY(target);
    int tk = getKoma(target, 2);
    int tp = 0;
    if (target.Length == 4)
    {
        if (target.Substring(3, 1) == "打")
        {
            tp = 1;
        }
    }
    //Console.WriteLine("{0},{1} koma={2} p={3} : {4}",
        tx, ty, tk, tp, tn);
    var moves = new List<Te>();
    int sn = -1;
    for (int i=0; i<kifu.Count(); i++)
    {
        Te move = kifu[i];
        if (move.N != tn) continue;
        if (move.X != tx) continue;
        if (move.Y != ty) continue;
        if (move.koma != tk) continue;
        if (move.put != tp) continue;
        sn = i;
        break;
    }
    if (sn == -1)

```



```

{
    Console.WriteLine("引数 {0} or {1}が間違っています!",
        args[1], args[2]);
    return;
}
for (int i = sn; i >= 0; i--)
{
    Te tt = kifu[i];
    moves.Add(tt);
    if (tt.branch == 2) // 分岐点
    {
        for (int j = i - 1; j >= 0; j--)
        {
            Te t2 = kifu[j];
            if (t2.N == tt.N)
            {
                i = j;
                break;
            }
        }
    }
}
/*
for (int i = 0; i < moves.Count(); i++)
{
    PrintTe(moves[i]);
}
*/
// PrintTe(moves[moves.Count()-1]);
// [手合割：二枚落ち] 等を決める
int[,] ban =
{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    {0, -11, -9, -7, -6, -1, -6, -7, -9, -11},
    {0, 0, -4, 0, 0, 0, 0, 0, -2, 0},
    {0, -13, -13, -13, -13, -13, -13, -13, -13, -13 },
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },

```

```

        {0, 13, 13, 13, 13, 13, 13, 13, 13, 13 },
        {0, 0, 2, 0, 0, 0, 0, 0, 4, 0},
        {0, 11, 9, 7, 6, 1, 6, 7, 9, 11},
    };
    int[] sente_motigoma =
    {
        0, 0, 0, 0, 0, 0, 0, 0,
    };
    int[] gote_motigoma =
    {
        0, 0, 0, 0, 0, 0, 0, 0,
    };
    bool komaoti = false;

    if (tesuwari == "二枚落ち")
    {
        ban[2, 2] = 0;
        ban[2, 8] = 0;
        komaoti = true;
    }
    else if (tesuwari == "飛香落ち")
    {
        ban[2, 8] = 0;
        ban[1, 1] = 0;
        komaoti = true;
    }
    else if (tesuwari == "飛落ち")
    {
        ban[2, 8] = 0;
        komaoti = true;
    }
    else if (tesuwari == "角落ち")
    {
        ban[2, 2] = 0;
        komaoti = true;
    }
    else if (tesuwari == "香落ち")
    {
        ban[1, 1] = 0;
    }

```

```

        komaoti = true;
    }
    if (komaoti == true)
    {
        // moves に応じて手を進める
        for (int i = moves.Count() - 1; i >= 0; i--)
        {
            Te t = moves[i];
            if (t.N % 2 == 0) // 後手
            {
                if (t.put == 1)
                {
                    ban[t.Y, t.X] = t.koma;
                    switch (t.koma)
                    {
                        case 2:
                            gote_motigoma[0] -= 1;
                            break;
                        case 4:
                            gote_motigoma[1] -= 1;
                            break;
                        case 6:
                            gote_motigoma[2] -= 1;
                            break;
                        case 7:
                            gote_motigoma[3] -= 1;
                            break;
                        case 9:
                            gote_motigoma[4] -= 1;
                            break;
                        case 11:
                            gote_motigoma[5] -= 1;
                            break;
                        case 13:
                            gote_motigoma[6] -= 1;
                            break;
                    }
                }
            }
            else

```

```

{
    if (ban[t.Y, t.X] != 0)
    {
        switch (ban[t.Y, t.X])
        {
            case -2:
                gote_motigoma[0] += 1;
                break;
            case -4:
                gote_motigoma[1] += 1;
                break;
            case -6:
                gote_motigoma[2] += 1;
                break;
            case -7:
                gote_motigoma[3] += 1;
                break;
            case -9:
                gote_motigoma[4] += 1;
                break;
            case -11:
                gote_motigoma[5] += 1;
                break;
            case -13:
                gote_motigoma[6] += 1;
                break;
            case -3:
                gote_motigoma[0] += 1;
                break;
            case -5:
                gote_motigoma[1] += 1;
                break;
            case -8:
                gote_motigoma[3] += 1;
                break;
            case -10:
                gote_motigoma[4] += 1;
                break;
            case -12:

```

```

        gote_motigoma[5] += 1;
        break;
    case -14:
        gote_motigoma[6] += 1;
        break;
    }
}
ban[t.Y, t.X] = t.koma;
ban[t.oldY, t.oldX] = 0;
}
}
else // 先手
{
    if (t.put == 1)
    {
        ban[t.Y, t.X] = -t.koma;
        switch (t.koma)
        {
            case 2:
                sente_motigoma[0] -= 1;
                break;
            case 4:
                sente_motigoma[1] -= 1;
                break;
            case 6:
                sente_motigoma[2] -= 1;
                break;
            case 7:
                sente_motigoma[3] -= 1;
                break;
            case 9:
                sente_motigoma[4] -= 1;
                break;
            case 11:
                sente_motigoma[5] -= 1;
                break;
            case 13:
                sente_motigoma[6] -= 1;
                break;
        }
    }
}

```

```

    }
}
else
{
    if (ban[t.Y, t.X] != 0)
    {
        switch (ban[t.Y, t.X])
        {
            case 2:
                sente_motigoma[0] += 1;
                break;
            case 4:
                sente_motigoma[1] += 1;
                break;
            case 6:
                sente_motigoma[2] += 1;
                break;
            case 7:
                sente_motigoma[3] += 1;
                break;
            case 9:
                sente_motigoma[4] += 1;
                break;
            case 11:
                sente_motigoma[5] += 1;
                break;
            case 13:
                sente_motigoma[6] += 1;
                break;
            case 3:
                sente_motigoma[0] += 1;
                break;
            case 5:
                sente_motigoma[1] += 1;
                break;
            case 8:
                sente_motigoma[3] += 1;
                break;
            case 10:

```

```

        sente_motigoma[4] += 1;
        break;
    case 12:
        sente_motigoma[5] += 1;
        break;
    case 14:
        sente_motigoma[6] += 1;
        break;
    }
}
ban[t.Y, t.X] = -t.koma;
ban[t.oldY, t.oldX] = 0;
}
}
}
// \TeX の棋譜を出力
Console.WriteLine("\\begin{center}");
Console.WriteLine("\\begin{myshogi}[1.8]");
Console.WriteLine("  \\banmen");
var motigoma = "  \\mochigoma{\\sente}";
for (int k=0; k<7; k++)
{
    motigoma += " {";
    motigoma += gote_motigoma[k].ToString();
    motigoma += "}";
}
Console.WriteLine(motigoma);
motigoma = "  \\mochigoma{\\gote}";
for (int k = 0; k < 7; k++)
{
    motigoma += " {";
    motigoma += sente_motigoma[k].ToString();
    motigoma += "}";
}
Console.WriteLine(motigoma);
for (int i=1; i<10; i++)
{
    for (int k=1; k<10; k++)
    {

```

```

        if (ban[i, k] != 0)
        {
            var te = "  \\koma{";
            te += k.ToString();
            te += i.ToString();
            te += "} ";
            te += ChangeKoma(ban[i, k]);
            Console.WriteLine(te);
        }
    }
    Console.WriteLine("\\end{myshogi}");
    Console.WriteLine("\\end{center}");
}
else // 平手
{
    // moves に応じて手を進める
    for (int i = moves.Count() - 1; i >= 0; i--)
    {
        Te t = moves[i];
        if (t.N % 2 == 0) // 後手
        {
            if (t.put == 1)
            {
                ban[t.Y, t.X] = -t.koma;
                switch (t.koma)
                {
                    case 2:
                        gote_motigoma[0] -= 1;
                        break;
                    case 4:
                        gote_motigoma[1] -= 1;
                        break;
                    case 6:
                        gote_motigoma[2] -= 1;
                        break;
                    case 7:
                        gote_motigoma[3] -= 1;
                        break;
                }
            }
        }
    }
}

```



```

        case 9:
            gote_motigoma[4] -= 1;
            break;
        case 11:
            gote_motigoma[5] -= 1;
            break;
        case 13:
            gote_motigoma[6] -= 1;
            break;
    }
}
else
{
    if (ban[t.Y, t.X] != 0)
    {
        switch (ban[t.Y, t.X])
        {
            case 2:
                gote_motigoma[0] += 1;
                break;
            case 4:
                gote_motigoma[1] += 1;
                break;
            case 6:
                gote_motigoma[2] += 1;
                break;
            case 7:
                gote_motigoma[3] += 1;
                break;
            case 9:
                gote_motigoma[4] += 1;
                break;
            case 11:
                gote_motigoma[5] += 1;
                break;
            case 13:
                gote_motigoma[6] += 1;
                break;
            case 3:

```

```

        gote_motigoma[0] += 1;
        break;
    case 5:
        gote_motigoma[1] += 1;
        break;
    case 8:
        gote_motigoma[3] += 1;
        break;
    case 10:
        gote_motigoma[4] += 1;
        break;
    case 12:
        gote_motigoma[5] += 1;
        break;
    case 14:
        gote_motigoma[6] += 1;
        break;
    }
}
ban[t.Y, t.X] = -t.koma;
ban[t.oldY, t.oldX] = 0;
}
}
else // 先手
{
    if (t.put == 1)
    {
        ban[t.Y, t.X] = t.koma;
        switch (t.koma)
        {
            case 2:
                sente_motigoma[0] -= 1;
                break;
            case 4:
                sente_motigoma[1] -= 1;
                break;
            case 6:
                sente_motigoma[2] -= 1;
                break;

```

```

        case 7:
            sente_motigoma[3] -= 1;
            break;
        case 9:
            sente_motigoma[4] -= 1;
            break;
        case 11:
            sente_motigoma[5] -= 1;
            break;
        case 13:
            sente_motigoma[6] -= 1;
            break;
    }
}
else
{
    if (ban[t.Y, t.X] != 0)
    {
        switch (ban[t.Y, t.X])
        {
            case -2:
                sente_motigoma[0] += 1;
                break;
            case -4:
                sente_motigoma[1] += 1;
                break;
            case -6:
                sente_motigoma[2] += 1;
                break;
            case -7:
                sente_motigoma[3] += 1;
                break;
            case -9:
                sente_motigoma[4] += 1;
                break;
            case -11:
                sente_motigoma[5] += 1;
                break;
            case -13:

```

```

        sente_motigoma[6] += 1;
        break;
    case -3:
        sente_motigoma[0] += 1;
        break;
    case -5:
        sente_motigoma[1] += 1;
        break;
    case -8:
        sente_motigoma[3] += 1;
        break;
    case -10:
        sente_motigoma[4] += 1;
        break;
    case -12:
        sente_motigoma[5] += 1;
        break;
    case -14:
        sente_motigoma[6] += 1;
        break;
    }
}
ban[t.Y, t.X] = t.koma;
ban[t.oldY, t.oldX] = 0;
}
}
}
/*
for (int i=1; i<10; i++)
{
    string temp = "";
    for (int k=1; k<10; k++)
    {
        temp += ban[i, k].ToString();
        temp += " ";
    }
    Console.WriteLine(temp);
}
*/

```

```

// \TeX の棋譜を出力
Console.WriteLine("\\begin{center}");
Console.WriteLine("\\begin{myshogi}[1.8]");
Console.WriteLine("  \\banmen");
var motigoma = "  \\mochigoma{\\sente}";
for (int k = 0; k < 7; k++)
{
    motigoma += " {";
    motigoma += sente_motigoma[k].ToString();
    motigoma += "}";
}
Console.WriteLine(motigoma);
motigoma = "  \\mochigoma{\\gote}";
for (int k = 0; k < 7; k++)
{
    motigoma += " {";
    motigoma += gote_motigoma[k].ToString();
    motigoma += "}";
}
Console.WriteLine(motigoma);
for (int i = 1; i < 10; i++)
{
    for (int k = 1; k < 10; k++)
    {
        if (ban[i, k] != 0)
        {
            var te = "  \\koma{";
            te += k.ToString();
            te += i.ToString();
            te += "} ";
            te += ChangeKoma(ban[i, k]);
            Console.WriteLine(te);
        }
    }
}
Console.WriteLine("\\end{myshogi}");
Console.WriteLine("\\end{center}");
}

```

```

}
static string ChangeKoma(int k)
{
    string result = "";
    if (k > 0)
    {
        result = "{\\sente} ";
    }
    else
    {
        result = "{\\gote} ";
    }

    switch (Math.Abs(k))
    {
        case 1:
            result += "{\\gyoku}";
            break;
        case 2:
            result += "{\\hi}";
            break;
        case 3:
            result += "{\\ryu}";
            break;
        case 4:
            result += "{\\kaku}";
            break;
        case 5:
            result += "{\\uma}";
            break;
        case 6:
            result += "{\\kin}";
            break;
        case 7:
            result += "{\\gin}";
            break;
        case 8:
            result += "{\\narigin}";
            break;
    }
}

```

```

        case 9:
            result += "{\\kei}";
            break;
        case 10:
            result += "{\\narikei}";
            break;
        case 11:
            result += "{\\kyou}";
            break;
        case 12:
            result += "{\\narikyou}";
            break;
        case 13:
            result += "{\\fu}";
            break;
        case 14:
            result += "{\\tokin}";
            break;
    }
    return result;
}

static void PrintTe(Te k)
{
    Console.WriteLine("{0}:({1},{2}) k={3} p={4} ({5},{6}) b={7}",
        k.N, k.X, k.Y, k.koma, k.put, k.oldX, k.oldY, k.branch);
}

static int getX(string word)
{
    switch (word.Substring(0,1))
    {
        case "1 ":
            return 1;
        case "2 ":
            return 2;
        case "3 ":
            return 3;
        case "4 ":
            return 4;
        case "5 ":

```

```

        return 5;
    case "6 ":
        return 6;
    case "7 ":
        return 7;
    case "8 ":
        return 8;
    case "9 ":
        return 9;
    }
    return 0;
}

static int getY(string word)
{
    switch (word.Substring(1, 1))
    {
        case "一":
            return 1;
        case "二":
            return 2;
        case "三":
            return 3;
        case "四":
            return 4;
        case "五":
            return 5;
        case "六":
            return 6;
        case "七":
            return 7;
        case "八":
            return 8;
        case "九":
            return 9;
    }
    return 0;
}

static int getKoma(string word, int i)
{

```



```

int koma = 0;
switch (word.Substring(i, 1))
{
    case "玉":
        koma = 1; break;
    case "飛":
        koma = 2; break;
    case "龍":
        koma = 3; break;
    case "角":
        koma = 4; break;
    case "馬":
        koma = 5; break;
    case "金":
        koma = 6; break;
    case "銀":
        koma = 7; break;
    case "桂":
        koma = 9; break;
    case "香":
        koma = 11; break;
    case "歩":
        koma = 13; break;
    case "と":
        koma = 14; break;
    case "成":
        switch(word.Substring(i+1,1))
        {
            case "銀":
                koma = 8; break;
            case "桂":
                koma = 10; break;
            case "香":
                koma = 12; break;
        }
        break;
}
if (word.Length>3 && word.Substring(i+1, 1)=="成")
{

```

```

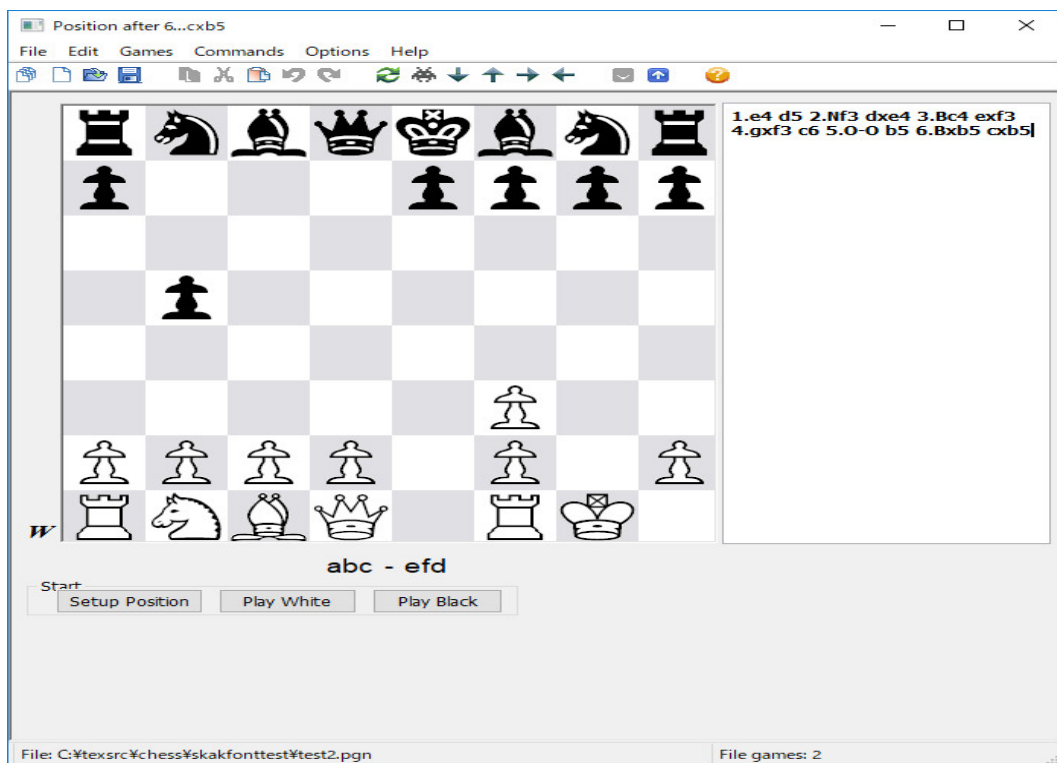
        koma += 1;
    }
    return koma;
}
}
}

```

です。

チェスの局面をカードに印刷するための $\text{T}_{\text{E}}\text{X}$ のプログラムを作るためのソフトも作ってみよう。チェスのフリーソフトには Stockfish というのがあります。対戦もチェスの棋譜づくりも局面の解析もできる優れものです。

出鱈目に動かすと



なります。棋譜を保存すると

```

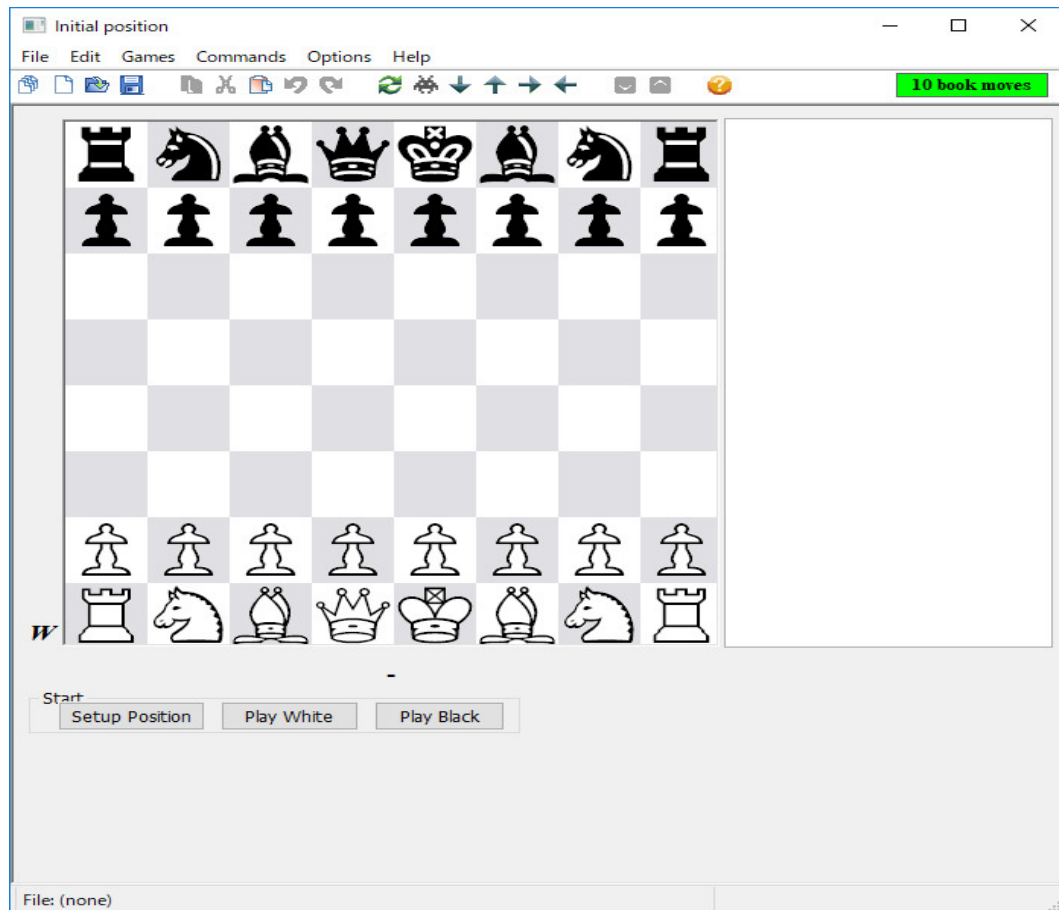
[Event "?"]
[Site "?"]
[Date "2019.03.24"]
[Round "?"]
[White "abc"]
[Black "efd"]
[Result "*"]

```

1. e4 d5 2. Nf3 dxe4 3. Bc4 exf3 4. gxf3 c6 5. 0-0 b5 6. Bxb5 cxb5 *

です。これを使って、局面をカードに印刷するための補助ソフトを作るのは元の位置の座標がないので少し厄介です。

プログラミングの勉強の為にチェスの棋譜を編集するソフトを C# で作ってみましょう。まず、画像の表示の方法を知らなければならない。



の画像を描く C# のプログラムを作ってみる。

デフォルトのデスクトップアプリケーションは

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```

using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}

```

のようなプログラムになっている。ここに PictureBox を配置し、

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)

```

```

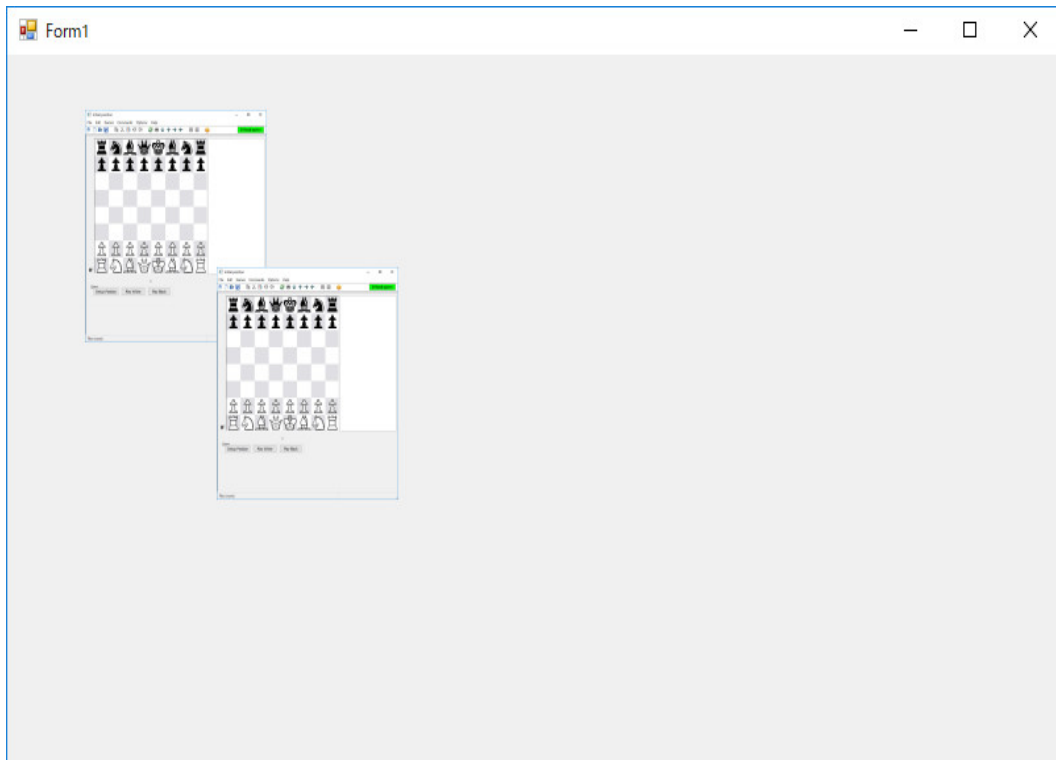
{
    //描画先とする Image オブジェクトを作成する
    Bitmap canvas = new Bitmap(pictureBox1.Width, pictureBox1.Height);
    //Image オブジェクトの Graphics オブジェクトを作成する
    Graphics g = Graphics.FromImage(canvas);

    //画像ファイルを読み込んで、Image オブジェクトとして取得する
    Image img = Image.FromFile(@"test.jpg");
    //画像を canvas の座標 (0, 0) の位置に描画する
    g.DrawImage(img, 0, 0, img.Width/5, img.Height/5);
    g.DrawImage(img, 100, 100, img.Width / 5, img.Height / 5);
    //Image オブジェクトのリソースを解放する
    img.Dispose();

    //Graphics オブジェクトのリソースを解放する
    g.Dispose();
    //PictureBox1 に表示する
    pictureBox1.Image = canvas;
}
}
}

```

と書き込むと



と PictureBox に画像を表示してくれます。チェスの skak のフォントを使って、チェスの駒の画像を作れば、なんとかチェスのそれなりのプログラムが作れそうです。「作って覚える Visual C# 2017 デスクトップアプリ入門」が役に立つかどうか分かりませんが、ちょっと眺めただけでは画像の表示の仕方が分からなかったもので、インターネットの断片的な情報と VC++ でのプログラミングの経験からによる類推で何とかここまでわかりました。これはフォームが作られるときに、画像を表示しているので、欲しいプログラミングの方法ではないが、とにかく画像が表示できた。

将棋のソフトのようにスラスラとは作れませんが、少しずつ実験しながら作っていけば、なんとかなるでしょう。

VC++ 2010 では、Form に PictureBox を配置し、PictureBox のイベントから一番下にある paint をダブルクリックし、

```
private: System::Void pictureBox1_Paint(System::Object^ sender,
    System::Windows::Forms::PaintEventArgs^ e) {
    Graphics^ g = e->Graphics;
    Pen^ pen = gcnew Pen(Color::Blue);
    double pi = Math::PI;
    g->DrawEllipse(pen, 50, 50, 260, 260);
    for (double t=0; t<2*pi; t += pi/80) {
        g->DrawLine(pen, (int)(180+130*cos(t)), (int)(180-130*sin(t)),
```

```

        (int)(180+130*cos(2*t)), (int)(180-130*sin(2*t)));
    }
}

```

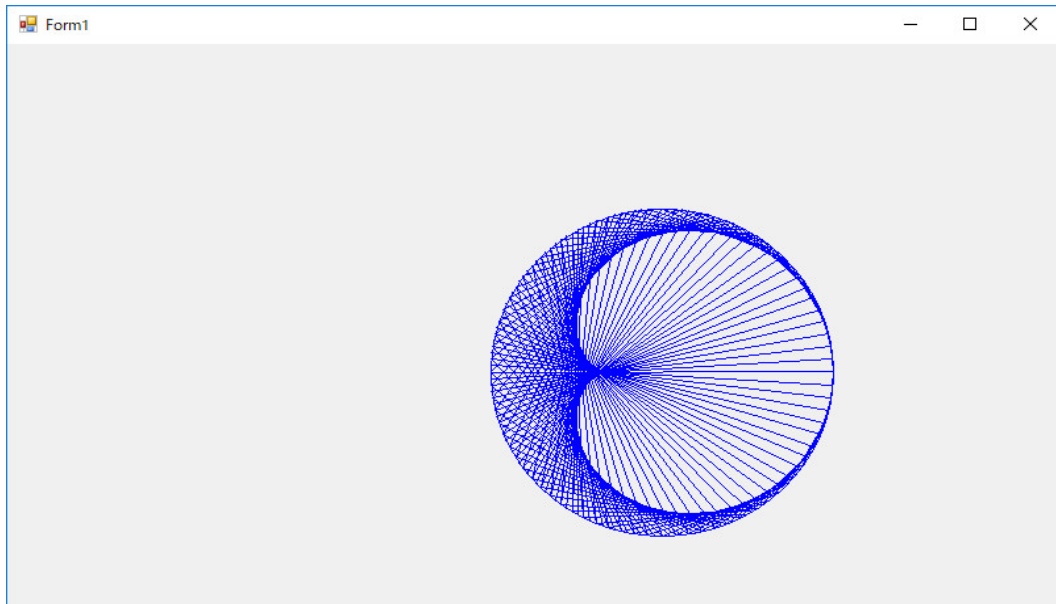
と中身を打ち込めばよかったが、C# でも同様に、Form に PictureBox を配置し、PictureBox のイベントから一番下にある paint をダブルクリックし、

```

private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Pen pen = new Pen(Color.Blue);
    double pi = Math.PI;
    g.DrawEllipse(pen, 50, 50, 260, 260);
    for (double t = 0; t < 2 * pi; t += pi / 80)
    {
        g.DrawLine(pen, (int)(180 + 130 * Math.Cos(t)),
                    (int)(180 - 130 * Math.Sin(t)),
                    (int)(180 + 130 * Math.Cos(2 * t)),
                    (int)(180 - 130 * Math.Sin(2 * t)));
    }
}

```

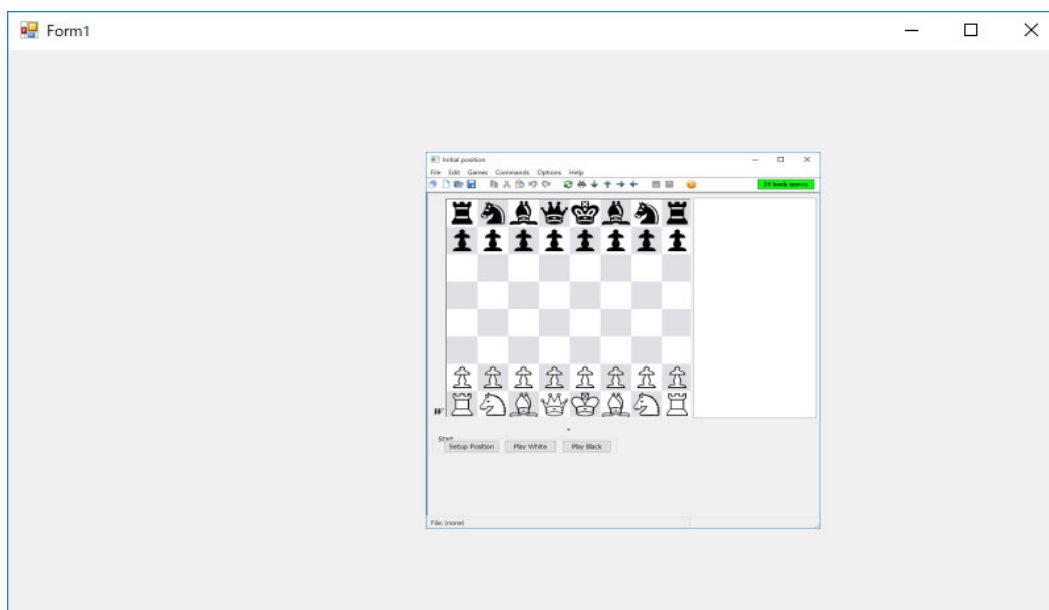
とプログラミングすれば、



を描いてくれます。VC++ のプログラムと C# のプログラムと一対一に対応している。基本的なグラフィックスのプログラミングは何とかなりそうです。

```
private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Image image = Image.FromFile("kifu21.jpg");
    g.DrawImage(image, 0, 0, 300, 300);
}
```

で、



を描いてくれます。これで良いなら VC++ のプログラミングの経験が生かせます。K とか Q とかの文字でなく、画像でコマを表示したいです。ソフトを公表するなら自分で画像を metafont などで作図すべきですが、自分で使うだけですから、skak のフォントを使わしてもらうことにしましょう。

```
\documentclass[a4paper,10pt]{jarticle}
```

```
\font\remark=skak30 scaled\magstep5
```

```
\pagestyle{plain}
```

```
%%%%%%%% TEXT START %%%%
```

```
\begin{document}
```

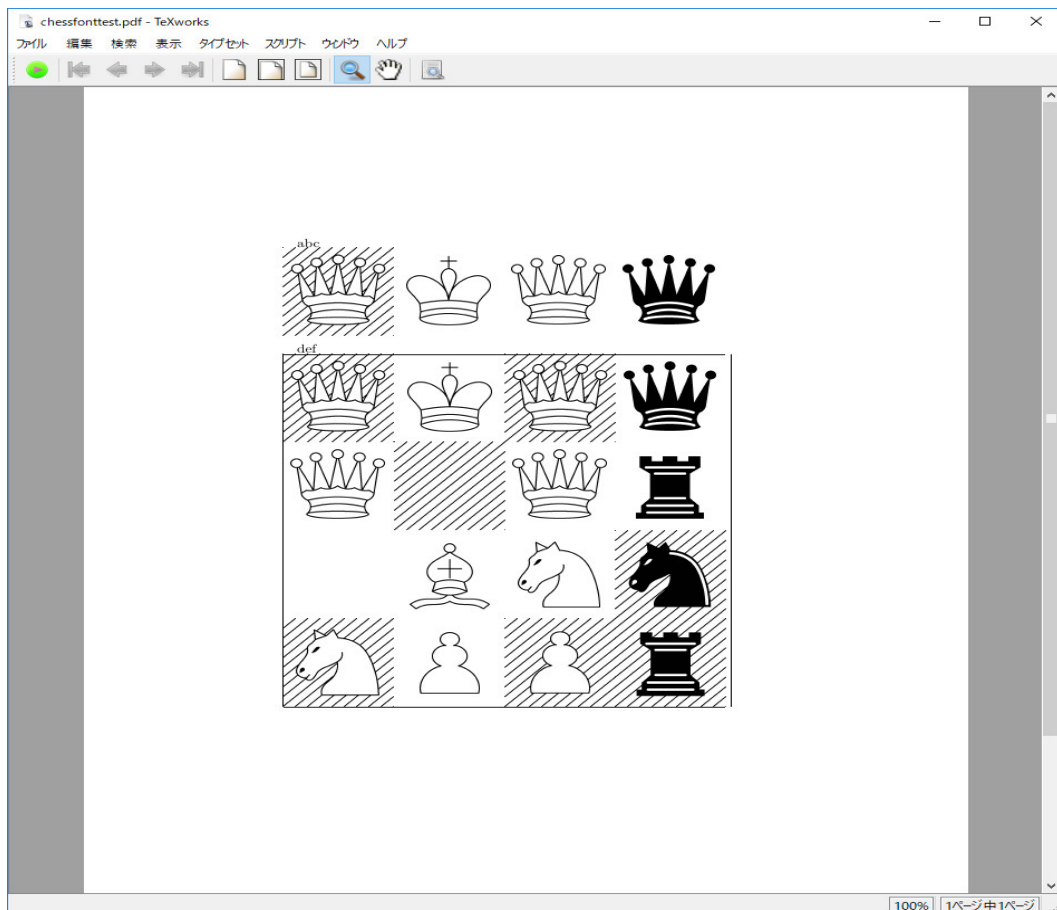
```
abc
```

```
\noindent{\remark \char76 \char75 Q q}
```


def

```
\hbox{\vrule
\vbox{\hrule \hbox{\remark \char76 \char75 L q }
\hbox{\remark \char81 \char90 Q r }
\hbox{\remark \char48 \char66 N m }
\hbox{\remark \char77 \char80 O s }
\hrule }
\vrule }
\end{document}
```

で



を描いてくれます。これをを GIMP で切り取り画像を作れば良いです。metafont や metapost（カラーの絵が描けます）の使いかたを勉強すれば、絵心がなくても（？）きれいな絵が描けます。例えば、トランプの自前の画像が欲しければ、これを勉強するのも一つの方法です。metapost の情報はインターネットで得られます。

画像とフォントの対応関係は

O : light square [48]
Z : dark square [90]
K : white king on light square [75]
k : black king on light square [107]
J : white king on dark square [75]
j : black king on dark square [106]
Q : white queen on light square [81]
q : black queen on light square [113]
L : white queen on dark square [76]
l : black queen on dark square [108]
R : white rook on light square [82]
r : black rook on light square [114]
S : white rook on dark square [83]
s : black rook on dark square [115]
B : white bishop on light square [66]
b : black bishop on light square [98]
A : white bishop on dark square [65]
a : black bishop on dark square [97]
N : white knight on light square [78]
n : black knight on light square [110]
M : white knight on dark square [77]
m : black knight on dark square [109]
P : white pawn on light square [80]
p : black pawn on light square [112]
O : white pawn on dark square [79]
o : black pawn on dark square [111]

です。これらを表示し、画像を作ります。

```
\documentclass[a4paper,10pt]{jarticle}
```

```
\font\remark=skak30 scaled\magstep5
```

```
\pagestyle{plain}
```

```
%%%%%%%% TEXT START %%%%%%%%%
```

```
\begin{document}
```

```
\noindent{\remark O Z}
```

```
\noindent{\remark K k J j}
```

```
\noindent{\remark Q q L l}
```

```
\noindent{\remark R r S s}
```

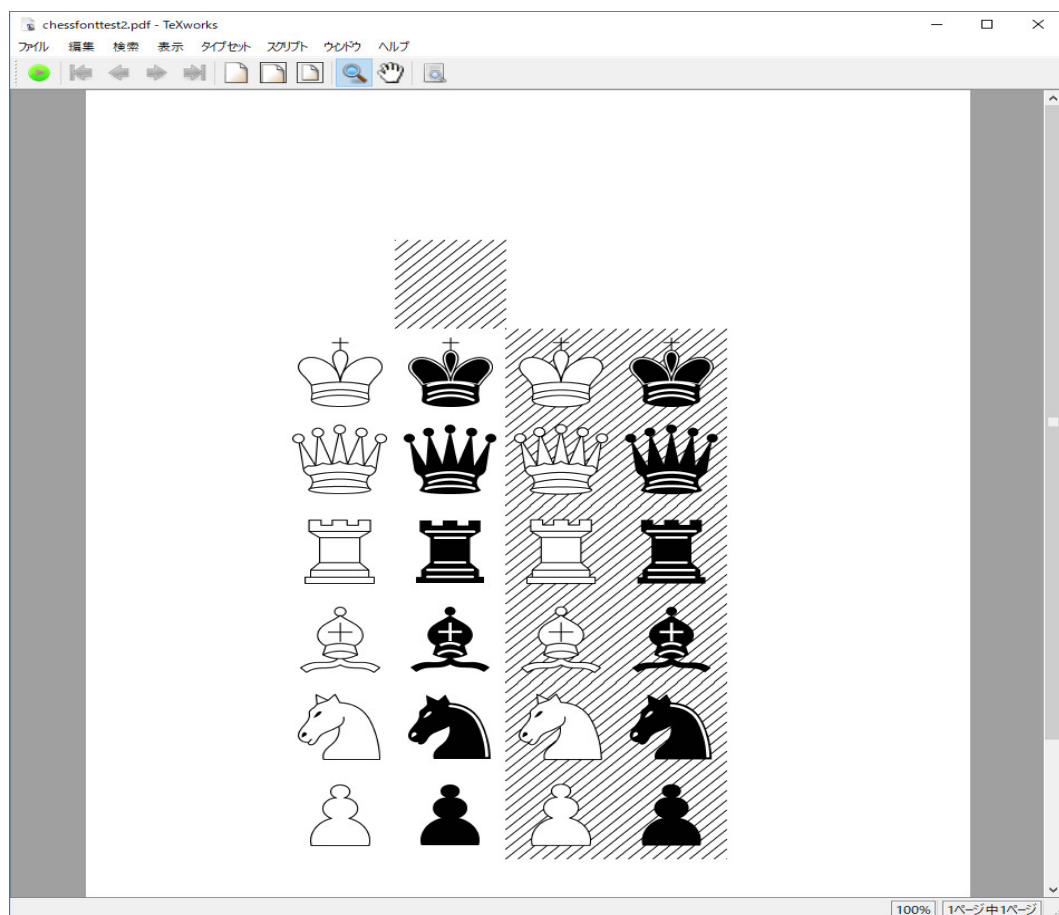
```
\noindent{\remark B b A a}
```

```
\noindent{\remark N n M m}
```

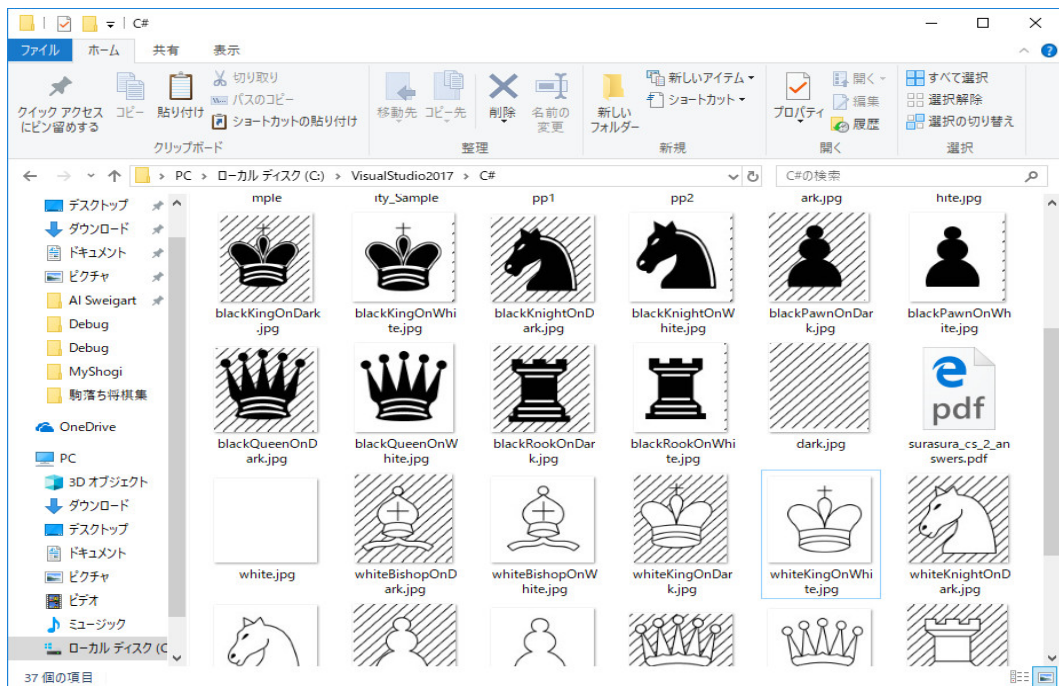
```
\noindent{\remark P p O o}
```

```
\end{document}
```

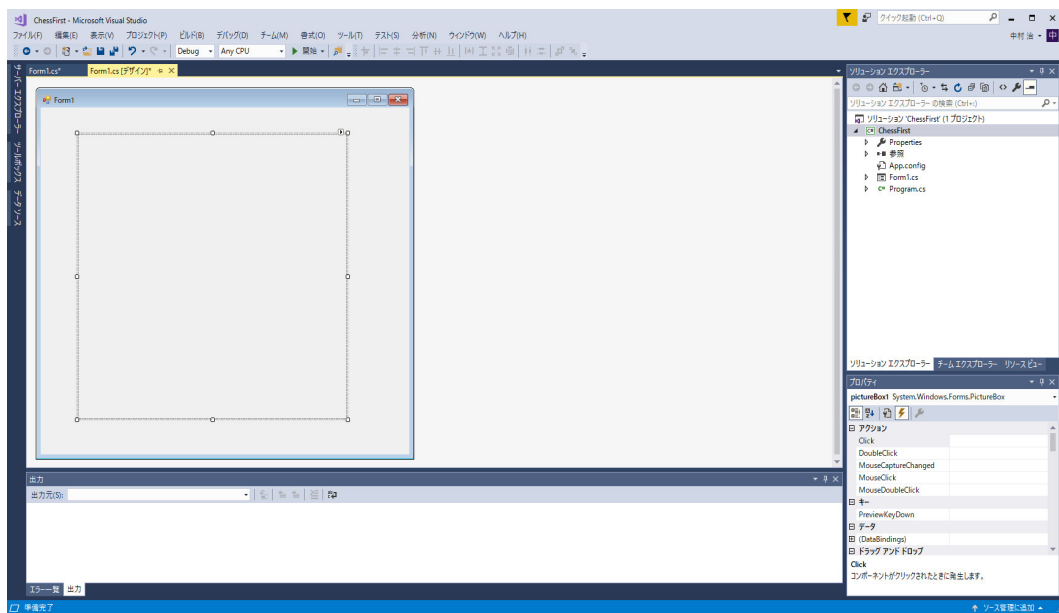
で



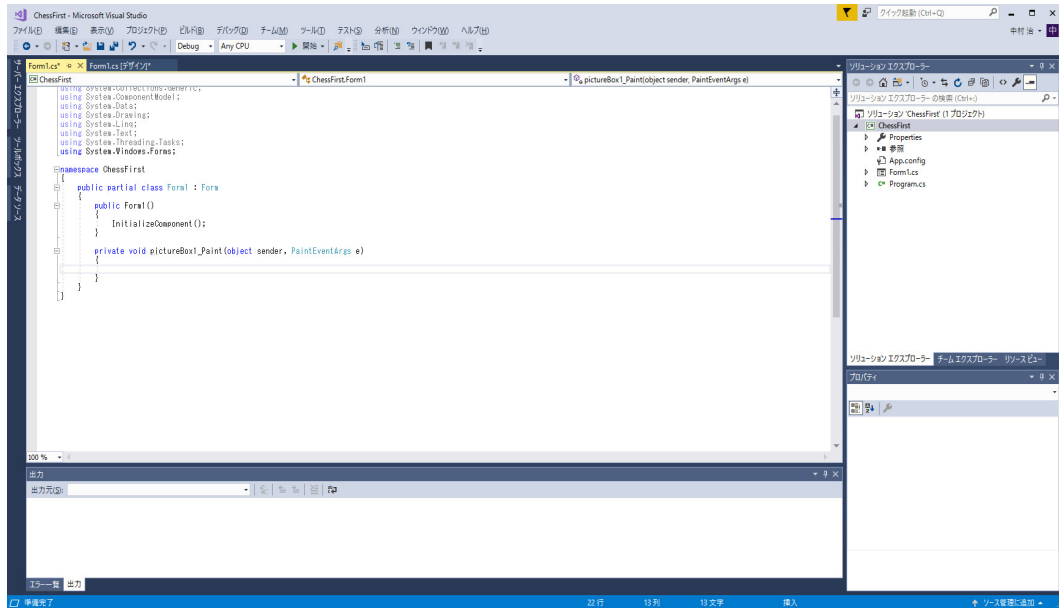
を描き、これでコマの画像を作ります。



まず、チェスの初期画面を描くプログラムを作ってみます。



のように、440 × 440 の PictureBox を貼り付けます。PictureBox のイベントから paint を選択肢、ダブルクリックします。



と Form1.cs のファイルが表示されます。ここにプログラムを書き込みます。

まず、駒のクラスを作ります。何が必要か分からないので、差し当って画像の Image だけ持たします。

色々試行錯誤した結果、Form1.cs のファイルに

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ChessFirst
{
    public partial class Form1 : Form
    {
        public class Koma
        {
            public Image image;
```

```

        public Koma(Image img)
        {
            this.image = img;
        }
    }

    public Koma koma = null;

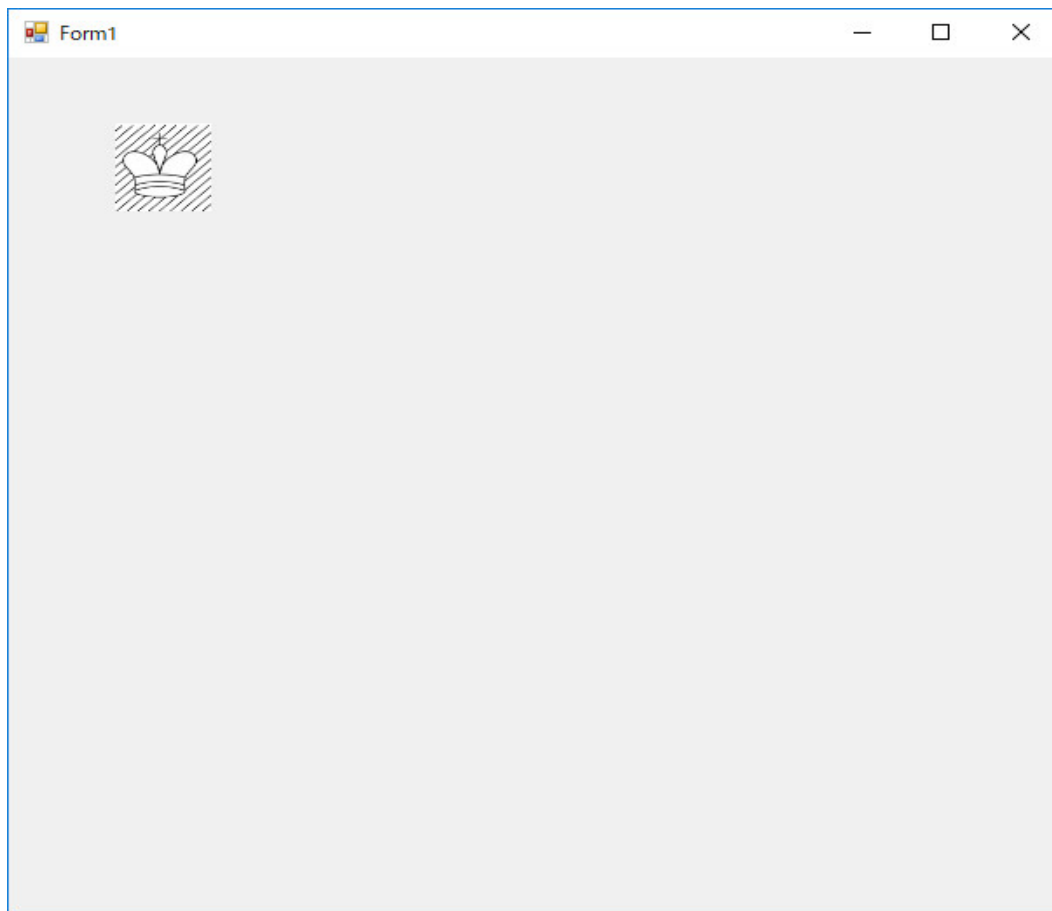
    public Form1()
    {
        InitializeComponent();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        Image img = Image.FromFile("whiteKingOnDark.jpg");
        koma = new Koma(img);

    }
    private void pictureBox1_Paint(object sender, PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        g.DrawImage(koma.image, 0, 0, 55, 55);
    }

    }
}

```

と書き込めば、



を描いてくれます。本やインターネットでこのようなプログラムをそのものを見つけたのは至難のことです。プログラムが間違っていればコンピュータが教えてくれます。Angela Duckworth 著「GRIT Why passion and resilience are the secrets to success」（翻訳：神崎朗子訳「やりぬく力」）にあるように

What is deliberate practice?

- Setting a specific stretch goal
- Concentrating 100 %
- Getting immediate and informative feedback
- Reflecting, refine, repeat

が大切みたいです。何をしたいかを明確にし、子供みたいに好奇心を持って、試行錯誤を繰り返せば、なんとかプログラムをでっちあげることが出来ます。プログラミングの学習では、どれか一つのプログラミング言語（C/C++ でも Java でも Python でも Ruby でも現在広く使われている言語であれば、最も自分で使うだけ

で他人に強制するのでなければ現在は使われなくなった BASIC でも何でもいいですが) の文法との習得と Niklaus Wirth 著「アルゴリズム+データ構造=プログラミング」にかかっているぐらいのアルゴリズムとデータ構造の理解と GUI プログラミングの技術を、自分の実力にあった一つの本の例題のプログラムをコツコツ、一つ一つ実際に打ち込み、実行して確かめるという単調な訓練を一通り経験しておけば、このように何とか試行錯誤でプログラミングをでっちあげることが出来るようになります。多分。「スキル」=「才能」×「努力」、「達成」=「スキル」×「努力」、即ち、「達成」=「才能」×「努力」×「努力」 だそうです。才能のある人が同じように努力すれば負けますが、プロスポーツ選手やプロ棋士ではないので勝ち負けではないです。自分が作りたいソフトを自力で作れるかどうかの問題です。

プログラミングを続けます。駒を List にセットして、チェスの初期配置を表示してみます。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ChessTest2
{
    public partial class Form1 : Form
    {
        public class Koma
        {
            public Image image;
            public Koma(Image img)
            {
                this.image = img;
            }
        }

        public Koma koma = null;
        public List<Koma> komas = new List<Koma>();
        public Form1()
```



```

{
    InitializeComponent();
    Image img =
        Image.FromFile(@"c:\VisualStudio2017\C#\white.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =
        Image.FromFile(@"c:\VisualStudio2017\C#\dark.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =
        Image.FromFile(@"c:\VisualStudio2017\C#\whiteKingOnWhite.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =
        Image.FromFile(@"c:\VisualStudio2017\C#\blackKingOnWhite.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =
        Image.FromFile(@"c:\VisualStudio2017\C#\whiteKingOnDark.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =
        Image.FromFile(@"c:\VisualStudio2017\C#\blackKingOnDark.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =
        Image.FromFile(@"c:\VisualStudio2017\C#\whiteQueenOnWhite.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =
        Image.FromFile(@"c:\VisualStudio2017\C#\blackQueenOnWhite.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =
        Image.FromFile(@"c:\VisualStudio2017\C#\whiteQueenOnDark.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img =

```

```

        Image.FromFile(@"c:\VisualStudio2017\C#\blackQueenOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteRookOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackRookOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteRookOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackRookOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteBishopOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackBishopOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteBishopOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackBishopOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteKnightOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);

```

```

        img =
            Image.FromFile(@"c:\VisualStudio2017\C#\blackKnightOnWhite.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img =
            Image.FromFile(@"c:\VisualStudio2017\C#\whiteKnightOnDark.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img =
            Image.FromFile(@"c:\VisualStudio2017\C#\blackKnightOnDark.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img =
            Image.FromFile(@"c:\VisualStudio2017\C#\whitePawnOnWhite.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img =
            Image.FromFile(@"c:\VisualStudio2017\C#\blackPawnOnWhite.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img =
            Image.FromFile(@"c:\VisualStudio2017\C#\whitePawnOnDark.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img =
            Image.FromFile(@"c:\VisualStudio2017\C#\blackPawnOnDark.jpg");
        koma = new Koma(img);
        komas.Add(koma);
    }

```

```

private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawImage(komas[11].image, 0, 0, 55, 55);
    g.DrawImage(komas[21].image, 55, 0, 55, 55);
    g.DrawImage(komas[15].image, 55 * 2, 0, 55, 55);
    g.DrawImage(komas[9].image, 55 * 3, 0, 55, 55);
    g.DrawImage(komas[3].image, 55 * 4, 0, 55, 55);
    g.DrawImage(komas[17].image, 55 * 5, 0, 55, 55);
}

```

```

g.DrawImage(komas[19].image, 55 * 6, 0, 55, 55);
g.DrawImage(komas[13].image, 55 * 7, 0, 55, 55);

g.DrawImage(komas[25].image, 0, 55, 55, 55);
g.DrawImage(komas[23].image, 55, 55, 55, 55);
g.DrawImage(komas[25].image, 55 * 2, 55, 55, 55);
g.DrawImage(komas[23].image, 55 * 3, 55, 55, 55);
g.DrawImage(komas[25].image, 55 * 4, 55, 55, 55);
g.DrawImage(komas[23].image, 55 * 5, 55, 55, 55);
g.DrawImage(komas[25].image, 55 * 6, 55, 55, 55);
g.DrawImage(komas[23].image, 55 * 7, 55, 55, 55);

g.DrawImage(komas[0].image, 0, 55 * 2, 55, 55);
g.DrawImage(komas[1].image, 55, 55 * 2, 55, 55);
g.DrawImage(komas[0].image, 55 * 2, 55 * 2, 55, 55);
g.DrawImage(komas[1].image, 55 * 3, 55 * 2, 55, 55);
g.DrawImage(komas[0].image, 55 * 4, 55 * 2, 55, 55);
g.DrawImage(komas[1].image, 55 * 5, 55 * 2, 55, 55);
g.DrawImage(komas[0].image, 55 * 6, 55 * 2, 55, 55);
g.DrawImage(komas[1].image, 55 * 7, 55 * 2, 55, 55);

g.DrawImage(komas[1].image, 0, 55 * 3, 55, 55);
g.DrawImage(komas[0].image, 55, 55 * 3, 55, 55);
g.DrawImage(komas[1].image, 55 * 2, 55 * 3, 55, 55);
g.DrawImage(komas[0].image, 55 * 3, 55 * 3, 55, 55);
g.DrawImage(komas[1].image, 55 * 4, 55 * 3, 55, 55);
g.DrawImage(komas[0].image, 55 * 5, 55 * 3, 55, 55);
g.DrawImage(komas[1].image, 55 * 6, 55 * 3, 55, 55);
g.DrawImage(komas[0].image, 55 * 7, 55 * 3, 55, 55);

g.DrawImage(komas[0].image, 0, 55 * 4, 55, 55);
g.DrawImage(komas[1].image, 55, 55 * 4, 55, 55);
g.DrawImage(komas[0].image, 55 * 2, 55 * 4, 55, 55);
g.DrawImage(komas[1].image, 55 * 3, 55 * 4, 55, 55);
g.DrawImage(komas[0].image, 55 * 4, 55 * 4, 55, 55);
g.DrawImage(komas[1].image, 55 * 5, 55 * 4, 55, 55);
g.DrawImage(komas[0].image, 55 * 6, 55 * 4, 55, 55);
g.DrawImage(komas[1].image, 55 * 7, 55 * 4, 55, 55);

```

```

g.DrawImage(komas[1].image, 0, 55 * 5, 55, 55);
g.DrawImage(komas[0].image, 55, 55 * 5, 55, 55);
g.DrawImage(komas[1].image, 55 * 2, 55 * 5, 55, 55);
g.DrawImage(komas[0].image, 55 * 3, 55 * 5, 55, 55);
g.DrawImage(komas[1].image, 55 * 4, 55 * 5, 55, 55);
g.DrawImage(komas[0].image, 55 * 5, 55 * 5, 55, 55);
g.DrawImage(komas[1].image, 55 * 6, 55 * 5, 55, 55);
g.DrawImage(komas[0].image, 55 * 7, 55 * 5, 55, 55);

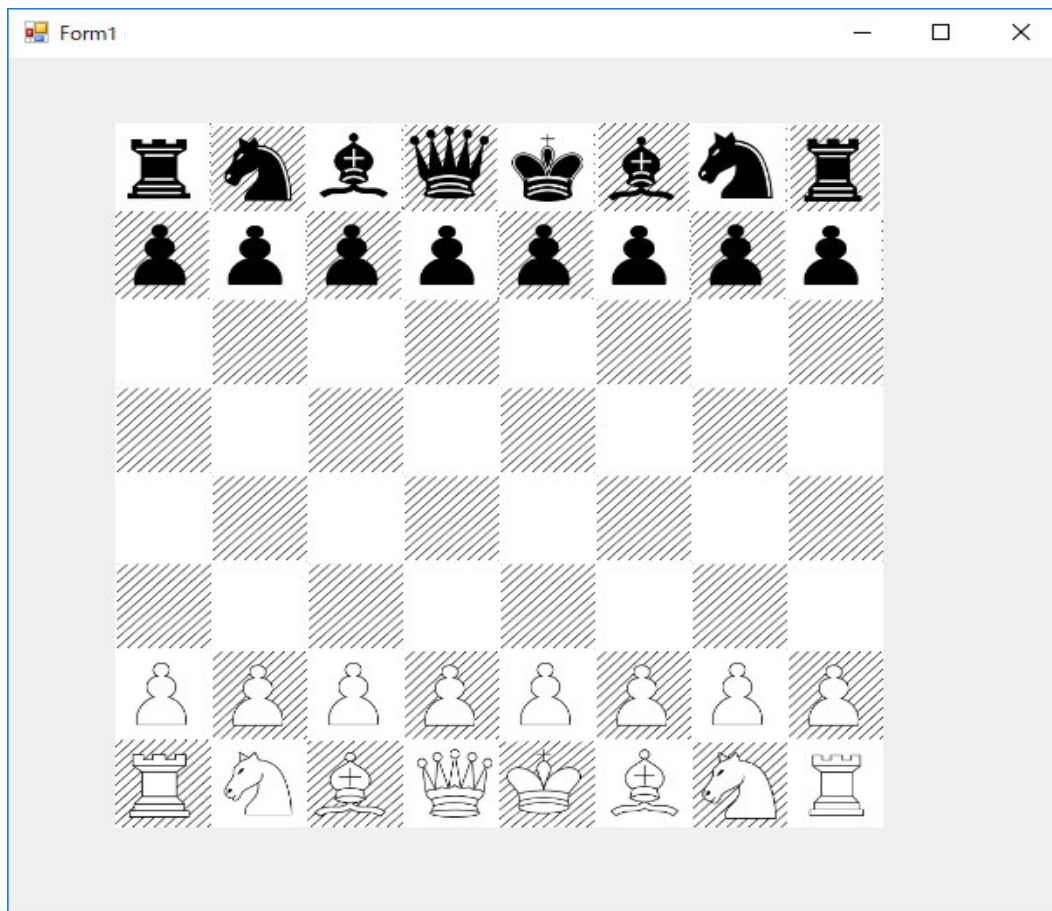
g.DrawImage(komas[22].image, 0, 55 * 6, 55, 55);
g.DrawImage(komas[24].image, 55, 55 * 6, 55, 55);
g.DrawImage(komas[22].image, 55 * 2, 55 * 6, 55, 55);
g.DrawImage(komas[24].image, 55 * 3, 55 * 6, 55, 55);
g.DrawImage(komas[22].image, 55 * 4, 55 * 6, 55, 55);
g.DrawImage(komas[24].image, 55 * 5, 55 * 6, 55, 55);
g.DrawImage(komas[22].image, 55 * 6, 55 * 6, 55, 55);
g.DrawImage(komas[24].image, 55 * 7, 55 * 6, 55, 55);

g.DrawImage(komas[12].image, 0, 55 * 7, 55, 55);
g.DrawImage(komas[18].image, 55, 55 * 7, 55, 55);
g.DrawImage(komas[16].image, 55 * 2, 55 * 7, 55, 55);
g.DrawImage(komas[6].image, 55 * 3, 55 * 7, 55, 55);
g.DrawImage(komas[4].image, 55 * 4, 55 * 7, 55, 55);
g.DrawImage(komas[14].image, 55 * 5, 55 * 7, 55, 55);
g.DrawImage(komas[20].image, 55 * 6, 55 * 7, 55, 55);
g.DrawImage(komas[10].image, 55 * 7, 55 * 7, 55, 55);

    }
}
}

```

がプログラムで、実行結果は



です。上手くいったみたいです。こんな長ったらしい下手くそなプログラムを作らなくても、うまい方法があるだろうとお思いでしょうが、素人が C# の文法書もろくに読まずに、VC++ や Java との類比で、手探りで作っているのがこのようになります。例えば、画像のファイル名を数字にしておけば、読込が簡単になりますが、このようなプログラムなら何番目に何の画像がセットされているか、プログラムを見れば判り、余計な記憶する必要がありません。但し、配列にファイル名をセットしておけば、繰り返しで読み込むことが出来ますが、どのやり方にも一長一短があります。また、素人は一時に沢山のことをやろうとせず、一つ一つ確かめながらプログラミングしていくのがいいと私は思います。

盤を 2 次元配列で表し、盤の情報を表示するようにしましょう。最終目的は

```
\documentclass[10pt,a4paper]{jarticle}
```

```
\font\remark=skak30 scaled\magstep2
```

```
%\paperwidth 210mm%用紙の幅 (A4 サイズ縦長の用紙の幅、210mm=597pt)
```

```
%\paperheight 297mm%用紙の高さ (A4 サイズ縦長の用紙の高さ、297mm=845pt)
```

`\voffset 0pt%`ヘッダ上端までの上余白 (実際には 1 インチ+`\voffset` の上余白となる)
`\hoffset -15pt%`文章領域までの左余白 (実際には 1 インチ+`\hoffset` の左余白となる)
`\topmargin -20pt%`上余白からヘッダ上端までの余白
`\headheight -20pt%`ヘッダの高さ
`\headsep 0pt%`ヘッダの下端から本文領域までの高さ
`\topskip 0pt%`文章領域内の本文開始位置までの高さ
`\oddsidemargin -15pt%`奇数ページのマージン (左余白から文章領域の左端までの余白)
`%\evensidemargin 0pt%`偶数ページのマージン (jsbook など左右ページでレイアウトが異なるときに指定する)
`\textwidth 176mm%`文章領域の幅 (A4 サイズ縦長の文章領域の幅、176mm=500pt)
`\textheight 264mm%`文章領域の高さ (A4 サイズ縦長の文章領域の高さ、264mm=750pt)
`%\textwidth 53zw%`文章領域の幅 (A4 サイズ縦長のとき) 文章領域の高さの pt × 0.107 = 全角文字数
`%\textheight 80zw%`文章領域の高さ (A4 サイズ縦長のとき) 文章領域の高さの pt × 0.107 = 全角文字数
`\footskip 20pt%`文章領域の下端からフッタの下端までの高さ
`\marginparsep 0pt%`文章領域の右端から欄外傍注の左端の幅
`\marginparwidth 0pt%`傍注の幅
`\marginparpush 0pt%`傍注間の高さ

`\begin{document}`
`\Huge`
`\centerline{`
`\hbox{\vrule`
`\vbox{\hrule`
`\hbox{\remark r m b l k a o s }`
`\hbox{\remark o p o p Z p o p }`
`\hbox{\remark 0 Z 0 Z 0 m 0 Z }`
`\hbox{\remark Z 0 Z 0 o 0 Z 0 }`
`\hbox{\remark 0 Z 0 Z P Z 0 Z}`
`\hbox{\remark Z 0 Z 0 Z N Z 0 }`
`\hbox{\remark P 0 P 0 0 0 P 0 }`
`\hbox{\remark S N A Q J B Z R }`
`\hrule }`
`\hskip -7pt \vrule }`
`}`

`\vskip 1cm`

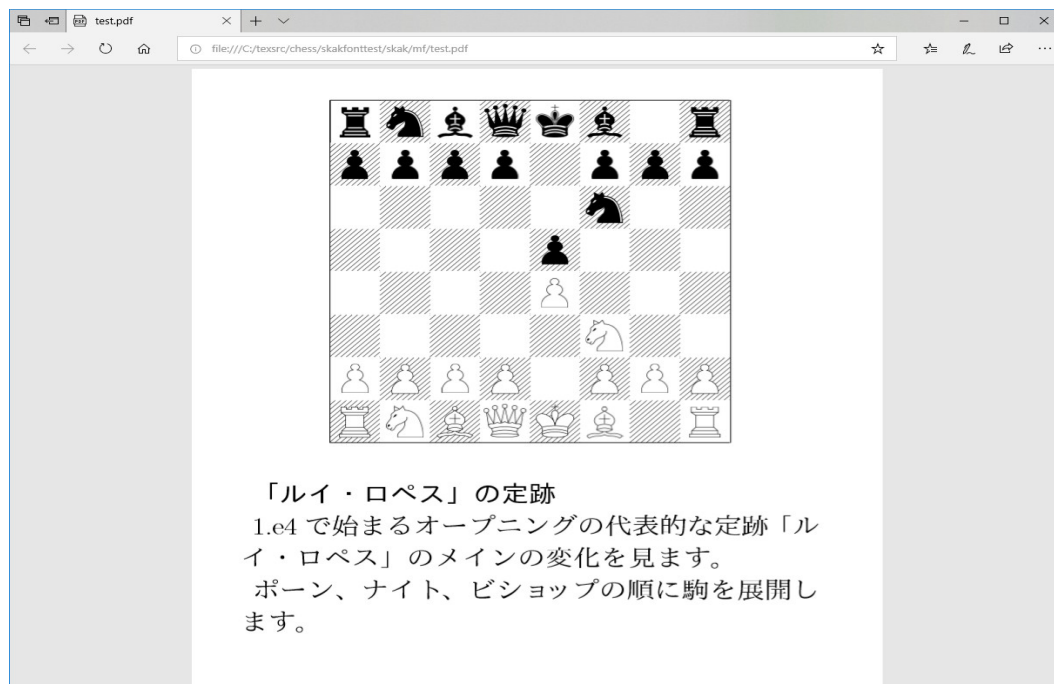
`{\bf 「ルイ・ロペス」の定跡 }`

1.e4 で始まるオープニングの代表的な定跡「ルイ・ロペス」のメインの変化を見
ます。

ポーン、ナイト、ビショップの順に駒を展開します。

`\end{document}`

のファイルを作り、



の pdf ファイルを得ることで、コンピュータと対戦するプログラムではないです
から、盤の配列は 8×8 で充分です。駒は

空 0

白キング 1 、 黒キング -1

白クイーン 2 、 黒クイーン -2

白ルーク 3 、 黒ルーク -3

白ビショップ 4 、 黒ビショップ -4

白ナイト 5、 黒ナイト -5

白ポーン 6、 黒ポーン -6

で表すことにします。チェスは将棋や囲碁と違い左下隅が原点なので、それに合わせることにします。

```
public int[,] ban =
{
    {3, 5, 4, 2, 1, 4, 5, 3 },
    {6, 6, 6, 6, 6, 6, 6, 6 },
    {0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0 },
    {-6, -6, -6, -6, -6, -6, -6, -6 },
    {-3, -5, -4, -2, -1, -4, -5, -3 },
};
```

で、チェスの初期配置を配列 ban にセットします。盤の情報と駒の座標の情報から、画像の番号を決定する関数 public int square(int n, int k, int i) を

```
public int square(int n, int k, int i)
{
    int result = 0;
    if (n > 0)
    {
        result = 4 * n - 2 + ((i + k) % 2) * 2;
    }
    else if (n < 0)
    {
        result = 4 * (-n) - 2 + ((i + k) % 2) * 2 + 1;
    }
    else
    {
        result = (i + k) % 2;
    }
    return result;
}
```

で定義します。すると初期配置を描く部分のプログラムは

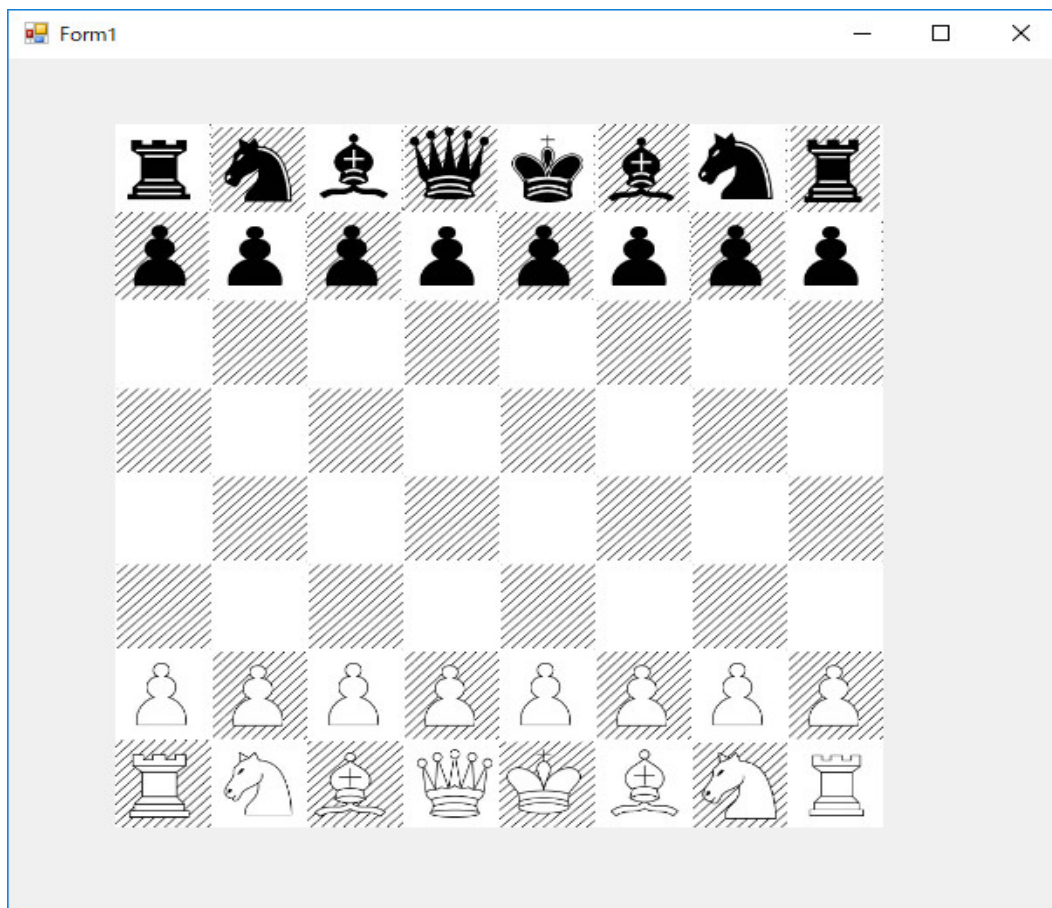
```
private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
```

```

        for (int k=0; k<8; k++)
        {
            for (int i=0; i<8; i++)
            {
                g.DrawImage(komas[square(ban[k, i], k, i)].image,
                    55 * i, 55 * (7 - k), 55, 55);
            }
        }
    }
}

```

となります。実行結果は



です。上手くいったみたいです。

次にすべきことは、マウスでコマを移動できるようにすることです。PictureBox のイベント `MouseDown` をダブルクリックします。

```

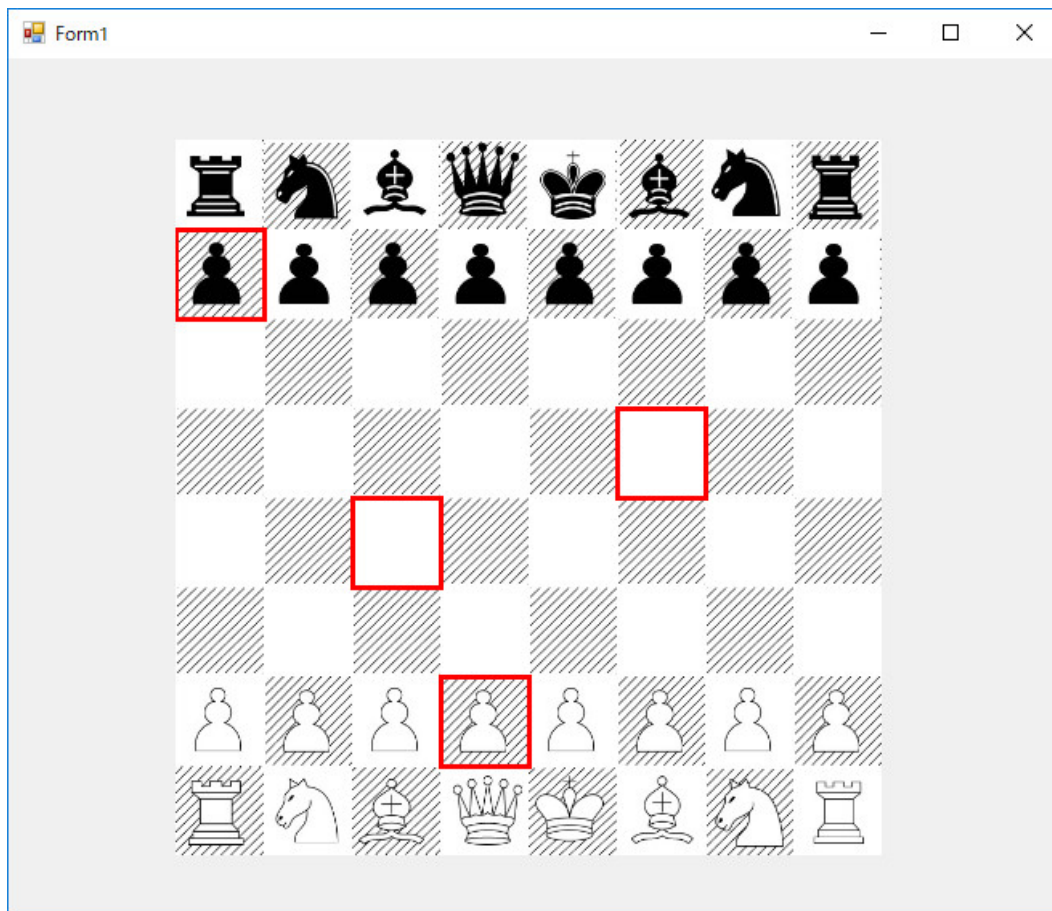
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    // ...
}

```

が作られます。ここに、盤面をマウスでクリックした時の処理を書き込みます。練習のため、

```
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    int x = e.X;
    int y = e.Y;
    int X = 0;
    int Y = 0;
    for (int i=0; i<8; i++)
    {
        X = 55 * i;
        if (x >= 55 * i && x < 55 * (i + 1)) break;
    }
    for (int i = 0; i < 8; i++)
    {
        Y = 55 * i;
        if (y >= 55 * i && y < 55 * (i + 1)) break;
    }
    Graphics g = pictureBox1.CreateGraphics();
    Pen p = new Pen(Color.Red, 3);
    g.DrawRectangle(p, X, Y, 55, 55);
}
```

というプログラムを作ってみます。実行し、マウスで何か所かクリックすると



となります。意図したように動いているみたいです。C# の GUI を解説した参考書を探して、それを読むことなしに、プログラミングが何とかできているのは、インターネットにアップしている私の「数独のヒントを表示してくれるプログラム」という pdf を見ながら、VC++ で出来たことと同じようなことが出来るはずだと思って、類推で、インターネットで確かめながらプログラミングしているからです。

1 回目のクリックで、移動させる駒を指示し、2 回目のクリックで移動場所を指示して、駒を移動させるようにします。本当に移動できるかをチェックするとか、棋譜を残し保存できるようにできるようにするには、プログラムが複雑になりますから、自分だけで使うプログラムですから、差し当って、正しい指示をするものと仮定して、単純に、一回目のクリックで、移動させる駒を指示し、2 回目のクリックで移動場所を指示して、駒を移動させるようにします。やっぱり駒の移動が正しいかチェックすべきだと思ったときに、プログラムを修正するようにします。クリックが 1 回目か、2 回目かなどの駒を 1 回目にクリックしたかを保持している変数が必要です。

```
public bool FirstClick = true;
public int oldX = 0;
```

```
public int oldY = 0;
```

とグローバル変数を定義し、

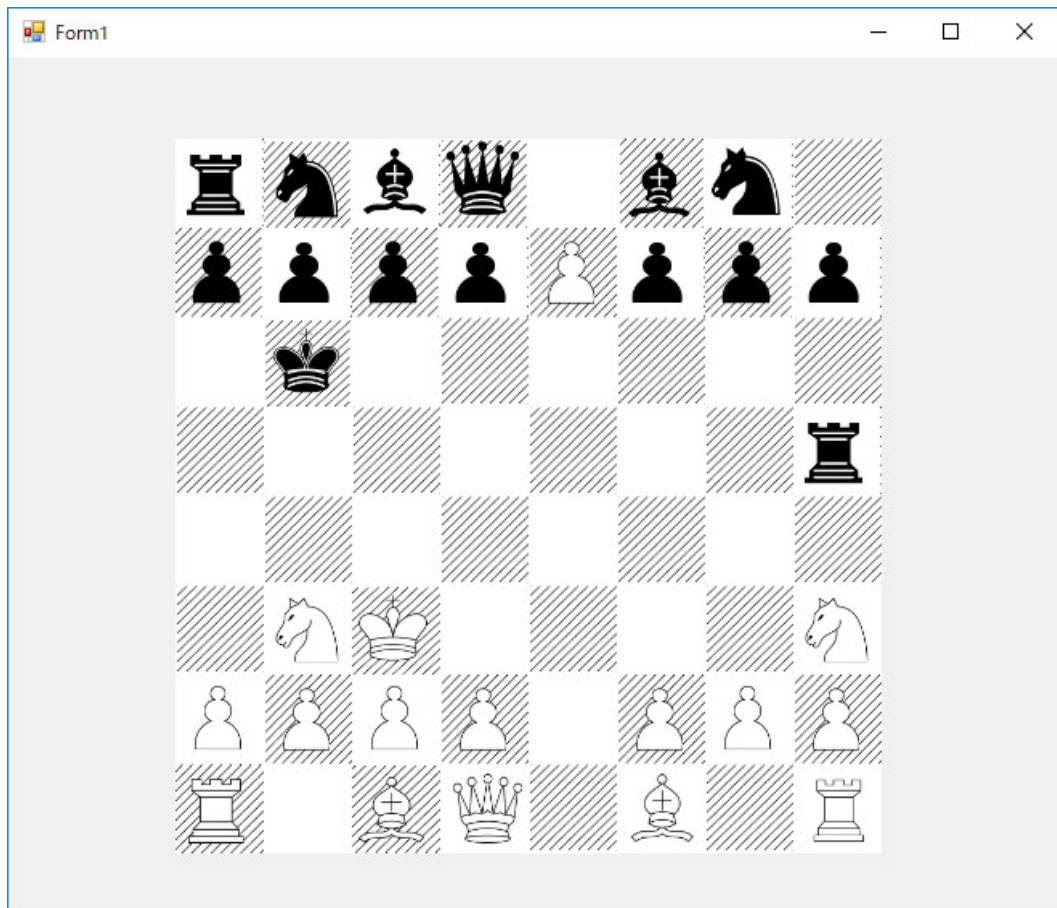
```
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    int x = e.X;
    int y = e.Y;
    int X = 0;
    int Y = 0;
    for (int i=0; i<8; i++)
    {
        X = 55 * i;
        if (x >= 55 * i && x < 55 * (i + 1)) break;
    }
    for (int i = 0; i < 8; i++)
    {
        Y = 55 * i;
        if (y >= 55 * i && y < 55 * (i + 1)) break;
    }
    if (FirstClick == true)
    {
        int K = 7 - Y / 55;
        int I = X / 55;
        if (ban[K, I] == 0) return;
        Graphics g = pictureBox1.CreateGraphics();
        Pen p = new Pen(Color.Red, 1);
        g.DrawRectangle(p, X, Y, 54, 54);
        FirstClick = false;
        oldX = X;
        oldY = Y;
    }
    else
    {
        int K = 7 - oldY / 55;
        int I = oldX / 55;
        int K2 = 7 - Y / 55;
        int I2 = X / 55;
        ban[K2, I2] = ban[K, I];
        ban[K, I] = 0;
    }
}
```

```

Graphics g = pictureBox1.CreateGraphics();
g.FillRectangle(Brushes.White, oldX, oldY, 55, 55);
g.DrawImage(komas[square(ban[K, I], K, I)].image,
    55 * I, 55 * (7 - K), 55, 55);
g.DrawImage(komas[square(ban[K2, I2], K2, I2)].image,
    55 * I2, 55 * (7 - K2), 55, 55);
FirstClick = true;
    }
}

```

とすれば、



と出鱈目に駒を動かすことが出来ますが、意図したとおりに動いています。1回目のクリックで描いた赤の矩形がうまく消えないので

```

Pen p = new Pen(Color.Red, 1);
g.DrawRectangle(p, X, Y, 54, 54);

```

と修正しています。駒をルール通り動かすために

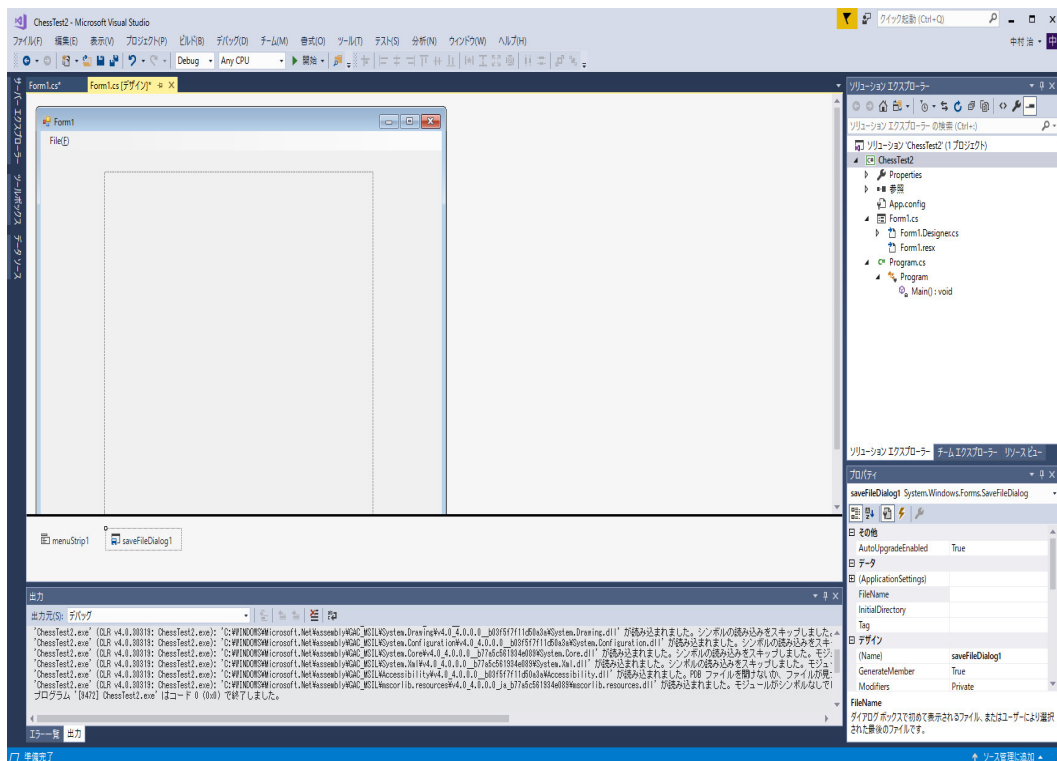
```
if (ban[K, I] == 0) return;
```

とプログラムしていますが、これを除くと空のセルを駒のある位置に動かすと駒を消すことが出来ます。棋譜を作るプログラムでなく、局面を作るプログラムですから、これがない方が正解です。欲しい局面が作れるようになりましたから、局面の TeX プログラムをファイルに打ち出す部分を作ります。

まず、メニューを作ります。「ツールボックス」から「MenuStrip」を探し、フォームヘドラッグ&ドロップします。「ここへ入力」に、「File(&F)」、その下の「ここへ入力」に、「TeXMacroSave(&S)」と入力し、ダブルクリックします。

```
private void texMacroSaveToolStripMenuItem_Click(
    object sender, EventArgs e)
{
}
}
```

が作られる。「ツールボックス」から「SaveFileDialog」を探し、フォームヘドラッグ&ドロップします。



と画面になったはずです。

```
private void texMacroSaveToolStripMenuItem_Click(
    object sender, EventArgs e)
```

```

    {
        SaveFileDialog dlg = new SaveFileDialog();
        dlg.Title = "保存するファイル名を入力してください。"; // ダイアログのタイトル。

        if (dlg.ShowDialog() == DialogResult.OK)
        {
            System.IO.Stream stream;
            stream = dlg.OpenFile();
            if (stream != null)
            {
                //ファイルに書き込む
                System.IO.StreamWriter sw =
                    new System.IO.StreamWriter(stream);
                sw.Write("Hello world!");
                //閉じる
                sw.Close();
                stream.Close();
            }
        }
    }
}

```

と打ち込みます。実行すると指定したファイルに「Hello world!」と書きだされています。

```

public string[] cord =
{
    "O", "Z", "K", "k", "J", "j", "Q", "q", "L", "l",
    "R", "r", "S", "s", "B", "b", "A", "a", "N", "n",
    "M", "m", "P", "p", "0", "o",
};

```

と配列を宣言し、

```
sw.Write("Hello world!");
```

の部分的局面の $\text{T}_{\text{E}}\text{X}$ マクロを打ち出すプログラムに変えます。

```

private void texMacroSaveSToolStripMenuItem_Click(
    object sender, EventArgs e)
{

```



```
SaveFileDialog dlg = new SaveFileDialog();
dlg.Title = "保存するファイル名を入力してください。"; // ダイアログのタイトル。
```

```
if (dlg.ShowDialog() == DialogResult.OK)
{
    System.IO.Stream stream;
    stream = dlg.OpenFile();
    if (stream != null)
    {
        //ファイルに書き込む
        System.IO.StreamWriter sw =
            new System.IO.StreamWriter(stream);
        sw.WriteLine("\\centerline{");
        sw.WriteLine("\\hbox{\\vrule");
        sw.WriteLine("\\vbox{\\hrule");
        for (int k=7; k>=0; k--)
        {
            string line = " \\hbox{\\remark";
            for (int i=0; i<8; i++)
            {
                line += " ";
                line += cord[square(ban[k, i], k, i)];
            }
            line += " }";
            sw.WriteLine(line);
        }
        sw.WriteLine(" \\hrule }");
        sw.WriteLine(" \\hskip - 7pt \\vrule }");
        sw.WriteLine("}");
        //閉じる
        sw.Close();
        stream.Close();
    }
}
```

で、初期配置で、保存すると

```
\centerline{
```

```

\hbox{\vrule
\vbox{\hrule
  \hbox{\remark r m b l k a n s }
  \hbox{\remark o p o p o p o p }
  \hbox{\remark 0 Z 0 Z 0 Z 0 Z }
  \hbox{\remark Z 0 Z 0 Z 0 Z 0 }
  \hbox{\remark 0 Z 0 Z 0 Z 0 Z }
  \hbox{\remark Z 0 Z 0 Z 0 Z 0 }
  \hbox{\remark P 0 P 0 P 0 P 0 }
  \hbox{\remark S N A Q J B M R }
  \hrule }
  \hskip - 7pt \vrule }
}

```

というファイルを書き出してくれます。

```
\documentclass[10pt,a4paper]{jarticle}
```

```
\font\remark=skak30 scaled\magstep2
```

```

%\paperwidth 210mm%用紙の幅 (A4 サイズ縦長の用紙の幅、210mm=597pt)
%\paperheight 297mm%用紙の高さ (A4 サイズ縦長の用紙の高さ、297mm=845pt)
\voffset 0pt%ヘッダ上端までの上余白 (実際には1インチ+\voffsetの上余白
となる)
\hoffset -15pt%文章領域までの左余白 (実際には1インチ+\hoffsetの左余白
となる)
\topmargin -20pt%上余白からヘッダ上端までの余白
\headheight -20pt%ヘッダの高さ
\headsep 0pt%ヘッダの下端から本文領域までの高さ
\topskip 0pt%文章領域内の本文開始位置までの高さ
\oddsidemargin -15pt%奇数ページのマージン (左余白から文章領域の左端まで
の余白)
%\evensidemargin 0pt%偶数ページのマージン (jsbookなど左右ページでレイア
ウトが異なるときに指定する)
\textwidth 176mm%文章領域の幅 (A4 サイズ縦長の文章領域の幅、176mm=500pt)
\textheight 264mm%文章領域の高さ (A4 サイズ縦長の文章領域の高さ、264mm=750pt)
%\textwidth 53zw%文章領域の幅 (A4 サイズ縦長のとき) 文章領域の高さの pt ×
0.107 = 全角文字数
%\textheight 80zw%文章領域の高さ (A4 サイズ縦長のとき) 文章領域の高さの
pt × 0.107 = 全角文字数

```

`\footskip 20pt%`文章領域の下端からフッタの下端までの高さ
`\marginparsep 0pt%`文章領域の右端から欄外傍注の左端の幅
`\marginparwidth 0pt%`傍注の幅
`\marginparpush 0pt%`傍注間の高さ

`\begin{document}`

`\Huge`

```

\centerline{
\hbox{\vrule
\vbox{\hrule
  \hbox{\remark r m b l k a n s }
  \hbox{\remark o p o p o p o p }
  \hbox{\remark 0 Z 0 Z 0 Z 0 Z }
  \hbox{\remark Z 0 Z 0 Z 0 Z 0 }
  \hbox{\remark 0 Z 0 Z 0 Z 0 Z }
  \hbox{\remark Z 0 Z 0 Z 0 Z 0 }
  \hbox{\remark P 0 P 0 P 0 P 0 }
  \hbox{\remark S N A Q J B M R }
  \hrule }
  \hskip - 7pt \vrule }
}

```

`\vskip 1cm`

`{\bf 「ルイ・ロペス」の定跡 }`

1.e4 で始まるオープニングの代表的な定跡「ルイ・ロペス」のメインの変化を見ます。

ポーン、ナイト、ビショップの順に駒を展開します。

`\end{document}`

の型紙の

```

\centerline{
\hbox{\vrule
\vbox{\hrule
  \hbox{\remark r m b l k a n s }

```

```

\hbox{\remark o p o p o p o p }
\hbox{\remark O Z O Z O Z O Z }
\hbox{\remark Z O Z O Z O Z O }
\hbox{\remark O Z O Z O Z O Z }
\hbox{\remark Z O Z O Z O Z O }
\hbox{\remark P O P O P O P O }
\hbox{\remark S N A Q J B M R }
\hrule }
\hskip - 7pt \vrule }
}

```

の部分を出してくれるプログラムができました。これが欲しかったプログラムです。プログラムの全体は

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ChessTest2
{
    public partial class Form1 : Form
    {
        public class Koma
        {
            public Image image;
            public Koma(Image img)
            {
                this.image = img;
            }
        }

        public Koma koma = null;
        public List<Koma> komas = new List<Koma>();
        public int[,] ban =

```

```

{
    {3, 5, 4, 2, 1, 4, 5, 3 },
    {6, 6, 6, 6, 6, 6, 6, 6 },
    {0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0 },
    {0, 0, 0, 0, 0, 0, 0, 0 },
    {-6, -6, -6, -6, -6, -6, -6, -6 },
    {-3, -5, -4, -2, -1, -4, -5, -3 },
};

public int square(int n, int k, int i)
{
    int result = 0;
    if (n > 0)
    {
        result = 4 * n - 2 + (1-(i + k) % 2) * 2;
    }
    else if (n < 0)
    {
        result = 4 * (-n) - 2 + (1-(i + k) % 2) * 2 + 1;
    }
    else
    {
        result = 1-(i + k) % 2;
    }
    return result;
}

public bool FirstClick = true;
public int oldX = 0;
public int oldY = 0;
public string[] cord =
{
    "O", "Z", "K", "k", "J", "j", "Q", "q", "L", "l",
    "R", "r", "S", "s", "B", "b", "A", "a", "N", "n",
    "M", "m", "P", "p", "O", "o",
};

public Form1()
{

```

```

InitializeComponent();
Image img =
    Image.FromFile(@"c:\VisualStudio2017\C#\white.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\dark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteKingOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackKingOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteKingOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackKingOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteQueenOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackQueenOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteQueenOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackQueenOnDark.jpg");

```

```

koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteRookOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackRookOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteRookOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackRookOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteBishopOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackBishopOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteBishopOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackBishopOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteKnightOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =

```

```

        Image.FromFile(@"c:\VisualStudio2017\C#\blackKnightOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whiteKnightOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackKnightOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whitePawnOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackPawnOnWhite.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\whitePawnOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
img =
    Image.FromFile(@"c:\VisualStudio2017\C#\blackPawnOnDark.jpg");
koma = new Koma(img);
komas.Add(koma);
}

```

```

private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    for (int k=0; k<8; k++)
    {
        for (int i=0; i<8; i++)
        {
            g.DrawImage(komas[square(ban[k, i], k, i)].image,
                55 * i, 55 * (7 - k), 55, 55);
        }
    }
}

```



```

    }
}

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    int x = e.X;
    int y = e.Y;
    int X = 0;
    int Y = 0;
    for (int i=0; i<8; i++)
    {
        X = 55 * i;
        if (x >= 55 * i && x < 55 * (i + 1)) break;
    }
    for (int i = 0; i < 8; i++)
    {
        Y = 55 * i;
        if (y >= 55 * i && y < 55 * (i + 1)) break;
    }
    if (FirstClick == true)
    {
        int K = 7 - Y / 55;
        int I = X / 55;
        //if (ban[K, I] == 0) return;
        Graphics g = pictureBox1.CreateGraphics();
        Pen p = new Pen(Color.Red, 1);
        g.DrawRectangle(p, X, Y, 54, 54);
        FirstClick = false;
        oldX = X;
        oldY = Y;
    }
    else
    {
        int K = 7 - oldY / 55;
        int I = oldX / 55;
        int K2 = 7 - Y / 55;
        int I2 = X / 55;
        ban[K2, I2] = ban[K, I];
        ban[K, I] = 0;
    }
}

```

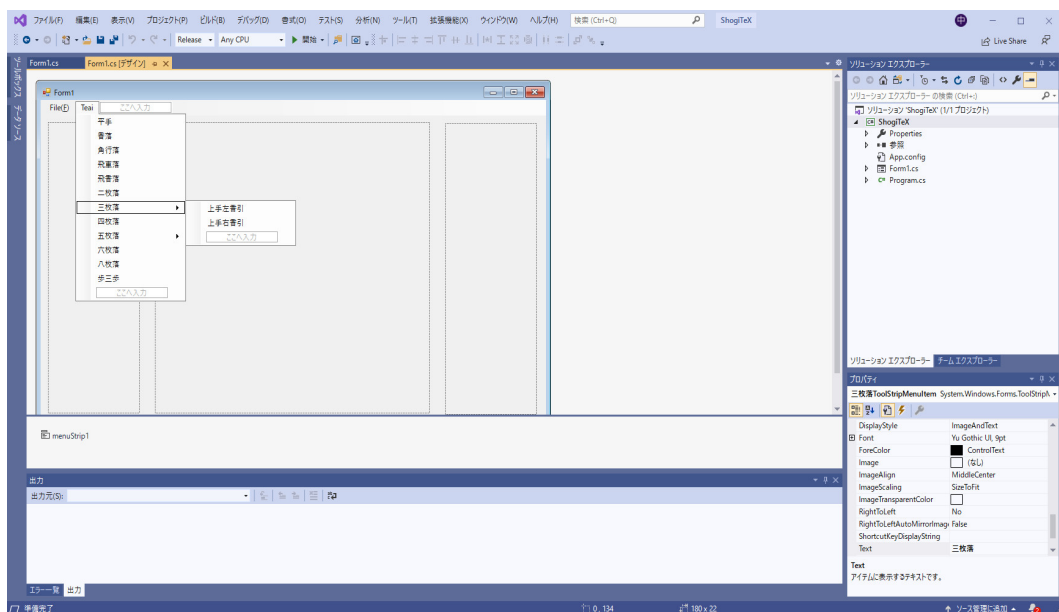
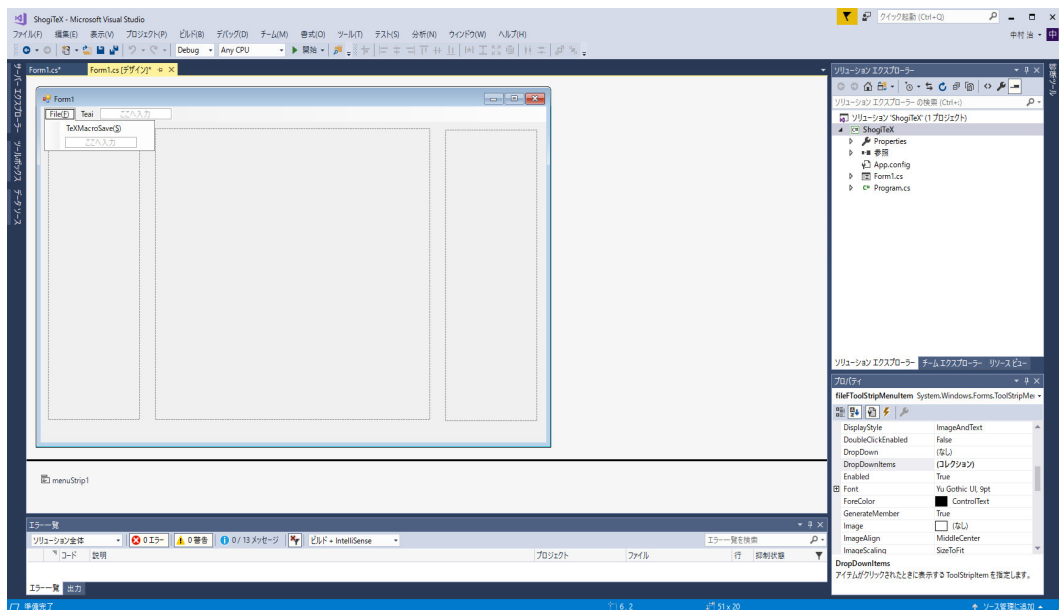
```

Graphics g = pictureBox1.CreateGraphics();
g.FillRectangle(Brushes.White, oldX, oldY, 55, 55);
g.DrawImage(komas[square(ban[K, I], K, I)].image,
    55 * I, 55 * (7 - K), 55, 55);
g.DrawImage(komas[square(ban[K2, I2], K2, I2)].image,
    55 * I2, 55 * (7 - K2), 55, 55);
FirstClick = true;
    }
}

private void texMacroSaveSToolStripMenuItem_Click(
    object sender, EventArgs e)
{
    SaveFileDialog dlg = new SaveFileDialog();
    dlg.Title = "保存するファイル名を入力してください。"; // ダイアログのタイトル。

    if (dlg.ShowDialog() == DialogResult.OK)
    {
        System.IO.Stream stream;
        stream = dlg.OpenFile();
        if (stream != null)
        {
            //ファイルに書き込む
            System.IO.StreamWriter sw =
                new System.IO.StreamWriter(stream);
            sw.WriteLine("\\centerline{");
            sw.WriteLine("\\hbox{\\vrule");
            sw.WriteLine("\\vbox{\\hrule");
            for (int k=7; k>=0; k--)
            {
                string line = " \\hbox{\\remark";
                for (int i=0; i<8; i++)
                {
                    line += " ";
                    line += cord[square(ban[k, i], k, i)];
                }
                line += " }";
                sw.WriteLine(line);
            }
        }
    }
}

```

としています。

Teai

平手
香落
角行落
飛車落
飛香落
二枚落
三枚落 -> 上手左香引

上手右香引

四枚落

五枚落 → 上手右桂引

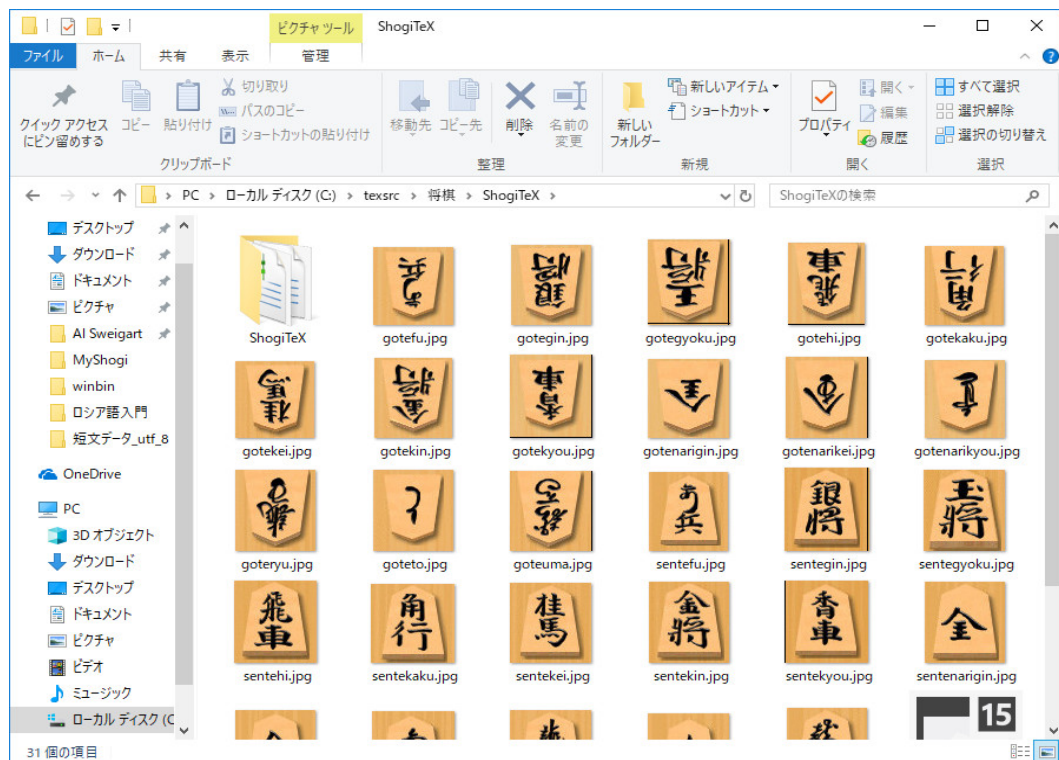
上手左桂引

六枚落

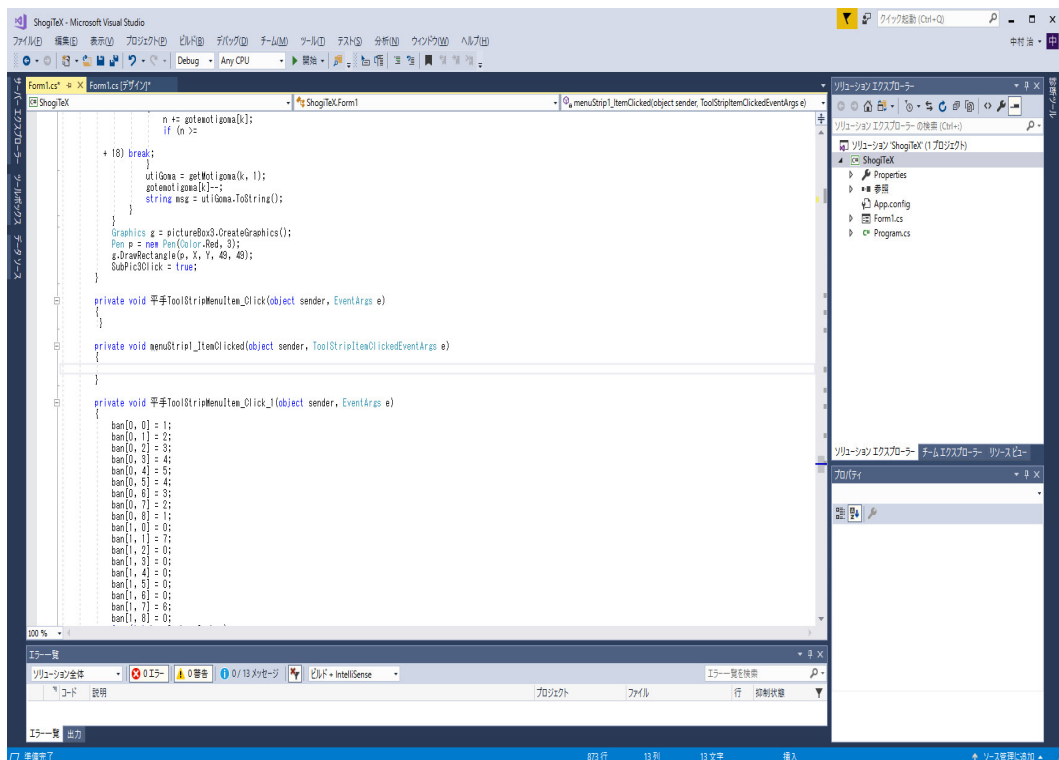
八枚落

歩三步

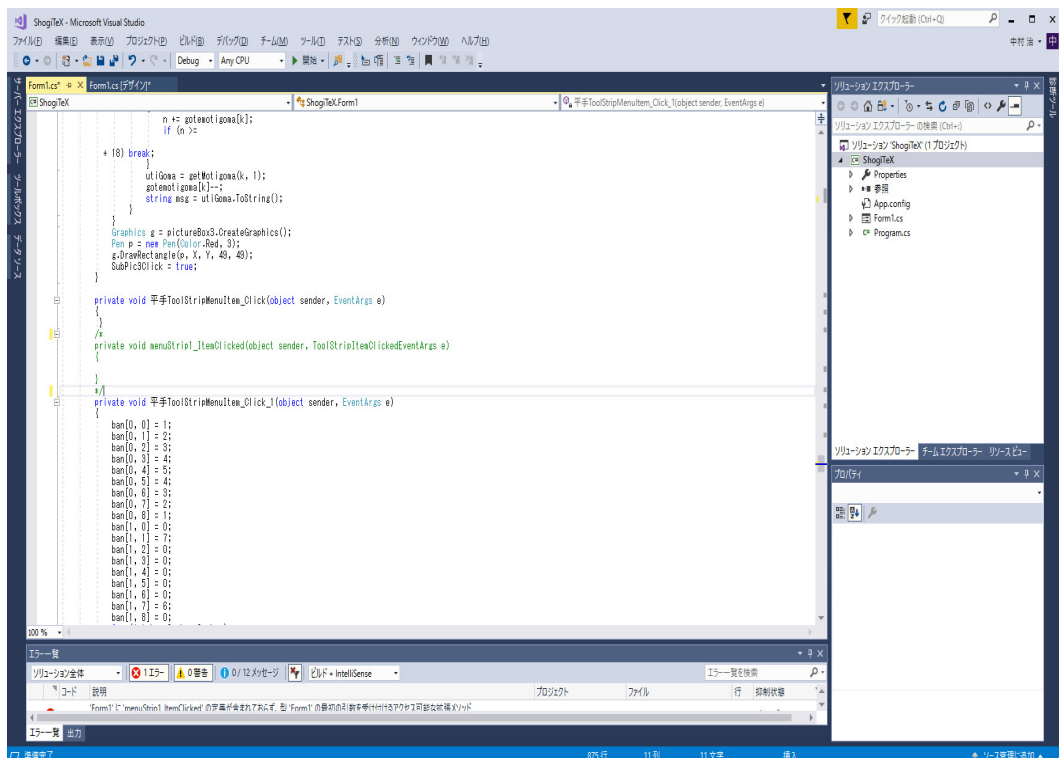
将棋の駒の画像は、自分だけで使うプログラムですから Kifu for Windows V7
の画像を使わしてもらいます。



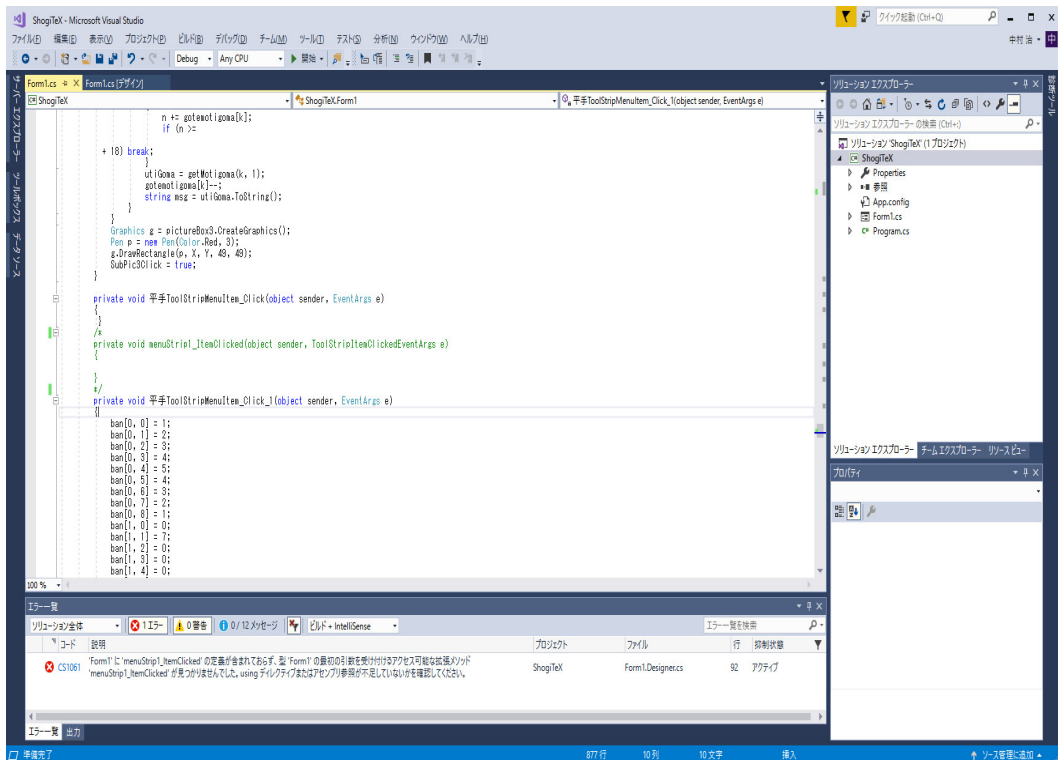
上手く切り取れていませんが、素人ですから気にしないことにします。作り方は基本的にチェスのプログラミングと同じですから、作っていて気付いたことだけ説明します。



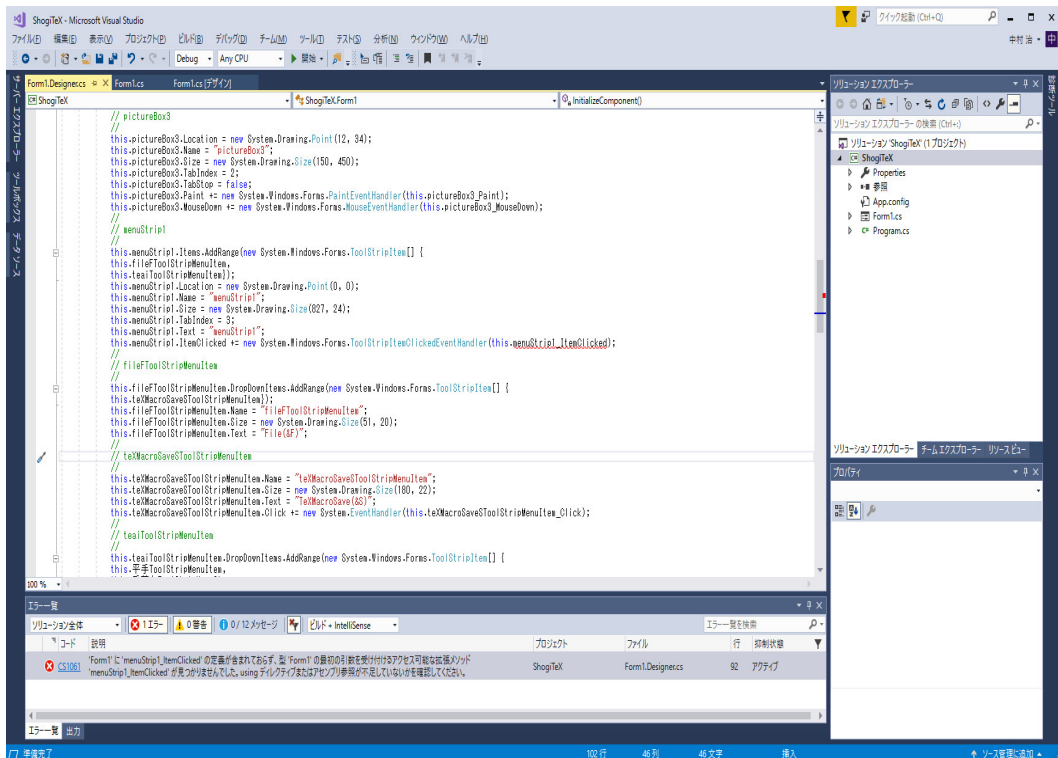
のように、操作ミスで、余計なイベントを作ってしまうことがありますが、そのままでも実害はないですが、気になるなら



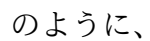
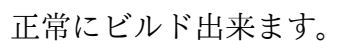
のようにコメントアウトし、ビルドすると



とエラーになるので、エラーの表示をダブルクリックすると



と Form1.Designer.cs が開くので、エラーの部分をコメントアウトすると




```

private void pictureBox1_DoubleClick(object sender, EventArgs e)
{
    int x = ((MouseEventArgs)e).X;
    int y = ((MouseEventArgs)e).Y;
    int X = 0;
    int Y = 0;
    for (int i = 0; i < 9; i++)
    {
        X = 50 * i;
        if (x >= 50 * i && x < 50 * (i + 1)) break;
    }
    for (int i = 0; i < 9; i++)
    {
        Y = 50 * i;
        if (y >= 50 * i && y < 50 * (i + 1)) break;
    }
    ...
}

```

ダブルクリックした位置の座標を得るには上のようにキャストを使います。インターネットには間違った（今では使えない）情報をありますが、気長に探していると正しい情報を見つけることができます。見つけるたびに、実際にプログラミングしてみて、それが自分のプログラムで使える（プログラミング言語のバージョンによっては使えなくなっていることがあるので、自分の使っているプログラミング言語で実際に使える）、正しい情報かどうか確かめてみれば良いです。

また

```

string msg = "K="+K.ToString()+"k="+k.ToString()+":"+utiGoma.ToString();
MessageBox.Show(msg,
    "エラー",
    MessageBoxButtons.OK);
msg = "";
for (k=0; k<7; k++)
{
    msg += sentemotigoma[k].ToString() + " ";
}
MessageBox.Show(msg,
    "エラー",
    MessageBoxButtons.OK);

```

を挿入することにより、変数の値を表示することが出来ます。

プログラムは

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ShogiTeX
{
    public partial class Form1 : Form
    {
        public class Koma
        {
            public Image image;
            public Koma(Image img)
            {
                this.image = img;
            }
        }

        public Koma koma = null;
        public List<Koma> komas = new List<Koma>();
        public int[,] ban =
        {
            {1, 2, 3, 4, 5, 4, 3, 2, 1 },
            {0, 7, 0, 0, 0, 0, 0, 6, 0 },
            {8, 8, 8, 8, 8, 8, 8, 8, 8 },
            {0, 0, 0, 0, 0, 0, 0, 0, 0 },
            {0, 0, 0, 0, 0, 0, 0, 0, 0 },
            {0, 0, 0, 0, 0, 0, 0, 0, 0 },
            {-8, -8, -8, -8, -8, -8, -8, -8, -8 },
            {0, -6, 0, 0, 0, 0, 0, -7, 0 },
            {-1, -2, -3, -4, -5, -4, -3, -2, -1 },
        };

        public bool FirstClick = true;
        public int oldX = 0;
    }
}
```

```

public int oldY = 0;
public int[] gotemotigoma = { 0, 0, 0, 0, 0, 0, 0, 0, };
public int[] sentemotigoma = { 0, 0, 0, 0, 0, 0, 0, 0, };
public bool SubPic2Click = false;
public bool SubPic3Click = false;
public int utiGoma = 0;

public Form1()
{
    InitializeComponent();
    Image img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\space.jpg");
    koma = new Koma(img);
    komas.Add(koma);

    img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotekyou.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotekai.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotegin.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotekin.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img = Image.FromFile(@"c:\texsrc\将棋\gotegyoku.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotekaku.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotehi.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotefu.jpg");
    koma = new Koma(img);
    komas.Add(koma);
    img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotenarikyoku.jpg");

```

```

koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotenarikei.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\gotenarigin.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\goteuma.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\goteryu.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\goteto.jpg");
koma = new Koma(img);
komas.Add(koma);

img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\sentekyou.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\sentekei.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\sentegin.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\senteikin.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\sentegyoku.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\senteakaku.jpg");
koma = new Koma(img);
komas.Add(koma);
img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\senteihi.jpg");
koma = new Koma(img);
komas.Add(koma);

```

```

        img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\senteфу.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\sentenarikyou.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\sentenarikei.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\sentenarigin.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\senteuma.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\senteryu.jpg");
        koma = new Koma(img);
        komas.Add(koma);
        img = Image.FromFile(@"c:\texsrc\将棋\ShogiTeX\senteto.jpg");
        koma = new Koma(img);
        komas.Add(koma);
    }

    static int index(int n)
    {
        if (n >= 0)
        {
            return n;
        }
        else
        {
            return 14 - n;
        }
    }

    private void pictureBox1_Paint(object sender, PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        for (int k = 0; k < 9; k++)

```

```

    {
        for (int i = 0; i < 9; i++)
            g.DrawImage(komas[index(ban[k, i])].image,
                        50 * i, 50 * k, 50, 50);
    }
    for (int k = 0; k < 9; k++)
        g.DrawLine(Pens.Black, 0, 50*k, 450, 50*k);
    for (int k = 0; k < 9; k++)
        g.DrawLine(Pens.Black, 50 * k, 0, 50 * k, 450);

}

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    int x = e.X;
    int y = e.Y;
    int X = 0;
    int Y = 0;
    for (int i = 0; i < 9; i++)
    {
        X = 50 * i;
        if (x >= 50 * i && x < 50 * (i + 1)) break;
    }
    for (int i = 0; i < 9; i++)
    {
        Y = 50 * i;
        if (y >= 50 * i && y < 50 * (i + 1)) break;
    }

    if (e.Button == MouseButton.Left) {

        if (SubPic2Click == true)
        {
            int K2 = Y / 50;
            int I2 = X / 50;
            if (ban[K2, I2] != 0)
            {
                if (ban[K2, I2] > 0)

```

```

{
    switch (ban[K2, I2])
    {
        case 7:
        case 13:
            sentemotigoma[0] += 1; break;
        case 6:
        case 12:
            sentemotigoma[1] += 1; break;
        case 4:
            sentemotigoma[2] += 1; break;
        case 3:
        case 11:
            sentemotigoma[3] += 1; break;
        case 2:
        case 10:
            sentemotigoma[4] += 1; break;
        case 1:
        case 9:
            sentemotigoma[5] += 1; break;
        case 8:
        case 14:
            sentemotigoma[6] += 1; break;
        case 5:
            sentemotigoma[7] += 1; break;
    }
}
else
{
    switch (ban[K2, I2])
    {
        case -7:
        case -13:
            sentemotigoma[0] += 1; break;
        case -6:
        case -12:
            sentemotigoma[1] += 1; break;
        case -4:
            sentemotigoma[2] += 1; break;
    }
}

```

```

        case -3:
        case -11:
            sentemotigoma[3] += 1; break;
        case -2:
        case -10:
            sentemotigoma[4] += 1; break;
        case -1:
        case -9:
            sentemotigoma[5] += 1; break;
        case -8:
        case -14:
            sentemotigoma[6] += 1; break;
        case -5:
            sentemotigoma[7] += 1; break;
    }
}
}
else // ban[K2, I2] == 0
{
    Graphics g2 = pictureBox1.CreateGraphics();
    g2.FillRectangle(Brushes.White, X, Y, 50, 50);
}
ban[K2, I2] = utiGoma;
Graphics g = pictureBox1.CreateGraphics();
g.DrawImage(komas[index(ban[K2, I2])].image,
    50 * I2, 50 * K2, 50, 50);
SubPic2Click = false;
FirstClick = true;
pictureBox1.Refresh();
pictureBox2.Refresh();
pictureBox3.Refresh();
}
else if (SubPic3Click == true)
{
    int K2 = Y / 50;
    int I2 = X / 50;
    if (ban[K2, I2] != 0)
    {
        if (ban[K2, I2] > 0)

```



```

{
    switch (ban[K2, I2])
    {
        case 7:
        case 13:
            gotemotigoma[0] += 1; break;
        case 6:
        case 12:
            gotemotigoma[1] += 1; break;
        case 4:
            gotemotigoma[2] += 1; break;
        case 3:
        case 11:
            gotemotigoma[3] += 1; break;
        case 2:
        case 10:
            gotemotigoma[4] += 1; break;
        case 1:
        case 9:
            gotemotigoma[5] += 1; break;
        case 8:
        case 14:
            gotemotigoma[6] += 1; break;
        case 5:
            gotemotigoma[7] += 1; break;
    }
}
else
{
    switch (ban[K2, I2])
    {
        case -7:
        case -13:
            gotemotigoma[0] += 1; break;
        case -6:
        case -12:
            gotemotigoma[1] += 1; break;
        case -4:
            gotemotigoma[2] += 1; break;
    }
}

```

```

        case -3:
        case -11:
            gotemotigoma[3] += 1; break;
        case -2:
        case -10:
            gotemotigoma[4] += 1; break;
        case -1:
        case -9:
            gotemotigoma[5] += 1; break;
        case -8:
        case -14:
            gotemotigoma[6] += 1; break;
        case -5:
            gotemotigoma[7] += 1; break;
    }
}
}
else // ban[K2, I2] == 0
{
    Graphics g2 = pictureBox1.CreateGraphics();
    g2.FillRectangle(Brushes.White, X, Y, 50, 50);
}
ban[K2, I2] = utiGoma;
Graphics g = pictureBox1.CreateGraphics();
g.DrawImage(komas[index(ban[K2, I2])].image,
    50 * I2, 50 * K2, 50, 50);
SubPic3Click = false;
FirstClick = true;
pictureBox1.Refresh();
pictureBox2.Refresh();
pictureBox3.Refresh();
}
else
{
    if (FirstClick == true)
    {
        Graphics g = pictureBox1.CreateGraphics();
        Pen p = new Pen(Color.Red, 3);
        g.DrawRectangle(p, X, Y, 49, 49);
    }
}

```

```

        FirstClick = false;
        oldX = X;
        oldY = Y;
    }
    else
    {
        int K = oldY / 50;
        int I = oldX / 50;
        int K2 = Y / 50;
        int I2 = X / 50;
        if (K != K2 || I != I2)
        {
            if (ban[K2, I2] != 0)
            {
                if (ban[K2, I2] > 0)
                {
                    switch (ban[K2, I2])
                    {
                        case 7:
                        case 13:
                            sentemotigoma[0] += 1; break;
                        case 6:
                        case 12:
                            sentemotigoma[1] += 1; break;
                        case 4:
                            sentemotigoma[2] += 1; break;
                        case 3:
                        case 11:
                            sentemotigoma[3] += 1; break;
                        case 2:
                        case 10:
                            sentemotigoma[4] += 1; break;
                        case 1:
                        case 9:
                            sentemotigoma[5] += 1; break;
                        case 8:
                        case 14:
                            sentemotigoma[6] += 1; break;
                        case 5:

```

```

                                sentemotigoma[7] += 1; break;
                                }
                                }
                                else
                                {
                                    switch (ban[K2, I2])
                                    {
                                        case -7:
                                        case -13:
                                            gotemotigoma[0] += 1; break;
                                        case -6:
                                        case -12:
                                            gotemotigoma[1] += 1; break;
                                        case -4:
                                            gotemotigoma[2] += 1; break;
                                        case -3:
                                        case -11:
                                            gotemotigoma[3] += 1; break;
                                        case -2:
                                        case -10:
                                            gotemotigoma[4] += 1; break;
                                        case -1:
                                        case -9:
                                            gotemotigoma[5] += 1; break;
                                        case -8:
                                        case -14:
                                            gotemotigoma[6] += 1; break;
                                        case -5:
                                            gotemotigoma[7] += 1; break;
                                    }
                                }
                                }
                                ban[K2, I2] = ban[K, I];
                                ban[K, I] = 0;
                                Graphics g = pictureBox1.CreateGraphics();
                                g.FillRectangle(Brushes.White, oldX, oldY, 50, 50);
                                g.DrawImage(komas[index(ban[K, I])].image,
                                    50 * I, 50 * K, 50, 50);
                                g.DrawImage(komas[index(ban[K2, I2])].image,

```

```

                    50 * I2, 50 * K2, 50, 50);
            }
            else
            {
                Graphics g = pictureBox1.CreateGraphics();
                g.FillRectangle(Brushes.White, oldX, oldY, 50, 50);
                g.DrawImage(komas[index(ban[K2, I2])].image,
                    50 * I2, 50 * K2, 50, 50);
            }
            FirstClick = true;
            pictureBox1.Refresh();
            pictureBox2.Refresh();
            pictureBox3.Refresh();
        }
    }
} else if (e.Button == MouseButton.Right) {
    int K2 = Y / 50;
    int I2 = X / 50;
    ban[K2, I2] = 0;
    FirstClick = true;
    pictureBox1.Refresh();
    pictureBox2.Refresh();
    pictureBox3.Refresh();
}
}

private void pictureBox1_DoubleClick(object sender, EventArgs e)
{
    int x = ((MouseEventArgs)e).X;
    int y = ((MouseEventArgs)e).Y;
    int X = 0;
    int Y = 0;
    for (int i = 0; i < 9; i++)
    {
        X = 50 * i;
        if (x >= 50 * i && x < 50 * (i + 1)) break;
    }
    for (int i = 0; i < 9; i++)
    {

```

```

        Y = 50 * i;
        if (y >= 50 * i && y < 50 * (i + 1)) break;
    }
    if (ban[Y / 50, X / 50] > 0)
    {
        switch (ban[Y / 50, X / 50])
        {
            case 1:
            case 2:
            case 3:
                ban[Y / 50, X / 50] = ban[Y / 50, X / 50] + 8; break;
            case 6:
            case 7:
            case 8:
                ban[Y / 50, X / 50] = ban[Y / 50, X / 50] + 6; break;
        }
    }
    else if (ban[Y / 50, X / 50] < 0)
    {
        switch (ban[Y / 50, X / 50])
        {
            case -1:
            case -2:
            case -3:
                ban[Y / 50, X / 50] = ban[Y / 50, X / 50] - 8; break;
            case -6:
            case -7:
            case -8:
                ban[Y / 50, X / 50] = ban[Y / 50, X / 50] - 6; break;
        }
    }
    pictureBox1.Refresh();
}

private void pictureBox2_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.FillRectangle(Brushes.White, 0, 0, 150, 450);
    int X = 0, Y = 0;

```

```

for (int i = 0; i < 8; i++)
{
    for (int k = 0; k < sentemotigoma[i]; k++)
    {
        switch (i)
        {
            case 0:
                g.DrawImage(komas[index(-7)].image,
                    50 * X, 50 * Y, 50, 50);
                break;
            case 1:
                g.DrawImage(komas[index(-6)].image,
                    50 * X, 50 * Y, 50, 50);
                break;
            case 2:
                g.DrawImage(komas[index(-4)].image,
                    50 * X, 50 * Y, 50, 50);
                break;
            case 3:
                g.DrawImage(komas[index(-3)].image,
                    50 * X, 50 * Y, 50, 50);
                break;
            case 4:
                g.DrawImage(komas[index(-2)].image,
                    50 * X, 50 * Y, 50, 50);
                break;
            case 5:
                g.DrawImage(komas[index(-1)].image,
                    50 * X, 50 * Y, 50, 50);
                break;
            case 6:
                g.DrawImage(komas[index(-8)].image,
                    50 * X, 50 * Y, 50, 50);
                break;
            case 7:
                g.DrawImage(komas[index(-5)].image,
                    50 * X, 50 * Y, 50, 50);
                break;
        }
    }
}

```

```

        if (Y < 8)
        {
            Y += 1;
        }
        else
        {
            X += 1;
            Y = 0;
        }
    }
}

private void pictureBox3_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.FillRectangle(Brushes.White, 0, 0, 150, 450);
    int X = 0, Y = 0;
    for (int i = 0; i < 8; i++)
    {
        for (int k = 0; k < gotemotigoma[i]; k++)
        {
            switch (i)
            {
                case 0:
                    g.DrawImage(komas[index(7)].image, 50 * X,
                                50 * (8 - Y), 50, 50);
                    break;
                case 1:
                    g.DrawImage(komas[index(6)].image, 50 * X,
                                50 * (8 - Y), 50, 50);
                    break;
                case 2:
                    g.DrawImage(komas[index(4)].image, 50 * X,
                                50 * (8 - Y), 50, 50);
                    break;
                case 3:
                    g.DrawImage(komas[index(3)].image, 50 * X,
                                50 * (8 - Y), 50, 50);

```



```

        break;
    case 4:
        g.DrawImage(komas[index(2)].image, 50 * X,
                    50 * (8 - Y), 50, 50);
        break;
    case 5:
        g.DrawImage(komas[index(1)].image, 50 * X,
                    50 * (8 - Y), 50, 50);
        break;
    case 6:
        g.DrawImage(komas[index(8)].image, 50 * X,
                    50 * (8 - Y), 50, 50);
        break;
    case 7:
        g.DrawImage(komas[index(5)].image, 50 * X,
                    50 * (8 - Y), 50, 50);
        break;
    }
    if (Y < 8)
    {
        Y += 1;
    }
    else
    {
        X += 1;
        Y = 0;
    }
}

}

static string ChangeKoma(int k)
{
    string result = "";
    if (k > 0)
    {
        result = "{\\gote} ";
    }
}

```

```

else
{
    result = "{\\sente} ";
}
switch (Math.Abs(k))
{
    case 5:
        result += "{\\gyoku}";
        break;
    case 7:
        result += "{\\hi}";
        break;
    case 13:
        result += "{\\ryu}";
        break;
    case 6:
        result += "{\\kaku}";
        break;
    case 12:
        result += "{\\uma}";
        break;
    case 4:
        result += "{\\kin}";
        break;
    case 3:
        result += "{\\gin}";
        break;
    case 11:
        result += "{\\narigin}";
        break;
    case 2:
        result += "{\\kei}";
        break;
    case 10:
        result += "{\\narikei}";
        break;
    case 1:
        result += "{\\kyou}";
        break;
}

```

```

        case 9:
            result += "{\\narikyou}";
            break;
        case 8:
            result += "{\\fu}";
            break;
        case 14:
            result += "{\\tokin}";
            break;
    }
    return result;
}

private void teXMacroSaveSToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    SaveFileDialog dlg = new SaveFileDialog();
    dlg.Title = "保存するファイル名を入力してください。"; // ダイアログのタイトル。
    if (dlg.ShowDialog() == DialogResult.OK)
    {
        System.IO.Stream stream;
        stream = dlg.OpenFile();
        if (stream != null)
        {
            //ファイルに書き込む
            System.IO.StreamWriter sw =
                new System.IO.StreamWriter(stream);
            sw.WriteLine("\\documentclass[10pt, a4paper]{ jarticle}");
            sw.WriteLine("\\usepackage[dvips]{ graphicx}");
            sw.WriteLine("\\usepackage[dvips]{ color}");
            sw.WriteLine("\\usepackage{ myshogi}");
            sw.WriteLine("\\voffset 0pt%ヘッダ上端までの上余白
(実際には1インチ+\\voffsetの上余白となる)");
            sw.WriteLine("\\hoffset -15pt%文章領域までの左余白
(実際には1インチ+\\hoffsetの左余白となる)");
            sw.WriteLine("\\topmargin -20pt%上余白からヘッダ上
端までの余白");
            sw.WriteLine("\\topskip 0pt%文章領域内の本文開始位
置までの高さ");

```

```

        sw.WriteLine("\\oddsidemargin -20pt%奇数ページのマー
ジン (左余白から文章領域の左端までの余白)");
        sw.WriteLine("\\textwidth 500pt%文章領域の幅 (A4 サ
イズ縦長の文章領域の幅、176mm=500pt)");
        sw.WriteLine("\\textheight 264mm%文章領域の高さ (A4
サイズ縦長の文章領域の高さ、264mm=750pt)");
        sw.WriteLine("\\footskip 20pt%文章領域の下端からフ
ッタの下端までの高さ");
        sw.WriteLine("\\marginparsep 0pt%文章領域の右端か
ら欄外傍注の左端の幅");
        sw.WriteLine("\\marginparwidth 0pt%傍注の幅");
        sw.WriteLine("\\marginparpush 0pt%傍注間の高さ");
        sw.WriteLine("");
        sw.WriteLine("\\begin{document}");
        sw.WriteLine("");
        sw.WriteLine("\\huge");
        sw.WriteLine("");
        sw.WriteLine("");

// \TeX の棋譜を出力
sw.WriteLine("\\begin{center}");
sw.WriteLine("\\begin{myshogi}[1.9]");
sw.WriteLine(" \\banmen");
var motigoma = " \\mochigoma{\\sente}";
for (int k = 0; k < 7; k++)
{
    motigoma += " {";
    motigoma += sentemotigoma[k].ToString();
    motigoma += "}";
}
sw.WriteLine(motigoma);
motigoma = " \\mochigoma{\\gote}";
for (int k = 0; k < 7; k++)
{
    motigoma += " {";
    motigoma += gotemotigoma[k].ToString();
    motigoma += "}";
}
sw.WriteLine(motigoma);

```

```

        for (int i = 0; i < 9; i++)
        {
            for (int k = 0; k < 9; k++)
            {
                if (ban[i, k] != 0)
                {
                    var te = " \\koma{";
                    te += (9-k).ToString();
                    te += (i+1).ToString();
                    te += "} ";
                    te += ChangeKoma(ban[i, k]);
                    sw.WriteLine(te);
                }
            }
        }
        sw.WriteLine("\\end{myshogi}");
        sw.WriteLine("\\end{center}"); //閉じる
        sw.WriteLine("");
        sw.WriteLine("");
        sw.WriteLine("");
        sw.WriteLine("\\end{document}");
        sw.Close();
        stream.Close();
    }
}

```

```

public int getMotigoma(int k, int ind)
{
    int result = 0;
    if (ind == 0)
    {
        switch(k)
        {
            case 0:
                result = -7; break;
            case 1:
                result = -6; break;
            case 2:

```

```

        result = -4; break;
    case 3:
        result = -3; break;
    case 4:
        result = -2; break;
    case 5:
        result = -1; break;
    case 6:
        result = -8; break;
    case 7:
        result = -5; break;
    }
} else
{
    switch (k)
    {
        case 0:
            result = 7; break;
        case 1:
            result = 6; break;
        case 2:
            result = 4; break;
        case 3:
            result = 3; break;
        case 4:
            result = 2; break;
        case 5:
            result = 1; break;
        case 6:
            result = 8; break;
        case 7:
            result = 5; break;
    }
}
return result;
}

private void pictureBox2_MouseDown(object sender, MouseEventArgs e)
{
    int x = e.X;

```

```

int y = e.Y;
int X = 0;
int Y = 0;
for (int i = 0; i < 3; i++)
{
    X = 50 * i;
    if (x >= 50 * i && x < 50 * (i + 1)) break;
}
for (int i = 0; i < 9; i++)
{
    Y = 50 * i;
    if (y >= 50 * i && y < 50 * (i + 1)) break;
}
int I = X / 50;
int K = Y / 50;
utiGoma = 0;
if (I == 0)
{
    if (K < sentemotigoma.Aggregate((sum, n) => sum+n))
    {
        int k = 0, n = 0;
        for (k=0; k<8; k++)
        {
            n += sentemotigoma[k];
            if (n >= K+1) break;
        }
        utiGoma = getMotigoma(k, 0);
        sentemotigoma[k]--;
        /*
        string msg = "K="+K.ToString()+"k="+k.ToString()+" ":"
            +utiGoma.ToString();
        MessageBox.Show(msg,
            "エラー",
            MessageBoxButtons.OK);
        msg = "";
        for (k=0; k<7; k++)
        {
            msg += sentemotigoma[k].ToString() + " ";
        }
    }
}

```

```

        MessageBox.Show(msg,
            "エラー",
            MessageBoxButtons.OK);
        */
    }
} else if (I==1)
{
    if (K + 9 < sentemotigoma.Aggregate((sum, n) => sum + n))
    {
        int k = 0, n = 0;
        for (k = 0; k < 8; k++)
        {
            n += sentemotigoma[k];
            if (n >= K+1 + 9) break;
        }
        utiGoma = getMotigoma(k, 0);
        sentemotigoma[k]--;
    }
} else
{
    if (K + 18 < sentemotigoma.Aggregate((sum, n) => sum + n))
    {
        int k = 0, n = 0;
        for (k = 0; k < 8; k++)
        {
            n += sentemotigoma[k];
            if (n >= K+1 + 18) break;
        }
        utiGoma = getMotigoma(k, 0);
        sentemotigoma[k]--;
    }
}
Graphics g = pictureBox2.CreateGraphics();
Pen p = new Pen(Color.Red, 3);
g.DrawRectangle(p, X, Y, 49, 49);
SubPic2Click = true;
}

private void pictureBox3_MouseDown(object sender, MouseEventArgs e)

```



```

{
    int x = e.X;
    int y = e.Y;
    int X = 0;
    int Y = 0;
    for (int i = 0; i < 3; i++)
    {
        X = 50 * i;
        if (x >= 50 * i && x < 50 * (i + 1)) break;
    }
    for (int i = 0; i < 9; i++)
    {
        Y = 50 * i;
        if (y >= 50 * i && y < 50 * (i + 1)) break;
    }
    int I = X / 50;
    int K = Y / 50;
    utiGoma = 0;
    if (I == 0)
    {
        if (8 - K < gotemotigoma.Aggregate((sum, n) => sum + n))
        {
            int k = 0, n = 0;
            for (k = 0; k < 8; k++)
            {
                n += gotemotigoma[k];
                if (n >= 9-K) break;
            }
            utiGoma = getMotigoma(k, 1);
            gotemotigoma[k]--;
        }
    }
    else if (I == 1)
    {
        if (8-K + 9 < gotemotigoma.Aggregate((sum, n) => sum + n))
        {
            int k = 0, n = 0;
            for (k = 0; k < 8; k++)
            {

```

```

        n += gotemotigoma[k];
        if (n >= 9-K + 9) break;
    }
    utiGoma = getMotigoma(k, 1);
    gotemotigoma[k]--;
}
}
else
{
    if (8 - K + 18 < gotemotigoma.Aggregate((sum, n) => sum + n))
    {
        int k = 0, n = 0;
        for (k = 0; k < 8; k++)
        {
            n += gotemotigoma[k];
            if (n >= 9-K + 18) break;
        }
        utiGoma = getMotigoma(k, 1);
        gotemotigoma[k]--;
        string msg = utiGoma.ToString();
    }
}
}
Graphics g = pictureBox3.CreateGraphics();
Pen p = new Pen(Color.Red, 3);
g.DrawRectangle(p, X, Y, 49, 49);
SubPic3Click = true;
}

private void 平手ToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    ban[0, 0] = 1;
    ban[0, 1] = 2;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 1;
}

```

```

ban[1, 0] = 0;
ban[1, 1] = 7;
ban[1, 2] = 0;
ban[1, 3] = 0;
ban[1, 4] = 0;
ban[1, 5] = 0;
ban[1, 6] = 0;
ban[1, 7] = 6;
ban[1, 8] = 0;
for (int k = 0; k < 9; k++)
    ban[2, k] = 8;
for (int k = 3; k < 6; k++)
    for (int i = 0; i < 9; i++)
        ban[k, i] = 0;
for (int k = 0; k < 9; k++)
    ban[6, k] = -8;
ban[7, 0] = 0;
ban[7, 1] = -6;
ban[7, 2] = 0;
ban[7, 3] = 0;
ban[7, 4] = 0;
ban[7, 5] = 0;
ban[7, 6] = 0;
ban[7, 7] = -7;
ban[7, 8] = 0;
ban[8, 0] = -1;
ban[8, 1] = -2;
ban[8, 2] = -3;
ban[8, 3] = -4;
ban[8, 4] = -5;
ban[8, 5] = -4;
ban[8, 6] = -3;
ban[8, 7] = -2;
ban[8, 8] = -1;
for (int k = 0; k < 8; k++)
{
    gotemotigoma[k] = 0;
    sentemotigoma[k] = 0;
}

```

```

        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

```

```

private void 香落ちToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 1;
    ban[0, 1] = 2;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 0;
    ban[1, 0] = 0;
    ban[1, 1] = 7;
    ban[1, 2] = 0;
    ban[1, 3] = 0;
    ban[1, 4] = 0;
    ban[1, 5] = 0;
    ban[1, 6] = 0;
    ban[1, 7] = 6;
    ban[1, 8] = 0;
    for (int k = 0; k < 9; k++)
        ban[2, k] = 8;
    for (int k = 3; k < 6; k++)
        for (int i = 0; i < 9; i++)
            ban[k, i] = 0;
    for (int k = 0; k < 9; k++)
        ban[6, k] = -8;
    ban[7, 0] = 0;
    ban[7, 1] = -6;
    ban[7, 2] = 0;
    ban[7, 3] = 0;
}

```

```

        ban[7, 4] = 0;
        ban[7, 5] = 0;
        ban[7, 6] = 0;
        ban[7, 7] = -7;
        ban[7, 8] = 0;
        ban[8, 0] = -1;
        ban[8, 1] = -2;
        ban[8, 2] = -3;
        ban[8, 3] = -4;
        ban[8, 4] = -5;
        ban[8, 5] = -4;
        ban[8, 6] = -3;
        ban[8, 7] = -2;
        ban[8, 8] = -1;
        for (int k=0; k<8; k++)
        {
            gotemotigoma[k] = 0;
            sentemotigoma[k] = 0;
        }
        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

private void 角行落ちToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 1;
    ban[0, 1] = 2;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 1;
    ban[1, 0] = 0;

```

```

ban[1, 1] = 7;
ban[1, 2] = 0;
ban[1, 3] = 0;
ban[1, 4] = 0;
ban[1, 5] = 0;
ban[1, 6] = 0;
ban[1, 7] = 0;
ban[1, 8] = 0;
for (int k = 0; k < 9; k++)
    ban[2, k] = 8;
for (int k = 3; k < 6; k++)
    for (int i = 0; i < 9; i++)
        ban[k, i] = 0;
for (int k = 0; k < 9; k++)
    ban[6, k] = -8;
ban[7, 0] = 0;
ban[7, 1] = -6;
ban[7, 2] = 0;
ban[7, 3] = 0;
ban[7, 4] = 0;
ban[7, 5] = 0;
ban[7, 6] = 0;
ban[7, 7] = -7;
ban[7, 8] = 0;
ban[8, 0] = -1;
ban[8, 1] = -2;
ban[8, 2] = -3;
ban[8, 3] = -4;
ban[8, 4] = -5;
ban[8, 5] = -4;
ban[8, 6] = -3;
ban[8, 7] = -2;
ban[8, 8] = -1;
for (int k = 0; k < 8; k++)
{
    gotemotigoma[k] = 0;
    sentemotigoma[k] = 0;
}
pictureBox1.Refresh();

```

```

        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

private void 飛車落ちToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 1;
    ban[0, 1] = 2;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 1;
    ban[1, 0] = 0;
    ban[1, 1] = 0;
    ban[1, 2] = 0;
    ban[1, 3] = 0;
    ban[1, 4] = 0;
    ban[1, 5] = 0;
    ban[1, 6] = 0;
    ban[1, 7] = 6;
    ban[1, 8] = 0;
    for (int k = 0; k < 9; k++)
        ban[2, k] = 8;
    for (int k = 3; k < 6; k++)
        for (int i = 0; i < 9; i++)
            ban[k, i] = 0;
    for (int k = 0; k < 9; k++)
        ban[6, k] = -8;
    ban[7, 0] = 0;
    ban[7, 1] = -6;
    ban[7, 2] = 0;
    ban[7, 3] = 0;
    ban[7, 4] = 0;

```

```

        ban[7, 5] = 0;
        ban[7, 6] = 0;
        ban[7, 7] = -7;
        ban[7, 8] = 0;
        ban[8, 0] = -1;
        ban[8, 1] = -2;
        ban[8, 2] = -3;
        ban[8, 3] = -4;
        ban[8, 4] = -5;
        ban[8, 5] = -4;
        ban[8, 6] = -3;
        ban[8, 7] = -2;
        ban[8, 8] = -1;
        for (int k = 0; k < 8; k++)
        {
            gotemotigoma[k] = 0;
            sentemotigoma[k] = 0;
        }
        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

```

```

private void 飛香落ちToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 1;
    ban[0, 1] = 2;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 0;
    ban[1, 0] = 0;
    ban[1, 1] = 0;
}

```



```

ban[1, 2] = 0;
ban[1, 3] = 0;
ban[1, 4] = 0;
ban[1, 5] = 0;
ban[1, 6] = 0;
ban[1, 7] = 6;
ban[1, 8] = 0;
for (int k = 0; k < 9; k++)
    ban[2, k] = 8;
for (int k = 3; k < 6; k++)
    for (int i = 0; i < 9; i++)
        ban[k, i] = 0;
for (int k = 0; k < 9; k++)
    ban[6, k] = -8;
ban[7, 0] = 0;
ban[7, 1] = -6;
ban[7, 2] = 0;
ban[7, 3] = 0;
ban[7, 4] = 0;
ban[7, 5] = 0;
ban[7, 6] = 0;
ban[7, 7] = -7;
ban[7, 8] = 0;
ban[8, 0] = -1;
ban[8, 1] = -2;
ban[8, 2] = -3;
ban[8, 3] = -4;
ban[8, 4] = -5;
ban[8, 5] = -4;
ban[8, 6] = -3;
ban[8, 7] = -2;
ban[8, 8] = -1;
for (int k = 0; k < 8; k++)
{
    gotemotigoma[k] = 0;
    sentemotigoma[k] = 0;
}
pictureBox1.Refresh();
pictureBox2.Refresh();

```

```

        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

```

```

private void 二枚落ちToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 1;
    ban[0, 1] = 2;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 1;
    ban[1, 0] = 0;
    ban[1, 1] = 0;
    ban[1, 2] = 0;
    ban[1, 3] = 0;
    ban[1, 4] = 0;
    ban[1, 5] = 0;
    ban[1, 6] = 0;
    ban[1, 7] = 0;
    ban[1, 8] = 0;
    for (int k = 0; k < 9; k++)
        ban[2, k] = 8;
    for (int k = 3; k < 6; k++)
        for (int i = 0; i < 9; i++)
            ban[k, i] = 0;
    for (int k = 0; k < 9; k++)
        ban[6, k] = -8;
    ban[7, 0] = 0;
    ban[7, 1] = -6;
    ban[7, 2] = 0;
    ban[7, 3] = 0;
    ban[7, 4] = 0;
    ban[7, 5] = 0;
}

```

```

        ban[7, 6] = 0;
        ban[7, 7] = -7;
        ban[7, 8] = 0;
        ban[8, 0] = -1;
        ban[8, 1] = -2;
        ban[8, 2] = -3;
        ban[8, 3] = -4;
        ban[8, 4] = -5;
        ban[8, 5] = -4;
        ban[8, 6] = -3;
        ban[8, 7] = -2;
        ban[8, 8] = -1;
        for (int k = 0; k < 8; k++)
        {
            gotemotigoma[k] = 0;
            sentemotigoma[k] = 0;
        }
        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

```

```

private void 上手左香引ToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 1;
    ban[0, 1] = 2;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 0;
    ban[1, 0] = 0;
    ban[1, 1] = 0;
    ban[1, 2] = 0;
}

```

```

ban[1, 3] = 0;
ban[1, 4] = 0;
ban[1, 5] = 0;
ban[1, 6] = 0;
ban[1, 7] = 0;
ban[1, 8] = 0;
for (int k = 0; k < 9; k++)
    ban[2, k] = 8;
for (int k = 3; k < 6; k++)
    for (int i = 0; i < 9; i++)
        ban[k, i] = 0;
for (int k = 0; k < 9; k++)
    ban[6, k] = -8;
ban[7, 0] = 0;
ban[7, 1] = -6;
ban[7, 2] = 0;
ban[7, 3] = 0;
ban[7, 4] = 0;
ban[7, 5] = 0;
ban[7, 6] = 0;
ban[7, 7] = -7;
ban[7, 8] = 0;
ban[8, 0] = -1;
ban[8, 1] = -2;
ban[8, 2] = -3;
ban[8, 3] = -4;
ban[8, 4] = -5;
ban[8, 5] = -4;
ban[8, 6] = -3;
ban[8, 7] = -2;
ban[8, 8] = -1;
for (int k = 0; k < 8; k++)
{
    gotemotigoma[k] = 0;
    sentemotigoma[k] = 0;
}
pictureBox1.Refresh();
pictureBox2.Refresh();
pictureBox3.Refresh();

```

```

        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

private void 上手右香引ToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 0;
    ban[0, 1] = 2;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 1;
    ban[1, 0] = 0;
    ban[1, 1] = 0;
    ban[1, 2] = 0;
    ban[1, 3] = 0;
    ban[1, 4] = 0;
    ban[1, 5] = 0;
    ban[1, 6] = 0;
    ban[1, 7] = 0;
    ban[1, 8] = 0;
    for (int k = 0; k < 9; k++)
        ban[2, k] = 8;
    for (int k = 3; k < 6; k++)
        for (int i = 0; i < 9; i++)
            ban[k, i] = 0;
    for (int k = 0; k < 9; k++)
        ban[6, k] = -8;
    ban[7, 0] = 0;
    ban[7, 1] = -6;
    ban[7, 2] = 0;
    ban[7, 3] = 0;
    ban[7, 4] = 0;
    ban[7, 5] = 0;
}

```

```

        ban[7, 6] = 0;
        ban[7, 7] = -7;
        ban[7, 8] = 0;
        ban[8, 0] = -1;
        ban[8, 1] = -2;
        ban[8, 2] = -3;
        ban[8, 3] = -4;
        ban[8, 4] = -5;
        ban[8, 5] = -4;
        ban[8, 6] = -3;
        ban[8, 7] = -2;
        ban[8, 8] = -1;
        for (int k = 0; k < 8; k++)
        {
            gotemotigoma[k] = 0;
            sentemotigoma[k] = 0;
        }
        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;

    }

    private void 四枚落ToolStripMenuItem_Click(object sender, EventArgs e)
    {
        ban[0, 0] = 0;
        ban[0, 1] = 0;
        ban[0, 2] = 3;
        ban[0, 3] = 4;
        ban[0, 4] = 5;
        ban[0, 5] = 4;
        ban[0, 6] = 3;
        ban[0, 7] = 0;
        ban[0, 8] = 0;
        ban[1, 0] = 0;
        ban[1, 1] = 0;
    }

```

```

ban[1, 2] = 0;
ban[1, 3] = 0;
ban[1, 4] = 0;
ban[1, 5] = 0;
ban[1, 6] = 0;
ban[1, 7] = 0;
ban[1, 8] = 0;
for (int k = 0; k < 9; k++)
    ban[2, k] = 8;
for (int k = 3; k < 6; k++)
    for (int i = 0; i < 9; i++)
        ban[k, i] = 0;
for (int k = 0; k < 9; k++)
    ban[6, k] = -8;
ban[7, 0] = 0;
ban[7, 1] = -6;
ban[7, 2] = 0;
ban[7, 3] = 0;
ban[7, 4] = 0;
ban[7, 5] = 0;
ban[7, 6] = 0;
ban[7, 7] = -7;
ban[7, 8] = 0;
ban[8, 0] = -1;
ban[8, 1] = -2;
ban[8, 2] = -3;
ban[8, 3] = -4;
ban[8, 4] = -5;
ban[8, 5] = -4;
ban[8, 6] = -3;
ban[8, 7] = -2;
ban[8, 8] = -1;
for (int k = 0; k < 8; k++)
{
    gotemotigoma[k] = 0;
    sentemotigoma[k] = 0;
}
pictureBox1.Refresh();
pictureBox2.Refresh();

```

```

        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

private void 上手右桂引ToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 0;
    ban[0, 1] = 0;
    ban[0, 2] = 3;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 3;
    ban[0, 7] = 2;
    ban[0, 8] = 0;
    ban[1, 0] = 0;
    ban[1, 1] = 0;
    ban[1, 2] = 0;
    ban[1, 3] = 0;
    ban[1, 4] = 0;
    ban[1, 5] = 0;
    ban[1, 6] = 0;
    ban[1, 7] = 0;
    ban[1, 8] = 0;
    for (int k = 0; k < 9; k++)
        ban[2, k] = 8;
    for (int k = 3; k < 6; k++)
        for (int i = 0; i < 9; i++)
            ban[k, i] = 0;
    for (int k = 0; k < 9; k++)
        ban[6, k] = -8;
    ban[7, 0] = 0;
    ban[7, 1] = -6;
    ban[7, 2] = 0;
    ban[7, 3] = 0;
    ban[7, 4] = 0;

```



```

        ban[7, 5] = 0;
        ban[7, 6] = 0;
        ban[7, 7] = -7;
        ban[7, 8] = 0;
        ban[8, 0] = -1;
        ban[8, 1] = -2;
        ban[8, 2] = -3;
        ban[8, 3] = -4;
        ban[8, 4] = -5;
        ban[8, 5] = -4;
        ban[8, 6] = -3;
        ban[8, 7] = -2;
        ban[8, 8] = -1;
        for (int k = 0; k < 8; k++)
        {
            gotemotigoma[k] = 0;
            sentemotigoma[k] = 0;
        }
        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

    private void 上手左桂引ToolStripMenuItem_Click(object sender, EventArgs e)
    {
        ban[0, 0] = 0;
        ban[0, 1] = 2;
        ban[0, 2] = 3;
        ban[0, 3] = 4;
        ban[0, 4] = 5;
        ban[0, 5] = 4;
        ban[0, 6] = 3;
        ban[0, 7] = 0;
        ban[0, 8] = 0;
        ban[1, 0] = 0;
    }

```

```

ban[1, 1] = 0;
ban[1, 2] = 0;
ban[1, 3] = 0;
ban[1, 4] = 0;
ban[1, 5] = 0;
ban[1, 6] = 0;
ban[1, 7] = 0;
ban[1, 8] = 0;
for (int k = 0; k < 9; k++)
    ban[2, k] = 8;
for (int k = 3; k < 6; k++)
    for (int i = 0; i < 9; i++)
        ban[k, i] = 0;
for (int k = 0; k < 9; k++)
    ban[6, k] = -8;
ban[7, 0] = 0;
ban[7, 1] = -6;
ban[7, 2] = 0;
ban[7, 3] = 0;
ban[7, 4] = 0;
ban[7, 5] = 0;
ban[7, 6] = 0;
ban[7, 7] = -7;
ban[7, 8] = 0;
ban[8, 0] = -1;
ban[8, 1] = -2;
ban[8, 2] = -3;
ban[8, 3] = -4;
ban[8, 4] = -5;
ban[8, 5] = -4;
ban[8, 6] = -3;
ban[8, 7] = -2;
ban[8, 8] = -1;
for (int k = 0; k < 8; k++)
{
    gotemotigoma[k] = 0;
    sentemotigoma[k] = 0;
}
pictureBox1.Refresh();

```

```

        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

    private void 六枚落ToolStripMenuItem_Click(object sender, EventArgs e)
    {
        ban[0, 0] = 0;
        ban[0, 1] = 0;
        ban[0, 2] = 3;
        ban[0, 3] = 4;
        ban[0, 4] = 5;
        ban[0, 5] = 4;
        ban[0, 6] = 3;
        ban[0, 7] = 0;
        ban[0, 8] = 0;
        ban[1, 0] = 0;
        ban[1, 1] = 0;
        ban[1, 2] = 0;
        ban[1, 3] = 0;
        ban[1, 4] = 0;
        ban[1, 5] = 0;
        ban[1, 6] = 0;
        ban[1, 7] = 0;
        ban[1, 8] = 0;
        for (int k = 0; k < 9; k++)
            ban[2, k] = 8;
        for (int k = 3; k < 6; k++)
            for (int i = 0; i < 9; i++)
                ban[k, i] = 0;
        for (int k = 0; k < 9; k++)
            ban[6, k] = -8;
        ban[7, 0] = 0;
        ban[7, 1] = -6;
        ban[7, 2] = 0;
        ban[7, 3] = 0;
    }

```

```

        ban[7, 4] = 0;
        ban[7, 5] = 0;
        ban[7, 6] = 0;
        ban[7, 7] = -7;
        ban[7, 8] = 0;
        ban[8, 0] = -1;
        ban[8, 1] = -2;
        ban[8, 2] = -3;
        ban[8, 3] = -4;
        ban[8, 4] = -5;
        ban[8, 5] = -4;
        ban[8, 6] = -3;
        ban[8, 7] = -2;
        ban[8, 8] = -1;
        for (int k = 0; k < 8; k++)
        {
            gotemotigoma[k] = 0;
            sentemotigoma[k] = 0;
        }
        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;

    }

    private void 步三步ToolStripMenuItem_Click(object sender, EventArgs e)
    {
        ban[0, 0] = 0;
        ban[0, 1] = 0;
        ban[0, 2] = 0;
        ban[0, 3] = 0;
        ban[0, 4] = 5;
        ban[0, 5] = 0;
        ban[0, 6] = 0;
        ban[0, 7] = 0;
        ban[0, 8] = 0;
    }

```

```

ban[1, 0] = 0;
ban[1, 1] = 0;
ban[1, 2] = 0;
ban[1, 3] = 0;
ban[1, 4] = 0;
ban[1, 5] = 0;
ban[1, 6] = 0;
ban[1, 7] = 0;
ban[1, 8] = 0;
for (int k = 0; k < 9; k++)
    ban[2, k] = 0;
for (int k = 3; k < 6; k++)
    for (int i = 0; i < 9; i++)
        ban[k, i] = 0;
for (int k = 0; k < 9; k++)
    ban[6, k] = -8;
ban[7, 0] = 0;
ban[7, 1] = -6;
ban[7, 2] = 0;
ban[7, 3] = 0;
ban[7, 4] = 0;
ban[7, 5] = 0;
ban[7, 6] = 0;
ban[7, 7] = -7;
ban[7, 8] = 0;
ban[8, 0] = -1;
ban[8, 1] = -2;
ban[8, 2] = -3;
ban[8, 3] = -4;
ban[8, 4] = -5;
ban[8, 5] = -4;
ban[8, 6] = -3;
ban[8, 7] = -2;
ban[8, 8] = -1;
for (int k = 0; k < 8; k++)
{
    gotemotigoma[k] = 0;
    sentemotigoma[k] = 0;
}

```

```

        gotemotigoma[6] = 3;
        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;
    }

private void 八枚落ToolStripMenuItem_Click(object sender, EventArgs e)
{
    ban[0, 0] = 0;
    ban[0, 1] = 0;
    ban[0, 2] = 0;
    ban[0, 3] = 4;
    ban[0, 4] = 5;
    ban[0, 5] = 4;
    ban[0, 6] = 0;
    ban[0, 7] = 0;
    ban[0, 8] = 0;
    ban[1, 0] = 0;
    ban[1, 1] = 0;
    ban[1, 2] = 0;
    ban[1, 3] = 0;
    ban[1, 4] = 0;
    ban[1, 5] = 0;
    ban[1, 6] = 0;
    ban[1, 7] = 0;
    ban[1, 8] = 0;
    for (int k = 0; k < 9; k++)
        ban[2, k] = 8;
    for (int k = 3; k < 6; k++)
        for (int i = 0; i < 9; i++)
            ban[k, i] = 0;
    for (int k = 0; k < 9; k++)
        ban[6, k] = -8;
    ban[7, 0] = 0;
    ban[7, 1] = -6;
}

```

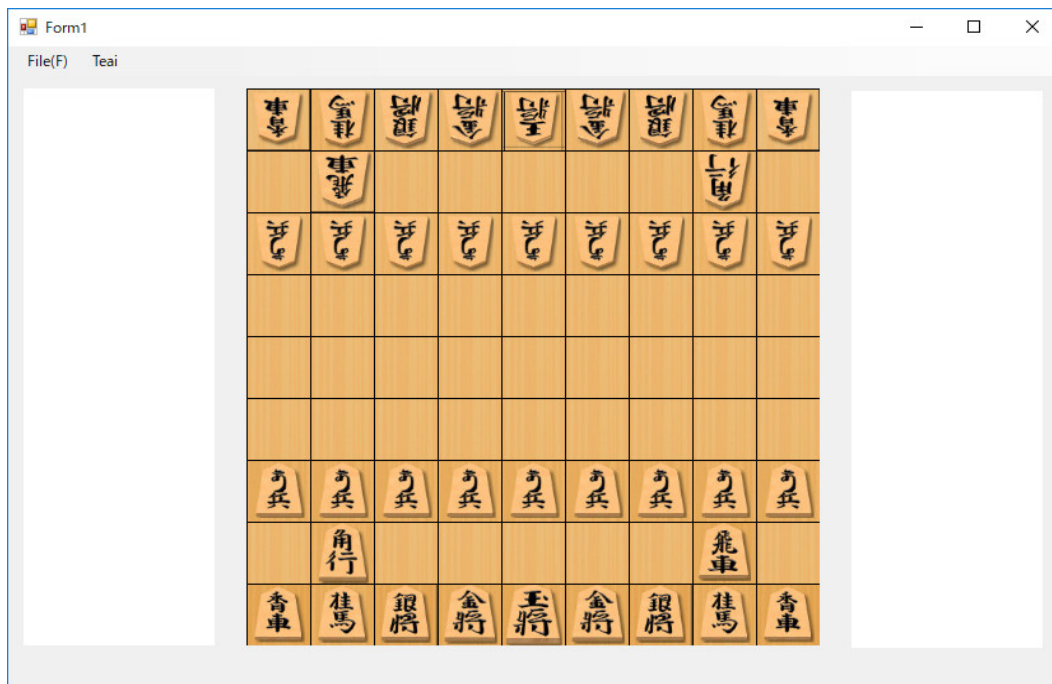
```

        ban[7, 2] = 0;
        ban[7, 3] = 0;
        ban[7, 4] = 0;
        ban[7, 5] = 0;
        ban[7, 6] = 0;
        ban[7, 7] = -7;
        ban[7, 8] = 0;
        ban[8, 0] = -1;
        ban[8, 1] = -2;
        ban[8, 2] = -3;
        ban[8, 3] = -4;
        ban[8, 4] = -5;
        ban[8, 5] = -4;
        ban[8, 6] = -3;
        ban[8, 7] = -2;
        ban[8, 8] = -1;
        for (int k = 0; k < 8; k++)
        {
            gotemotigoma[k] = 0;
            sentemotigoma[k] = 0;
        }
        pictureBox1.Refresh();
        pictureBox2.Refresh();
        pictureBox3.Refresh();
        FirstClick = true;
        SubPic2Click = false;
        SubPic3Click = false;

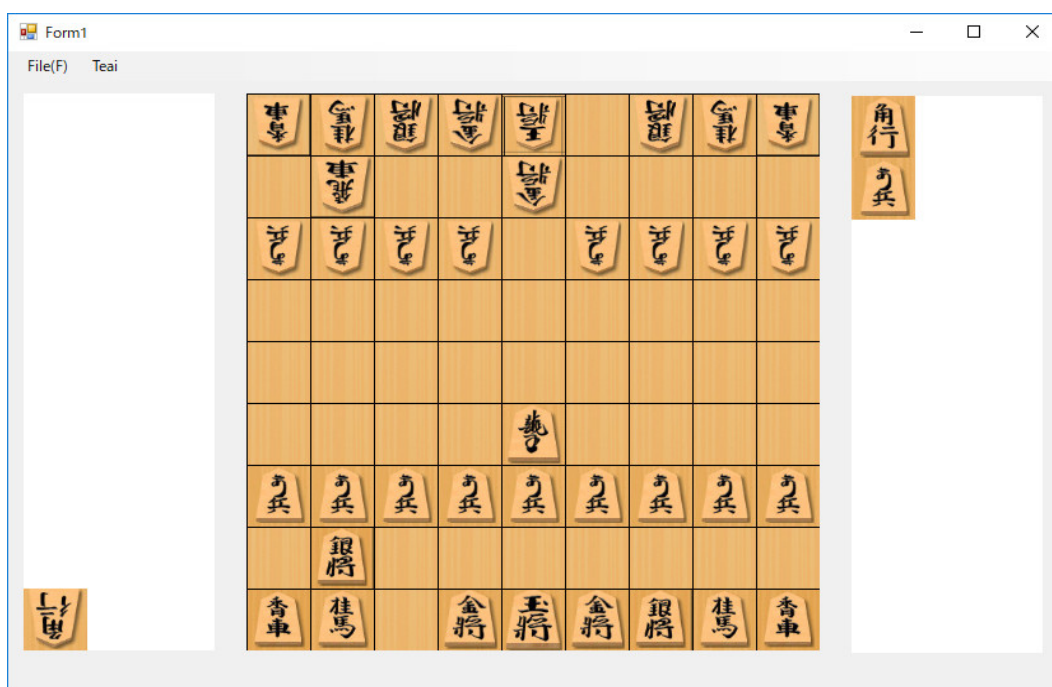
    }
}

```

です。
実行すると

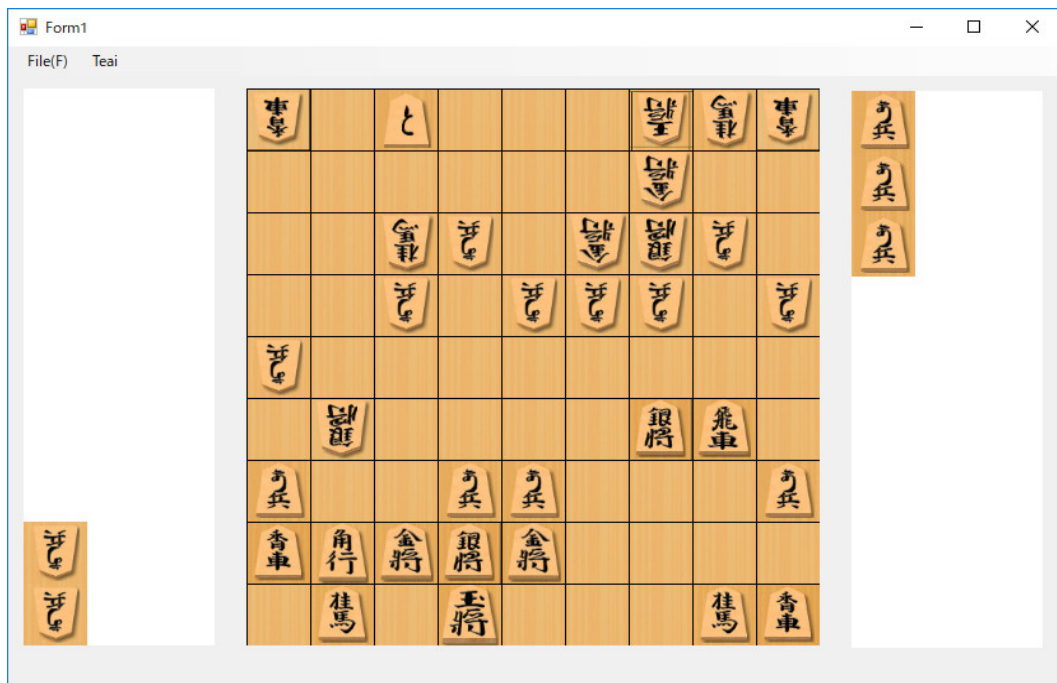


です。盤面を左クリックして局面を作れます。移動したい駒を左クリックし、移動したい場所を左クリックします。移動先に駒があれば、持ち駒になります。駒台の駒を左クリックして、打ちたい場所を左クリックすると駒を打ちことができます。盤面の駒を駒台に移動したいときは、まず駒台の開いてる場所を左クリックしてから駒台に移動したい駒を左クリックします。将棋のルールに縛られません。5手で



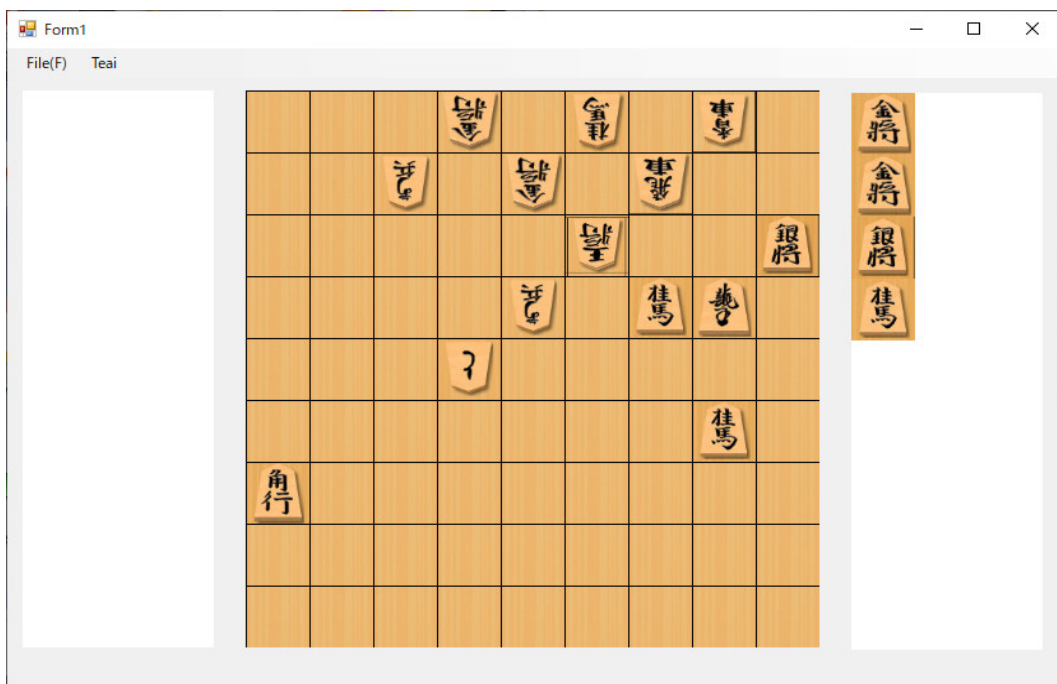
の局面が作れます。持ち駒を左クリックして打つこともできます。自分の使いやすいように修正してください。

「屋根裏王」との対局で、



の局面で、間違えて一気に悪くなりましたが、このような局面を $\text{T}_{\text{E}}\text{X}$ を使って A6 のカードに印刷して、どうすればよかったかじっくり考えたかったです。

更に、右クリックすれば、盤面から駒が消えますから



のような詰将棋の画面（伊藤宗看著「将棋無双」の第一番）や必至の画面を作ることが出来ます。但し、 $\text{\LaTeX}$ では

	9	8	7	6	5	4	3	2	1	
持駒			歩	金	歩	金	銀	皇		一
持駒			歩		金	王	桂		銀	二
持駒					歩		桂	龍		三
持駒				マ						四
持駒								桂		五
持駒	角									六
持駒										七
持駒										八
持駒										九

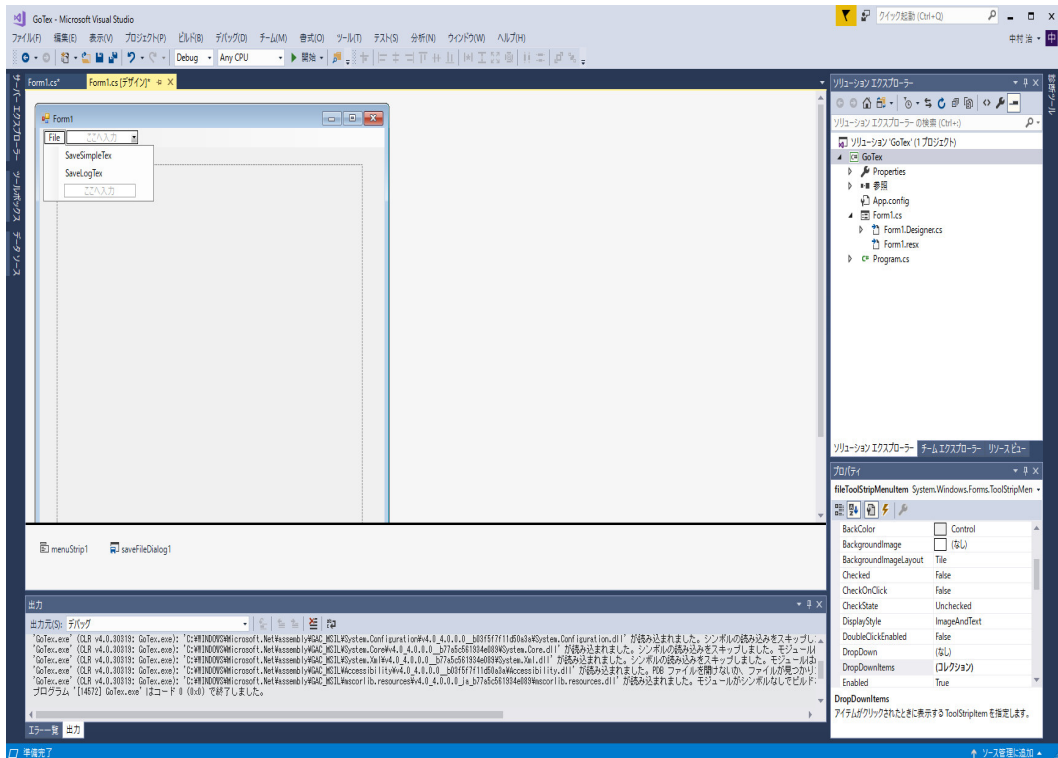
▲持駒
金
銀
桂

のような pdf ファイルになります。残りの駒全部とは書いてくれないので、駒を盤面から消すのではなく、持ち駒に加える必要がありました。これでは煩雑なら、カードに印刷してから、万年筆で修正するなり、各自工夫してください。また、勿論、STY ファイル myshogi.sty を「詰将棋・必至」用に修正することでも対処できますから、興味ある人は各自挑戦してみてください。センター試験の数学の問題用紙と全く同じものを $\text{\LaTeX}$ で作成できる人もいます。世の中にはすごい人がいっぱいいます。

更に、私は、AI と対局すると定跡通りに打ってくれませんかから、大山康晴著「定本大山の駒落ち」などの定跡の変化図を A6 カードに印刷して、定跡の丸暗記ではなく、手筋を覚えようとしています。即ち、所司和晴著「次の一手」で覚える駒落ち定跡コレクション 404」や高橋道雄著「丸暗記原始棒銀戦法」にある「局面」達を印刷したカード集を自分で作ろうとしています。

もっとも、こんなソフトを作ることに時間を費やすより、「屋根裏王」や「技巧」で手を進めてみて、検討の方がはるかに勉強になるような気がします。勉強の仕方は人それぞれで、本人がそれで満足していればそれでいいと思います。

囲碁のプログラムも C# で作ってみます。



のように、Form の Size を、580, 620 とし、Size 500、500 の PictureBox を貼り付け、File メニュー : SaveSimpleTex, SaveLogTex を作ります。

ここでは、互先の一手目から順番に打ち込むプログラムだけを作ります。黑白ランダムに打ち込んで盤面を作るプログラムは自分で作ってください。山下宏さんのアルゴリズムをプログラミングします。結果だけをお示しします。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace GoTex
{
    public partial class Form1 : Form
```

```

{
    const int SPACE = 0;
    const int BLACK = 1;
    const int WHITE = 2;
    const int OUTSIDE = 3;

    public class log
    {
        public int color;
        public int x;
        public int y;
        public log(int C, int X, int Y) { color = C; x = X; y = Y; }
    };

    public class Go
    {
        public int size;
        public int move;
        public int black_prisoner, white_prisoner;
        public int koX, koY, koNum;
        public int[,] board = new int[21, 21];
        public bool[,] checkboard = new bool[21, 21];
        public log[] logData = new log[500];

        public Go(int n)
        {
            size = n + 2;
            move = 1;
            koX = koY = 999;
            koNum = -999;
            black_prisoner = white_prisoner = 0;

            for (int x = 1; x < size - 1; x++)
            {
                for (int y = 1; y < size - 1; y++)
                {
                    board[x, y] = SPACE;
                }
            }
        }
    }
}

```

```

    for (int z = 0; z < size; z++)
    {
        board[0, z] = OUTSIDE;
        board[size - 1, z] = OUTSIDE;
        board[z, 0] = OUTSIDE;
        board[z, size - 1] = OUTSIDE;
    }
}

public bool LegalMoveP(int color, int x, int y)
{
    if (board[x, y] != SPACE)
        return false;
    if (move > 1) {
        if (koX == x && koY == y && koNum == move-1) {
            return false;
        }
    }
    if (SuicideMoveP(color, x, y))
        return false;
    return true;
}

public void ClearCheckBoard()
{
    for (int x = 1; x < size - 1; x++)
        for (int y = 1; y < size - 1; y++)
            checkboard[x, y] = false;
}

public bool EnclosedStoneP(int color, int x, int y)
{
    // 座標 (x,y) にある石 (color) が相手に囲まれているか調べ

    bool rtn;

    if (checkboard[x, y])

```

る

```

        return true;
    checkboard[x, y] = true;
    if (board[x, y] == SPACE)
        return false;
    if (board[x, y] == color)
    {
        if (x > 1)
        {
            rtn = EnclosedStoneP(color, x - 1, y);
            if (!rtn)
                return false;
        }
        if (y > 1)
        {
            rtn = EnclosedStoneP(color, x, y - 1);
            if (!rtn)
                return false;
        }
        if (x < size - 2)
        {
            rtn = EnclosedStoneP(color, x + 1, y);
            if (!rtn)
                return false;
        }
        if (y < size - 2)
        {
            rtn = EnclosedStoneP(color, x, y + 1);
            if (!rtn)
                return false;
        }
    }
    return true;
}

public bool SuicideMoveP(int color, int x, int y)
{
    bool rtnVal;
    int opponentColor;

```

```

board[x, y] = color;
ClearCheckBoard();
rtnVal = EnclosedStoneP(color, x, y);
if (rtnVal)
{
    opponentColor = (color == BLACK) ? WHITE : BLACK;
    if (x > 1)
    {
        if (board[x - 1, y] == opponentColor)
        {
            ClearCheckBoard();
            rtnVal = EnclosedStoneP(opponentColor, x - 1, y);
            if (rtnVal)
            {
                board[x, y] = SPACE;
                return false;
            }
        }
    }
    if (y > 1)
    {
        if (board[x, y - 1] == opponentColor)
        {
            ClearCheckBoard();
            rtnVal = EnclosedStoneP(opponentColor, x, y - 1);
            if (rtnVal)
            {
                board[x, y] = SPACE;
                return false;
            }
        }
    }
    if (x < size - 2)
    {
        if (board[x + 1, y] == opponentColor)
        {
            ClearCheckBoard();
            rtnVal = EnclosedStoneP(opponentColor, x + 1, y);
            if (rtnVal)

```

```

        {
            board[x, y] = SPACE;
            return false;
        }
    }
}
if (y < size - 2)
{
    if (board[x, y + 1] == opponentColor)
    {
        ClearCheckBoard();
        rtnVal = EnclosedStoneP(opponentColor, x, y + 1);
        if (rtnVal)
        {
            board[x, y] = SPACE;
            return false;
        }
    }
}
board[x, y] = SPACE;
return true;
}
else
{
    board[x, y] = SPACE;
    return false;
}
}

public void add_stone(int color, int x, int y)
{
    int prisonerN; // 取り除かれた石の数 (上)
    int prisonerE; // 取り除かれた石の数 (右)
    int prisonerS; // 取り除かれた石の数 (下)
    int prisonerW; // 取り除かれた石の数 (左)
    int prisonerAll;
    bool koFlag;   // コウかどうか

    board[x, y] = color;

```



```

// values[x, y] = 0;

if (board[x, y + 1] != color && board[x, y - 1] != color &&
    board[x + 1, y] != color && board[x - 1, y] != color)
{
    koFlag = true;
}
else
{
    koFlag = false;
}
prisonerN = prisonerE = prisonerS = prisonerW = 0;
if (y > 1)
    prisonerN = RemoveStone(color, x, y - 1);
if (x > 1)
    prisonerW = RemoveStone(color, x - 1, y);
if (y < size - 2)
    prisonerS = RemoveStone(color, x, y + 1);
if (x < size - 2)
    prisonerE = RemoveStone(color, x + 1, y);
prisonerAll = prisonerN + prisonerE + prisonerS + prisonerW;
if (koFlag && prisonerAll == 1)
{
    koNum = move;
    if (prisonerE == 1)
    {
        koX = x + 1;
        koY = y;
    }
    else if (prisonerS == 1)
    {
        koX = x;
        koY = y + 1;
    }
    else if (prisonerW == 1)
    {
        koX = x - 1;
        koY = y;
    }
}

```

```

        else if (prisonerN == 1)
        {
            koX = x;
            koY = y - 1;
        }
    }
    if (prisonerAll > 0)
        if (color == BLACK)
            black_prisoner += prisonerAll;
        else if (color == WHITE)
            white_prisoner += prisonerAll;
}

public int RemoveStone(int color, int x, int y)
{
    // 座標 (x,y) の石が死んでいれば碁盤から取り除く

    int prisoner; // 取り除かれた石の数

    if (board[x, y] == color)
        return 0;
    if (board[x, y] == SPACE)
        return 0;
    ClearCheckBoard();
    if (EnclosedStoneP(board[x, y], x, y))
    {
        prisoner = DoRemoveStone(board[x, y], x, y, 0);
        return prisoner;
    }
    return 0;
}

public int DoRemoveStone(int color, int x, int y, int prisoner)
{
    if (board[x, y] == color)
    {
        prisoner++;
        board[x, y] = SPACE;
        if (x > 1)

```

```

        prisoner = DoRemoveStone(color, x - 1, y, prisoner);
    if (y > 1)
        prisoner = DoRemoveStone(color, x, y - 1, prisoner);
    if (x < size - 2)
        prisoner = DoRemoveStone(color, x + 1, y, prisoner);
    if (y < size - 2)
        prisoner = DoRemoveStone(color, x, y + 1, prisoner);
    }
    return prisoner;
}

public void RecordMove(int color, int x, int y)
{
    if (move < 500)
    {
        log L = new log(color, x, y);
        logData[move] = L;
    }
}

public Go game;
public int CurrentColor = BLACK;

public Form1()
{
    InitializeComponent();
    game = new Go(19);
    CurrentColor = BLACK;
}

public string changeNum(int k)
{
    switch (k)
    {
        case 1: return "a";
        case 2: return "b";
        case 3: return "c";
        case 4: return "d";
    }
}

```

```

        case 5: return "e";
        case 6: return "f";
        case 7: return "g";
        case 8: return "h";
        case 9: return "i";
        case 10: return "k";
        case 11: return "l";
        case 12: return "m";
        case 13: return "n";
        case 14: return "o";
        case 15: return "p";
        case 16: return "q";
        case 17: return "r";
        case 18: return "s";
        case 19: return "t";
    }
    return " ";
}

private void saveSimpleTexToolStripMenuItem_Click(
    object sender, EventArgs e)
{
    SaveFileDialog dlg = new SaveFileDialog();
    dlg.Title = "保存するファイル名を入力してください。"; // ダイアログのタイトル。
    if (dlg.ShowDialog() == DialogResult.OK)
    {
        System.IO.Stream stream;
        stream = dlg.OpenFile();
        if (stream != null)
        {
            //ファイルに書き込む
            System.IO.StreamWriter sw =
                new System.IO.StreamWriter(stream);
            // \TeX の棋譜を出力
            sw.WriteLine("\\documentclass[a4paper, 12pt]{jarticle}");
            sw.WriteLine("\\usepackage{go2}");
            sw.WriteLine("\\textwidth=6.5truein");
            sw.WriteLine(" \\textheight=10.9truein");
            sw.WriteLine("\\oddsidemargin=0pt");
        }
    }
}

```

```

sw.WriteLine("\\topmargin=-72pt");
sw.WriteLine(" \\begin{document}");
sw.WriteLine(" \\inifullldiagram");
for (int i = 1; i <= 19; i++)
{
    for (int k = 1; k <= 19; k++)
    {
        if (game.board[i, k] == BLACK ||
            game.board[i, k] == WHITE)
        {
            var te = " \\pos{";
            te += changeNum(i);
            te += "}{";
            te += k.ToString();
            te += "}=";
            if (game.board[i, k] == BLACK)
            {
                te += "\\black.";
            }
            else if (game.board[i, k] == WHITE)
            {
                te += "\\white.";
            }
            sw.WriteLine(te);
        }
    }
}
sw.WriteLine("\\begin{figure}");
sw.WriteLine("\\gofontsize{20}");
sw.WriteLine("\\showfullldiagram");
sw.WriteLine("\\end{figure}");
sw.WriteLine(" \\end{document}");
sw.Close();
stream.Close();
}
}
}

```

```

private void pictureBox1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    for (int k = 1; k <= 19; k++)
        g.DrawLine(Pens.Black, 25, 25 * k, 475, 25 * k);
    for (int k = 1; k <= 19; k++)
        g.DrawLine(Pens.Black, 25 * k, 25, 25 * k, 475);
    for (int k=1; k<=19; k++)
    {
        for (int i=1; i<=19; i++)
        {
            if (i == 4 && k == 4 || i == 4 && k == 10 ||
                i == 4 && k == 16 ||
                i == 10 && k == 4 || i == 10 && k == 10 ||
                i == 10 && k == 16 ||
                i == 16 && k == 4 || i == 16 && k == 10 ||
                i == 16 && k == 16 )
            {
                SolidBrush b = new SolidBrush(Color.Black);
                g.FillEllipse(b, 25 * i - 4, 25 * k - 4, 8, 8);
                b.Dispose();
            }
            if (game.board[k, i] == 0) {
                continue;
            } else if (game.board[k, i] == 1)
            {
                SolidBrush b = new SolidBrush(Color.Black);
                g.FillEllipse(b, 25 * i - 12, 25 * k - 12, 25, 25);
                b.Dispose();
            }
            else if (game.board[k, i] == 2)
            {
                SolidBrush b = new SolidBrush(Color.White);
                g.FillEllipse(b, 25 * i - 12, 25 * k - 12, 25, 25);
                b.Dispose();
                g.DrawEllipse(Pens.Black, 25 * i - 12, 25 * k - 12,
                    25, 25);
            }
        }
    }
}

```

```

    }
}

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    int x = e.X;
    int y = e.Y;
    int X = 0;
    int Y = 0;
    for (int i = 1; i <= 19; i++)
    {
        X = i;
        if (x >= 25 * i - 12 && x < 25 * i + 12) break;
    }
    for (int i = 1; i <= 19; i++)
    {
        Y = i;
        if (y >= 25 * i - 12 && y < 25 * i + 12) break;
    }
    if (game.LegalMoveP(CurrentColor, Y, X))
    {
        game.add_stone(CurrentColor, Y, X);
        game.RecordMove(CurrentColor, Y, X);
        game.move++;
    }
    else
    {
        return;
    }
    CurrentColor = 3- CurrentColor;
    pictureBox1.Refresh();
}

private void saveLogTexToolStripMenuItem_Click(object sender, EventArgs e)
{
    SaveFileDialog dlg = new SaveFileDialog();
    dlg.Title = "保存するファイル名を入力してください。"; // ダ
    ログのタイトル。
    if (dlg.ShowDialog() == DialogResult.OK)
    {

```

```

System.IO.Stream stream;
stream = dlg.OpenFile();
if (stream != null)
{
    //ファイルに書き込む
    System.IO.StreamWriter sw =
        new System.IO.StreamWriter(stream);
    // \TeX の棋譜を出力
    sw.WriteLine("\\documentclass[a4paper, 12pt]{jarticle}");
    sw.WriteLine("\\usepackage{go2}");
    sw.WriteLine("\\textwidth=6.5truein");
    sw.WriteLine("\\textheight=10.9truein");
    sw.WriteLine("\\oddsidemargin=0pt");
    sw.WriteLine("\\topmargin=-72pt");
    sw.WriteLine("\\begin{document}");
    sw.WriteLine("\\inifulldiagram");

    game.ClearCheckBoard();

    for (int k = game.move - 1; k > 0; k--)
    {
        if (game.checkboard[game.logData[k].x,
            game.logData[k].y] == false)
        {
            if (game.board[game.logData[k].x,
                game.logData[k].y] == game.logData[k].color)
            {
                var te = " \\pos{";
                te += changeNum(game.logData[k].x);
                te += "}{";
                te += game.logData[k].y.ToString();
                te += "}=";
                if (game.logData[k].color == BLACK)
                {
                    te += "\\black{";
                    te += k.ToString();
                    te += "}";
                }
                else if (game.logData[k].color == WHITE)

```



```

        {
            te += "\\white{";
            te += k.ToString();
            te += "}";
        }
        sw.WriteLine(te);
        game.checkboard[game.logData[k].x,
            game.logData[k].y] = true;
    }
}

sw.WriteLine("\\begin{figure}");
sw.WriteLine("\\gofontsize{20}");
sw.WriteLine("\\showfulldiagram");
sw.WriteLine("\\end{figure}");
sw.WriteLine("");

game.ClearCheckBoard();

sw.WriteLine("\\gofontsize{20}");
sw.WriteLine("\\LARGE");
sw.WriteLine("補足 : ");
for (int k = game.move - 1; k > 0; k--)
{
    if (game.checkboard[game.logData[k].x,
        game.logData[k].y] == true)
    {
        var te = "";
        if (game.logData[k].color == BLACK)
        {
            te += "\\textblack{";
            te += k.ToString();
            te += "}=(";
        }
        else if (game.logData[k].color == WHITE)
        {
            te += "\\textwhite{";
            te += k.ToString();
            te += "}=(";
        }
    }
}

```

```

    }
    te += changeNum(game.logData[k].x);
    te += ",";
    te += game.logData[k].y.ToString();
    te += ")";
    sw.WriteLine(te);
} else
{
    if (game.board[game.logData[k].x,
        game.logData[k].y] != game.logData[k].color)
    {
        var te = "";
        if (game.logData[k].color == BLACK)
        {
            te += "\\textblack{";
            te += k.ToString();
            te += "} =(";
        }
        else if (game.logData[k].color == WHITE)
        {
            te += "\\textwhite{";
            te += k.ToString();
            te += "} =(";
        }
        te += changeNum(game.logData[k].x);
        te += ",";
        te += game.logData[k].y.ToString();
        te += ")";
        sw.WriteLine(te);
    }
}
game.checkboard[game.logData[k].x,
    game.logData[k].y] = true;
}
sw.WriteLine("\\end{document}");
sw.Close();
stream.Close();
}
}

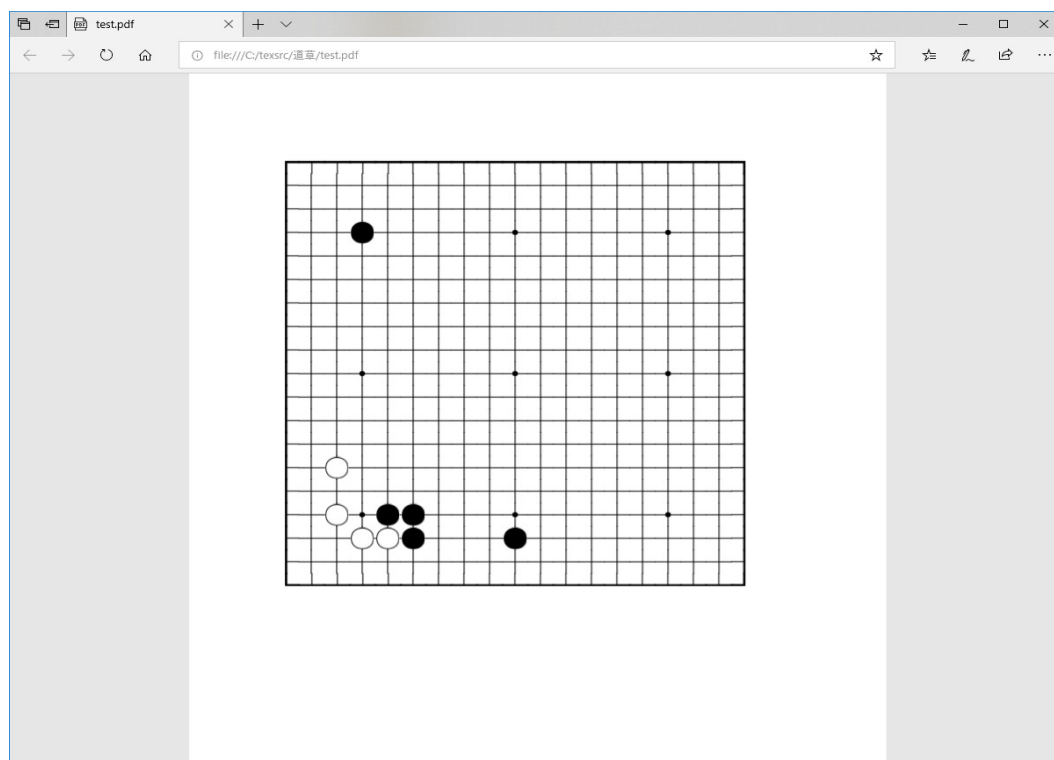
```

```

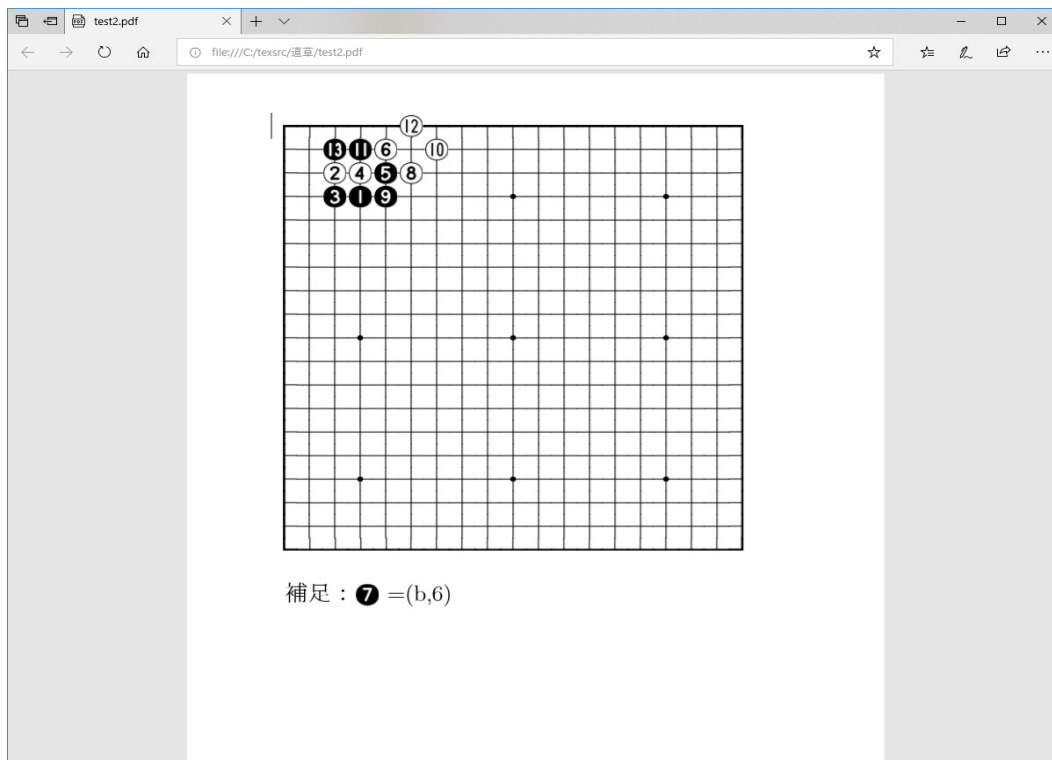
    }
  }
}

```

589 行の簡単なプログラムです。



や



のような局面図が $\text{\LaTeX}$ で作れます。

```

\documentclass[a4paper, 12pt]{jarticle}
\usepackage{go2}
\textwidth=6.5truein
\textheight=10.9truein
\oddsidemargin=0pt
\topmargin=-72pt
\begin{document}
\inifulldiagram
\pos{b}{3}=\black{13}
\pos{a}{6}=\white{12}
\pos{b}{4}=\black{11}
\pos{b}{7}=\white{10}
\pos{d}{5}=\black{9}
\pos{c}{6}=\white{8}
\pos{b}{5}=\white{6}
\pos{c}{5}=\black{5}
\pos{c}{4}=\white{4}
\pos{d}{3}=\black{3}
\pos{c}{3}=\white{2}

```

```

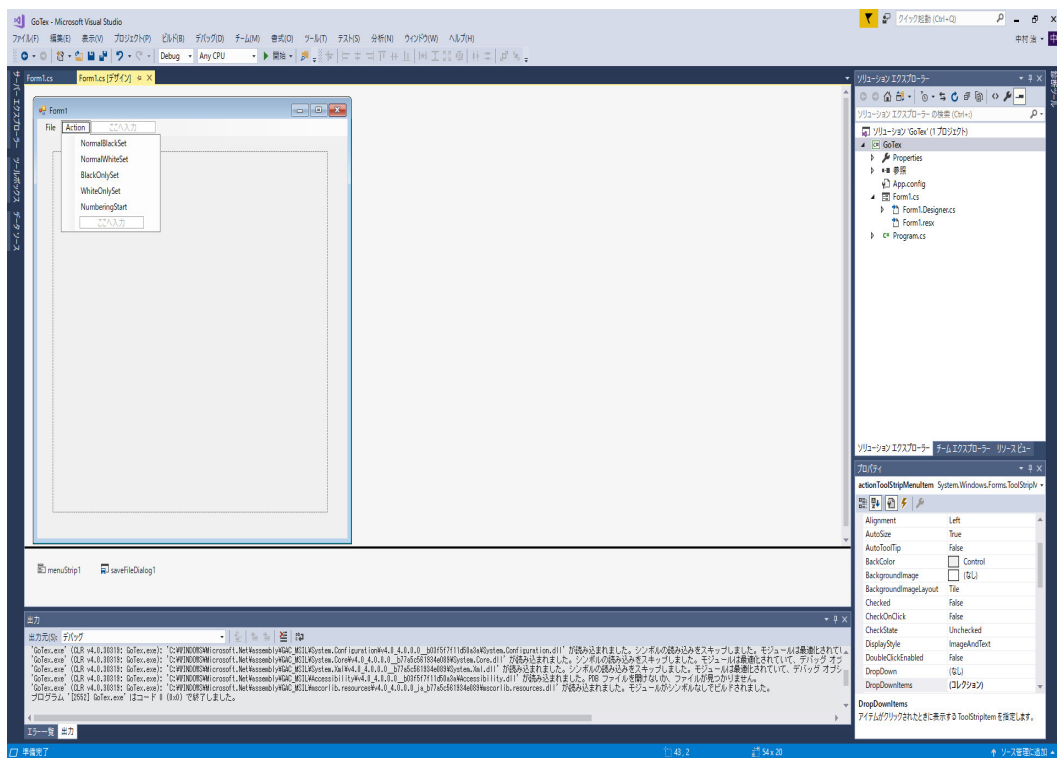
\pos{d}{4}=\black{1}
\begin{figure}
\gofontsize{20}
\showfulldiagram
\end{figure}

\gofontsize{20}
\LARGE
補足：
\textblack{7} =(b,6)
\end{document}

```

が、二番目の図の L^AT_EX の原稿ですが、`\textblack{7} =(b,6)` のような表現は単独では機能せず、文の中に組み込まなければ機能しないので、注意が必要です。

部分的に、石に番号を振るようには対応できていません。手で L^AT_EX の原稿を修正してください。手作業が大変になれば、プログラミングを考えると良いです。又、打った石の順番が分からないまたはその情報が必要ない場合に、黒石や白石を纏めて配置する場合にも対応するようにプログラミングするのがいいと思います。私は次のようにしました。



のように、メニューに、Action : NormalBlackSet, NormalWhiteSet, BlackOnlySet, WhiteOnlySet, NumberingStart を作ります。また、メニュー File に、new を追加します。

```

private void normalBlackSetToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    normalP = true;
    CurrentColor = BLACK;
}

private void normalWhiteSetToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    normalP = true;
    CurrentColor = WHITE;
}

private void blackOnlySetToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    normalP = false;
    CurrentColor = BLACK;
}

private void whiteOnlySetToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    normalP = false;
    CurrentColor = WHITE;
}

private void numberingStartToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    NumberingPos = game.move;
}

private void newToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    game = new Go(19);
    normalP = true;
}

```

```

        CurrentColor = BLACK;
        NumberingPos = game.move;
        pictureBox1.Refresh();
    }

```

と定義します。

```

public Go game;
public int CurrentColor = BLACK;
public bool normalP = true;
public int NumberingPos = 1;

```

と, normalP と NumberingPos の宣言を追加します。

pictureBox1_MouseDown の定義の後半を修正します。

```

private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    int x = e.X;
    int y = e.Y;
    int X = 0;
    int Y = 0;
    for (int i = 1; i <= 19; i++)
    {
        X = i;
        if (x >= 25 * i - 12 && x < 25 * i + 12) break;
    }
    for (int i = 1; i <= 19; i++)
    {
        Y = i;
        if (y >= 25 * i - 12 && y < 25 * i + 12) break;
    }
    if (game.LegalMoveP(CurrentColor, Y, X))
    {
        game.add_stone(CurrentColor, Y, X);
        game.RecordMove(CurrentColor, Y, X);
        game.move++;
    }
    else
    {
        return;
    }
}

```

```

    }
    if (normalP == true)
    {
        CurrentColor = 3 - CurrentColor;
    }
    pictureBox1.Refresh();
}

```

と normalP == true の時だけ CurrentColor を変更するように修正します。

saveLogTexToolStripMenuItem_Click の定義を次のように修正します。

```

private void saveLogTexToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    SaveFileDialog dlg = new SaveFileDialog();
    dlg.Title = "保存するファイル名を入力してください。"; // ダ
    イアログのタイトル。
    if (dlg.ShowDialog() == DialogResult.OK)
    {
        System.IO.Stream stream;
        stream = dlg.OpenFile();
        if (stream != null)
        {
            //ファイルに書き込む
            System.IO.StreamWriter sw =
                new System.IO.StreamWriter(stream);
            // \TeX の棋譜を出力
            sw.WriteLine("\\documentclass[a4paper, 12pt]{jarticle}");
            sw.WriteLine("\\usepackage{go2}");
            sw.WriteLine("\\textwidth=6.5truein");
            sw.WriteLine("\\textheight=10.9truein");
            sw.WriteLine("\\oddsidemargin=0pt");
            sw.WriteLine("\\topmargin=-72pt");
            sw.WriteLine("\\begin{document}");
            sw.WriteLine("\\inifulldiagram");

            game.ClearCheckBoard();

            for (int k = game.move - 1; k >= NumberingPos; k--)
            {

```



```

        if (game.checkboard[game.logData[k].x,
            game.logData[k].y] == false)
        {
            if (game.board[game.logData[k].x,
                game.logData[k].y] == game.logData[k].color)
            {
                var te = " \\pos{";
                te += changeNum(game.logData[k].x);
                te += "}{";
                te += game.logData[k].y.ToString();
                te += "}=";
                if (game.logData[k].color == BLACK)
                {
                    te += "\\black{";
                    te += (k - NumberingPos + 1).ToString();
                    te += "}";
                }
                else if (game.logData[k].color == WHITE)
                {
                    te += "\\white{";
                    te += (k - NumberingPos + 1).ToString();
                    te += "}";
                }
                sw.WriteLine(te);
                game.checkboard[game.logData[k].x,
                    game.logData[k].y] = true;
            }
        }
    }
}
for (int k = NumberingPos - 1; k > 0 ; k--)
{
    if (game.checkboard[game.logData[k].x,
        game.logData[k].y] == false)
    {
        if (game.board[game.logData[k].x,
            game.logData[k].y] == game.logData[k].color)
        {
            var te = " \\pos{";
            te += changeNum(game.logData[k].x);

```

```

        te += "}{";
        te += game.logData[k].y.ToString();
        te += "}= ";
        if (game.logData[k].color == BLACK)
        {
            te += "\\black.";
        }
        else if (game.logData[k].color == WHITE)
        {
            te += "\\white.";
        }
        sw.WriteLine(te);
        game.checkboard[game.logData[k].x,
            game.logData[k].y] = true;
    }
}
}

```

```

sw.WriteLine("\\begin{figure}");
sw.WriteLine("\\gofontsize{20}");
sw.WriteLine("\\showfulldiagram");
sw.WriteLine("\\end{figure}");
sw.WriteLine("");

```

```

game.ClearCheckBoard();

```

```

sw.WriteLine("\\gofontsize{20}");
sw.WriteLine("\\LARGE");
sw.WriteLine("補足 : ");
for (int k = game.move - 1; k >= NumberingPos; k--)
{
    if (game.checkboard[game.logData[k].x,
        game.logData[k].y] == true)
    {
        var te = "";
        if (game.logData[k].color == BLACK)
        {
            te += "\\textblack{";
            te += (k - NumberingPos + 1).ToString();

```

```

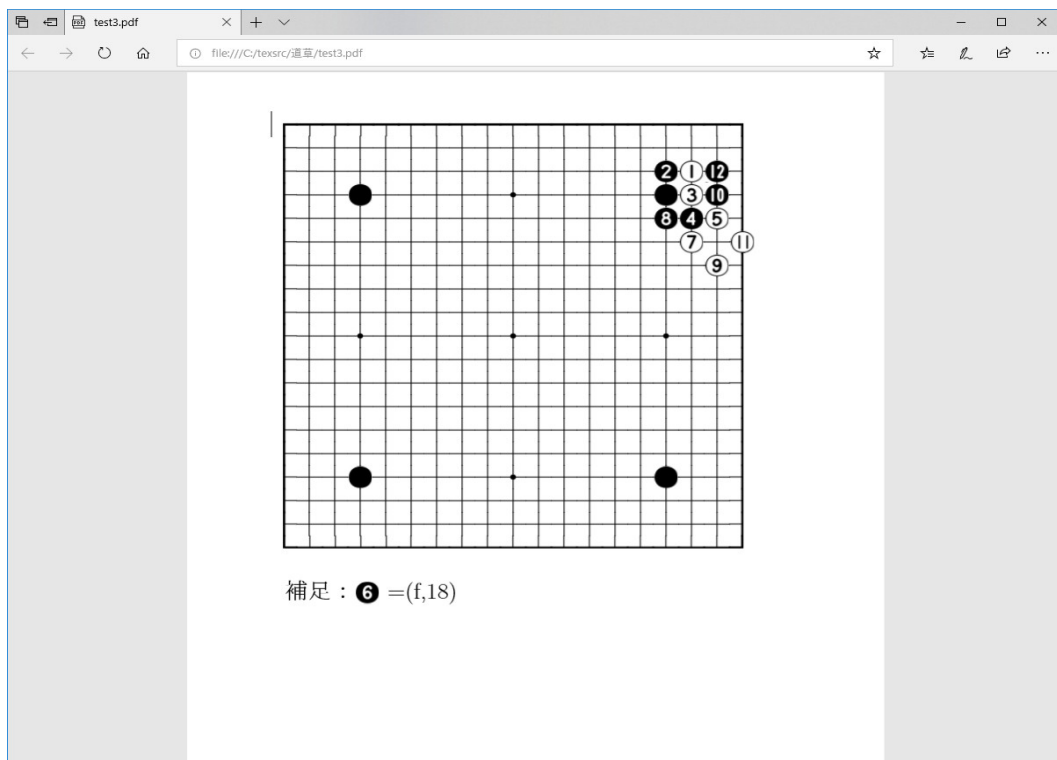
        te += "}=(";
    }
    else if (game.logData[k].color == WHITE)
    {
        te += "\\textwhite{";
        te += (k - NumberingPos + 1).ToString();
        te += "}=(";
    }
    te += changeNum(game.logData[k].x);
    te += ",";
    te += game.logData[k].y.ToString();
    te += ")";
    sw.WriteLine(te);
}
else
{
    if (game.board[game.logData[k].x,
        game.logData[k].y] != game.logData[k].color)
    {
        var te = "";
        if (game.logData[k].color == BLACK)
        {
            te += "\\textblack{";
            te += (k - NumberingPos + 1).ToString();
            te += "} =(";
        }
        else if (game.logData[k].color == WHITE)
        {
            te += "\\textwhite{";
            te += (k - NumberingPos + 1).ToString();
            te += "} =(";
        }
        te += changeNum(game.logData[k].x);
        te += ",";
        te += game.logData[k].y.ToString();
        te += ")";
        sw.WriteLine(te);
    }
}
}

```

```

        game.checkboard[game.logData[k].x,
            game.logData[k].y] = true;
    }
    sw.WriteLine("\\end{document}");
    sw.Close();
    stream.Close();
}
}
}

```



のような pdf ファイルが作れるようになりました。普通の棋譜のように、黒の 6 を「座標 (f,18)」でなく、「白 5 の下」のように表示するのは、手で修正してください。これもプログラミングでやりたければ、自分でアルゴリズムを考えて下さい。これで、囲碁のプログラムは終わりです。

また、今まで二人でプレイするカード・ゲーム「ジャーマン・ホイスト」を VC++, Python, Ruby, Javascript 及び Java で作ってきましたが、インターネットで無料素材のトランプの画像 (.gif でも .png でも .jpg でも良いです) を手に入れて、自分で使うソフトを作ってくださいと述べてきましたが、 $\text{T}_{\text{E}}\text{X}$ を勉強すれば、比較的簡単にカードの画像（絵札をどうするかが大問題です）が *metapost* を使って自分で作ることが可能です。最後に、これをやってみましょう。

Windows に $\text{T}_{\text{E}}\text{X}$ を導入しているとします。

```

def drawDia(expr x,y,s) =
pair d[];
d0 := (x-4*s, y);
d1 := (x, y+5*s);
d2 := (x+4*s, y);
d3 := (x, y-5*s);
fill d0--d1--d2--d3--cycle withcolor red;
enddef;

def drawInd(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
if n < 9:
    p:=substring(n,n+1)of s ifont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
    draw p shifted (5mm, 120mm) withcolor (1,0,0);
else:
    if n = 9:
        p:=substring(n,n+2)of s ifont "ptmb8r" scaled 5r;
        draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
        draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (1,0,0);
    else:
        p:=substring(n+1,n+2)of s ifont "ptmb8r" scaled 5r;
        draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
        draw p shifted (5mm, 120mm) withcolor (1,0,0);
    fi
fi
drawDia(10mm, 110mm, 1mm);
drawDia(84mm, 35mm, 1mm);
enddef;

def drawFrame =
z0=(0mm, 0mm);
z1=(95mm, 0mm);
z2=(95mm, 146mm);
z3=(0mm, 146mm);
draw z0--z1--z2--z3--cycle;
enddef;

```

```

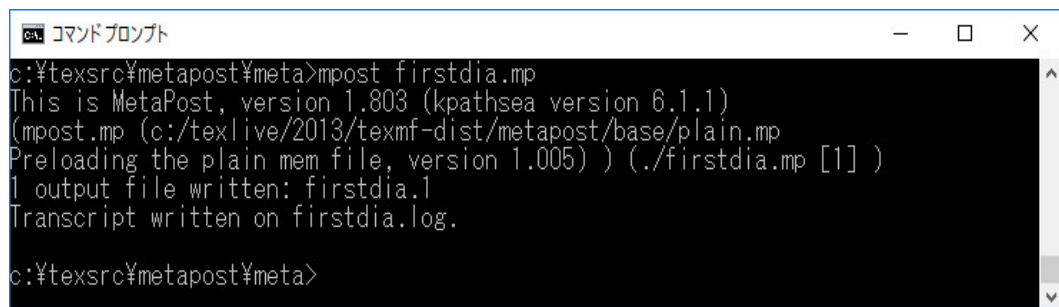
beginfig(1);
drawDia(47mm, 73mm, 2mm);
drawInd(0);
drawFrame;
endfig;
end

```

とプログラミングし、例えば、firstdia.mp と名前を付け、コマンドブロックラインで

```
mpost firstdia.mp
```

とすると



のように、firstdia.1 という名前の画像を作ってくれます。

```
beginfig(1);
```

だから、firstdia.1 という名前で、

```
beginfig(2);
```

とすれば、firstdia.2 という名前になります。一つの mp ファイルで複数の絵を描くことが出来ます。この画像を見るためには、

```

\documentclass[a4j,10pt]{jarticle}
\usepackage[dvipdfmx,hiresbb]{graphicx}

```

```
\pagestyle{plain}
```

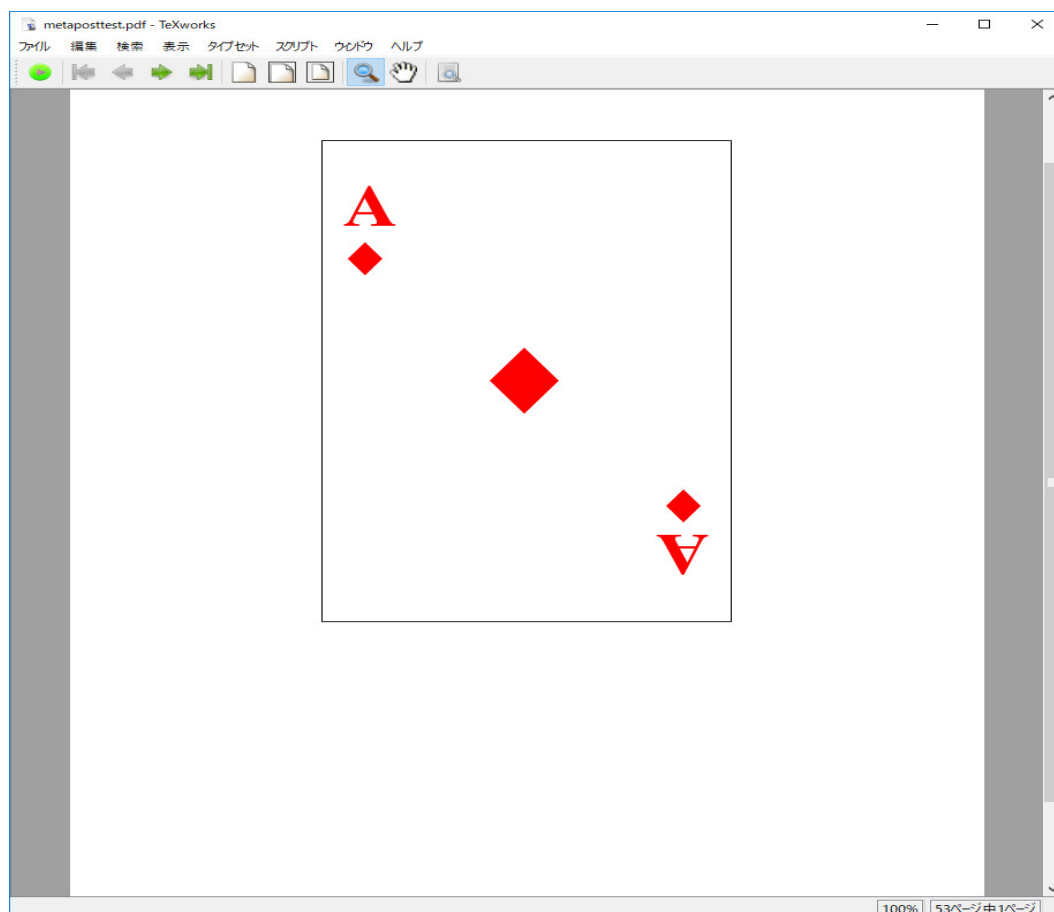
```
%%%%% TEXT START %%%%%
```

```
\begin{document}
```

```
\begin{center}
```

```
\includegraphics[width=94mm,height=146mm]{firstdia.1}
\end{center}
\end{document}
```

という L^AT_EX のファイルを作り、コンパイルすれば

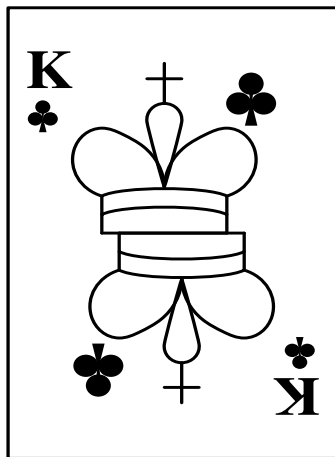
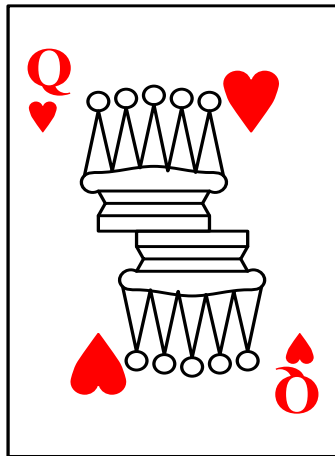
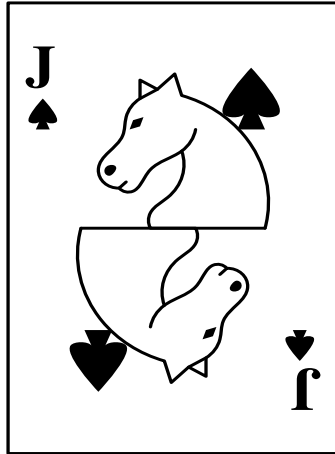


と表示してくれます。ダイヤのエースの札ができました。これを Gimp で処理して、.jpg ファイルなどを作れば良いです。

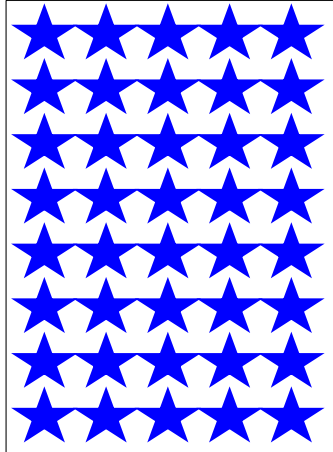
metapost のプログラミングは、metafont のプログラミングと殆ど同じで、文法は

John D. Hobby 著「METAPOST A USER'S MANUAL」

がインターネットから手に入りますが、文法は広範囲にわたり、正確な絵を簡単に描くために、細かな技法が使えるようになっていて複雑ですから、サンプル集をインターネットで探し、それを実際に打ち込み勉強するのがいいと思います。今の場合のように、何がしたいか目標がきちんと定まっているので、それに必要なことだけ勉強すればいいです。経験を積んでから、より良い方法がないか、文法書を精読する方が良いでしょう。最初から、完全な知識をと考えると失敗します。絵札でなければ、比較的簡単ですが、市販のトランプのような絵札を描くのは大変です。ここでは



のように、チェスのナイト、クイーン、キングの絵文字を描くことにしました。自分で好みの絵にしてください。又、裏面は



としました。使っている文法を一々説明するのは煩雑ですから、結果だけ表示します。この metapost のプログラムは

```
def drawDia(expr x,y,s) =
pair d[];
d0 := (x-4*s, y);
d1 := (x, y+5*s);
d2 := (x+4*s, y);
d3 := (x, y-5*s);
fill d0--d1--d2--d3--cycle withcolor red;
enddef;

def drawSpade(expr x,y,s,f) =
path spade;
pair ss[];
ss0 := (0, 5*s);
ss1 := (4*s, -2.5*s);
ss2 := (s, -2.5*s);
ss3 := (2*s, -5*s);
spade := ss0{curl 0}..tension2..ss1..tension1.5..ss2--ss3;
spade := spade -- reverse spade xscaled -1 --cycle;
if f = 0:
    fill spade shifted (x, y) withcolor black;
else:
    fill spade rotated 180 shifted (x, y) withcolor black;
fi
enddef;
```

```

def drawHeart(expr x,y,s,f) =
path heart;
pair hh[];
hh0 := (0.1*s, -5*s);
hh1 := (3.2*s, 0);
hh2 := (4*s, 2.7*s);
hh3 := (2*s, 5*s);
hh4 := (0.1*s, 3.6*s);
heart := hh0{curl 0}..tension2..hh1..tension1.5..hh2..hh3..hh4;
heart := heart -- reverse heart xscaled -1 --cycle;
if f = 0:
    fill heart shifted (x, y) withcolor red;
else:
    fill heart rotated 180 shifted (x, y) withcolor red;
fi
enddef;

def drawClub(expr x,y,s,f) =
pair cc[];
cc0 := (0, 5*s);
cc1 := (1.8*s, 3.2*s);
cc2 := (0, 1.4*s);
cc3 := (-1.8*s, 3.2*s);

cc4 := (1.8*s, 1.8*s);
cc5 := (3.6*s, 0);
cc6 := (1.8*s, -1.8*s);
cc7 := (0, 0);

cc8 := (-1.8*s, 1.8*s);
cc9 := (-3.6*s, 0);
cc10 := (-1.8*s, -1.8*s);
cc11 := (0, 0);

cc12 := (0.9*s, -3.6*s);
cc13 := (0.1*s, 0);
cc14 := (0.3*s, 1.8*s);
cc15 := (-0.3*s, 1.8*s);
cc16 := (-0.1*s, 0);

```

```

cc17 := (-0.9*s, -3.6*s);

if f = 0:
  fill cc0..cc1..cc2..cc3..cycle shifted (x, y) withcolor black;
  fill cc4..cc5..cc6..cc7..cycle shifted (x, y) withcolor black;
  fill cc8..cc9..cc10..cc11..cycle shifted (x, y) withcolor black;
  fill cc12--cc13--cc14--cc15--cc16--cc17--cycle shifted (x, y)
    withcolor black;
else:
  fill cc0..cc1..cc2..cc3..cycle rotated 180 shifted (x, y) withcolor black;
  fill cc4..cc5..cc6..cc7..cycle rotated 180 shifted (x, y) withcolor black;
  fill cc8..cc9..cc10..cc11..cycle rotated 180 shifted (x, y) withcolor black;
  fill cc12--cc13--cc14--cc15--cc16--cc17--cycle rotated 180 shifted (x, y)
    withcolor black;
fi
enddef;

def drawStar(expr x,y) =
  pair A,B,C,D,E;
  A := (0,1cm);
  B := A rotated 72;
  C := B rotated 72;
  D := C rotated 72;
  E := D rotated 72;
  fill A--C--E--B--D--cycle shifted (x, y) withcolor blue;
enddef;

def drawInd(expr n) =
  numeric r; picture p; string s;
  r := 1;
  s := "A2345678910JQK";
  if n < 9:
    p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
    draw p shifted (5mm, 120mm) withcolor (1,0,0);
  else:
    if n = 9:
      p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
      draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);

```

```

        draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (1,0,0);
    else:
        p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
        draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
        draw p shifted (5mm, 120mm) withcolor (1,0,0);
    fi
fi
drawDia(10mm, 110mm, 1mm);
drawDia(84mm, 35mm, 1mm);
enddef;

def drawIndSpade(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
if n < 9:
    p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
    draw p shifted (5mm, 120mm) withcolor (0,0,0);
else:
    if n = 9:
        p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
        draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
        draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (0,0,0);
    else:
        p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
        draw p rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
        draw p shifted (5mm, 120mm) withcolor (0,0,0);
    fi
fi
drawSpade(10mm, 110mm, 1mm, 0);
drawSpade(84mm, 35mm, 1mm, 1);
enddef;

def drawIndHeart(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
if n < 9:

```

```

    p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
    draw p shifted (5mm, 120mm) withcolor (1,0,0);
else:
    if n = 9:
        p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
        draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
        draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (1,0,0);
    else:
        p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
        draw p rotated 180 shifted (90mm, 26mm) withcolor (1,0,0);
        draw p shifted (5mm, 120mm) withcolor (1,0,0);
    fi
fi
drawHeart(10mm, 110mm, 1mm, 0);
drawHeart(84mm, 35mm, 1mm, 1);
enddef;

def drawIndClub(expr n) =
numeric r; picture p; string s;
r := 1;
s := "A2345678910JQK";
if n < 9:
    p:=substring(n,n+1)of s infont "ptmb8r" scaled 5r;
    draw p rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
    draw p shifted (5mm, 120mm) withcolor (0,0,0);
else:
    if n = 9:
        p:=substring(n,n+2)of s infont "ptmb8r" scaled 5r;
        draw p xscaled 0.5 rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
        draw p xscaled 0.5 shifted (5mm, 120mm) withcolor (0,0,0);
    else:
        p:=substring(n+1,n+2)of s infont "ptmb8r" scaled 5r;
        draw p rotated 180 shifted (90mm, 26mm) withcolor (0,0,0);
        draw p shifted (5mm, 120mm) withcolor (0,0,0);
    fi
fi
drawClub(10mm, 110mm, 1.2mm, 0);
drawClub(84mm, 35mm, 1.2mm, 1);

```

```

enddef;

def drawFrame =
z0=(0mm, 0mm);
z1=(95mm, 0mm);
z2=(95mm, 146mm);
z3=(0mm, 146mm);
draw z0--z1--z2--z3--cycle;
enddef;

def drawKnight =
pair j[];
j0 := (54mm, 0mm);
j1 := (48mm, 30mm);
j2 := (30mm, 43mm);
j3 := (28mm, 50mm);
j4 := (23mm, 45mm);
j5 := (18mm, 42mm);
j6 := (14mm, 36mm);
j7 := (13mm, 30mm);
j8 := (6mm, 19mm);
j9 := (12mm, 13mm);
j10 := (14mm, 15mm);
j11 := (16mm, 12mm);
j12 := (18mm, 14mm);
j13 := (21mm, 19mm);
j14 := (30mm, 25mm);
j15 := (34mm, 32mm);
j16 := (26mm, 10mm);
j17 := (21mm, 5mm);
j18 := (21mm, 0mm);
j19 := (18mm, 48mm);
j20 := (15mm, 32mm);
j21 := (18mm, 33mm);
j22 := (19mm, 36mm);
j23 := (16mm, 35mm);
j24 := (9mm, 17mm);
j25 := (10.5mm, 18mm);
j26 := (11mm, 20mm);

```

```

j27 := (10mm, 20.5mm);
pickup pencircle scaled 0.01w;
path q[];
q1 = j0..j1..j2--j3--j4--j5..j6..j7..j8..j9--j10;
draw q1;
q2 = j9..j11..j12..j13..j14..j15;
draw q2;
q3 = j14..j16..j17..j18--j0;
draw q3;
q4 = j4--j19--j5;
draw q4;
q5 = j20--j21--j22--j23--cycle;
fill q5 withcolor black;
q6 = j24..j25..j26..j27..cycle;
fill q6 withcolor black;
picture knightpicture;
knightpicture := currentpicture;
currentpicture := nullpicture;
draw knightpicture shifted (20mm, 73mm);
draw knightpicture rotated 180 shifted (75mm, 73mm);
enddef;

```

```

def drawQueen =
pair q[];
q0 := (12mm, 0mm);
q1 := (44mm, 0mm);
q2 := (44mm, 5mm);
q3 := (12mm, 5mm);
q4 := (14mm, 9mm);
q5 := (42mm, 9mm);
q6 := (12mm, 13mm);
q7 := (44mm, 13mm);

q8 := (8mm, 18mm);
q9 := (12mm, 19mm);
q10 := (20mm, 20mm);
q11 := (28mm, 21mm);
q12 := (36mm, 20mm);
q13 := (44mm, 19mm);

```

```

q14 := (48mm, 18mm);

q15 := (12mm, 39.0mm);
q16 := (16mm, 19.5mm);
q17 := (20mm, 40mm);
q18 := (24mm, 20.5mm);
q19 := (28mm, 41mm);
q20 := (32mm, 19.5mm);
q21 := (36mm, 40mm);
q22 := (40mm, 19mm);
q23 := (44mm, 39.0mm);

pickup pencircle scaled 0.01w;

draw q0--q1--q2--q3--cycle;
draw q3--q4..q5--q2;
draw q4--q6..q7--q5;
draw q6..q8..q9..q10..q11..q12..q13..q14..q7;
draw q8--q15--q16--q17--q18--q19--q20--q21--q22--q23--q14;
draw fullcircle scaled 6mm shifted q15 shifted (0, 3mm);
draw fullcircle scaled 6mm shifted q17 shifted (0, 3mm);
draw fullcircle scaled 6mm shifted q19 shifted (0, 3mm);
draw fullcircle scaled 6mm shifted q21 shifted (0, 3mm);
draw fullcircle scaled 6mm shifted q23 shifted (0, 3mm);
picture queenpicture;
queenpicture := currentpicture;
currentpicture := nullpicture;
draw queenpicture shifted (14mm, 73mm);
draw queenpicture rotated 180 shifted (81mm, 73mm);
enddef;

def drawKing =
pair k[];
k0 := (15mm, 0mm);
k1 := (51mm, 0mm);
k2 := (15mm, 6mm);
k3 := (51mm, 6mm);
k4 := (15mm, 12mm);
k5 := (51mm, 12mm);

```



```

k6 := (7mm, 27mm);
k7 := (15mm, 34mm);
k8 := (21mm, 32mm);
k9 := (27mm, 28mm);
k10 := (33mm, 15mm);
k11 := (39mm, 28mm);
k12 := (45mm, 32mm);
k13 := (51mm, 34mm);
k14 := (59mm, 27mm);

k15 := (28mm, 36mm);
k16 := (33mm, 42mm);
k17 := (38mm, 36mm);

k18 := (33mm, 55mm);
k19 := (28mm, 50mm);
k20 := (38mm, 50mm);

pickup pencircle scaled 0.01w;
draw k0..tension 4..k1;
draw k0--k2{dir 30}..tension 2..k3--k1;
draw k2--k4{dir 30}..tension 2..k5--k3;
draw k4..k6..k7..k8..k9..k10;
draw k5..k14..k13..k12..k11..k10;
draw k10..tension 3..k15..k16..k17..tension 3..k10;
draw k16--k18;
draw k19--k20;
picture kingpicture;
kingpicture := currentpicture;
currentpicture := nullpicture;
draw kingpicture shifted (12mm, 73mm);
draw kingpicture rotated 180 shifted (83mm, 73mm);
enddef;

w = 95mm;

beginfig(1);

```

```

drawDia(47mm, 73mm, 2mm);
drawInd(0);
drawFrame;
endfig;
beginfig(2);
drawDia(47mm, 30mm, 2mm);
drawDia(47mm, 115mm, 2mm);
drawInd(1);
drawFrame;
endfig;
beginfig(3);
drawDia(47mm, 30mm, 2mm);
drawDia(47mm, 73mm, 2mm);
drawDia(47mm, 115mm, 2mm);
drawInd(2);
drawFrame;
endfig;
beginfig(4);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawInd(3);
drawFrame;
endfig;
beginfig(5);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(47mm, 73mm, 2mm);
drawInd(4);
drawFrame;
endfig;
beginfig(6);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);

```

```

drawDia(26mm, 73mm, 2mm);
drawDia(70mm, 73mm, 2mm);
drawInd(5);
drawFrame;
endfig;
beginfig(7);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 73mm, 2mm);
drawDia(70mm, 73mm, 2mm);
drawDia(47mm, 50mm, 2mm);
drawInd(6);
drawFrame;
endfig;
beginfig(8);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 73mm, 2mm);
drawDia(70mm, 73mm, 2mm);
drawDia(47mm, 50mm, 2mm);
drawDia(47mm, 95mm, 2mm);
drawInd(7);
drawFrame;
endfig;
beginfig(9);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 58mm, 2mm);
drawDia(70mm, 58mm, 2mm);
drawDia(26mm, 86mm, 2mm);
drawDia(70mm, 86mm, 2mm);
drawDia(47mm, 73mm, 2mm);
drawInd(8);

```

```

drawFrame;
endfig;
beginfig(10);
drawDia(26mm, 30mm, 2mm);
drawDia(26mm, 115mm, 2mm);
drawDia(70mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawDia(26mm, 58mm, 2mm);
drawDia(70mm, 58mm, 2mm);
drawDia(26mm, 86mm, 2mm);
drawDia(70mm, 86mm, 2mm);
drawDia(47mm, 44mm, 2mm);
drawDia(47mm, 101mm, 2mm);
drawInd(9);
drawFrame;
endfig;
beginfig(11);
drawKnight;
drawDia(26mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawInd(10);
drawFrame;
endfig;
beginfig(12);
drawQueen;
drawDia(26mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawInd(11);
drawFrame;
endfig;
beginfig(13);
drawKing;
drawDia(26mm, 30mm, 2mm);
drawDia(70mm, 115mm, 2mm);
drawInd(12);
drawFrame;
endfig;

beginfig(14);

```

```

drawSpade(47mm, 73mm, 4mm, 0);
drawIndSpade(0);
drawFrame;
endfig;
beginfig(15);
drawSpade(47mm, 30mm, 2mm, 1);
drawSpade(47mm, 115mm, 2mm, 0);
drawIndSpade(1);
drawFrame;
endfig;
beginfig(16);
drawSpade(47mm, 30mm, 2mm, 1);
drawSpade(47mm, 73mm, 2mm, 0);
drawSpade(47mm, 115mm, 2mm, 0);
drawIndSpade(2);
drawFrame;
endfig;
beginfig(17);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawIndSpade(3);
drawFrame;
endfig;
beginfig(18);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(47mm, 73mm, 2mm, 0);
drawIndSpade(4);
drawFrame;
endfig;
beginfig(19);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);

```

```

drawSpade(26mm, 73mm, 2mm, 0);
drawSpade(70mm, 73mm, 2mm, 0);
drawIndSpade(5);
drawFrame;
endfig;
beginfig(20);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 73mm, 2mm, 0);
drawSpade(70mm, 73mm, 2mm, 0);
drawSpade(47mm, 50mm, 2mm, 1);
drawIndSpade(6);
drawFrame;
endfig;
beginfig(21);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 73mm, 2mm, 0);
drawSpade(70mm, 73mm, 2mm, 0);
drawSpade(47mm, 50mm, 2mm, 1);
drawSpade(47mm, 95mm, 2mm, 0);
drawIndSpade(7);
drawFrame;
endfig;
beginfig(22);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 01);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 58mm, 2mm, 1);
drawSpade(70mm, 58mm, 2mm, 1);
drawSpade(26mm, 86mm, 2mm, 0);
drawSpade(70mm, 86mm, 2mm, 0);
drawSpade(47mm, 73mm, 2mm, 0);
drawIndSpade(8);

```

```

drawFrame;
endfig;
beginfig(23);
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(26mm, 115mm, 2mm, 0);
drawSpade(70mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawSpade(26mm, 58mm, 2mm, 1);
drawSpade(70mm, 58mm, 2mm, 1);
drawSpade(26mm, 86mm, 2mm, 0);
drawSpade(70mm, 86mm, 2mm, 0);
drawSpade(47mm, 44mm, 2mm, 1);
drawSpade(47mm, 101mm, 2mm, 0);
drawIndSpade(9);
drawFrame;
endfig;
beginfig(24);
drawKnight;
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawIndSpade(10);
drawFrame;
endfig;
beginfig(25);
drawQueen;
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawIndSpade(11);
drawFrame;
endfig;
beginfig(26);
drawKing;
drawSpade(26mm, 30mm, 2mm, 1);
drawSpade(70mm, 115mm, 2mm, 0);
drawIndSpade(12);
drawFrame;
endfig;

beginfig(27);

```

```

drawHeart(47mm, 73mm, 2mm, 0);
drawIndHeart(0);
drawFrame;
endfig;
beginfig(28);
drawHeart(47mm, 30mm, 2mm, 1);
drawHeart(47mm, 115mm, 2mm, 0);
drawIndHeart(1);
drawFrame;
endfig;
beginfig(29);
drawHeart(47mm, 30mm, 2mm, 1);
drawHeart(47mm, 73mm, 2mm, 0);
drawHeart(47mm, 115mm, 2mm, 0);
drawIndHeart(2);
drawFrame;
endfig;
beginfig(30);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawIndHeart(3);
drawFrame;
endfig;
beginfig(31);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(47mm, 73mm, 2mm, 0);
drawIndHeart(4);
drawFrame;
endfig;
beginfig(32);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);

```



```

drawHeart(26mm, 73mm, 2mm, 0);
drawHeart(70mm, 73mm, 2mm, 0);
drawIndHeart(5);
drawFrame;
endfig;
beginfig(33);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 73mm, 2mm, 0);
drawHeart(70mm, 73mm, 2mm, 0);
drawHeart(47mm, 50mm, 2mm, 1);
drawIndHeart(6);
drawFrame;
endfig;
beginfig(34);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 73mm, 2mm, 0);
drawHeart(70mm, 73mm, 2mm, 0);
drawHeart(47mm, 50mm, 2mm, 1);
drawHeart(47mm, 95mm, 2mm, 0);
drawIndHeart(7);
drawFrame;
endfig;
beginfig(35);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 01);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 58mm, 2mm, 1);
drawHeart(70mm, 58mm, 2mm, 1);
drawHeart(26mm, 86mm, 2mm, 0);
drawHeart(70mm, 86mm, 2mm, 0);
drawHeart(47mm, 73mm, 2mm, 0);
drawIndHeart(8);

```

```

drawFrame;
endfig;
beginfig(36);
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(26mm, 115mm, 2mm, 0);
drawHeart(70mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawHeart(26mm, 58mm, 2mm, 1);
drawHeart(70mm, 58mm, 2mm, 1);
drawHeart(26mm, 86mm, 2mm, 0);
drawHeart(70mm, 86mm, 2mm, 0);
drawHeart(47mm, 44mm, 2mm, 1);
drawHeart(47mm, 101mm, 2mm, 0);
drawIndHeart(9);
drawFrame;
endfig;
beginfig(37);
drawKnight;
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawIndHeart(10);
drawFrame;
endfig;
beginfig(38);
drawQueen;
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawIndHeart(11);
drawFrame;
endfig;
beginfig(39);
drawKing;
drawHeart(26mm, 30mm, 2mm, 1);
drawHeart(70mm, 115mm, 2mm, 0);
drawIndHeart(12);
drawFrame;
endfig;

beginfig(40);

```

```

drawIndClub(0);
drawClub(47mm, 73mm, 2mm, 0);
drawFrame;
endfig;
beginfig(41);
drawClub(47mm, 30mm, 2mm, 1);
drawClub(47mm, 115mm, 2mm, 0);
drawIndClub(1);
drawFrame;
endfig;
beginfig(42);
drawClub(47mm, 30mm, 2mm, 1);
drawClub(47mm, 73mm, 2mm, 0);
drawClub(47mm, 115mm, 2mm, 0);
drawIndClub(2);
drawFrame;
endfig;
beginfig(43);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawIndClub(3);
drawFrame;
endfig;
beginfig(44);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(47mm, 73mm, 2mm, 0);
drawIndClub(4);
drawFrame;
endfig;
beginfig(45);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);

```

```

drawClub(26mm, 73mm, 2mm, 0);
drawClub(70mm, 73mm, 2mm, 0);
drawIndClub(5);
drawFrame;
endfig;
beginfig(46);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 73mm, 2mm, 0);
drawClub(70mm, 73mm, 2mm, 0);
drawClub(47mm, 50mm, 2mm, 1);
drawIndClub(6);
drawFrame;
endfig;
beginfig(47);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 73mm, 2mm, 0);
drawClub(70mm, 73mm, 2mm, 0);
drawClub(47mm, 50mm, 2mm, 1);
drawClub(47mm, 95mm, 2mm, 0);
drawIndClub(7);
drawFrame;
endfig;
beginfig(48);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 01);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 58mm, 2mm, 1);
drawClub(70mm, 58mm, 2mm, 1);
drawClub(26mm, 86mm, 2mm, 0);
drawClub(70mm, 86mm, 2mm, 0);
drawClub(47mm, 73mm, 2mm, 0);
drawIndClub(8);

```

```

drawFrame;
endfig;
beginfig(49);
drawClub(26mm, 30mm, 2mm, 1);
drawClub(26mm, 115mm, 2mm, 0);
drawClub(70mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawClub(26mm, 58mm, 2mm, 1);
drawClub(70mm, 58mm, 2mm, 1);
drawClub(26mm, 86mm, 2mm, 0);
drawClub(70mm, 86mm, 2mm, 0);
drawClub(47mm, 44mm, 2mm, 1);
drawClub(47mm, 101mm, 2mm, 0);
drawIndClub(9);
drawFrame;
endfig;
beginfig(50);
drawKnight;
drawClub(26mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawIndClub(10);
drawFrame;
endfig;
beginfig(51);
drawQueen;
drawClub(26mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawIndClub(11);
drawFrame;
endfig;
beginfig(52);
drawKing;
drawClub(26mm, 30mm, 2mm, 1);
drawClub(70mm, 115mm, 2mm, 0);
drawIndClub(12);
drawFrame;
endfig;

beginfig(53);

```

```

for x=1 upto 5:
  for y=1 upto 8:
    drawStar(50*x-19, 50*y-17);
  endfor
endfor
drawFrame;
endfig;

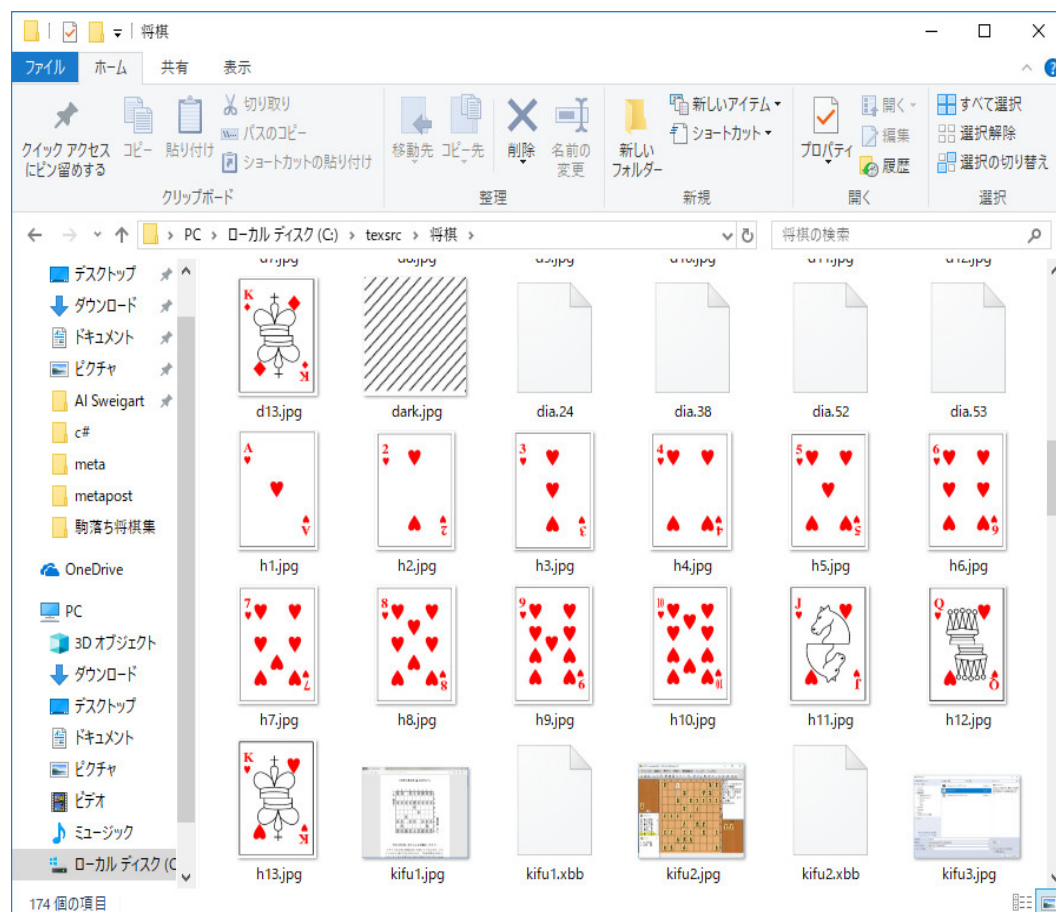
end

```

です。

素人が作った長々と同じようなことが書いてあるプログラムで、本当はプログラムを短くする技法があると思いますが、一度しか使わないプログラムなので、これで良いことにします。コピー アンド ペーストで、同じことを繰り返しているだけなので、丸二日で作ることが出来ます。

上のプログラムで作った画像を $\text{T}_\text{E}\text{X}$ で表示し、Gimp で切り取れば、



のように、オリジナルなカードの画像を作ることが出来ました。やっつけ仕事ですから、スーツのスペードやクラブやハートの形や大きさや絵札の絵が気に入ら

なければ、自分で上のプログラムを修正してください。

これで、オリジナルなカードの画像が作れたので、二人でプレイするカード・ゲーム「ジャーマン・ホイスト」を C# で作っていきますとすべきですが、既に見たように VC++ 2010 のプログラムを真似すれば良いはずですから、今回はやめておきます。

私のような素人のプログラマーは、自分の欲しい道具（ソフト）を最低限の労力で作るために、このようにして作っています。自分の作りたいものを正確に見極め、そのために必要な知識・技能の習得をまず考えます。差し当って必要もないのに関連する知識を完全に習得しなければいけない、理解できなければ恥ずかしいと頑張れば失敗します。

私がプログラミングを始めたころと異なり、インターネットに沢山の情報があるので、更に有利です。その場合、やりたいことがすべてプログラミングされている完全なプログラムが見つかることはまずないです。そのプログラムがどのような構造になるか、これをして、これをして、…、これをすればいいということを考え、それぞれの部分を実行するにはどうすればいいかをインターネットで調べます。音声データの再生の説明で述べたように、カレントフォルダーにあるすべてのファイルを取り出すにはどうすればいいか、その中の .wma ファイルを選び出すにはどうすればいいか、ファイル名のある部分を抜き出すにはどうすればいいか、…、と言った単純な機能のプログラミングのやり方をインターネットで調べます。そして、それらの情報を組み合わせて、作りたかったプログラムをでっちあげます。学生さんを教育するためには、独りよがりの、昔はこうだったと言って、もう世の中では使っていないことを教えたのでは、学生さんが可哀そうですから、できるだけ最新の情報を隅から隅まで読んでおかなければならぬ大変でしたが、もう役目が終わりました。約 40 年間、暇を見つけて学んできたプログラミングの断片的な知識が、定年を過ぎて、暇な時間ができて、今まで何を学んできたか見直す時間ができて、バラバラだった知識が結びつき、やっと役に立ってきました。

外国語（アラビア語やロシア語やラテン語やギリシャ語など余り人が興味を持たないもの）と将棋と囲碁と 3 つも趣味があると人の三倍努力しないといけないので、この pdf ファイルで解説したプログラムを作ってみました。C# のプログラムは実行可能なソフトをダウンロードできるようにすべきではないかとも思いましたが、私は教師をやってきましたから、釣った魚を配るのではなく、魚の釣り方を教えるのが仕事だと思っているので、実行可能なソフトの配布はしません。自分で解説を読んでプログラミングしてください。定年になると働いていた時の 3 倍の自由時間があると言いますから何とかかなと思っています。更に、小学生の時からあこがれてはいましたが、不器用で、ブロックで作ったラジオ以外、作ったラジオが一度も鳴ったことがない、苦手な電子工作に力を入れて勉強したいです。長生きも悪いものではないです。

インターネットの情報は、プログラミングに関してはまともな情報がアップさ

れていますが、それでも古くてあるいは間違っていて使えないものも結構沢山ありますが、プログラミングに関する情報だけは、どれが正しい情報か実際にプログラミングして実行してみればコンピュータが結果を教えてくれるので、私のような適切な情報を教えてくれる友人が周りにいない人間でも独学が可能です。大抵のことは親切な人がインターネットに情報をアップしてくれています。ものを作るのは楽しいものです。知ることは楽しいことです。プロ野球選手の打撃スタイルがみんな異なっていたり、女子マラソンの高橋尚子さんと有森裕子さんの練習方法が異なっていたように、勉強の仕方も、誰にも通用する普遍的なやり方があると言うよりも、色々な人の勉強方法を参考に、何をしたいかに応じて、達成可能な目標を定めて、人それぞれ自分の個性に合った自分なりの勉強方法を自分で見つけるしかないみたいです。「練習は嘘をつかないって言葉があるけど、頭を使って練習しないと普通に嘘をつくよ」(ダルビッシュ有)。師匠の工藤達二先生の図書カードに英文タイプライターで打ち込んだ他人には理解できない暗号のようなカードや高知大学の京大卒の教授が京大式カードを使っているのを見てきましたが、私も70歳を過ぎてやっと $\text{\LaTeX}$ で将棋や囲碁の局面やロシア語やアラビア語やドイツ語やフランス語などの文法規則などのリストを作ってB6やA6のカードに印刷して利用し始めました。この文章が誰かの参考になれば幸いです。

動詞の完了形の活用

3人称男性単数	كَتَبَ
3人称男性双数	كَتَبَا
3人称男性複数	كَتَبُوا
3人称女性単数	كَتَبَتْ
3人称女性双数	كَتَبَتَا
3人称女性複数	كَتَبْنَ
2人称男性単数	كَتَبْتَ
2人称男性双数	كَتَبْتُمَا
2人称男性複数	كَتَبْتُمْ
2人称女性単数	كَتَبْتِ
2人称女性双数	كَتَبْتُمَا
2人称女性複数	كَتَبْتُنَّ
1人称単数	كَتَبْتُ
1人称複数	كَتَبْنَا

кро́ме тобо́	その他に
дабнó	ずっと前から
теплó	暖かだ
весна́	春
пого́да	天気
се́вер	北
юг	南
востóк	東
за́пад	西
зимóй	冬に
хóлодно	寒い
ти́хо	小さな声で
ну́жно	…しなければならない
гро́мко	大きな声で
нельз́я	…してはいけない
так	そんなに
бы́стро	速く
мо́жно	…してよい・かまわない
сейча́с	いますぐ
изуча́ть	勉強する
тру́дно	難しい
сад	庭園
фонта́н	噴水
гуля́ть(гуля́ем)	散歩する
отдыха́ть(отдыха́ем)	休息する