2019/07/19 実験実習

目的

この演習においてはオブジェクト指向プログラミングの理解を深める。

装置/ツール

- Visual Studio
- Windows 10 Proト

実験

問題 11.1

Fighter クラスの Initialize メソッドを参考にして図 11.10 中に問題 11.1 追加箇所とコメントされた箇所に三角形のサイズである定数 MAX_X および MAX_Y と、適切に命令を追加して、敵オブジェクトを逆三角形として表示されるようにしなさい。また、上記ソースコードを実行してスクリーンショットを示しなさい。
実行結果を図 11.1 に示す。

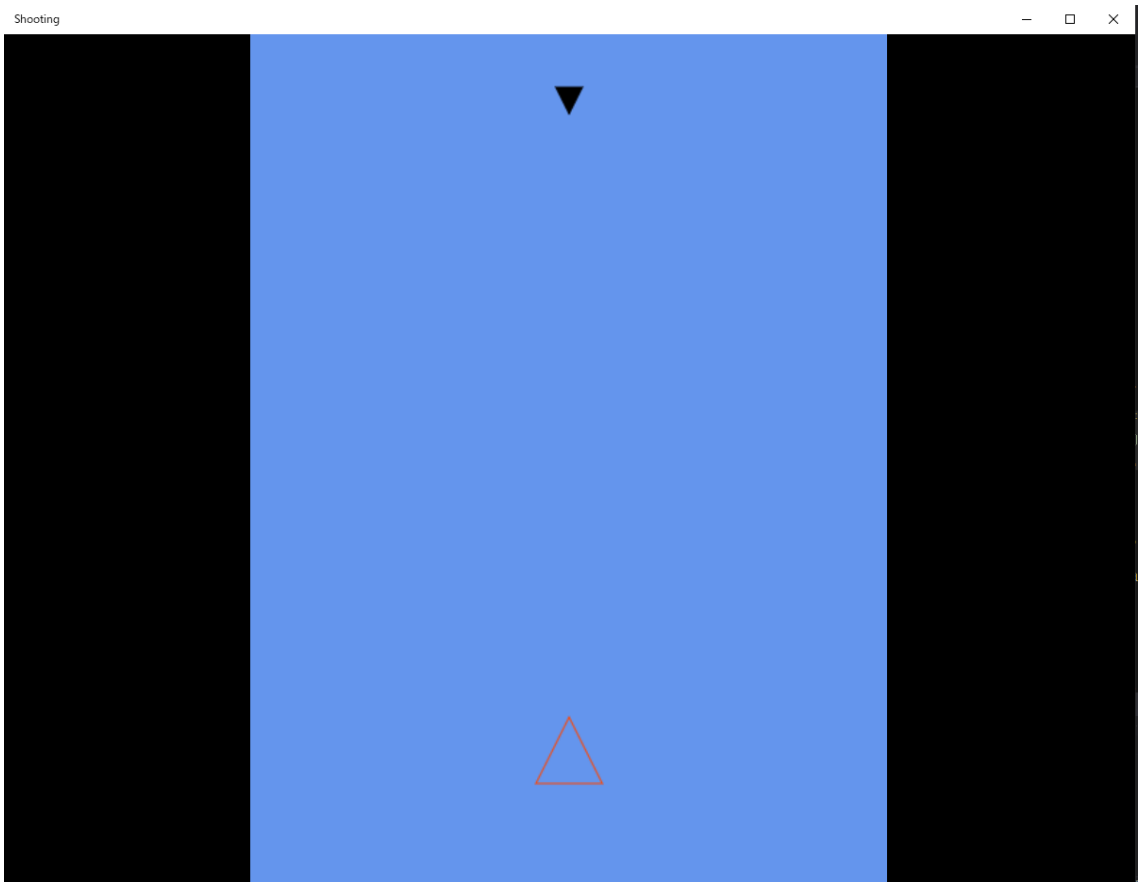

図 11.1 敵オブジェクトを表示した

問題 11.2

すべての変更を追加してプログラムのスクリーンショットを報告しなさい。
実行結果を 11.2 に示す。

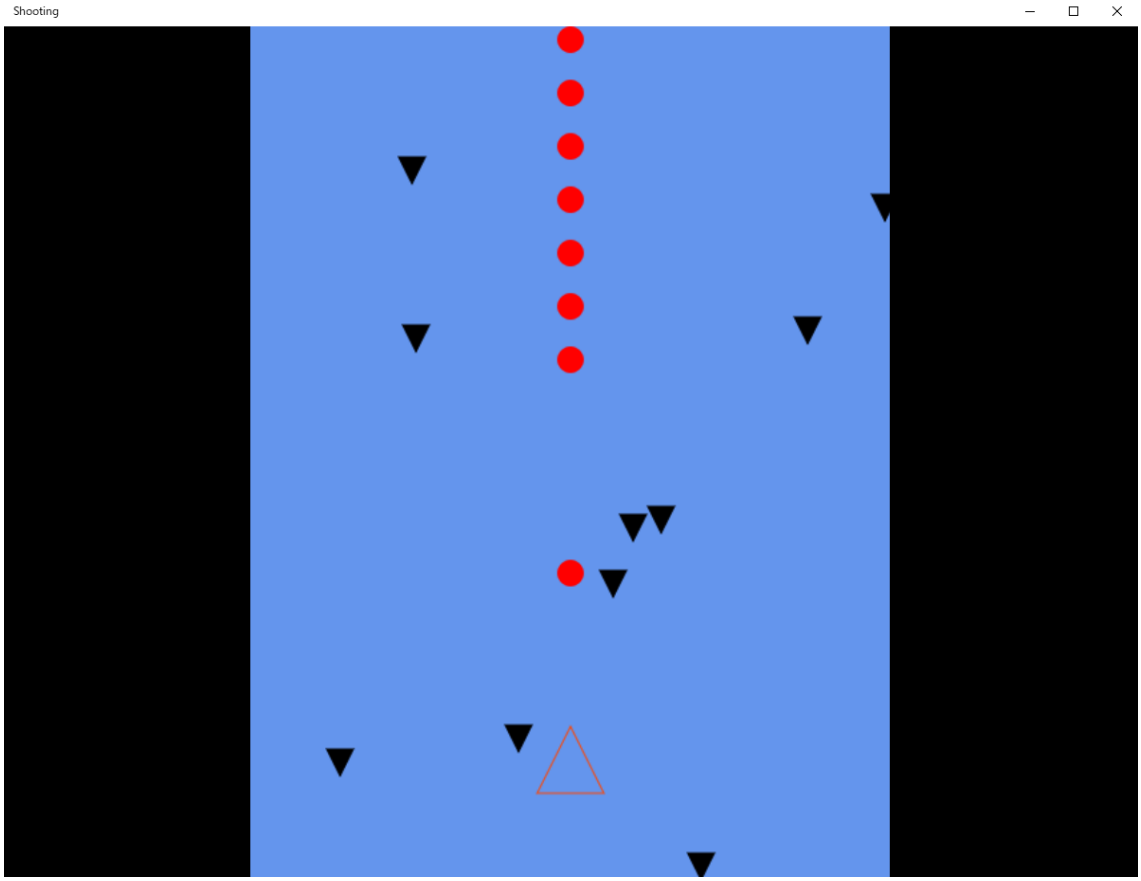

図 11.2 すべての変更を追加した

課題

レポート課題 11.1

クラス図を報告しなさい
クラス図を図 11.3 に示す。

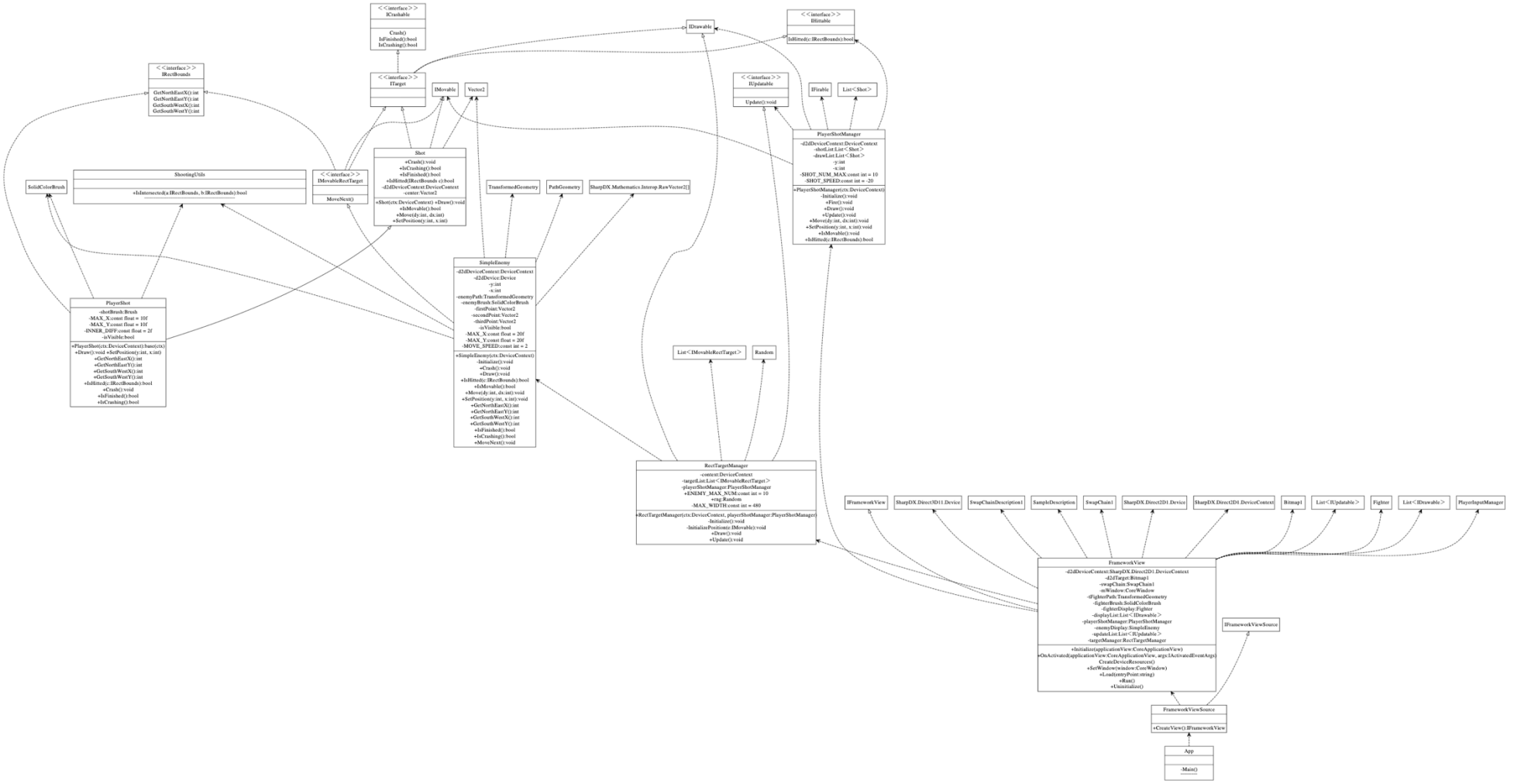
<<interface>>
ICrushable
Crash()
IsFinished():bool
IsCrushing():bool
<<interface>>
IDrawable
<<interface>>
IHittable
IsHitted(c:IRectBounds):bool
<<interface>>
IRectBounds
GetNorthEastX():int
GetNorthEastY():int
GetSouthWestX():int
GetSouthWestY():int
<<interface>>
ITarget
IMovable
Vector2
<<interface>>
IUpdatable
Update():void
IFirable
List<Shot>
Shot
+Crash():void
+IsCrushing():bool
+IsFinished():bool
+IsHitted(IRectBounds c):bool
-d2dDeviceContext:DeviceContext
-center:Vector2
+Shot(ctx:DeviceContext) +Draw():void
+IsMovable():bool
+Move(dy:int, dx:int)
+SetPosition(y:int, x:int)
PlayerShotManager
-d2dDeviceContext:DeviceContext
-shotList:List<Shot>
-drawList:List<Shot>
-y:int
-x:int
-SHOT_NUM_MAX:const int = 10
-SHOT_SPEED:const int = -20
+PlayerShotManager(ctx:DeviceContext)
-Initialize():void
+Fire():void
+Draw():void
+Update():void
+Move(dy:int, dx:int):void
+SetPosition(y:int, x:int):void
+IsMovable():void
+IsHitted(c:IRectBounds):bool
ShootingUtils
+IsIntersected(a:IRectBounds, b:IRectBounds):bool
SolidColorBrush
<<interface>>
IMovableRectTarget
MoveNext()
TransformedGeometry
PathGeometry
SharpDX.Mathematics.Interop.RawVector2[]
PlayerShot
-shotBrush:Brush
-MAX_X:const float = 10f
-MAX_Y:const float = 10f
-INNER_DIFF:const float = 2f
-isVisible:bool
+PlayerShot(ctx.DeviceContext):base(ctx)
+Draw():void +SetPosition(y:int, x:int)
+GetNorthEastX():int
+GetNorthEastY():int
+GetSouthWestX():int
+GetSouthWestY():int
+IsHitted(c:IRectBounds):bool
+Crash():void
+IsFinished():bool
+IsCrushing():bool
SimpleEnemy
-d2dDeviceContext:DeviceContext
-d2dDevice:Device
-y:int
-x:int
-enemyPath:TransformedGeometry
-enemyBrush:SolidColorBrush
-firstPoint:Vector2
-secondPoint:Vector2
-thirdPoint:Vector2
-isVisible:bool
-MAX_X:const float = 20f
-MAX_Y:const float = 20f
-MOVE_SPEED:const int = 2
+SimpleEnemy(ctx.DeviceContext)
-Initialize():void
+Crash():void
+Draw():void
+IsHitted(c:IRectBounds):bool
+IsMovable():bool
+Move(dy:int, dx:int):void
+SetPosition(y:int, x:int):void
+GetNorthEastX():int
+GetNorthEastY():int
+GetSouthWestX():int
+GetSouthWestY():int
+IsFinished():bool
+IsCrushing():bool
+MoveNext():void
List<IMovableRectTarget>
Random
RectTargetManager
-context:DeviceContext
-targetList:List<IMovableRectTarget>
-playerShotManager:PlayerShotManager
+ENEMY_MAX_NUM:const int = 10
+rng:Random
-MAX_WIDTH:const int = 480
+RectTargetManager(ctx:DeviceContext, playerShotManager:PlayerShotManager)
-Initialize():void
-InitializePosition(c:IMovable):void
+Draw():void
+Update():void
IFrameworkView
SharpDX.Direct3D11.Device
SwapChainDescription1
SampleDescription
SwapChain1
SharpDX.Direct2D1.Device
SharpDX.Direct2D1.DeviceContext
Bitmap1
List<IUpdatable>
Fighter
List<IDrawable>
PlayerInputManager
FrameworkView
-d2dDeviceContext:SharpDX.Direct2D1.DeviceContext
-d2dTarget:Bitmap1
-swapChain:SwapChain1
-nsWindow:CoreWindow
-fighterPath:TransformedGeometry
-fighterBrush:SolidColorBrush
-fighterDisplay:Fighter
-displayList:List<IDrawable>
-playerShotManager:PlayerShotManager
-enemyDisplay:SimpleEnemy
-updateList:List<IUpdatable>
-targetManager:RectTargetManager
+Initialize(applicationView:CoreApplicationView)
+OnActivated(applicationView:CoreApplicationView, args:IActivatedEventArgs)
CreateDeviceResources()
+SetWindow(window:CoreWindow)
+Load(entryPoint:string)
+Run()
+Uninitialize()
IFrameworkViewSource
FrameworkViewSource
+CreateView():IFrameworkView
App
-Main()

図 11.3 クラス図

非常にクラス図が大きく、見にくいので、以下にクラス図のコードを示す。
```graphviz
digraph obj{
	node[shape=record]
	rankdir="BT"

	IHittable[label="{
	＜＜interface＞＞¥n
	IHittable||
	IsHitted(c:IRectBounds):bool
	}"]

	ICrashable[label="{
	＜＜interface＞＞¥n
	ICrashable||
	Crash() ¥n
	IsFinished():bool ¥n
	IsCrashing():bool ¥n
	}"]

	ITarget[label="{
	＜＜interface＞＞¥n
	ITarget||
	}"]

	IRectBounds[label="{
	＜＜interface＞＞¥n
	IRectBounds||
	GetNorthEastX():int ¥n
	GetNorthEastY():int ¥n
	GetSouthWestX():int ¥n
	GetSouthWestY():int ¥n
	}"]

	IMovableRectTarget[label="{
	＜＜interface＞＞¥n
	IMovableRectTarget||
	MoveNext()
	}"]

	ShootingUtils[label="{
	ShootingUtils||
```

```
+IsIntersected(a:IRectBounds, b:IRectBounds):bool¥n
―――――――――――――――――――――――
}"]

SimpleEnemy[label="{
SimpleEnemy|
-d2dDeviceContext:DeviceContext ¥n
-d2dDevice:Device ¥n
-y:int ¥n
-x:int ¥n
-enemyPath:TransformedGeometry ¥n
-enemyBrush:SolidColorBrush ¥n
-firstPoint:Vector2 ¥n
-secondPoint:Vector2 ¥n
-thirdPoint:Vector2 ¥n
-isVisible:bool ¥n
-MAX_X:const float = 20f ¥n
-MAX_Y:const float = 20f ¥n
-MOVE_SPEED:const int = 2 ¥n|
+SimpleEnemy(ctx:DeviceContext) ¥n
-Initialize():void ¥n
+Crash():void ¥n
+Draw():void ¥n
+IsHitted(c:IRectBounds):bool ¥n
+IsMovable():bool ¥n
+Move(dy:int, dx:int):void ¥n
+SetPosition(y:int, x:int):void ¥n
+GetNorthEastX():int ¥n
+GetNorthEastY():int ¥n
+GetSouthWestX():int ¥n
+GetSouthWestY():int ¥n
+IsFinished():bool ¥n
+IsCrashing():bool ¥n
+MoveNext():void
}"]

IUpdatable[label="{
＜＜interface＞＞¥n
IUpdatable||
Update():void
}"]
```

```
RectTargetManager[label="{
RectTargetManager|
-context:DeviceContext ¥n
-targetList:List＜IMovableRectTarget＞ ¥n
-playerShotManager:PlayerShotManager ¥n
+ENEMY_MAX_NUM:const int = 10 ¥n
+rng:Random ¥n
-MAX_WIDTH:const int = 480¥n |
+RectTargetManager(ctx:DeviceContext, playerShotManager:PlayerShotManager) ¥n
-Initialize():void ¥n
-InitializePosition(e:IMovable):void ¥n
+Draw():void¥n
+Update():void¥n
}"]

PlayerShotManager[label="{
PlayerShotManager|
-d2dDeviceContext:DeviceContext ¥n
-shotList:List＜Shot＞ ¥n
-drawList:List＜Shot＞ ¥n
-y:int ¥n
-x:int ¥n
-SHOT_NUM_MAX:const int = 10 ¥n
-SHOT_SPEED:const int = -20 ¥n|
+PlayerShotManager(ctx:DeviceContext)¥n
-Initialize():void¥n
+Fire():void¥n
+Draw():void¥n
+Update():void¥n
+Move(dy:int, dx:int):void¥n
+SetPosition(y:int, x:int):void¥n
+IsMovable():void¥n
+IsHitted(c:IRectBounds):bool
}"]

PlayerShot[label="{
PlayerShot|
-shotBrush:Brush ¥n
-MAX_X:const float = 10f ¥n
-MAX_Y:const float = 10f ¥n
-INNER_DIFF:const float = 2f ¥n
-isVisible:bool ¥n|
```

```
+PlayerShot(ctx:DeviceContext):base(ctx) ¥n
+Draw():void
+SetPosition(y:int, x:int) ¥n
+GetNorthEastX():int ¥n
+GetNorthEastY():int ¥n
+GetSouthWestX():int ¥n
+GetSouthWestY():int ¥n
+IsHitted(c:IRectBounds):bool ¥n
+Crash():void ¥n
+IsFinished():bool ¥n
+IsCrashing():bool ¥n
}"]

Shot[label="{
Shot|
+Crash():void ¥n
+IsCrashing():bool ¥n
+IsFinished():bool ¥n
+IsHitted(IRectBounds c):bool ¥n
-d2dDeviceContext:DeviceContext ¥n
-center:Vector2 ¥n|
+Shot(ctx:DeviceContext)
+Draw():void ¥n
+IsMovable():bool ¥n
+Move(dy:int, dx:int) ¥n
+SetPosition(y:int, x:int)¥n
}"]

App [label="{
App||
-Main()¥n
———
}"]

FrameworkViewSource[label="{
FrameworkViewSource||
+CreateView():IFrameworkView
}"]

FrameworkView[label="{
FrameworkView|
-d2dDeviceContext:SharpDX.Direct2D1.DeviceContext ¥n
```

```
-d2dTarget:Bitmap1 ¥n
-swapChain:SwapChain1 ¥n
-mWindow:CoreWindow ¥n
-tFighterPath:TransformedGeometry ¥n
-fighterBrush:SolidColorBrush ¥n
-fighterDisplay:Fighter ¥n
-displayList:List＜IDrawable＞ ¥n
-playerShotManager:PlayerShotManager ¥n

-enemyDisplay:SimpleEnemy ¥n
-updateList:List＜IUpdatable＞ ¥n
-targetManager:RectTargetManager ¥n|
+Initialize(applicationView:CoreApplicationView)¥n
+OnActivated(applicationView:CoreApplicationView, args:IActivatedEventArgs)¥n
CreateDeviceResources()¥n
+SetWindow(window:CoreWindow)¥n
+Load(entryPoint:string)¥n
+Run()¥n
+Uninitialize()
}"]

// クラス継承
    edge [arrowhead = "empty"]
PlayerShot -> Shot

// インターフェース
    edge [arrowhead = "empty" style="dashed"]
ITarget->IHittable, ICrashable, IDrawable
IMovableRectTarget->ITarget, IRectBounds, IMovable
SimpleEnemy->IMovableRectTarget
RectTargetManager->IUpdatable, IDrawable
PlayerShot -> IRectBounds
Shot -> ITarget, IMovable
FrameworkViewSource -> IFrameworkViewSource
FrameworkView -> IFrameworkView

// 依存
    edge [arrowhead = "vee" style="dashed"]
SimpleEnemy-
>PathGeometry,Vector2,"SharpDX.Mathematics.Interop.RawVector2[]",TransformedGeometry,So
lidColorBrush,ShootingUtils
RectTargetManager->List＜IMovableRectTarget＞,Random,SimpleEnemy
```

```
    PlayerShotManager->IDrawable, IMovable, IFirable, IUpdatable, IHittable
    PlayerShotManager->List＜Shot＞
    PlayerShot -> SolidColorBrush,ShootingUtils
    Shot -> Vector2
    App -> FrameworkViewSource
    FrameworkViewSource -> FrameworkView
    FrameworkView ->
"SharpDX.Direct3D11.Device",SwapChainDescription1,SampleDescription,SwapChain1,"SharpD
X.Direct2D1.Device","SharpDX.Direct2D1.DeviceContext","Bitmap1",List＜IUpdatable
＞,PlayerShotManager,Fighter,List＜IDrawable＞,RectTargetManager,PlayerInputManager
}
```

図 11.4 graphviz のコード

レポート課題 11.2

新しく機能を追加するなら何を追加するか

2 種類目の弾丸を実装する。例えば、緑で大きくて遅い弾丸があれば、プレイの幅が広がって楽しくなると思う。