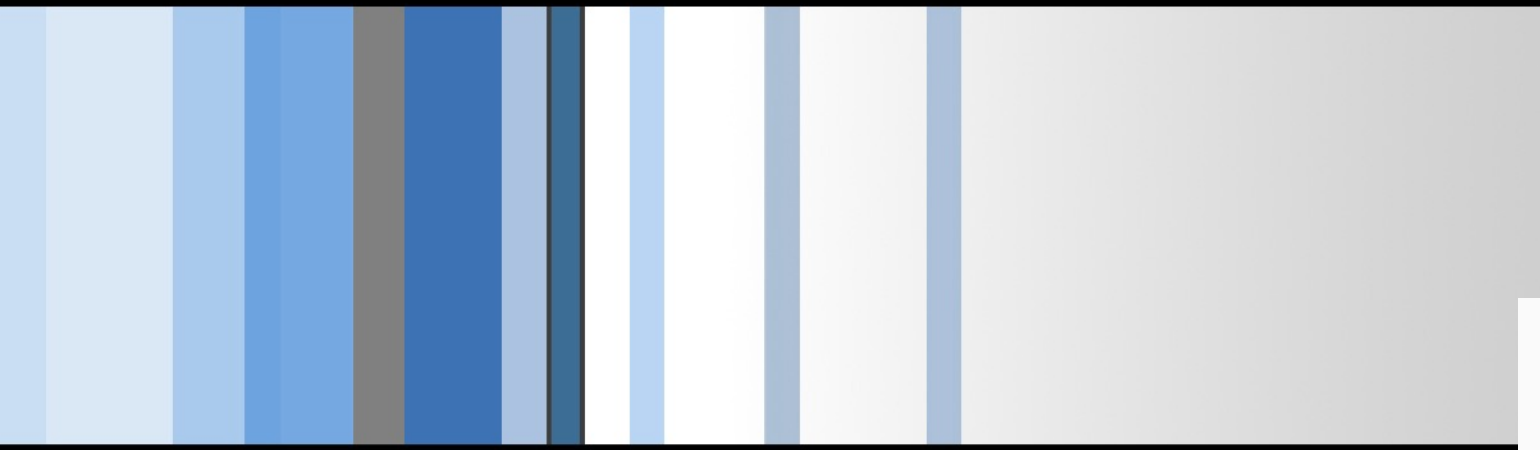
Class 24

CSC 496: iOS App Development
SpriteKit (6): Entities and Components

Si Chen (schen@wcupa.edu)



SpriteKit

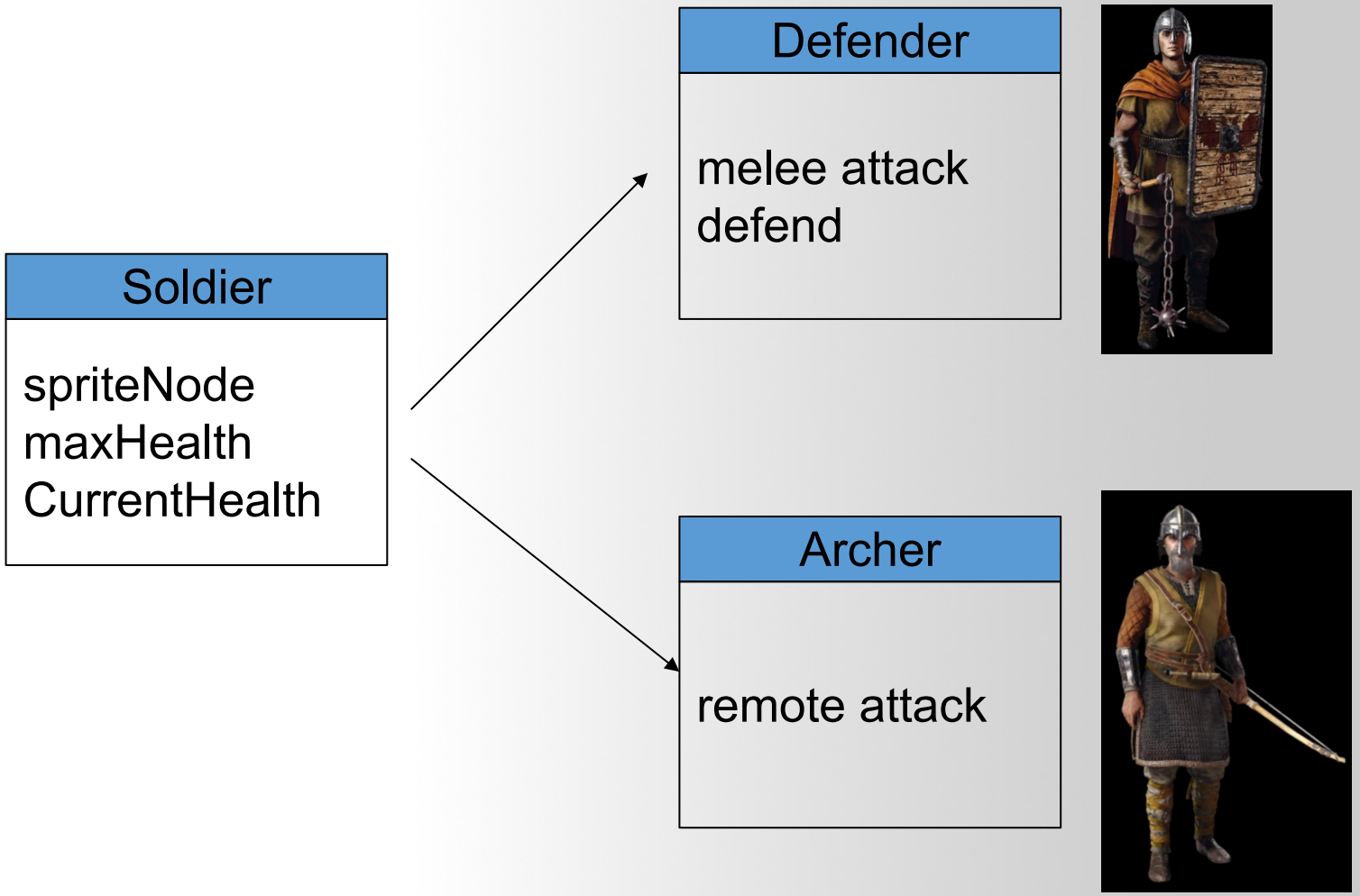
The Entity-Component Design Pattern

- The Entity-Component design pattern, also known as Entity Component System (ECS) is a common architecture in game design.
- This type of architecture pattern uses a ***composition-based*** design pattern instead of an ***inheritance-based*** design pattern.
- What's the difference between the two?

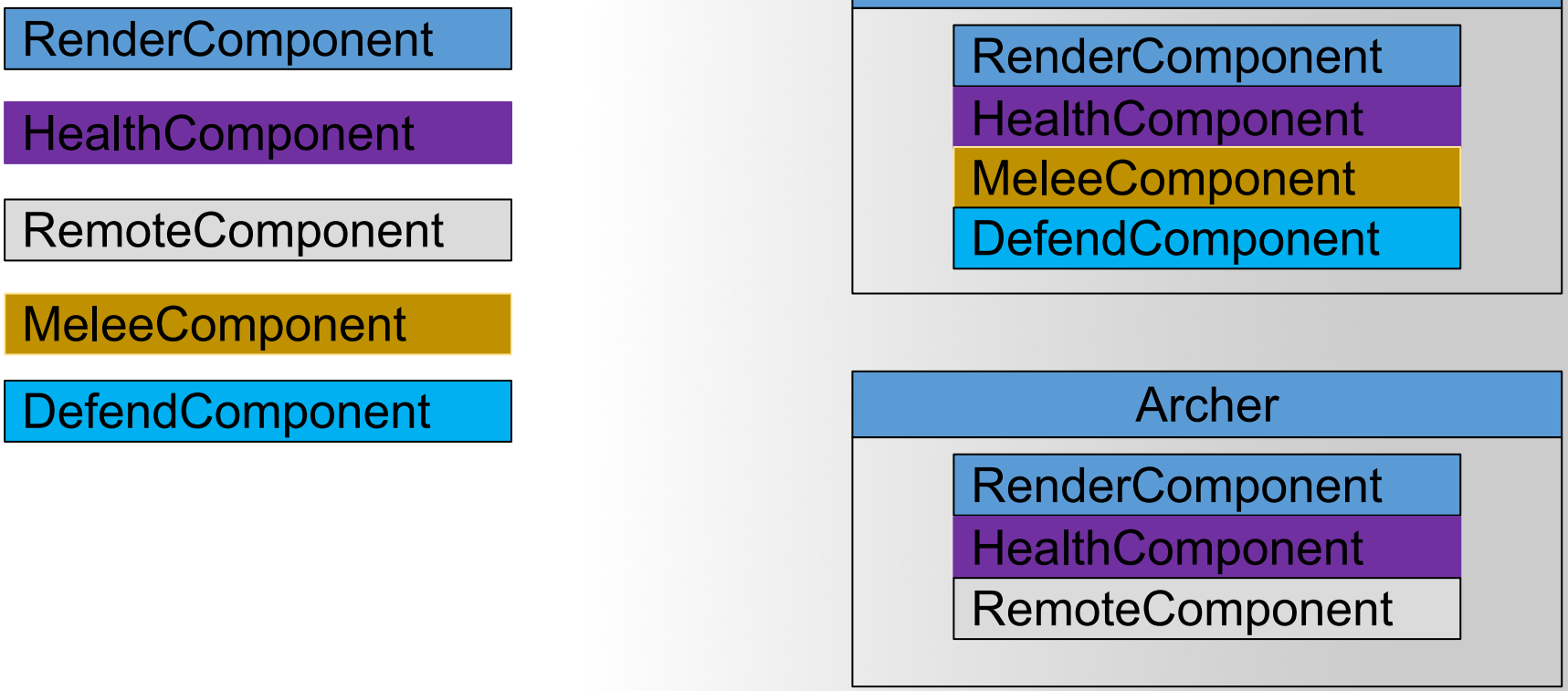
Defender

Archer

Inheritance-based Design Pattern

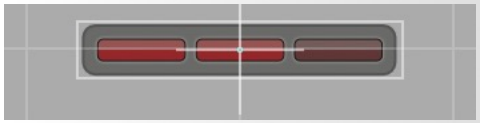


Composition-based Design Pattern



Our First Component – HP (health point) Component

- First, download Pokemon2DII_Template.zip from D2L.
- You can notice that I added the image assets of the HP bar to the project



HP Component

- Creating the HP component:
- First, create a new file, name the new file "**HPComponent.swift**"

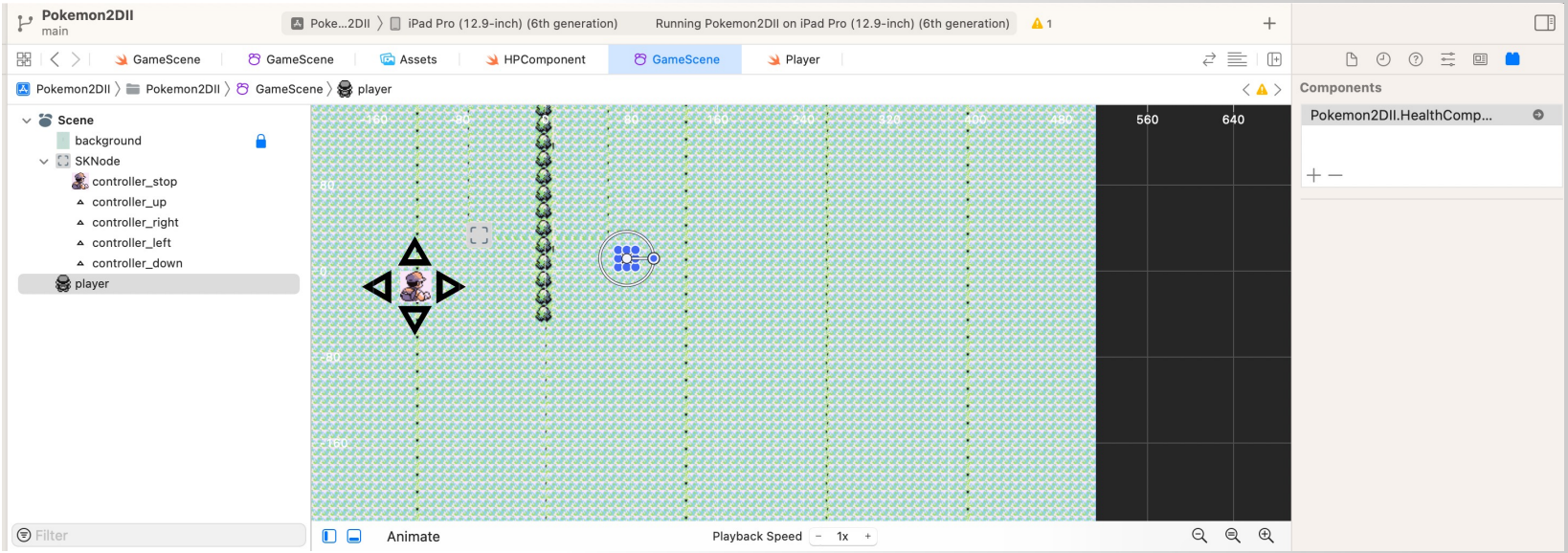
```
import SpriteKit
import GameplayKit

class HPComponent: GKComponent{

    override func didAddToEntity() {

    }

    override func willRemoveFromEntity() {

    }

    override func update(deltaTime seconds: TimeInterval) {
        // update entities
    }

    override class var supportsSecureCoding: Bool{
        true
    }
}
```

HP Component

```
import SpriteKit
import GameplayKit

class HPComponent: GKComponent{

    override func didAddToEntity() {
        guard let node = entity?.component(ofType: GKSKNodeComponent.self)?.node
        else{
            return
        }
        if let HPMeter = SKReferenceNode(fileNamed: "HealthMeter"){
            HPMeter.position = CGPoint(x:0, y:50)
            node.addChild(HPMeter)
        }
    }

    override func willRemoveFromEntity() {

    }

    override func update(deltaTime seconds: TimeInterval) {
        // update entities
    }

    override class var supportsSecureCoding: Bool{
        true
    }
}
```

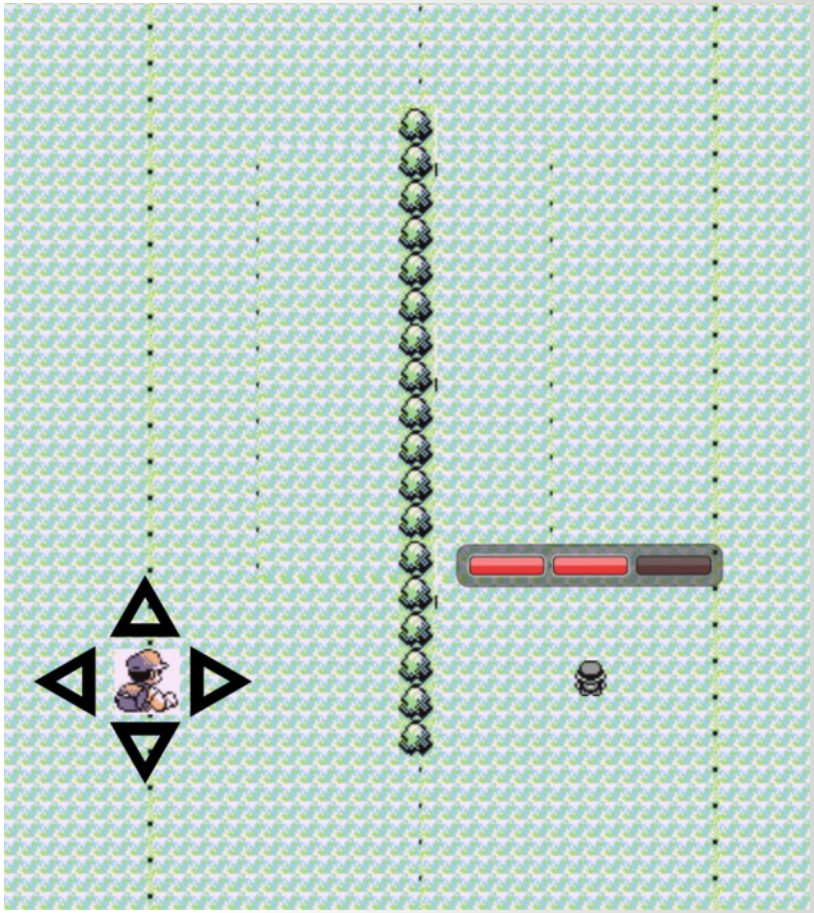
HP Component

- In GameScene.sks, select the player node. Click the button to show the inspectors and then switch to the Component Inspector.



- Click the + button to reveal the list of available components. Add the HealthComponent

Test it



Adding Options

```
import SpriteKit
import GameplayKit

class HPComponent: GKComponent{
    @GKInspectable var currentHealth: Int = 3
    @GKInspectable var maxHealth: Int = 3

    override func didAddToEntity() {
        guard let node = entity?.component(ofType: GKSKNodeComponent.self)?.node
        else{
            return
        }
        if let HPMeter = SKReferenceNode(fileNamed: "HealthMeter"){
            HPMeter.position = CGPoint(x:0, y:50)
            node.addChild(HPMeter)
        }
    }

    override func willRemoveFromEntity() {

    }

    override func update(deltaTime seconds: TimeInterval) {
        // update entities
    }

    override class var supportsSecureCoding: Bool{
        true
    }
}
```

Adding textures

```
private let healthFull = SKTexture(imageNamed: "health_full")
private let healthEmpty = SKTexture(imageNamed: "health_empty")
```

Q & A