## DEAKIN UNIVERSITY

## OBJECT ORIENTED DEVELOPMENT

ONTRACK SUBMISSION

## Different Robots

Submitted By: Akashdeep Akashdeep s223040483 2023/05/02 01:57

 $Tutor: \\ Son NGUYEN$ 

Outcome	$\mathbf{Weight}$
Build Programs	****

Learned how to create different types of robots and use them in the game

May 2, 2023



```
using System;
   using SplashKitSDK;
   using System. IO;
   using System.Collections.Generic;
   public class RobotDodge
6
       private Player Player;
        //private Robot _TestRobot;
       private Window _GameWindow;
10
       private List<Robot> _Robots = new List<Robot>();
        private List<Robot> _removedRobots = new List<Robot>();
12
13
       private List<Bullet> _Bullets = new List<Bullet>();
15
       private List<Bullet> _removedBullets = new List<Bullet>();
17
       private Bitmap HeartBitmap = new Bitmap("Heart", "heart.png");
18
        public SplashKitSDK.Timer myTimer;
19
20
22
23
        public bool Quit
24
25
            get
26
            {
27
                return _Player.Quit;
            }
29
        }
30
31
       public RobotDodge(Window window)
32
        {
            _GameWindow = window;
34
            _Player = new Player(window);
35
            //_TestRobot = RandomRobot();
36
            SplashKit.LoadMusic("background", "background.mp3");
37
            SplashKit.LoadSoundEffect("gameover", "gameover.wav");
            SplashKit.LoadSoundEffect("background", "background.wav");
39
            SplashKit.PlayMusic("background");
40
            myTimer = new SplashKitSDK.Timer("My Timer");
41
            myTimer.Start();
42
43
       }
       public void HandleInput()
46
47
            _Player.HandleInput();
48
            _Player.StayOnWindow(_GameWindow);
49
        }
51
       public void Draw()
52
        {
53
```

```
GameWindow.Clear(Color.Black);
54
            foreach (Robot robot in _Robots)
55
56
                 robot.Draw();
            }
58
59
             _Player.Draw();
60
            foreach (Bullet bullet in _Bullets)
61
                 bullet.Draw();
            }
            DisplayHUD();
65
            if (_Player.Lives <= 0)</pre>
66
                 _GameWindow.Clear(Color.Black);
68
                 Bitmap _GameOver = new Bitmap("Game Over", "gameover.png");
                 SplashKit.DrawBitmap(_GameOver, 200, 100);
70
                 SplashKit.StopMusic();
                 SplashKit.PlaySoundEffect("gameover");
72
73
            }
             _GameWindow.Refresh(60);
        }
76
78
        public Robot RandomRobot()
79
            // _TestRobot = new Robot(_GameWindow, _Player);
            //return _TestRobot;
82
            Robot _RandomRobotOne = new Robot.Boxy(_GameWindow, _Player);
            Robot _RandomRobotTwo = new Robot.Roundy(_GameWindow, _Player);
84
            Robot _RandomRobotThree = new Robot.Custom(_GameWindow, _Player);
85
            double randomNumber = SplashKit.Rnd(900);
87
            if (randomNumber < 300)</pre>
             {
89
                 return _RandomRobotOne;
90
            else if (randomNumber > 300 & randomNumber < 600)</pre>
92
             {
                 return _RandomRobotTwo;
94
            }
95
            else
96
             {
                 return _RandomRobotThree;
            }
99
100
        public void Update()
101
102
            CheckCollisions();
103
             _Player.Score = Convert.ToInt32(myTimer.Ticks / 1000);
104
            foreach (Robot robot in _Robots)
105
             {
106
```

```
robot.Update();
107
             }
108
             //add random number of robots into the list
109
             double randomNumber = SplashKit.Rnd(1000);
             if (randomNumber < 25)</pre>
111
             {
112
                 _Robots.Add(RandomRobot());
113
             }
114
             if (SplashKit.MouseClicked(MouseButton.LeftButton))
116
             {
117
                 _Bullets.Add(AddBullet());
118
119
                 SplashKit.PlaySoundEffect("bullet");
120
             }
121
             foreach (Bullet bullet in _Bullets)
123
                 bullet.Update();
124
125
        }
126
        public Bullet AddBullet()
128
129
             Bullet _RandomBullet = new Bullet(_GameWindow, _Player);
130
             return RandomBullet;
131
         }
132
133
        private void CheckCollisions()
134
         {
135
136
             foreach (Robot robot in _Robots)
137
138
                 //check the player and robot collision to remove the robot from main
140
                     list
                 if (_Player.CollidedWith(robot) || robot.IsOffscreen(_GameWindow))
141
142
                      _removedRobots.Add(robot);
144
                 if (_Player.CollidedWith(robot) & _Player.Lives > 0)
145
146
                      _Player.Lives = _Player.Lives - 1;
147
148
                 //check the bullet and robot collision
149
                 foreach (Bullet bullet in _Bullets)
151
                      if (bullet.BulletCollidedWith(robot))
152
                      {
153
                          _removedBullets.Add(bullet);
154
                          _removedRobots.Add(robot);
155
                      }
156
                      if (bullet.IsOffscreen(_GameWindow))
157
                      {
158
```

```
_removedBullets.Add(bullet);
159
                      }
160
                  }
161
             }
             foreach (Robot robot in _removedRobots)
163
             {
164
                  _Robots.Remove(robot);
165
             }
166
             foreach (Bullet bullet in _removedBullets)
167
             {
168
                  _Bullets.Remove(bullet);
169
             }
170
171
         }
172
173
         public void DisplayHUD()
175
             DrawHearts(_Player.Lives);
176
             SplashKit.DrawText("SCORE: " + _Player.Score, Color.White, 0, 40);
177
         }
178
         public void DrawHearts(int numberOfHearts)
180
181
             int heartX = 0;
182
             for (int i = 0; i < numberOfHearts; i++)</pre>
183
184
                  if (heartX < 300)
185
                  {
186
                      SplashKit.DrawBitmap(HeartBitmap, heartX, 0);
187
                      heartX = heartX + 40;
188
                  }
189
             }
190
         }
191
192
    }
193
194
195
196
197
```

```
using System;
   using SplashKitSDK;
2
   public abstract class Robot
   {
5
        public double X
6
7
             get;
             set;
10
        public double Y
11
12
             get;
13
             set;
        }
15
        public Color MainColor
17
             get;
18
             set;
19
        }
20
        private Vector2D Velocity { get; set; }
22
23
        public int Width
24
25
26
             get
             {
27
                 return 50;
28
             }
29
        }
30
31
        //read only property
32
        public int Height
        {
34
             get
35
             {
36
                 return 50;
37
             }
38
        }
39
40
        public Circle CollisionCircle
41
42
             get
43
             {
44
                 return SplashKit.CircleAt(X+25, Y+25, 20);
             }
46
        }
47
48
       public Robot(Window gameWindow, Player player)
49
50
             //Initial position of robot
51
             //Randomly pick.. Top/bottom or Left/Right
52
             if (SplashKit.Rnd() < 0.5)</pre>
53
```

```
{
54
                 //We picked...Top/Bottom
55
56
                 //Start by picking the random position left to right (X)
                 X = SplashKit.Rnd(gameWindow.Width);
58
59
                 //Now work out if we are top or bottom?
60
                 if (SplashKit.Rnd() < 0.5)</pre>
61
62
                      Y = -Height; //Top...so above top
63
                 }
64
                 else
65
                 {
66
                      Y = gameWindow.Height; //Bottom so below bottom
67
68
             }
             else
70
             {
71
                 //We picked..Left // Right
72
                 Y = SplashKit.Rnd(gameWindow.Height);
73
                 if (SplashKit.Rnd() < 0.5)</pre>
75
                 {
76
                      X = -Width;
77
                 }
78
                 else
79
                 {
                      X = gameWindow.Width;
                 }
82
             }
83
84
             //process to add the velocity to robot
85
             const int SPEED = 4;
87
             //Get a point from Robot
88
             Point2D fromPt = new Point2D()
89
90
                 X = X,
                 Y = Y
92
             };
93
94
             //Get a point from Player
95
             Point2D toPt = new Point2D()
96
             {
                 X = player.X,
                 Y = player.Y
99
             };
100
101
             //Calculate the direction to head
102
             Vector2D dir;
103
             dir = SplashKit.UnitVector(SplashKit.VectorPointToPoint(fromPt, toPt));
104
105
             //Set the speed and assign the velocity
106
```

```
Velocity = SplashKit.VectorMultiply(dir, SPEED);
107
             //generate a random color for robot
108
             MainColor = Color.RandomRGB(200);
109
110
111
     public bool IsOffscreen(Window screen)
112
113
             return (X < -Width | | X > screen.Width | | Y < -Height | | Y > screen.Height);
114
        }
116
     public abstract void Draw();
117
        public void Update()
118
119
             X = X + Velocity.X;
120
             Y = Y + Velocity.Y;
121
        }
122
123
124
    internal class Boxy : Robot
125
    {
126
        public Boxy(Window gameWindow, Player player) : base(gameWindow, player)
127
128
129
130
        public override void Draw()
131
132
             double eyeY, mouthY;
133
             double leftX = X + 12;
134
             double rightX = X + 27;
135
             eyeY = Y + 10;
136
             mouthY = Y + 30;
137
138
             SplashKit.FillRectangle(Color.Gray, X, Y, Width, Height);
139
             SplashKit.FillRectangle(MainColor, leftX, eyeY, 10, 10);
140
             SplashKit.FillRectangle(MainColor, rightX, eyeY, 10, 10);
141
             SplashKit.FillRectangle(MainColor, leftX, mouthY, 25, 10);
142
             SplashKit.FillRectangle(MainColor, leftX + 2, mouthY + 2, 21, 6);
143
        }
145
    }
146
    internal class Roundy : Robot
147
148
        public Roundy(Window gameWindow, Player player) : base(gameWindow, player)
149
        {
150
        }
151
        public override void Draw()
152
153
             double leftX, midX, rightX;
154
             double midY, eyeY, mouthY;
155
156
             leftX = X + 17;
157
             midX = X + 25;
158
             rightX = X + 33;
159
```

```
160
             midY = Y + 25;
161
             eyeY = Y + 20;
162
             mouthY = Y + 35;
164
             SplashKit.FillCircle(Color.White, midX, midY, 25);
165
             SplashKit.DrawCircle(Color.Gray, midX, midY, 25);
166
             SplashKit.FillCircle(MainColor, leftX, eyeY, 5);
167
             SplashKit.FillCircle(MainColor, rightX, eyeY, 5);
168
             SplashKit.FillEllipse(Color.Gray, X, eyeY, 50, 30);
169
             SplashKit.DrawLine(Color.Black, X, mouthY, X + 50, Y + 35);
170
        }
171
    }
172
       public class Custom : Robot
173
174
    {
        public Custom(Window gameWindow, Player player) : base(gameWindow, player)
175
176
        }
177
        public override void Draw()
178
179
             double leftX, midX, rightX;
             double midY, eyeY, mouthY;
181
182
             leftX = X + 17;
183
             midX = X + 25;
184
             rightX = X + 33;
185
186
             midY = Y + 25;
187
             eyeY = Y + 20;
188
             mouthY = Y + 35;
189
190
             SplashKit.FillCircle(Color.Red, midX, midY, 25);
191
             SplashKit.FillCircle(Color.Blue, midX, midY, 20);
             SplashKit.FillCircle(Color.Green, midX, midY, 15);
193
             SplashKit.FillCircle(Color.Yellow, midX, midY, 10);
194
             SplashKit.FillCircle(Color.White, midX, midY, 5);
195
        }
196
    }
197
198
199
    }
200
```

File 3 of 3 Screenshot

