

情報科学演習_第4期課題_
並行プログラミング、アニメーション

6321120 横溝尚也

提出日：1月20日（金）

ファイル名

Gamemake.java(実行可能なプログラム)

Gamemake1.java(描画されないがよりプログラムを完成に近づけようとしたプログラム)

1 ゲーム内容と自分のプログラムの完成度

1.1 ゲーム内容について

- ・下にブロックを横1列に、その上にキャラを描画し、左右カーソルキーによってキャラが動く
- ・上からランダムにコインが降ってきて（テトリスのように）そのコインをキャラが獲得できるようにカーソルを動かしコインをより多く獲得する

1.2 自分が作成できたプログラムの内容

- ・ブロックとキャラ、コインの出力やキャラがブロックの上を左右に移動することの実現
- ・スペースボタンの入力によるアニメーション終了

1.3 プログラムの作成を試みたが実現できなかった内容

今回の課題では年末年始かなりの時間を割き、課題に取り組みましたが、ゲームの実現には程遠い完成度のプログラムしか作成できませんでした。以下が実現できなかった内容である。

- ・コインの上から下への自由落下（等速直線運動）
- ・コインを獲得した時のコインの削除

2 プログラムの説明

実行しても描画されないがより多くの機能を記述しようとした Gameamkel.java のプログラムの説明を行う。

いかがそのプログラムの内容である。説明で使用するために行数もふってある。

```
1import javafx.application.Application;  
2import javafx.scene.Scene;
```

```

3import javafx.stage.Stage;
4import javafx.scene.Group;
5import javafx.scene.canvas.Canvas;
6import javafx.scene.paint.Color;
7import javafx.scene.canvas.GraphicsContext;
8import javafx.animation.AnimationTimer;
9import javafx.scene.image.Image;
10import javafx.scene.image.ImageView;
11import javafx.scene.shape.Rectangle;
12import javafx.scene.input.KeyEvent;
13import javafx.scene.input.KeyCode;

14public class Gamemake1 extends Application{
15    private int initialSceneWidth = 400;
16    private int initialSceneHeight = 400;
17    private MapAndChars mapAndChars;

17    @Override
18    public void start(Stage st) throws Exception {
19        Group root = new Group();

20        Scene scene = new Scene(root, initialSceneWidth, 21initialScene
22        st.setTitle("Tile-based Game");
23        st.setScene(scene);
24        st.show();

25        scene.setOnKeyPressed(this::keyPressed);
26        mapAndChars = new MapAndChars(root);

27        drawScreen();

```

```

28     AnimationTimer timer = new AnimationTimer() {
29         @Override
30         public void handle(long t) {
31     drawScreen();
32         }
33     };
34     timer.start();
35 }

36     public static void main(String[] a) {
37 launch(a);
38     }

39     public void drawScreen(){
40 mapAndChars.drawboy();
41 mapAndChars.drawCoin();
42     }

43     public void keyPressed(KeyEvent e) {
44 KeyCode key = e.getCode();
45 int dir = -1;
46 switch ( key ) {
47 case LEFT: dir = 2; break;
48 case RIGHT: dir = 0; break;
49 case UP: dir = 1; break;
50 case DOWN: dir = 3; break;
51 }
52 if ( dir >= 0 ) mapAndChars.boyMove(dir);
53     }
54}

```

```

55class MapAndChars{
56    char[] [] map;
57    int MX = 6, MY = 6;
58    String[] initialMap = {
59        "      ",
60        "   C   ",
61        "C       ",
62        "      ",
63        "   M   ",
64        "BBBBBB",
65    };

66    int boyX, boyY;
67    ImageView boyView;
68    Coin coin;
69    boolean hasCoin = false;
70    ImageView coinView;

71    MapAndChars(Group root){
72    map = new char[MY+2][MX+2];
73        for (int x = 0; x <= MX+1; x++){
74    map[0][x] = 'B';
75    map[MY+1][x] = 'B';
76        }
77        for(int y = 0; y <= MY+1; y++){
78    map[y][0] = 'B';
79    map[y][MY+1] = 'B';
80        }

```

```

81     for (int y = 1; y <= MY; y++) {
82 for (int x = 1; x <= MX; x++) {
83     map[y][x] = initialMap[y-1].charAt(x-1);
84 }
85     }
86     drawInitialMapAndChars(root);
87 }

88     public void drawInitialMapAndChars(Group root) {
89 Image image;
90 for(int y = 1; y <= MX; y++){
91     for (int x = 1; x <= MX; x++){
92 int xx = 50*x, yy = 50*y;
93 switch ( map[y][x] ) {
94 case 'C':
95     image = new Image("./coin.gif");
96     coinView = new ImageView(image);
97     coin.draw();
98     root.getChildren().add(coinView);
99     break;
100 case 'B':
101     Block b = new Block(xx, yy);
102     root.getChildren().add(b.getParts());
103     break;
104 case 'M':
105     image = new Image("./boy.gif");
106     boyView = new ImageView(image);
107     boyX = x;
108     boyY = y;
109     drawboy();

```

```

110     root.getChildren().add(boyView);
111     break;
112 }
113     }
114 }
115     }

116     public void drawboy() {
117 boyView.setX(50*boyX);
118 boyView.setY(50*boyY);
119 boyView.toFront();
120 boyView.setFitWidth(50.);
121 boyView.setFitHeight(50.);
122     }

123     public void boyMove(int dir) {
124 int dx = 0, dy = 0;
125 switch ( dir ) {
126 case 0: dx = 1; break;
127 case 1: dy = -1; break;
128 case 2: dx = -1; break;
129 case 3: dy = 1; break;
130 }
131 if ( dx == 0 && dy == 0 ) return;
132 if ( map[boyY+dy][boyX+dx] == 'B' ) return;
133 boyX += dx; boyY += dy;
134 if ( map[boyY][boyX] == ' ' && map[boyY+1][boyX] == ' ' ) 135boyMov
136 if(map[boyY][boyX] == 'C') getItem();
137     }

```

```

138     public void drawCoin() {
139 if(hasCoin){
140     coin.clear();
141 }else{
142     coin.draw();
143     }
144 }

145     public void getItem() {
146 hasCoin = true;
147     }
148}

149abstract class Tile {
150     protected Group parts;
151     Tile(int x, int y) {
152         parts = new Group();
153         parts.setTranslateX(x);
154         parts.setTranslateY(y);
155         construct();
156     }

    //public void setX(double x) {parts.setTranslateX(x)};
    //public void setY(double y) {parts.setTranslateY(y)};
    //public double getX() {return parts.getTranslateX()};
    //public double getY() {return parts.getTranslateY()};
157     public Group getParts() {return parts;};
158     abstract public void construct();
159}

```



```

160class Block extends Tile {
161    Block(int x, int y) {
162        super(x, y);
163    }

164    @Override
165    public void construct() {
166        Color c = Color.GREEN;

167        Rectangle r1 = new Rectangle(0, 0, 26, 10);
168        r1.setFill(c);
169        Rectangle r2 = new Rectangle(32, 0, 18, 10);
170        r2.setFill(c);
171        Rectangle r3 = new Rectangle(0, 15, 10, 10);
172        r3.setFill(c);
173        Rectangle r4 = new Rectangle(16, 15, 34, 10);
174        r4.setFill(c);
175        Rectangle r5 = new Rectangle(0, 30, 18, 10);
176        r5.setFill(c);
177        Rectangle r6 = new Rectangle(24, 30, 26, 10);
178        r6.setFill(c);
179        parts.getChildren().add(r1);
180        parts.getChildren().add(r2);
181        parts.getChildren().add(r3);
182        parts.getChildren().add(r4);
183        parts.getChildren().add(r5);
184        parts.getChildren().add(r6);
185    }
186}

```

```

187class Coin{
188    Canvas canvas;
189    GraphicsContext gc;
190    ImageView coinView;
191    int coinX, coinY;

192    public void draw(){
193 coinView.setX(50*coinX);
194 coinView.setY(50*coinY);
195 coinView.toFront();
196 coinView.setFitWidth(50.);
197 coinView.setFitHeight(50.);
199    }

200    public void clear(){
201 gc.clearRect(coinX,coinY, 50.,50. );
202    }
203}

```

- ・ 1～13行目：クラスのインポート
- ・ 14～54行目：メインクラスの記述

drawScreen が画像の出力するメソッド、keyPressed がキーボード入力されたものを数値として簡潔にした。また、スペースキーが入力されたときは window 出力を終了する記述

- ・ 55～148行目：MapAndChars クラスの記述。初めに window に出力する内容を2次元配列として考え、外周は侵入不可として壁を作成した。配列の要素が M ならば計算された座標にキャラを出力、同様に壁 B とコイン C についても記述した。

121,122 行目では出力したい画像のサイズを指定して一つのブロックと同じ

サイズにした。drawMove では keyPressed で受け取った数値をもとに次のキャラの出力する座標を計算し、更新している。コインとキャラの座標が一致しているかどうかを判別するために hascoin,getItem を用意した。

- ・ 149～186行目：Tile クラスと、Block クラスの記述。construct メソッドで長方形の描画を利用してブロックの描画をおこなっている。

- ・ 187～203行目：コインとキャラの座標が一致したときにコインを削除するためにクラス coin を記述。clear は指定した範囲の出力を透明化している。

3 実行結果



図 1 出力結果

4 考察

Gamemake1.java を実行すると window が一瞬出力されすぐに消えてしまう。コンパイルエラーが出るわけではないため、画像出力をするときに何かしらのエラーが起きていると考える。

コインとキャラの座標が一致した時のプログラム (Gamemake1.java) がうまく出力されたとしたら、bool 型の hascoin=true となったときに、新たな変数 count を用意し、コインの獲得枚数を計算し、window に出力すれば、ゲームの得点結果がわかると考える。

コインを上から移動させるときにはコインを等速直線運動させればいいと考え、プログラムの記述を試した見たがうまく記述できませんでした。