Processingの作品の作成過程について

5455　情報科学科　難波学之

1. 作品の発想から作成の意図

この作品を作るにあたって、何か今までにないものを作り出そうと考えて、題材として何かないかなとサンプルを調べていたらテトリスにたどり着いたので、ここから何か新しいゲームを作り出せないかと考えたところ、いっそのこと普通のテトリスとは真逆のものにしてしまおうという普通のテトリスからは考えられないある意味恐ろしい考えができたのでそのまま採用しました。名前である「Reversal Tetris」も『逆転した』の英語である『Reversal』をつけたシンプルな名前にしています。

1. 遊び方について

真逆だといわれてもわからないと思うので説明しておきます。これは『横一列を完成させずに、どれだけ多くのブロックを敷き詰めることができるか』というゲームです。決して横一列を完成させないようにお願いします(すぐにGAMEOVERになってしまいます)

1. この作品の作成時に引用させてもらったソースについて

　この作品の作成時にテトリスを成り立たせる基本ソースを引用させてもらっています(表1)。全ては長すぎて載せられないので下記の主な変更点の一部を例に記載しています。

主な変更・追加点は

* ブロックの移動範囲の拡大などに対応できるように、ソースを理解し、変数a,bに変更しました(結局void setup以降で数値を変更するとエラーが起き、ほかのMODEに活用させることはできませんでした)
* draw関数にあったtetrisの起動ソースを単独でvoid化して、条件式を満たすと起動できるようにした上で、選択画面を追加しました。また、プレイ画面に移行した時に発生するグラフィックの変化も四角形を上に貼り付ける方針でカバーしました
* ゲームの遊び方がそもそも違うので、GAMEOVER時とCLEAR時のソースを逆転させ、CLEAR時には成績によって表示内容を変化させるランク付け機能を自作して追加しました
* 選択したMODEによって操作方法の変更などの追加を条件式によって行いました
* 出力する数値の値を遊び方の変更により『ブロックの数』に変更しました
* 条件式を使い、隠しモードの開放の機能を追加しました。
* テトリスにあい、かつ著作権的に(おそらく)大丈夫な民謡曲、式曲を流しました
* その他グラフィックの色の改善を行いました

1. 本来実装したかったこと

　本来は実行したかったけどエラーの発生などにより実現できなかったことがありました。

* ブロックの移動範囲の変更をMODEごとに実行すること(3章参照)
* 二人プレイ機能を実装すること。各得点と合計得点を表示し、横一列そろえたときに得点を減らす機能にすることで、二人で協力してハイスコアを目指すのも良し、相手のブロックを揃えさせて相手の得点を減らしたりなどして対立や裏切りなどをするのも良しという自由あふれる機能を追加したかったのですが、一度に二つのブロックの表示を同じフィールドに出現させることがエラー多発によりできませんでした
* START画面に戻すこと。Void drawのリセットの方法がわからず断念しました

1. 参考資料

表1:ソースの引用元(<http://www.ct-robo.sakura.ne.jp/550>)

|  |  |
| --- | --- |
| 引用元のプログラム | 変更後のプログラム |
| ・a,bの変数による対応(a=縦の数,b=横の数です。記載しているのは一部です。) | |
| |  | | --- | | **void** clear\_lines**(){** | | 23: **int** line=0**;** | | 24: **int[]** line\_gc\_y=**new** **int** **[**20**];** | | 25: **int** count\_line=0**;** | | 26: **boolean** flag=false**;** | | 27: **for(**y=1**;**y<21**;**y++**){** | | 28: **for(**x=1**;**x<11**;**x++**){** | | 29: line+=field**[**y**][**x**];** | | 30: **}** | | 31: **if(**line==30**){** | | 32: flag=true**;** | | 33: **for(**x=1**;**x<11**;**x++**){** | | 34: field**[**y**][**x**]**=1**;** | | 35: **}** | | 36: line\_gc\_y**[**count\_line++**]**=y**;** | | 37: **}** | | 38: line=0**;** | | 39: **}** | |  | | |  | | --- | | **void** clear\_lines**(){** | | 2: **int** line=0**;** | | 3: **int[]** line\_gc\_y=**new** **int** **[**a-2**];** | | 4: **int** count\_line=0**;** | | 5: **boolean** flag=false**;** | | 6: **for(**y=1**;**y<a-1**;**y++**){** | | 7: **for(**x=1**;**x<b-1**;**x++**){** | | 8: line+=field**[**y**][**x**];** | | 9: **}** | | 11: **if(**line==**(**b-2**)**\*3**){** | | 12: flag=true**;** | | 13: **for(**x=1**;**x<b-1**;**x++**){** | | 14: field**[**y**][**x**]**=1**;** | | 15: **}** | | 16: line\_gc\_y**[**count\_line++**]**=y**;** | | 17: **}** | | 18: line=0**;** | | 19: **}** | |
| ・Void化による条件起動(起動引数が条件を満たすと常に動くようにしました) | |
| |  | | --- | | **void** draw**(){** | | 24: **if(**field**[**1**][**5**]**==3||field**[**1**][**6**]**==3**)**gameover=true**;** | | 25: **if(**millis**()**-ms>delay\_time**()**&&!gameover**){** | | 26: **if(**!create\_block**())**move\_block**();** | | 27: draw\_field**();** | | 28: clear\_lines**();** | | 29: ms=millis**();** | | 30: message\_box**();** | | 31: **}** | | 32: **else** **if(**gameover**){** | | 33: rect**(**70**,**310**,**260**,**60**);** | | 34: textAlign**(**CENTER**);** | | 35: textSize**(**40**);** | | 36: fill**(**#ffffff**);** | | 37: text**(**"gameover"**,**200**,**350**);** | | 38: fill**(**#000000**);** | | 39: **}** | | 40: **}** | | |  | | --- | | **void** draw**(){** | | 2: | | 3: **if** **((**keyPressed == true**)** && **((**key == 'n'**)** || **(**key == 'N'**)))** **{** | | 4: pointn++**;** | | 5: **}** | | 6: **else** **if** **(**hiddenmodepoint>=1 && **(**keyPressed == true**)** && **((**key == 'h'**)** || **(**key == 'H'**)))** **{** | | 7: pointh++**;** | | 8: **}** | | 9: **else** **if** **(**hiddenmodepoint>=1 && **(**keyPressed == true**)** && **((**key == 'm'**)** || **(**key == 'M'**)))** **{** | | 10: pointm++**;** | | 11: **}** | | 12: **else** **if** **((**keyPressed == true**)** && **((**key == 'r'**)** || **(**key == 'R'**)))** **{** | | 13: PImage img = loadImage**(**"rule.png"**);** | | 14: image**(**img**,** 0**,** 0**);** | | 15: **}** | | 16: **else** **{** | | 17: **}** | | 18: **if** **(**pointn>=1 || pointh>=1 || pointm>=1**)** **{** | | 19: Tetrisbase**();** | | 20: **}** | |
| ・MODE選択別の入力方法の変更(hiddenmodepoint=hardとmirrorの開放引数) | |
| |  | | --- | | **void** keyPressed**(){** | | 47: **if(**key==CODED&&!gameover**)** **{** | | 48: **if(**keyCode==UP**){** | | 49: turn\_block**();** | | 50: draw\_field**();** | | 51: **}** | | 52: **else** **if(**keyCode==DOWN**){** | | 53: move\_block\_down**();** | | 54: draw\_field**();** | | 55: **}** | | 56: **else** **if(**keyCode==RIGHT**){** | | 57: move\_block\_right**();** | | 58: draw\_field**();** | | 59: **}** | | 60: **else** **if(**keyCode==LEFT**){** | | 61: move\_block\_left**();** | | 62: draw\_field**();** | | 63: **}** | | 64: **}** | | 65: **else** **if(**key==' '**){** | | 66: move\_block\_left**();** | | 67: draw\_field**();** | | 68: **}** | | 69: **}** | | |  | | --- | | **void** keyPressed**(){** | | 2: **if(**key==CODED&&!gameover**)** **{** | | 3: **if(**keyCode==UP && pointh<1**){** | | 4: turn\_block**();** | | 5: draw\_field**();** | | 6: **}** | | 7: **else** **if(**keyCode==DOWN**){** | | 8: move\_block\_down**();** | | 9: draw\_field**();** | | 10: **}** | | 11: **else** **if(**pointm<=0**){** | | 12: **if(**keyCode==RIGHT**){** | | 13: move\_block\_right**();** | | 14: draw\_field**();** | | 15: **}** | | 16: **else** **if(**keyCode==LEFT**){** | | 17: move\_block\_left**();** | | 18: draw\_field**();** | | 19: **}** | | 20: **}** | | 21: **else** **if(**pointm>=1**){** | | 22: **if(**keyCode==RIGHT**){** | | 23: move\_block\_left**();** | | 24: draw\_field**();** | | 25: **}** | | 26: **else** **if(**keyCode==LEFT**){** | | 27: move\_block\_right**();** | | 28: draw\_field**();** | | 29: **}** | | 30: **}** | | 31: **}** | | 32: **else** **if(**key==' '**){** | | 33: move\_block\_left**();** | | 34: draw\_field**();** | | 35: **}** | | 36: **else** **if** **((**keyPressed == true**)** && **((**key == 'p'**)** || **(**key == 'P'**)))** **{** | | 37: fill**(**255**);** | | 38: rect**(**150**,**400**,**500**,**120**);** | | 39: fill**(**0**);** | | 40: text**(**"1 Play mode(Hard) Press H key"**,**400**,**430**);** | | 41: text**(**"1 Play mode(Mirror) Press M key"**,**400**,**510**);** | | 42: hiddenmodepoint++ **;** | | 43: **}** | | 44: **}** | |