

Whitebox Test Game

Erwartete, und getestete Ergebnisse:

Breakpoints:

1. Nach der Methode viewGame_Click(Object, sender, EventArgs e)
 - a. Der Flappy Bird Fliegt mit einem Linksklick 75 Pixel nach oben

```
30  
31  
32 private void ViewGame_Click(object sender, EventArgs e)  
33 {  
34     pannelBoxVogel.Location = new Point(pannelBoxVogel.Location.X, pannelBoxVogel.Location.Y - 75);  
35 }
```

2. Nach der Methode viewGame_KeyDown(Object, sender, EventArgs e)
 - a. Der Flappy Bird fliegt mit betätigung der Leertaste 75 Pixel nach oben
 - b. Wenn der Spieler tot ist, wird mit Leertaste der Button „Neustart“ ausgeführt.

```
36  
37 private void ViewGame_KeyDown(object sender, KeyEventArgs e)  
38 {  
39     if (!ControllerGame.spielerLebt)  
40     {  
41         buttonNeustart_Click(sender, e);  
42     }  
43  
44     if (e.KeyValue == (char)Keys.Space)  
45     {  
46         if (ControllerGame.leertasteDruecken && ControllerGame.spielerLebt)  
47         {  
48             pannelBoxVogel.Location = new Point(pannelBoxVogel.Location.X, pannelBoxVogel.Location.Y - 75);  
49         }  
50     }  
51 }  
52  
53  
54  
55
```

3. Nach der Methode ueberpruefungObFlappyBirdAnRoehrelst()
 - a. Wenn Flappy Bird einer der Röhren berührt, werden die Methoden playerDeath() und deathScreen() ausgeführt.

```
56  
57 private void ueberpruefungObFlappyBirdAnRoehrelst()  
58 {  
59     while (true)  
60     {  
61         if (ControllerGame.spielerLebt == true)  
62         {  
63             if (pannelBoxRoehre1.Location.X <= 100 + 70 && pannelBoxRoehre1.Location.X >= 30)  
64             {  
65                 if (pannelBoxVogel.Location.Y <= pannelBoxRoehre1.Location.Y + 60)  
66                 {  
67                     playerDeath();  
68                     deathScreen();  
69                 }  
70             }  
71  
72             if (pannelBoxRoehre2.Location.X <= 100 + 70 && pannelBoxRoehre2.Location.X >= 30)  
73             {  
74                 if (pannelBoxVogel.Location.Y >= 75 && pannelBoxRoehre2.Location.Y <= 100)  
75                 {  
76                     playerDeath();  
77                     deathScreen();  
78                 }  
79             }  
80  
81             if (pannelBoxRoehre3.Location.X <= 100 + 70 && pannelBoxRoehre3.Location.X >= 30)  
82             {  
83                 if (pannelBoxVogel.Location.Y <= pannelBoxRoehre3.Location.Y + 60)  
84                 {  
85                     playerDeath();  
86                     deathScreen();  
87                 }  
88             }  
89  
90             if (pannelBoxRoehre4.Location.X <= 100 + 70 && pannelBoxRoehre4.Location.X >= 30)  
91             {  
92                 if (pannelBoxVogel.Location.Y >= 75 && pannelBoxRoehre4.Location.Y <= 100)  
93                 {  
94                     playerDeath();  
95                     deathScreen();  
96                 }  
97             }  
98  
99             ControllerGame.vogelFlatter(pannelBoxVogel);  
100             if (pannelBoxRoehre1.Location.X == 100 || pannelBoxRoehre2.Location.X == 100)  
101             {  
102                 ControllerGame.score += 1;  
103                 LabelScore.Text = "Score : " + ControllerGame.score;  
104             }  
105         }  
106     }  
107 }
```

4. Nach der Methode timerRoehrenBewegen_Tick(objekt, seder, EventArgs e)
 - a. Die Röhren werden alle 100ms um 10 Pixel nach links bewegt.

```
108  
109 private void timerRoehrenBewegen_Tick(object sender, EventArgs e)  
110 {  
111     if (ControllerGame.spielerLebt)  
112     {  
113         pannelBoxRoehre1.Location = new Point(pannelBoxRoehre1.Location.X - 10, pannelBoxRoehre1.Location.Y);  
114         pannelBoxRoehre2.Location = new Point(pannelBoxRoehre2.Location.X - 10, pannelBoxRoehre2.Location.Y);  
115         pannelBoxRoehre3.Location = new Point(pannelBoxRoehre3.Location.X - 10, pannelBoxRoehre3.Location.Y);  
116         pannelBoxRoehre4.Location = new Point(pannelBoxRoehre4.Location.X - 10, pannelBoxRoehre4.Location.Y);  
117         ControllerGame.vogelFlatter(pannelBoxVogel);  
118         if (pannelBoxRoehre1.Location.X == 100 || pannelBoxRoehre2.Location.X == 100)  
119         {  
120             ControllerGame.score += 1;  
121             LabelScore.Text = "Score : " + ControllerGame.score;  
122         }  
123     }  
124 }
```

5. Nach der Methode timerRoehrenNeuGenerieren_Tick(objekt, sender, EventArgs e)
 - a. Die Röhren werden nachdem sie links angekommen sind, rechts neu generiert

```

125 private void timerRoehrenNeuGenerieren_Tick(object sender, EventArgs e)
126 {
127     switch (ControllerGame.ueberpruefenVonRoehren(panelBoxRoehre1, panelBoxRoehre2))
128     {
129     case 5:
130
131         panelBoxRoehre3.Location = new Point(730, ControllerGame.positionRoehren - 600);
132         panelBoxRoehre1.Location = new Point(730, panelBoxRoehre3.Location.Y + 600 + 225);
133         panelBoxRoehre1.Refresh();
134         panelBoxRoehre3.Refresh();
135
136         break;
137     case 6:
138
139         panelBoxRoehre4.Location = new Point(730, -600 + ControllerGame.positionRoehren);
140         panelBoxRoehre2.Location = new Point(730, panelBoxRoehre4.Location.Y + 600 + 225);
141         panelBoxRoehre2.Refresh();
142         panelBoxRoehre4.Refresh();
143
144         break;
145     case 0:
146         break;
147     }
148 }
149

```

6. Nach der Methode playerDeath()
 - a. spielerLebt wird false gesetzt und der Highscore wird in der Datenbank gespeichert.

```

170 }
171
172 internal static void playerDeath()
173 {
174     ControllerGame.spielerLebt = false;
175     if (ControllerGame.score > ControllerGame.userData.Highscore)
176     {
177         ControllerGame.userData.Highscore = ControllerGame.score;
178         //APPLAUS NEUER HIGHSCORE (Einfügen, Danke<3)
179         ControllerGame.changeScore();
180     }
181 }
182
183
184
185
186
187

```

7. Nach der Methode deathScreen()
 - a. Das Label „labelGestorben“ und die Button „buttonNeustart“ und „buttonSchliessen“ werden visible gemacht.

```

187
188 private void deathScreen()
189 {
190     labelGestorben.Invoke(new Action(() =>
191     {
192         labelGestorben.Visible = true;
193         buttonNeustart.Visible = true;
194         buttonSchliessen.Visible = true;
195     }));
196 }
197
198
199

```

8. Nach der Methode buttonNeustart_Click((objekt, sender, EventArgs e)
 - a. Alles wird auf die Startposition zurückgesetzt, die Button der methode deathScreen() werden wieder versteckt, spielerLebt wird = true gesetzt und der Focus wird auf das hauptspiel gesetzt.

```

199
200 private void buttonNeustart_Click(object sender, EventArgs e)
201 {
202
203
204     panelBoxRoehre1.Location = new Point(700, 506);
205     panelBoxRoehre2.Location = new Point(300, 625);
206     panelBoxRoehre3.Location = new Point(700, -319);
207     panelBoxRoehre4.Location = new Point(300, -200);
208     panelBoxVogel.Location = new Point(100, 400);
209     ControllerGame.score = 0;
210     labelScore.Text = "Score : 0";
211     ControllerGame.spielerLebt = true;
212     labelGestorben.Hide();
213     buttonSchliessen.Hide();
214     buttonNeustart.Hide();
215     Form.ActiveForm.Focus();
216
217
218
219

```

9. Nach der Methode formSchliessen(object, sender, EventArgs e)
- a. Hier wird die Form geschlossen

```
219
220 private void formSchliessen(object sender, EventArgs e)
221 {
222     Environment.Exit(0);
223 }
```

10. Nach der Methode formSchliessen(object, sender, FormClosedEventArgs e)
- a. Hier wird die Form geschlossen.

```
225 private void formSchliessen(object sender, FormClosedEventArgs e)
226 {
227     Environment.Exit(0);
228 }
```

11. Nach der Methode ueberpruefungVonRoehren(PictureBox roehre1, PictureBox roehre2)
- a. Setzt roehre1 und roehre2 mit dem selben abstand zueinander wieder nach rechts.

```
21 public static int ueberpruefenVonRoehren(PictureBox roehre1, PictureBox roehre2)
22 {
23     if(roehre1.Location.X <= -120)
24     {
25         Random random = new Random();
26         positionRoehren = random.Next(300, 601);
27         return 5;
28     }
29     if(roehre2.Location.X <= -120)
30     {
31         Random random = new Random();
32         positionRoehren = random.Next(300, 601);
33         return 6;
34     }
35     return 0;
36 }
37
38
39
40
```

12. Nach der Methode vogelFallen(PictureBox vogel)
- a. Der Vogel fällt kontinuierlich.
- b. Wenn man ebene 0 oder 862 erreicht, stirbt man

```
42 public static void vogelFallen(PictureBox vogel)
43 {
44     if (ControllerGame.spielerLebt == true)
45     {
46
47         if (vogel.Location.Y >= 862)
48         {
49             ViewGame.playerDeath();
50         }
51         if (vogel.Location.Y <= 0)
52         {
53             ViewGame.playerDeath();
54         }
55         if (ControllerGame.spielerLebt == true)
56         {
57             vogel.Location = new Point(vogel.Location.X, vogel.Location.Y + 10);
58         }
59     }
60 }
61
62
```

13. Nach der Methode changeScore()
- a. Der Highscore wird in die Tabelle „User“ eingetragen.

```
63 internal static void changeScore()
64 {
65     using(MySqlConnection conn = new MySqlConnection(connectionString))
66     {
67         try
68         {
69             conn.Open();
70             MySqlCommand cmd = new MySqlCommand("UPDATE User SET Highscore = " + score + " WHERE Username = '"+ userData.Benutzername+ "'");
71             cmd.Connection = conn;
72             cmd.ExecuteNonQuery();
73         } catch (Exception e)
74         {
75
76         }
77     }
78 }
79
80
```