3.2佐藤担当分

3.2.1CharactorControllerクラス

　このクラスはJPanelを継承し、 ActionListenerを実装しているクラスで、主にゲーム自体の構成やゲームの描画などを行っているクラスである。工夫した点はTimerに対するactionPerformed内にModelクラス内の自機の動きに関するメソッドを入れることにより動きをスムーズにすることができたことである。

(1)コンストラクタ

29 public CharactorController()

30 {

31 timer = new javax.swing.Timer(50,this); //timerにtimer挿入(0.05秒おき)

32

33 model = new Model(); //modelオブジェクトの生成

34

35 this.setBackground(Color.BLACK); //背景設定

36 this.setFocusable(true); //キー入力をGUI部品が受け付ける

37 this.addKeyListener(this); //キー入力を追加

38 }

　このクラスのコンストラクタでは、Modelクラスのオブジェクトを変数modelに生成し、その他上のようにtimerの生成、背景の設定、キー入力が受け付けるように関数を指定した。

(2)timerstartメソッド、timerstopメソッド

　これらのメソッドはそれぞれコンストラクタで生成したtimerをスタート、ストップするメソッドである。これらは次のTitleFrameクラスで使う。

(3)paintComponentメソッド

　ここではModelクラスで作ったメソッドを利用して自機、ライフ、敵、そして自機、敵の弾の描画を実行できるようにした。

(4)actionerPerformedメソッド

　これはtimerが更新するごとに実行されるメソッドである。つまり、時間が経つごとにこのメソッドは実行される。

86 public void actionPerformed(ActionEvent e)

87 {

88 time++; //timerを0.1秒に換算

89

90 model.JikiMove(jrflag,jlflag,jdflag,juflag);

91 if(jshotflag==true && time%5==0)

92 {

93 model.createBullet(model.xJiki()+model.wJiki()/3,model.yJiki(),1); //自機弾の発射

94 model.createBullet(model.xJiki()-model.wJiki()/3,model.yJiki(),1); //自機弾の発射

95 }

96

97 Game();

98

99 /\*敵機の出現\*/

100 if(time == 10) //1秒たったら

101 {

102 for(int i=-320;i<=0;i+=80)

103 model.createEnemy(100,i,1);

104

105 for(int i=-320;i<=0;i+=80)

106 model.createEnemy(350,i,1);

107 }

108 if(time == 150)

109 {

110 for(int i=-320;i<=0;i+=80)

111 model.createEnemy(50,i,2);

112

113 for(int i=-320;i<=0;i+=80)

114 model.createEnemy(400,i,3);

115 }

116 if(time == 200)

117 for(int i=-160;i<=0;i+=80)

118 model.createEnemy(300,i,1);

119 if(time == 230)

120 for(int i=-160;i<=0;i+=80)

121 model.createEnemy(150,i,1);

122 if(time == 260)

123 for(int i=-160;i<=0;i+=80)

124 model.createEnemy(350,i,3);

125 if(time == 290)

126 for(int i=-160;i<=0;i+=80)

127 model.createEnemy(100,i,2);

128 if(time == 360)

129 for(int i=-160;i<=0;i+=80)

130 model.createEnemy(270,i,1);

131

132

133

134

135 model.EnemyMove();

136 model.BulletMove();

137 model.HitCheck();

138 model.DeleteFigure();

139

140 repaint(); //そして再描画

141

142 }

　まず、timerの更新を視覚化するためにint型のtime変数を 更新ごとに1加算している。そして、model内のJikiMoveやcreateBulletを使って自機の動き、自機弾の発射を指定している。また、次に説明するGameメソッドを使用することにより、このあとのTitleFrameクラスでGameOver画面、GameClear画面に遷移できるようにしている。このあとの100行目から130行目に関しては、時間が経てばmodelクラス内の敵を出現するメソッドであるcreateEnemyメソッドを用いて敵を出現させるようにした。135行目から138行目のメソッドは当たり判定に関するメソッドで、詳しくはmodelクラスの説明内で解説してある。

最後、時間が経つごとに再描画してある。

(5)Gameメソッド

　このメソッドはゲームオーバーになったらint型変数iを1に、ゲームクリアになったらiを2にして、その後のTitleFrameクラスで画面の遷移ができるようにしている。

3.2.2TitleFrameクラス

　このクラスはTitle画面、GameOver画面、GameClear画面も全て含めたこのプログラム全体の構成を担っているクラスである。工夫した点はCardLayoutというレイアウトを使い、画面の遷移を可能にすることができたことである。このクラスはJFrameを継承し、ActionListener及びKeyListenerを実装している。キー操作に関するメソッドは飯野が担当分であるため、ここではそれ以外のプログラムに関して説明していく。

(1)コンストラクタ

21 /\*コンストラクタ(フレーム内の設定)\*/

22 public TitleFrame()

23 {

24 /\*JPanel設定\*/

25 p1 = new JPanel(); //大本のパネル

26 p2 = new JPanel(); //１ページ目のパネル（タイトル画面）

27 p3 = new JPanel(); //２ページ目のパネル（ゲーム画面）

28 p4 = new JPanel(); //3ページ目のパネル（ゲームオーバー画面）

29 p5 = new JPanel(); //4ページ目のパネル（クリア画面）

30 chara = new CharactorController(); //2ページ目に貼り付けるゲーム画面

31 timer = new javax.swing.Timer(1,this);

32 timer.start();

33

34 p2.setBackground(Color.BLACK); //背景設定

35 p4.setBackground(Color.BLACK); //背景設定

36 p5.setBackground(Color.BLACK); //背景設定

37

38

39 /\*CardLayoutの設定\*/

40 layout = new CardLayout();

41

42 /\*KeyListener追加\*/

43 p1.addKeyListener(this);

44 p1.setFocusable(true);

45

46 /\*p2,p4をBorderLayoutで設定\*/

47 p2.setLayout(new BorderLayout());

48 p4.setLayout(new BorderLayout());

49 p5.setLayout(new BorderLayout());

50

51 /\*p2にJLabelを貼り付ける\*/

52 JLabel l2 = new JLabel("<html>"

53 +"<span style='font-size:110pt;font-family:Impact,Charcoal;"

54 +"color:yellow;'>"

55 +"Shooting</span></html>",

56 +JLabel.CENTER);

57

58

59 JLabel l3 = new JLabel("<html>"

60 +"<span style='font-size:30pt;color:white;'>"

61 +"十字キーで移動、zで弾発射。<br/>"

62 +"<span style='font-size:30pt;color:white;'>"

63 +"Enterでゲームスタート!!"

64 +"</span></html>",

65 +JLabel.CENTER);

66

67 p2.add(l2,BorderLayout.CENTER);

68 p2.add(l3,BorderLayout.SOUTH);

69

70 /\*2ページ目\*/

71 p3.setLayout(new BorderLayout());

72 p3.add(chara,BorderLayout.CENTER);

73

74 /\*p4にJLabelを貼り付ける\*/

75 JLabel l4 = new JLabel("<html>"

76 +"<span style='font-size:90pt;font-family:Impact,Charcoal;"

77 +"color:blue;'>"

78 +"GAMEOVER</span></html>",

79 +JLabel.CENTER);

80

81

82 JLabel l5 = new JLabel("<html>"

83 +"<span style='font-size:30pt;color:white'>"

84 +"2秒後、タイトル画面を表示します。"

85 +"</span></html>",

86 +JLabel.CENTER);

87

88 p4.add(l4,BorderLayout.CENTER);

89 p4.add(l5,BorderLayout.SOUTH);

90

91 /\*p4にJLabelを貼り付ける\*/

92 JLabel l6 = new JLabel("<html>"

93 +"<span style='font-size:120pt;font-family:Impact,Charcoal;"

94 +"color:red;'>"

95 +"GAME<br/>CLEAR!!</span></html>",

96 +JLabel.CENTER);

97

98

99 JLabel l7 = new JLabel("<html>"

100 +"<span style='font-size:30pt;color:white'>"

101 +"2秒後、タイトル画面を表示します。"

102 +"</span></html>",

103 +JLabel.CENTER);

104

105

106 p5.add(l6,BorderLayout.CENTER);

107 p5.add(l7,BorderLayout.SOUTH);

108

109 /\*p1をCardLayoutで設定し、p1にパネルを貼り付ける。\*/

110 p1.setLayout(layout);

111 p1.add(p2,BorderLayout.CENTER);

112 p1.add(p3,BorderLayout.CENTER);

113 p1.add(p4,BorderLayout.CENTER);

114 p1.add(p5,BorderLayout.CENTER);

115 getContentPane().add(p1,BorderLayout.CENTER); //これがないと表示されない(はず)

116

117

118 /\*Frame設定\*/

119 this.setSize(550,550); //Frameは550\*550(仮)

120 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

121 this.setTitle("Shooting");

122 this.setVisible(true);

123 }

46 /\*p2,p4をBorderLayoutで設定\*/

47 p2.setLayout(new BorderLayout());

48 p4.setLayout(new BorderLayout());

49 p5.setLayout(new BorderLayout());

50

51 /\*p2にJLabelを貼り付ける\*/

52 JLabel l2 = new JLabel("<html>"

53 +"<span style='font-size:110pt;font-family:Impact,Charcoal;"

54 +"color:yellow;'>"

55 +"Shooting</span></html>",

56 +JLabel.CENTER);

57

58

59 JLabel l3 = new JLabel("<html>"

60 +"<span style='font-size:30pt;color:white;'>"

61 +"十字キーで移動、zで弾発射。<br/>"

62 +"<span style='font-size:30pt;color:white;'>"

63 +"Enterでゲームスタート!!"

64 +"</span></html>",

65 +JLabel.CENTER);

66

67 p2.add(l2,BorderLayout.CENTER);

68 p2.add(l3,BorderLayout.SOUTH);

69

70 /\*2ページ目\*/

71 p3.setLayout(new BorderLayout());

72 p3.add(chara,BorderLayout.CENTER);

73

74 /\*p4にJLabelを貼り付ける\*/

75 JLabel l4 = new JLabel("<html>"

76 +"<span style='font-size:90pt;font-family:Impact,Charcoal;"

77 +"color:blue;'>"

78 +"GAMEOVER</span></html>",

79 +JLabel.CENTER);

80

81

82 JLabel l5 = new JLabel("<html>"

83 +"<span style='font-size:30pt;color:white'>"

84 +"2秒後、タイトル画面を表示します。"

85 +"</span></html>",

86 +JLabel.CENTER);

87

88 p4.add(l4,BorderLayout.CENTER);

89 p4.add(l5,BorderLayout.SOUTH);

90

91 /\*p4にJLabelを貼り付ける\*/

92 JLabel l6 = new JLabel("<html>"

93 +"<span style='font-size:120pt;font-family:Impact,Charcoal;"

94 +"color:red;'>"

95 +"GAME<br/>CLEAR!!</span></html>",

96 +JLabel.CENTER);

97

98

99 JLabel l7 = new JLabel("<html>"

100 +"<span style='font-size:30pt;color:white'>"

101 +"2秒後、タイトル画面を表示します。"

102 +"</span></html>",

103 +JLabel.CENTER);

104

105

106 p5.add(l6,BorderLayout.CENTER);

107 p5.add(l7,BorderLayout.SOUTH);

108

109 /\*p1をCardLayoutで設定し、p1にパネルを貼り付ける。\*/

110 p1.setLayout(layout);

111 p1.add(p2,BorderLayout.CENTER);

112 p1.add(p3,BorderLayout.CENTER);

113 p1.add(p4,BorderLayout.CENTER);

114 p1.add(p5,BorderLayout.CENTER);

115 getContentPane().add(p1,BorderLayout.CENTER); //これがないと表示されない(はず)

116

117

118 /\*Frame設定\*/

119 this.setSize(550,550); //Frameは550\*550(仮)

120 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

121 this.setTitle("Shooting");

122 this.setVisible(true);

123 }

99 JLabel l7 = new JLabel("<html>"

100 +"<span style='font-size:30pt;color:white'>"

101 +"2秒後、タイトル画面を表示します。"

102 +"</span></html>",

103 +JLabel.CENTER);

104

105

106 p5.add(l6,BorderLayout.CENTER);

107 p5.add(l7,BorderLayout.SOUTH);

108

109 /\*p1をCardLayoutで設定し、p1にパネルを貼り付ける。\*/

110 p1.setLayout(layout);

111 p1.add(p2,BorderLayout.CENTER);

112 p1.add(p3,BorderLayout.CENTER);

113 p1.add(p4,BorderLayout.CENTER);

114 p1.add(p5,BorderLayout.CENTER);

115 getContentPane().add(p1,BorderLayout.CENTER);

116

117

118 /\*Frame設定\*/

119 this.setSize(550,550); //Frameは550\*550(仮)

120 this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

121 this.setTitle("Shooting");

122 this.setVisible(true);

123 }

　JPanelは全体、タイトル画面、ゲーム画面、ゲームオーバー画面、ゲームクリア画面の5つのPanelを用意した。そして、timerの生成、背景設定、キー入力が受け付けるための設定、レイアウト設定をした。

　51行目から68行目に関してはTitle画面の生成をしている。JLabelを使い、HTMLでフォントや文字の大きさなどを指定しながらゲームタイトルと簡単なゲーム操作を記載した。また、74行目から89行目、91行目から107行目はそれぞれGameOver画面、GameClear画面をTitle画面と同様にJLabel、HTMLを使用しながらゲームオーバーもしくはゲームクリアである旨と画面が一定時間後に自動的に遷移するということを記載した。また、変数charaに前記のCharacterControllerクラスのオブジェクトを生成し、そのcharaをp3に貼り付けた。

　そして、p1に複数の部品を切り替えて表示できるCardLayoutを設定し、p2~p5を貼り付け、それが表示されるように115行目のようなメソッド記述をした。

　最後にはFrameのサイズなどの設定をした。

(2)actionPerformedメソッド

　このメソッドはtimerが更新されるごとに実行されるメソッドで、主にここでは画面の遷移について設定してある。

　前記のGameメソッドにより、ゲームオーバーになればi==1、ゲームクリアになればi==2となるように指定したある。このメソッドではこのi==1である時、layout.nextメソッドを使ってページを遷移し、GameOver画面に移行するように設定してある。i==2のときも同様にlayout.nextメソッドを使ってGameClear画面に移行するようにしている。そして、変数iを4に変えて、int型のtime変数を更新されるごとに加算。一定時間たったらsetVisibleメソッドでこのフレームを消し、新たなフレームを生成することによって 新たなゲームがプレイできるようにしている。

(3)mainメソッド

　このメソッドではフレームの生成しか行っていない。

[文責：佐藤]