

DEAKIN UNIVERSITY

OBJECT ORIENTED DEVELOPMENT

ONTRACK SUBMISSION

Different Robots

Submitted By:

Akashdeep AKASHDEEP

s223040483

2023/05/02 01:57

Tutor:

Son NGUYEN

Outcome	Weight
Build Programs	◆◆◆◆◆

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

May 2, 2023



```
1  using System;
2  using SplashKitSDK;
3  using System.IO;
4  using System.Collections.Generic;
5
6  public class RobotDodge
7  {
8      private Player _Player;
9      //private Robot _TestRobot;
10     private Window _GameWindow;
11     private List<Robot> _Robots = new List<Robot>();
12     private List<Robot> _removedRobots = new List<Robot>();
13
14     private List<Bullet> _Bullets = new List<Bullet>();
15
16     private List<Bullet> _removedBullets = new List<Bullet>();
17
18     private Bitmap HeartBitmap = new Bitmap("Heart", "heart.png");
19     public SplashKitSDK.Timer myTimer;
20
21
22
23
24     public bool Quit
25     {
26         get
27         {
28             return _Player.Quit;
29         }
30     }
31
32     public RobotDodge(Window window)
33     {
34         _GameWindow = window;
35         _Player = new Player(window);
36         //_TestRobot = RandomRobot();
37         SplashKit.LoadMusic("background", "background.mp3");
38         SplashKit.LoadSoundEffect("gameover", "gameover.wav");
39         SplashKit.LoadSoundEffect("background", "background.wav");
40         SplashKit.PlayMusic("background");
41         myTimer = new SplashKitSDK.Timer("My Timer");
42         myTimer.Start();
43
44     }
45
46     public void HandleInput()
47     {
48         _Player.HandleInput();
49         _Player.StayOnWindow(_GameWindow);
50     }
51
52     public void Draw()
53     {
```

```
54     _GameWindow.Clear(Color.Black);
55     foreach (Robot robot in _Robots)
56     {
57         robot.Draw();
58     }
59
60     _Player.Draw();
61     foreach (Bullet bullet in _Bullets)
62     {
63         bullet.Draw();
64     }
65     DisplayHUD();
66     if (_Player.Lives <= 0)
67     {
68         _GameWindow.Clear(Color.Black);
69         Bitmap _GameOver = new Bitmap("Game Over", "gameover.png");
70         SplashKit.DrawBitmap(_GameOver, 200, 100);
71         SplashKit.StopMusic();
72         SplashKit.PlaySoundEffect("gameover");
73
74     }
75     _GameWindow.Refresh(60);
76 }
77
78
79 public Robot RandomRobot()
80 {
81     // _TestRobot = new Robot(_GameWindow, _Player);
82     //return _TestRobot;
83     Robot _RandomRobotOne = new Robot.Boxy(_GameWindow, _Player);
84     Robot _RandomRobotTwo = new Robot.Roundy(_GameWindow, _Player);
85     Robot _RandomRobotThree = new Robot.Custom(_GameWindow, _Player);
86
87     double randomNumber = SplashKit.Rnd(900);
88     if (randomNumber < 300)
89     {
90         return _RandomRobotOne;
91     }
92     else if (randomNumber > 300 & randomNumber < 600)
93     {
94         return _RandomRobotTwo;
95     }
96     else
97     {
98         return _RandomRobotThree;
99     }
100 }
101 public void Update()
102 {
103     CheckCollisions();
104     _Player.Score = Convert.ToInt32(myTimer.Ticks / 1000);
105     foreach (Robot robot in _Robots)
106     {
```

```
107         robot.Update();
108     }
109     //add random number of robots into the list
110     double randomNumber = SplashKit.Rnd(1000);
111     if (randomNumber < 25)
112     {
113         _Robots.Add(RandomRobot());
114     }
115
116     if (SplashKit.MouseClicked(MouseButton.LeftButton))
117     {
118         _Bullets.Add(AddBullet());
119
120         SplashKit.PlaySoundEffect("bullet");
121     }
122     foreach (Bullet bullet in _Bullets)
123     {
124         bullet.Update();
125     }
126 }
127
128 public Bullet AddBullet()
129 {
130     Bullet _RandomBullet = new Bullet(_GameWindow, _Player);
131     return _RandomBullet;
132 }
133
134 private void CheckCollisions()
135 {
136
137     foreach (Robot robot in _Robots)
138     {
139
140         //check the player and robot collision to remove the robot from main
141         ↪ list
142         if (_Player.CollidedWith(robot) || robot.IsOffscreen(_GameWindow))
143         {
144             _removedRobots.Add(robot);
145         }
146         if (_Player.CollidedWith(robot) & _Player.Lives > 0)
147         {
148             _Player.Lives = _Player.Lives - 1;
149         }
150         //check the bullet and robot collision
151         foreach (Bullet bullet in _Bullets)
152         {
153             if (bullet.BulletCollidedWith(robot))
154             {
155                 _removedBullets.Add(bullet);
156                 _removedRobots.Add(robot);
157             }
158             if (bullet.IsOffscreen(_GameWindow))
159             {
```

```
159         _removedBullets.Add(bullet);
160     }
161 }
162 }
163 foreach (Robot robot in _removedRobots)
164 {
165     _Robots.Remove(robot);
166 }
167 foreach (Bullet bullet in _removedBullets)
168 {
169     _Bullets.Remove(bullet);
170 }
171
172 }
173
174 public void DisplayHUD()
175 {
176     DrawHearts(_Player.Lives);
177     SplashKit.DrawText("SCORE: " + _Player.Score, Color.White, 0, 40);
178 }
179
180 public void DrawHearts(int numberOfHearts)
181 {
182     int heartX = 0;
183     for (int i = 0; i < numberOfHearts; i++)
184     {
185         if (heartX < 300)
186         {
187             SplashKit.DrawBitmap(HeartBitmap, heartX, 0);
188             heartX = heartX + 40;
189         }
190     }
191 }
192
193 }
194
195
196
197
```

```
1  using System;
2  using SplashKitSDK;
3
4  public abstract class Robot
5  {
6      public double X
7      {
8          get;
9          set;
10     }
11     public double Y
12     {
13         get;
14         set;
15     }
16     public Color MainColor
17     {
18         get;
19         set;
20     }
21
22     private Vector2D Velocity { get; set; }
23
24     public int Width
25     {
26         get
27         {
28             return 50;
29         }
30     }
31
32     //read only property
33     public int Height
34     {
35         get
36         {
37             return 50;
38         }
39     }
40
41     public Circle CollisionCircle
42     {
43         get
44         {
45             return SplashKit.CircleAt(X+25, Y+25, 20);
46         }
47     }
48
49     public Robot(Window gameWindow, Player player)
50     {
51         //Initial position of robot
52         //Randomly pick.. Top/bottom or Left/Right
53         if (SplashKit.Rnd() < 0.5)
```

```
54     {
55         //We picked...Top/Bottom
56
57         //Start by picking the random position left to right (X)
58         X = SplashKit.Rnd(gameWindow.Width);
59
60         //Now work out if we are top or bottom?
61         if (SplashKit.Rnd() < 0.5)
62         {
63             Y = -Height; //Top...so above top
64         }
65         else
66         {
67             Y = gameWindow.Height; //Bottom so below bottom
68         }
69     }
70     else
71     {
72         //We picked..Left // Right
73         Y = SplashKit.Rnd(gameWindow.Height);
74
75         if (SplashKit.Rnd() < 0.5)
76         {
77             X = -Width;
78         }
79         else
80         {
81             X = gameWindow.Width;
82         }
83     }
84
85     //process to add the velocity to robot
86     const int SPEED = 4;
87
88     //Get a point from Robot
89     Point2D fromPt = new Point2D()
90     {
91         X = X,
92         Y = Y
93     };
94
95     //Get a point from Player
96     Point2D toPt = new Point2D()
97     {
98         X = player.X,
99         Y = player.Y
100     };
101
102     //Calculate the direction to head
103     Vector2D dir;
104     dir = SplashKit.UnitVector(SplashKit.VectorPointToPoint(fromPt, toPt));
105
106     //Set the speed and assign the velocity
```

```
107     Velocity = SplashKit.VectorMultiply(dir, SPEED);
108     //generate a random color for robot
109     MainColor = Color.RandomRGB(200);
110
111 }
112 public bool IsOffscreen(Window screen)
113 {
114     return (X < -Width || X > screen.Width || Y < -Height || Y > screen.Height);
115 }
116
117 public abstract void Draw();
118 public void Update()
119 {
120     X = X + Velocity.X;
121     Y = Y + Velocity.Y;
122 }
123
124
125 internal class Boxy : Robot
126 {
127     public Boxy(Window gameWindow, Player player) : base(gameWindow, player)
128     {
129
130     }
131     public override void Draw()
132     {
133         double eyeY, mouthY;
134         double leftX = X + 12;
135         double rightX = X + 27;
136         eyeY = Y + 10;
137         mouthY = Y + 30;
138
139         SplashKit.FillRectangle(Color.Gray, X, Y, Width, Height);
140         SplashKit.FillRectangle(MainColor, leftX, eyeY, 10, 10);
141         SplashKit.FillRectangle(MainColor, rightX, eyeY, 10, 10);
142         SplashKit.FillRectangle(MainColor, leftX, mouthY, 25, 10);
143         SplashKit.FillRectangle(MainColor, leftX + 2, mouthY + 2, 21, 6);
144     }
145 }
146
147 internal class Roundy : Robot
148 {
149     public Roundy(Window gameWindow, Player player) : base(gameWindow, player)
150     {
151     }
152     public override void Draw()
153     {
154         double leftX, midX, rightX;
155         double midY, eyeY, mouthY;
156
157         leftX = X + 17;
158         midX = X + 25;
159         rightX = X + 33;
```



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

