

Jonathan Frances

7:17 AM

is it ok to think of the child class as a special case of the parent? with optional added functionality

Anthony Bogli

7:17 AM

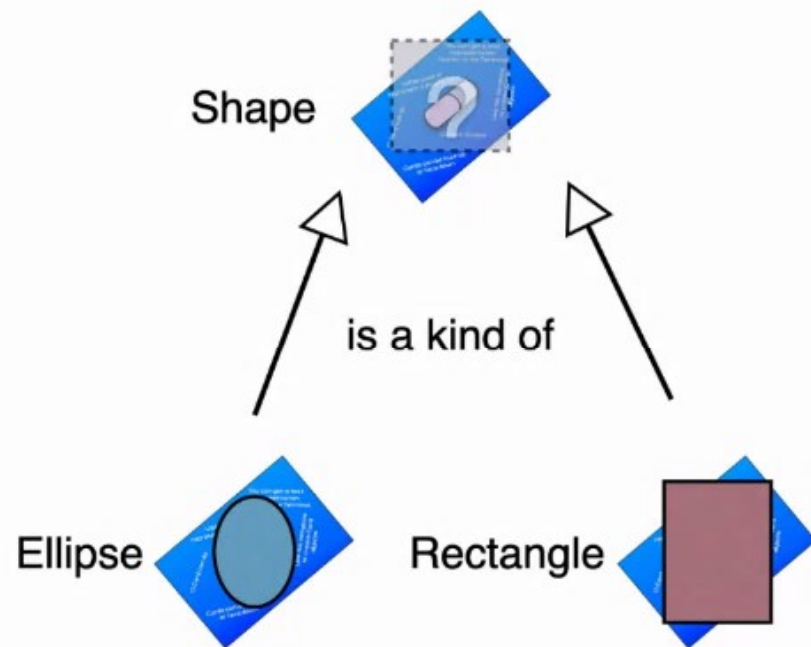
the idea of the inheritance system is for extendability.

like in an mmorpg

an item class could have a position and a image,

and we can say that a health potion or a mana potion is a type of item etc etc... with different uses

# Parent classes can have *placeholder* methods that **must** be overridden



How does Shape Draw?  
It doesn't; Draw is a placeholder = **abstract**  
abstract classes **cannot create objects**

Rectangle must override draw

Ellipse must override draw

So abstract methods are a guarantee that all children must implement that method, and any class with an abstract method can't be created as objects?

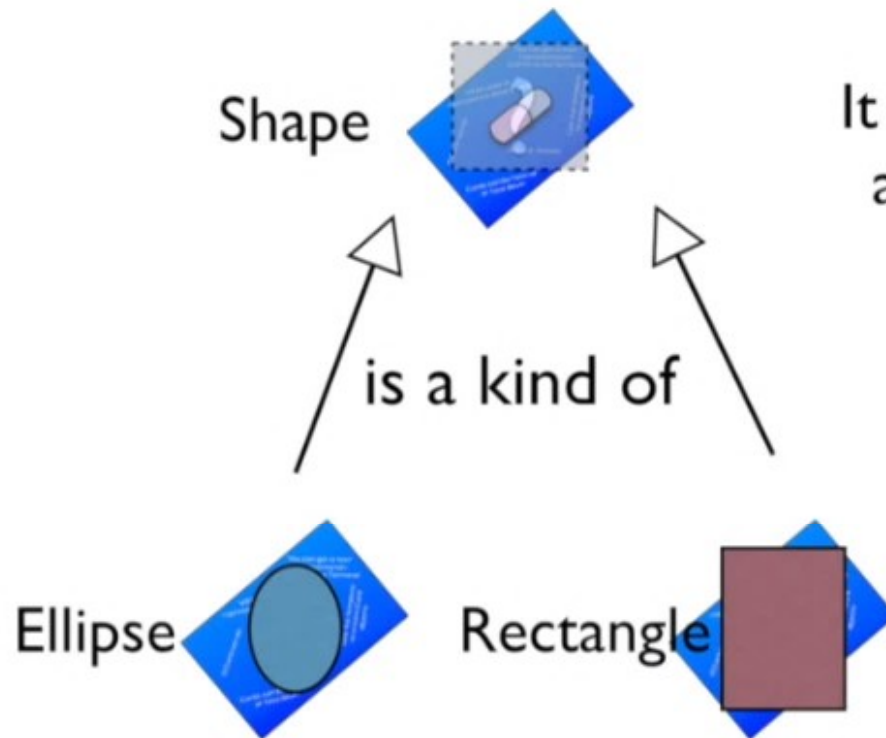
- - - - -  
If shape and rectangle both have a method with same name and we call it via object of rectangle than which method will be called?

Rect

# Parent classes can provide *placeholder* methods that **must** be overridden

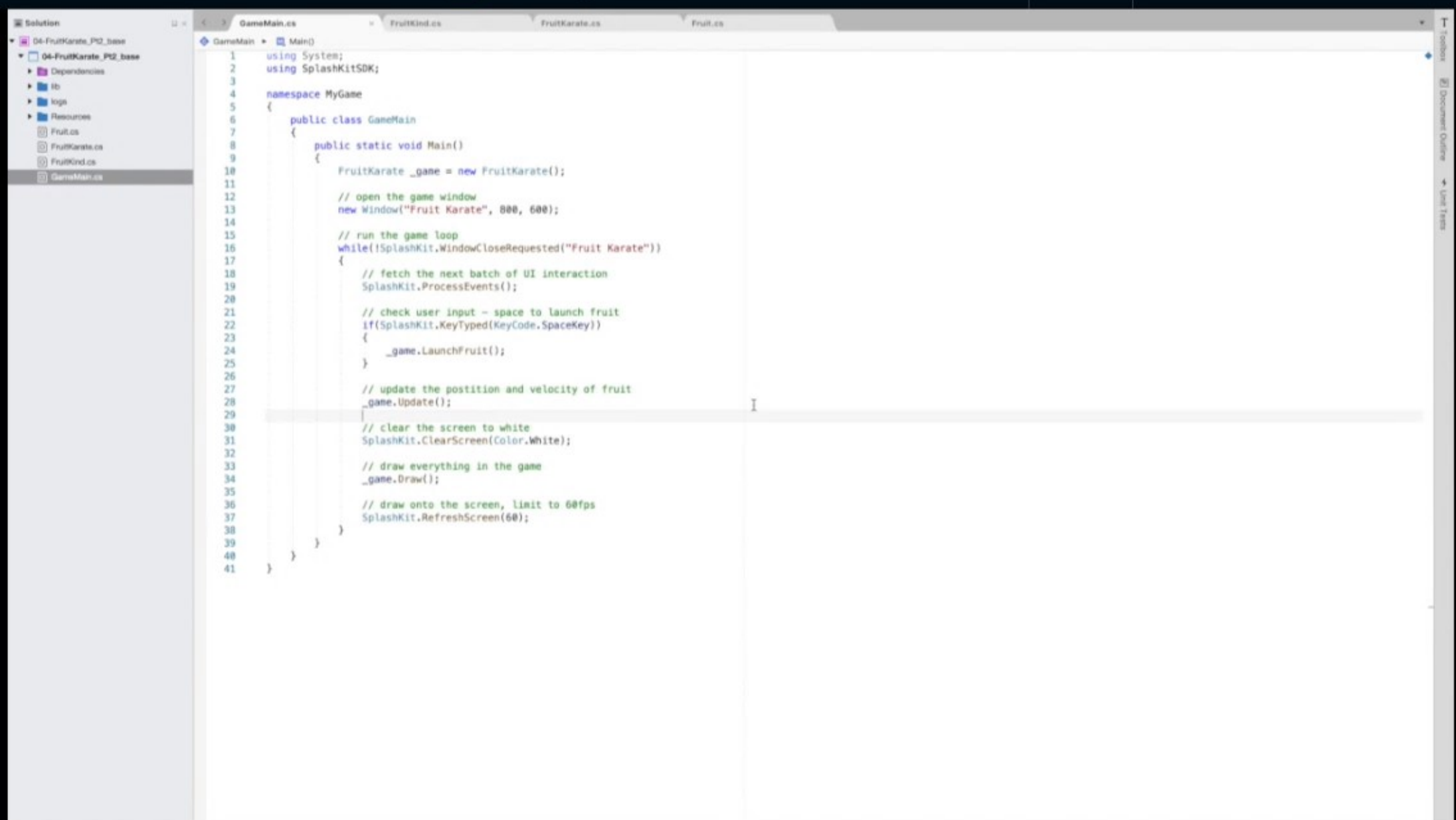
How does Shape Draw?

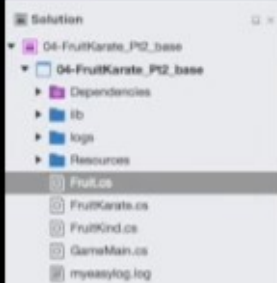
It doesn't; Draw is a placeholder = **abstract**  
abstract classes **cannot create objects**



Rectangle must override draw

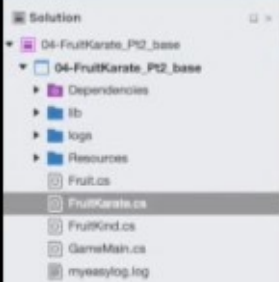
Ellipse must override draw





```
GameMain.cs FruitKind.cs FruitKarate.cs Fruit.cs
Fruit * IsAt(Point2D pt)
22 case FruitKind.Blueberry:
23     return SplashKit.LoadBitmap("Blueberry", "Blueberry.png");
24 case FruitKind.Pomegranate:
25     return SplashKit.LoadBitmap("Pomegranate", "Pomegranate.png");
26 case FruitKind.Apricot:
27     return SplashKit.LoadBitmap("Apricot", "Apricot.png");
28 case FruitKind.Raspberry:
29     return SplashKit.LoadBitmap("Raspberry", "Raspberry.png");
30 case FruitKind.Blackberry:
31     return SplashKit.LoadBitmap("Blackberry", "Blackberry.png");
32 case FruitKind.Strawberry:
33     return SplashKit.LoadBitmap("Strawberry", "Strawberry.png");
34 case FruitKind.Currant:
35     return SplashKit.LoadBitmap("Currant", "Currant.png");
36 default:
37     return SplashKit.LoadBitmap("Currant", "Currant.png");
38 }
39
40 public Fruit()
41 {
42     _position.X = 0;
43     _position.Y = SplashKit.ScreenHeight();
44
45     _velocity.X = 3.0;
46     _velocity.Y = -7.0 + SplashKit.Rnd(2) - 1;
47
48     _kind = (FruitKind) SplashKit.Rnd(9);
49 }
50
51 public void Update()
52 {
53     // update my position
54     _position.X += _velocity.X;
55     _position.Y += _velocity.Y;
56     // decay the velocity
57     _velocity = SplashKit.VectorAdd(_velocity, SplashKit.VectorTo(0, 0.05));
58 }
59
60 public void Draw()
61 {
62     SplashKit.DrawBitmap(MyBitmap(), _position.X, _position.Y);
63 }
64
65 public bool IsAt(Point2D pt)
66 {
67     return SplashKit.BitmapPointCollision(MyBitmap(), _position, pt);
68 }
69
70 }
71
```

class SplashKitSDK.SplashKit



```
1 using System;
2 using System.Collections.Generic;
3 using SplashKitSDK;
4
5 namespace MyGame
6 {
7     public class FruitKarate
8     {
9         private List<Fruit> _fruit = new List<Fruit>();
10
11         public FruitKarate() { }
12
13         public void LaunchFruit()
14         {
15             Fruit f = new Fruit();
16             _fruit.Add(f);
17         }
18
19         public void Update()
20         {
21             foreach (Fruit f in _fruit)
22             {
23                 f.Update();
24             }
25         }
26
27         public void Draw()
28         {
29             foreach (Fruit f in _fruit)
30             {
31                 f.Draw();
32             }
33         }
34
35         public void PunchAt(Point2D pt)
36         {
37             List<Fruit> toRemove = new List<Fruit>();
38
39             foreach (Fruit f in _fruit)
40             {
41                 if (f.IsAt(pt))
42                 {
43                     toRemove.Add(f);
44                 }
45             }
46
47             foreach (Fruit f in toRemove)
48             {
49                 _fruit.Remove(f);
50             }
51         }
52     }
53 }
54
55
```





Solution

- 04-FruitKarate\_Pt2\_base
  - Dependencies
  - lib
  - logs
  - Resources
    - Fruit.cs
    - FruitKarate.cs
    - FruitKind.cs
    - GameMain.cs
    - myeasylog.log

GameMain.cs

```
1 using System;
2 using SplashKitSDK;
3
4 namespace MyGame
5 {
6     public class GameMain
7     {
8         public static void Main()
9         {
10             FruitKarate _game = new FruitKarate();
11
12             // open the game window
13             new Window("Fruit Karate", 800, 600);
14
15             // run the game loop
16             while(!SplashKit.WindowCloseRequested("Fruit Karate"))
17             {
18                 // fetch the next batch of UI interaction
19                 SplashKit.ProcessEvents();
20
21                 // check user input - space to launch fruit
22                 if(SplashKit.KeyTyped(KeyCode.SpaceKey))
23                 {
24                     _game.LaunchFruit();
25                 }
26
27                 // punch fruit if mouse is clicked
28                 if (SplashKit.MouseClicked(MouseButton.LeftButton))
29                 {
30                     _game.PunchAt(SplashKit.MousePosition());
31                 }
32
33                 // update the position and velocity of fruit
34                 _game.Update();
35
36                 // clear the screen to white
37                 SplashKit.ClearScreen(Color.White);
38
39                 // draw everything in the game
40                 _game.Draw();
41
42                 // draw onto the screen, limit to 60fps
43                 SplashKit.RefreshScreen(60);
44             }
45         }
46     }
47 }
```

```
+ 04-FruitKarate_Pt2_base open
+ 04-FruitKarate_Pt2_base open
+ 04-FruitKarate_Pt2_base sln dotnet run
+ 04-FruitKarate_Pt2_base sln dotnet run
+ 04-FruitKarate_Pt2_base sln dotnet run
```

00  
ru  
00  
K1  
th  
.P  
US  
K1  
.L  
fr  
HK  
.P  
t  
at  
th  
.C  
ve  
w(  
nt  
.R



Solution

04-FruitKarate\_PS2\_base

04-FruitKarate\_PS2\_base

Dependencies

lib

logs

Resources

Fruit.cs

FruitKarate.cs

FruitKind.cs

GameMain.cs

myeasylog.log

GameMain.cs

FruitKind.cs

FruitKarate.cs

Fruit.cs

GameMain

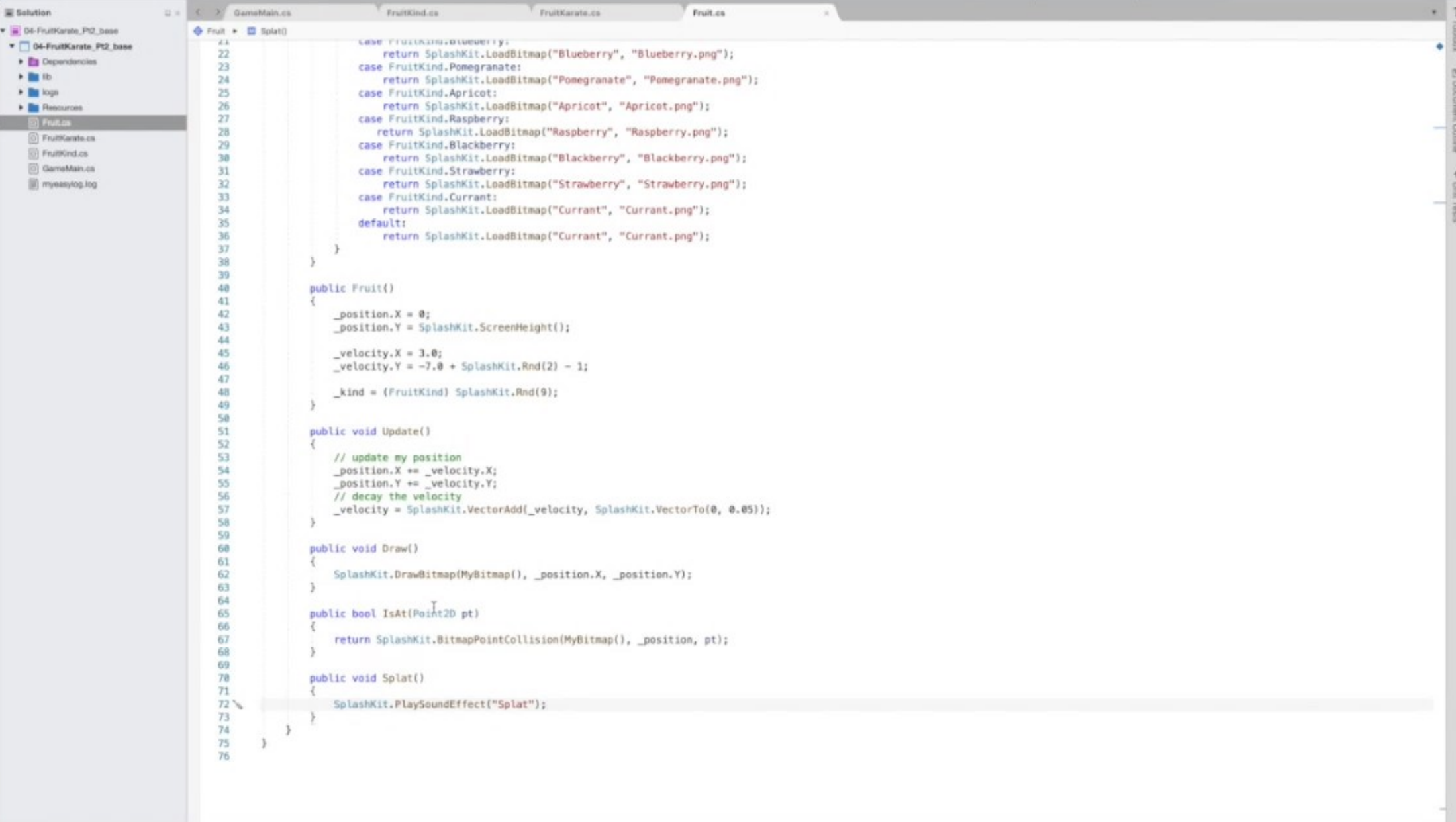
Main()

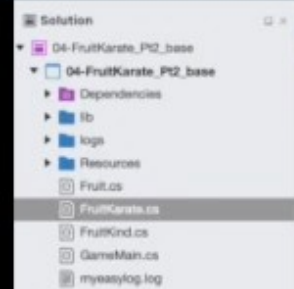
```
1 using System;
2 using SplashKitSDK;
3
4 namespace MyGame
5 {
6     public class GameMain
7     {
8         public static void Main()
9         {
10             SplashKit.LoadSoundEffect("Splat", "Splat.wav");
11
12             FruitKarate _game = new FruitKarate();
13
14             // open the game window
15             new Window("Fruit Karate", 800, 600);
16
17             // run the game loop
18             while(!SplashKit.WindowCloseRequested("Fruit Karate"))
19             {
20                 // fetch the next batch of UI interaction
21                 SplashKit.ProcessEvents();
22
23                 // check user input - space to launch fruit
24                 if(SplashKit.KeyTyped(KeyCode.SpaceKey))
25                 {
26                     _game.LaunchFruit();
27                 }
28
29                 // punch fruit if mouse is clicked
30                 if (SplashKit.MouseClicked(MouseButton.LeftButton))
31                 {
32                     _game.PunchAt(SplashKit.MousePosition());
33                 }
34
35                 // update the postition and velocity of fruit
36                 _game.Update();
37
38                 // clear the screen to white
39                 SplashKit.ClearScreen(Color.White);
40
41                 // draw everything in the game
42                 _game.Draw();
43
44                 // draw onto the screen, limit to 60fps
45                 SplashKit.RefreshScreen(60);
46             }
47         }
48     }
49 }
```

Toolbox

Document Outline

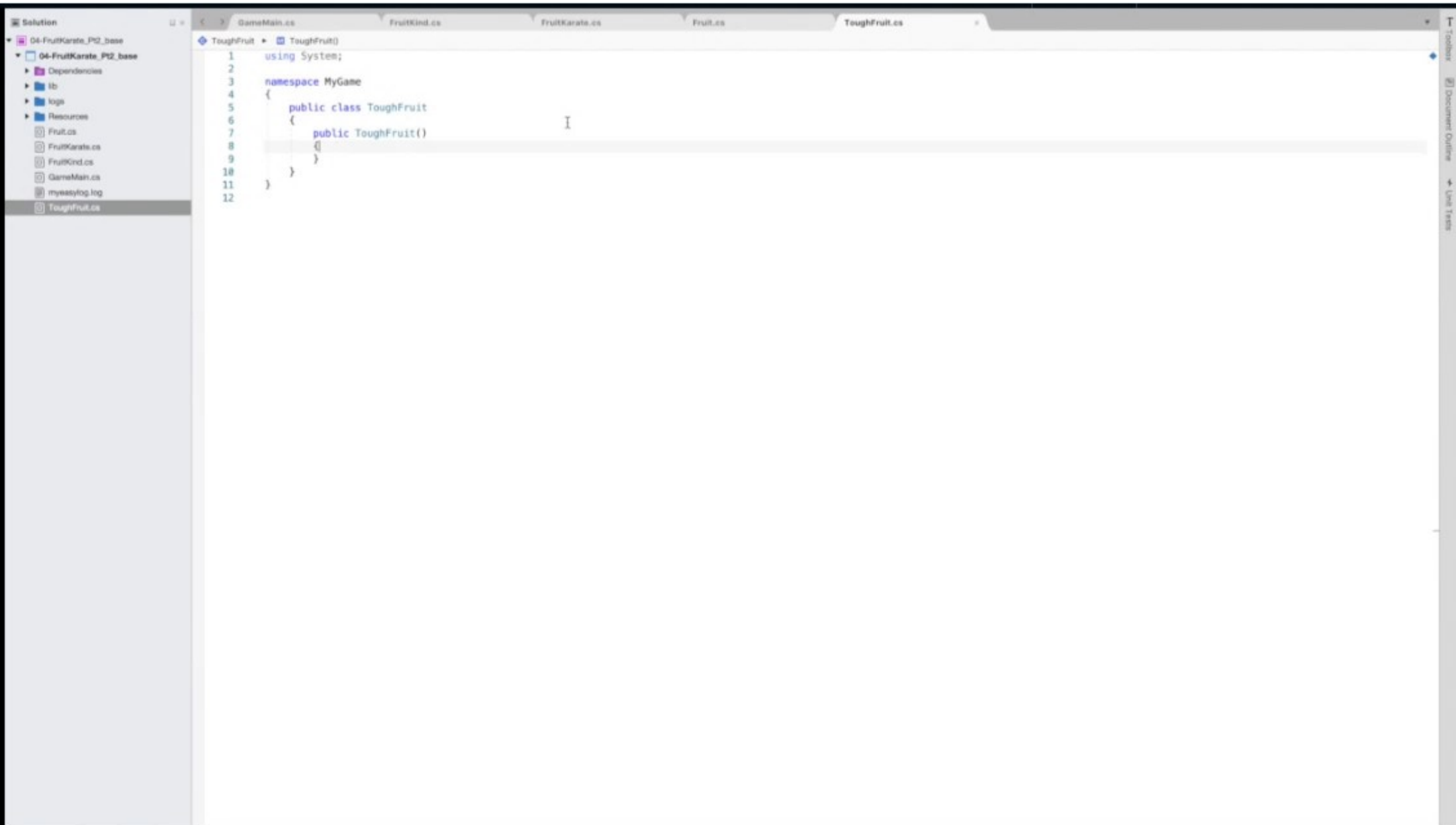
Unit Tests

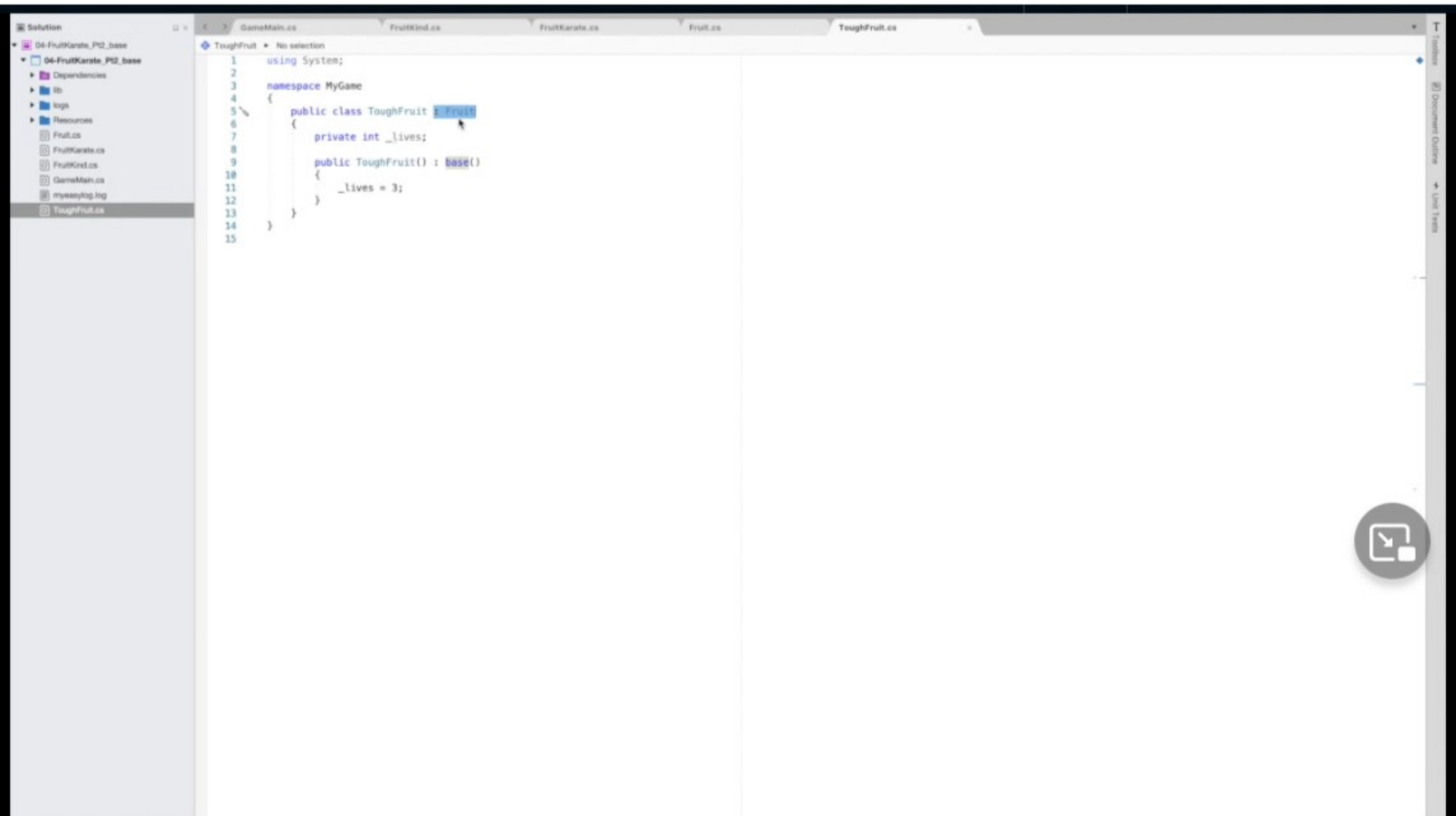




```
1 using System;
2 using System.Collections.Generic;
3 using SplashKitSDK;
4
5 namespace MyGame
6 {
7     public class FruitKarate
8     {
9         private List<Fruit> _fruit = new List<Fruit>();
10
11         public FruitKarate() { }
12
13         public void LaunchFruit()
14         {
15             Fruit f = new Fruit();
16             _fruit.Add(f);
17         }
18
19         public void Update()
20         {
21             foreach (Fruit f in _fruit)
22             {
23                 f.Update();
24             }
25         }
26
27         public void Draw()
28         {
29             foreach (Fruit f in _fruit)
30             {
31                 f.Draw();
32             }
33         }
34
35         public void PunchAt(Point2D pt)
36         {
37             List<Fruit> toRemove = new List<Fruit>();
38
39             foreach (Fruit f in _fruit)
40             {
41                 if (f.IsAt(pt))
42                 {
43                     f.Splat();
44                     toRemove.Add(f);
45                 }
46             }
47
48             foreach (Fruit f in toRemove)
49             {
50                 _fruit.Remove(f);
51             }
52         }
53     }
54 }
55
56
```





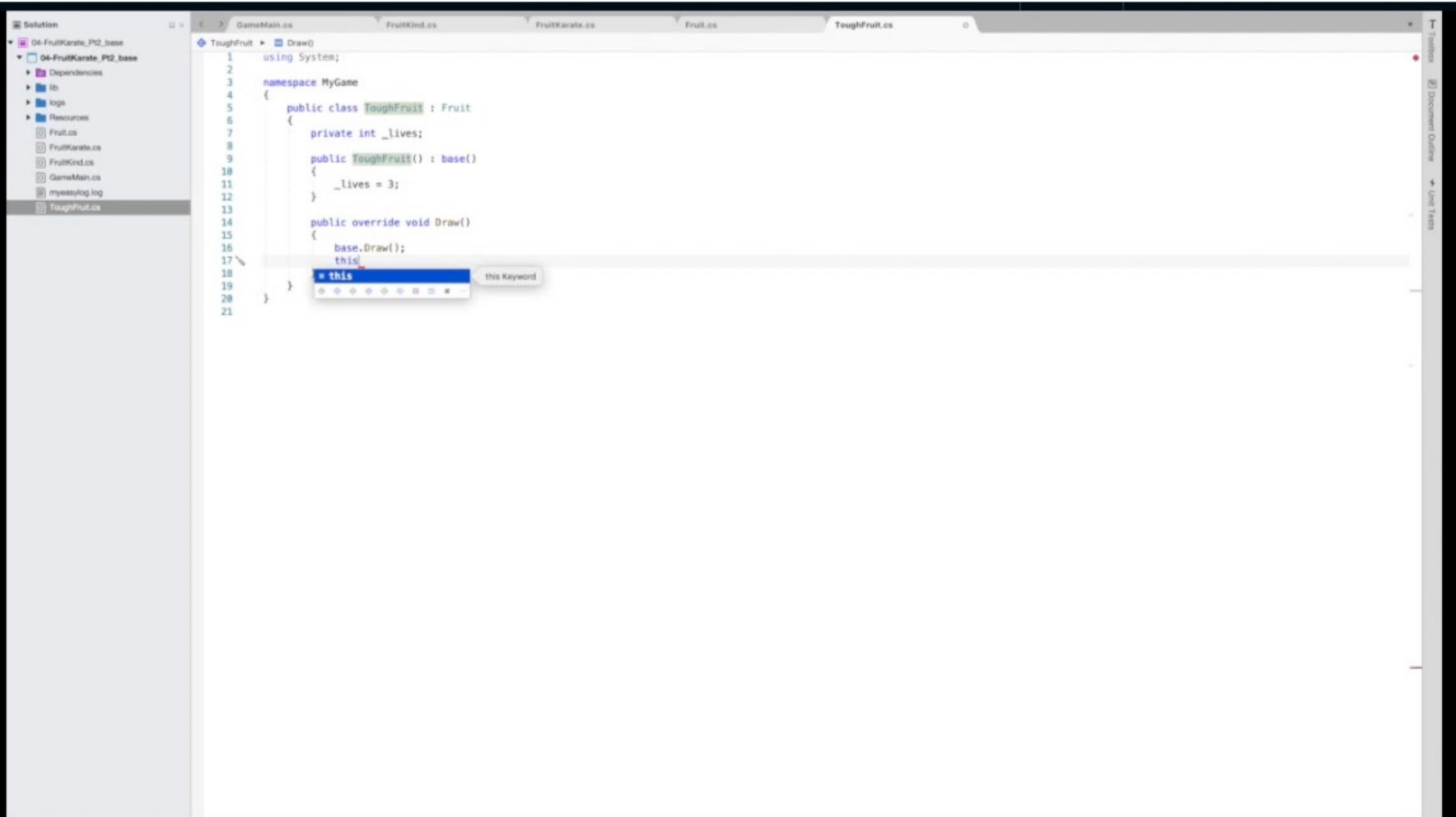


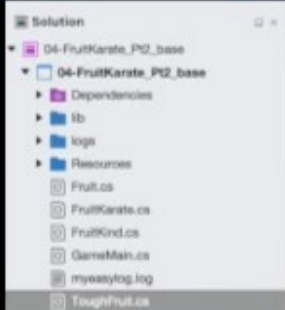
```

20         return SplashKit.LoadBitmap("Gooseberry", "Gooseberry.png");
21     case FruitKind.Blueberry:
22         return SplashKit.LoadBitmap("Blueberry", "Blueberry.png");
23     case FruitKind.Pomegranate:
24         return SplashKit.LoadBitmap("Pomegranate", "Pomegranate.png");
25     case FruitKind.Apricot:
26         return SplashKit.LoadBitmap("Apricot", "Apricot.png");
27     case FruitKind.Raspberry:
28         return SplashKit.LoadBitmap("Raspberry", "Raspberry.png");
29     case FruitKind.Blackberry:
30         return SplashKit.LoadBitmap("Blackberry", "Blackberry.png");
31     case FruitKind.Strawberry:
32         return SplashKit.LoadBitmap("Strawberry", "Strawberry.png");
33     case FruitKind.Currant:
34         return SplashKit.LoadBitmap("Currant", "Currant.png");
35     default:
36         return SplashKit.LoadBitmap("Currant", "Currant.png");
37     }
38 }
39
40 public Fruit()
41 {
42     _position.X = 0;
43     _position.Y = SplashKit.ScreenHeight();
44
45     _velocity.X = 3.0;
46     _velocity.Y = -7.0 + SplashKit.Rnd(2) - 1;
47
48     _kind = (FruitKind) SplashKit.Rnd(9);
49 }
50
51 public void Update()
52 {
53     // update my position
54     _position.X += _velocity.X;
55     _position.Y += _velocity.Y;
56     // decay the velocity
57     _velocity = SplashKit.VectorAdd(_velocity, SplashKit.VectorTo(0, 0.05));
58 }
59
60 public virtual void Draw()
61 {
62     SplashKit.DrawBitmap(MyBitmap(), _position.X, _position.Y);
63 }
64
65 public bool IsAt(Point2D pt)
66 {
67     return SplashKit.BitmapPointCollision(MyBitmap(), _position, pt);
68 }
69
70 public void Splat()
71 {
72     SplashKit.PlaySoundEffect("Splat");
73 }
74
75 }
76

```



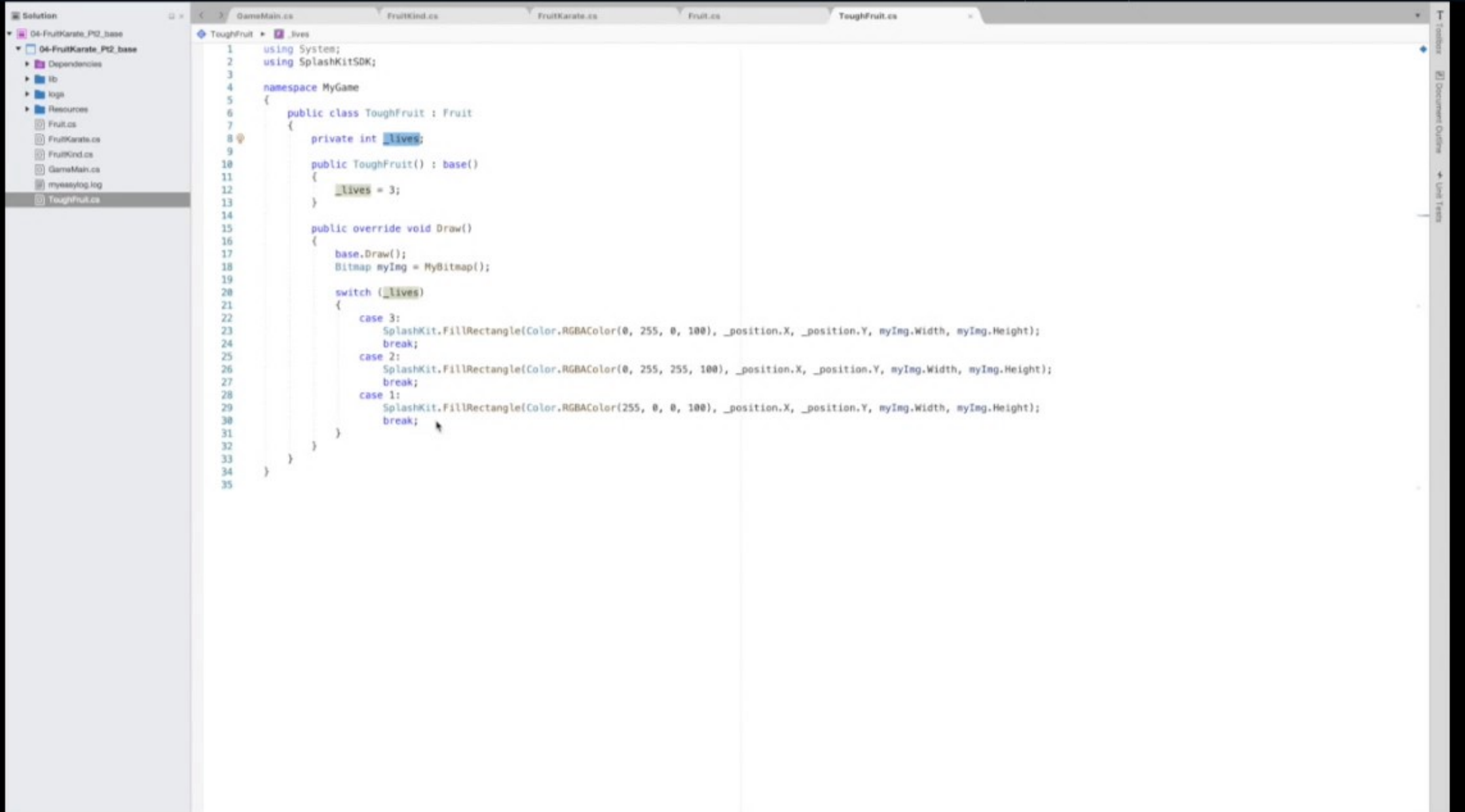




```
GameMain.cs | FruitKind.cs | FruitKarate.cs | Fruit.cs | ToughFruit.cs
ToughFruit | Draw()
1  using System;
2  using SplashKitSDK;
3
4  namespace MyGame
5  {
6      public class ToughFruit : Fruit
7      {
8          private int _lives;
9
10         public ToughFruit() : base()
11         {
12             _lives = 3;
13         }
14
15         public override void Draw()
16         {
17             base.Draw();
18
19             switch (_lives)
20             {
21                 case 3:
22                     SplashKit.FillRectangle(colour, positionx, positiony, width, height);
23                     break;
24                 case 2:
25                     break;
26                 case 1:
27                     break;
28             }
29         }
30     }
31 }
32
```



```
GameMain.cs    FruitKind.cs    FruitKarate.cs    Fruit.cs    ToughFruit.cs
Fruit.cs
    _velocity
1  using System;
2  using SplashKitSDK;
3
4  namespace MyGame
5  {
6      public class Fruit
7      {
8          private FruitKind _kind;
9
10         protected Point2D _position;
11         private Vector2D _velocity;
12
13         protected Bitmap MyBitmap()
14         {
15             switch (_kind)
16             {
17                 case FruitKind.Cherry:
18                     return SplashKit.LoadBitmap("Cherry", "Cherry.png");
19                 case FruitKind.Gooseberry:
20                     return SplashKit.LoadBitmap("Gooseberry", "Gooseberry.png");
21                 case FruitKind.Blueberry:
22                     return SplashKit.LoadBitmap("Blueberry", "Blueberry.png");
23                 case FruitKind.Pomegranate:
24                     return SplashKit.LoadBitmap("Pomegranate", "Pomegranate.png");
25                 case FruitKind.Apricot:
26                     return SplashKit.LoadBitmap("Apricot", "Apricot.png");
27                 case FruitKind.Raspberry:
28                     return SplashKit.LoadBitmap("Raspberry", "Raspberry.png");
29                 case FruitKind.Blackberry:
30                     return SplashKit.LoadBitmap("Blackberry", "Blackberry.png");
31                 case FruitKind.Strawberry:
32                     return SplashKit.LoadBitmap("Strawberry", "Strawberry.png");
33                 case FruitKind.Currant:
34                     return SplashKit.LoadBitmap("Currant", "Currant.png");
35                 default:
36                     return SplashKit.LoadBitmap("Currant", "Currant.png");
37             }
38         }
39
40         public Fruit()
41         {
42             _position.X = 0;
43             _position.Y = SplashKit.ScreenHeight();
44
45             _velocity.X = 3.0;
46             _velocity.Y = -7.0 + SplashKit.Rnd(2) - 1;
47
48             _kind = (FruitKind) SplashKit.Rnd(9);
49         }
50
51         public void Update()
52         {
53             // update my position
54             _position.X += _velocity.X;
55             _position.Y += _velocity.Y;
56             // decay the velocity
57             _velocity = SplashKit.VectorAdd(_velocity, SplashKit.VectorTo(0, 0.05));
58         }
59
60         public virtual void Draw()
```

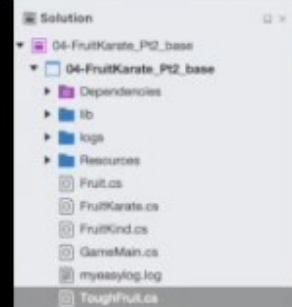


Solution

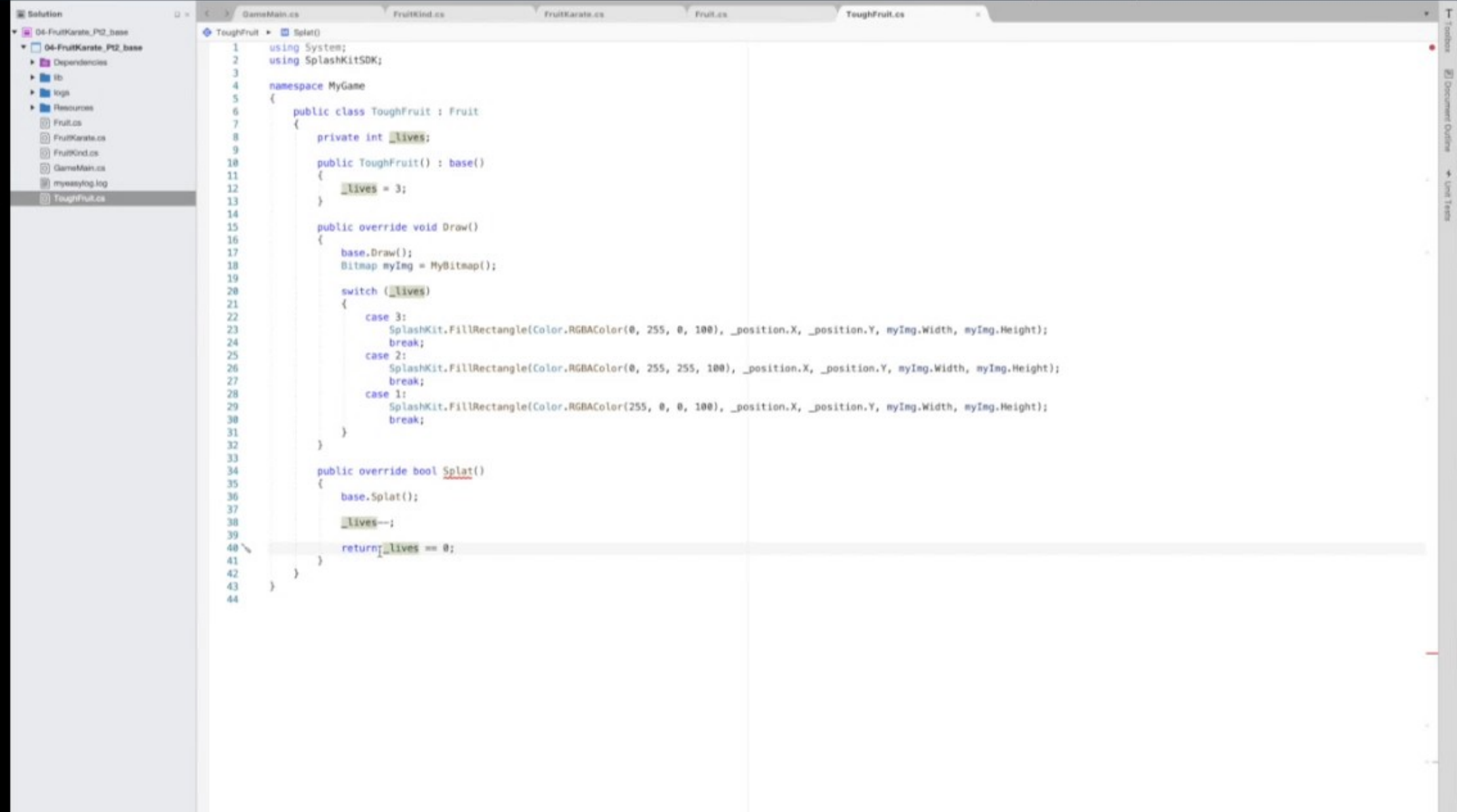
- 04-FruitKarate\_P02\_base
  - 04-FruitKarate\_P02\_base
    - Dependencies
    - lib
    - logs
    - Resources
    - Fruit.cs
    - FruitKarate.cs
    - FruitKind.cs
    - GameMain.cs
    - myeasylog.log
    - ToughFruit.cs

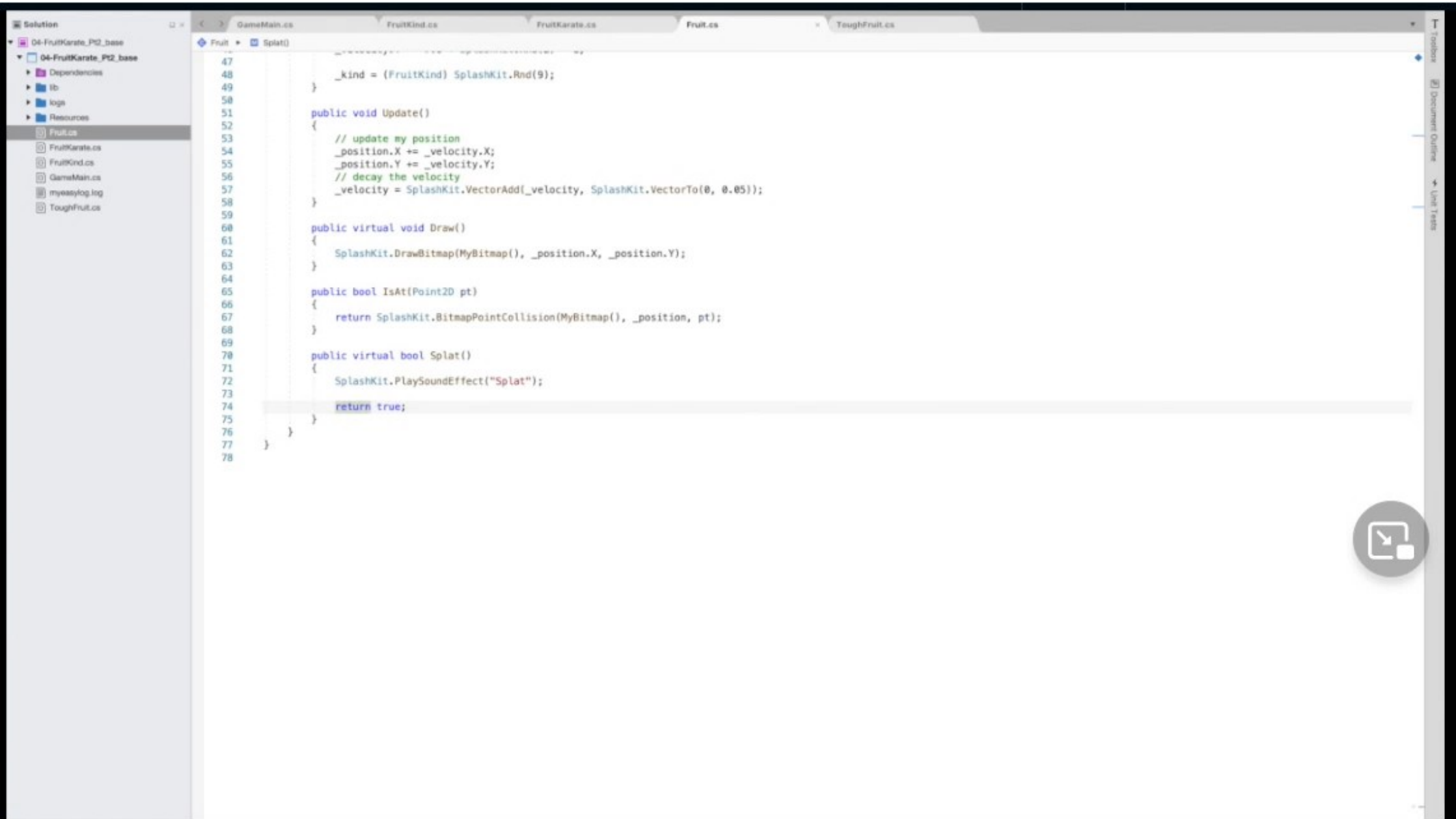
```
GameMain.cs  FruitKind.cs  FruitKarate.cs  Fruit.cs  ToughFruit.cs
Fruit  Splat()
47
48     _kind = (FruitKind) SplashKit.Rnd(9);
49 }
50
51 public void Update()
52 {
53     // update my position
54     _position.X += _velocity.X;
55     _position.Y += _velocity.Y;
56     // decay the velocity
57     _velocity = SplashKit.VectorAdd(_velocity, SplashKit.VectorTo(0, 0.05));
58 }
59
60 public virtual void Draw()
61 {
62     SplashKit.DrawBitmap(MyBitmap(), _position.X, _position.Y);
63 }
64
65 public bool IsAt(Point2D pt)
66 {
67     return SplashKit.BitmapPointCollision(MyBitmap(), _position, pt);
68 }
69
70 public virtual void Splat()
71 {
72     SplashKit.PlaySoundEffect("Splat");
73 }
74
75 }
76
```





```
GameMain.cs    FruitKind.cs    FruitKarate.cs    Fruit.cs    ToughFruit.cs
ToughFruit    Splat()
1  using System;
2  using SplashKitSDK;
3
4  namespace MyGame
5  {
6      public class ToughFruit : Fruit
7      {
8          private int _lives;
9
10         public ToughFruit() : base()
11         {
12             _lives = 3;
13         }
14
15         public override void Draw()
16         {
17             base.Draw();
18             Bitmap myImg = MyBitmap();
19
20             switch (_lives)
21             {
22                 case 3:
23                     SplashKit.FillRectangle(Color.RGBAColor(0, 255, 0, 100), _position.X, _position.Y, myImg.Width, myImg.Height);
24                     break;
25                 case 2:
26                     SplashKit.FillRectangle(Color.RGBAColor(0, 255, 255, 100), _position.X, _position.Y, myImg.Width, myImg.Height);
27                     break;
28                 case 1:
29                     SplashKit.FillRectangle(Color.RGBAColor(255, 0, 0, 100), _position.X, _position.Y, myImg.Width, myImg.Height);
30                     break;
31             }
32         }
33
34         public override void Splat()
35         {
36             base.Splat();
37
38             _lives--;
39         }
40     }
41 }
42
```







Solution

- 04-FruitKarate\_Pt2\_base
  - 04-FruitKarate\_Pt2\_base
    - Dependencies
    - lib
    - logs
    - Resources
    - Fruit.cs
    - FruitKarate.cs
    - FruitKind.cs
    - GameMain.cs
    - myeasylog.log
    - ToughFruit.cs

GameMain.cs FruitKind.cs FruitKarate.cs Fruit.cs ToughFruit.cs

FruitKarate.cs

```
1 using System;
2 using System.Collections.Generic;
3 using SplashKitSDK;
4
5 namespace MyGame
6 {
7     public class FruitKarate
8     {
9         private List<Fruit> _fruit = new List<Fruit>();
10
11         public FruitKarate() { }
12
13         public void LaunchFruit()
14         {
15             Fruit f = new Fruit();
16             _fruit.Add(f);
17         }
18
19         public void Update()
20         {
21             foreach (Fruit f in _fruit)
22             {
23                 f.Update();
24             }
25         }
26
27         public void Draw()
28         {
29             foreach (Fruit f in _fruit)
30             {
31                 f.Draw();
32             }
33         }
34
35         public void PunchAt(Point2D pt)
36         {
37             List<Fruit> toRemove = new List<Fruit>();
38
39             foreach (Fruit f in _fruit)
40             {
41                 if (f.IsAt(pt))
42                 {
43                     if (f.Splat())
44                     {
45                         toRemove.Add(f);
46                     }
47                 }
48             }
49
50             foreach (Fruit f in toRemove)
51             {
52                 _fruit.Remove(f);
53             }
54         }
55     }
56 }
57
58
```



```
ashKit.Rnd();
```

