GAME 3004

SpriteKit

Lesson 4

Expectation

Introduction to **SpriteKit**

Outcome

Understanding SKScene's **Rendering Loop** and SKScene's **Node Tree**

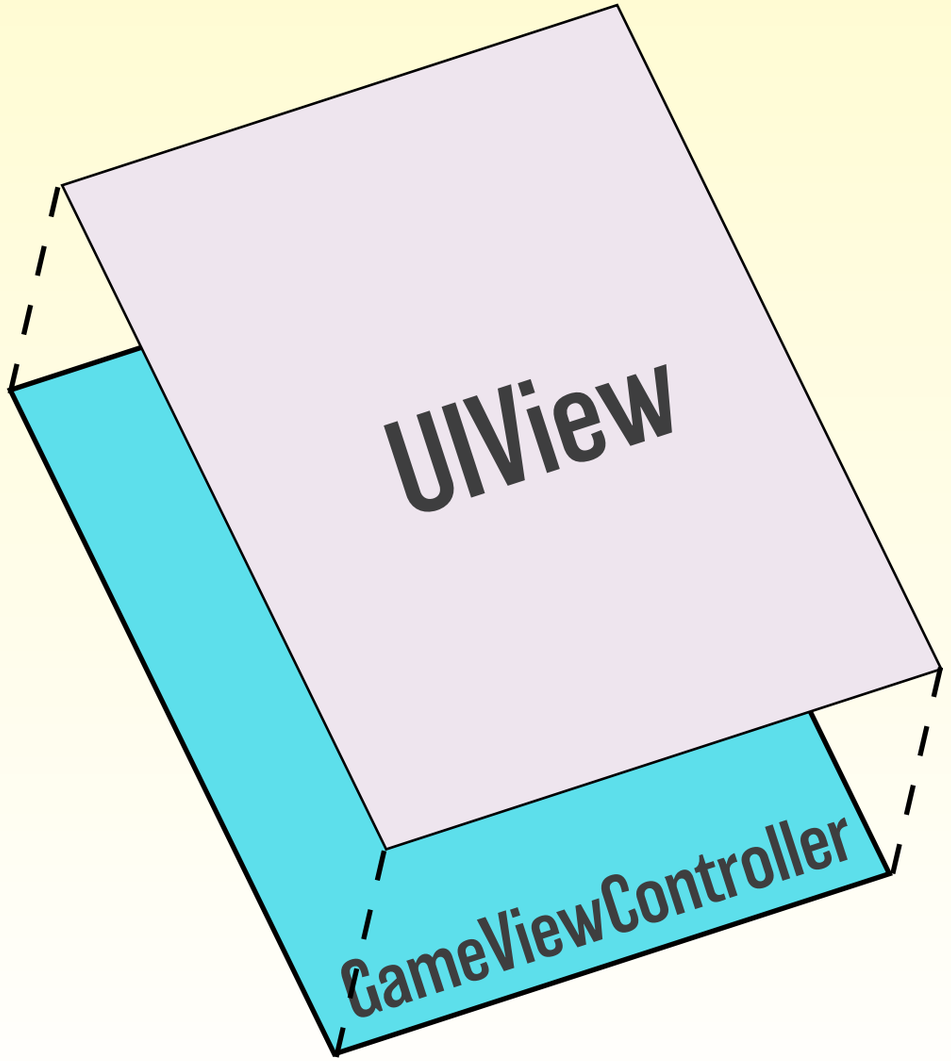
Key Concepts

SKScene Review
SKScene Rendering Loop
SKScene Node Tree
Rendering Nodes
Searching Nodes
Coordinates
Anchor Points

SKScene Review

1. Create the **GameViewController**

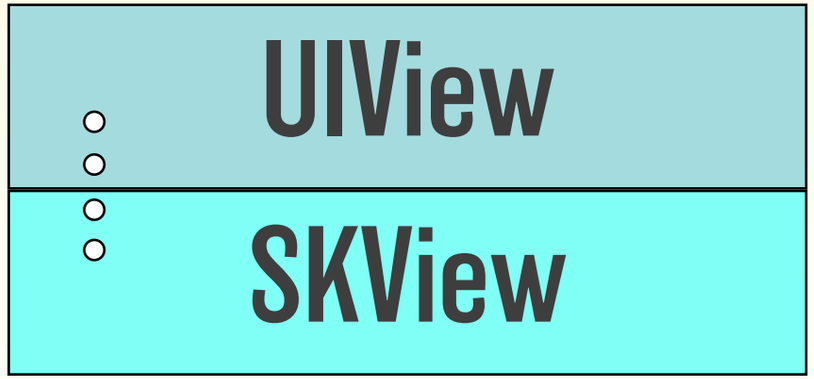
2. Have the **GameViewController** create its **UIView**



SKScene Review

3. Inside the **GameViewController.viewDidLoad()**, down-cast the **UIView** to an **SKView** and set the **showFPS** property to **true**:

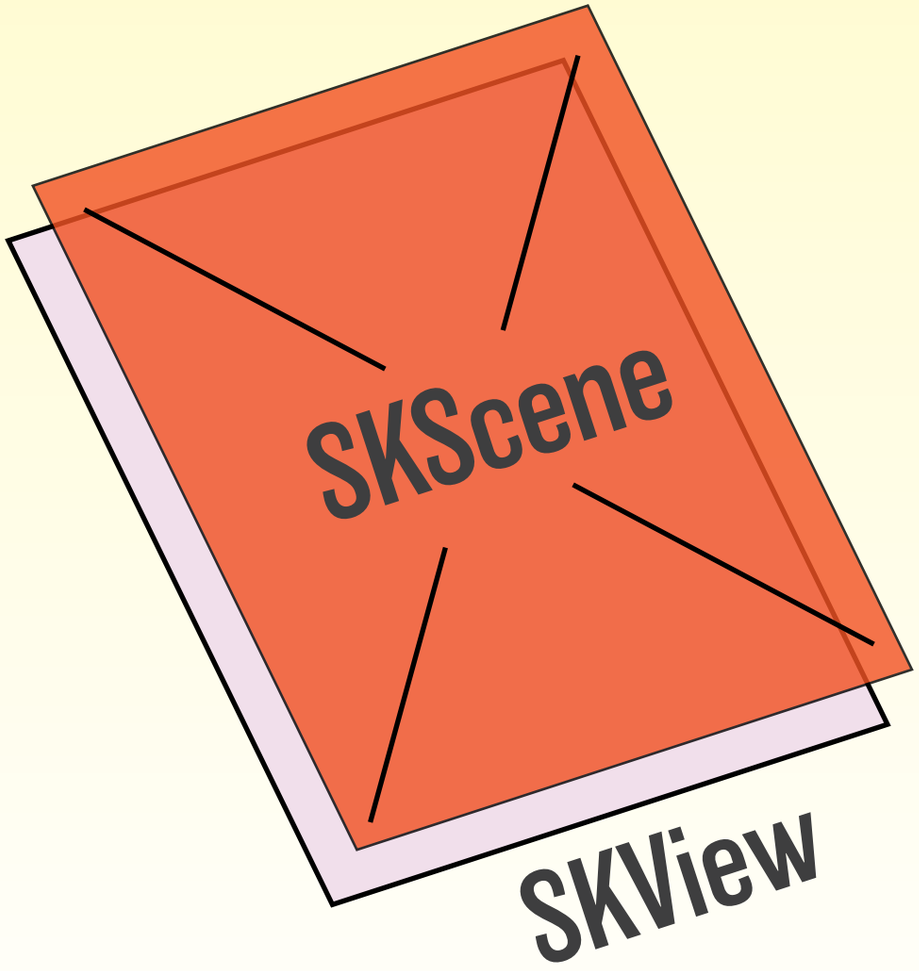
```
let skView = view as! SKView
skView.showFPS = true
```



SKScene Review

4. Create an instance of the SKScene named **scene**, passing it its size in the constructor and setting the scaleMode property:

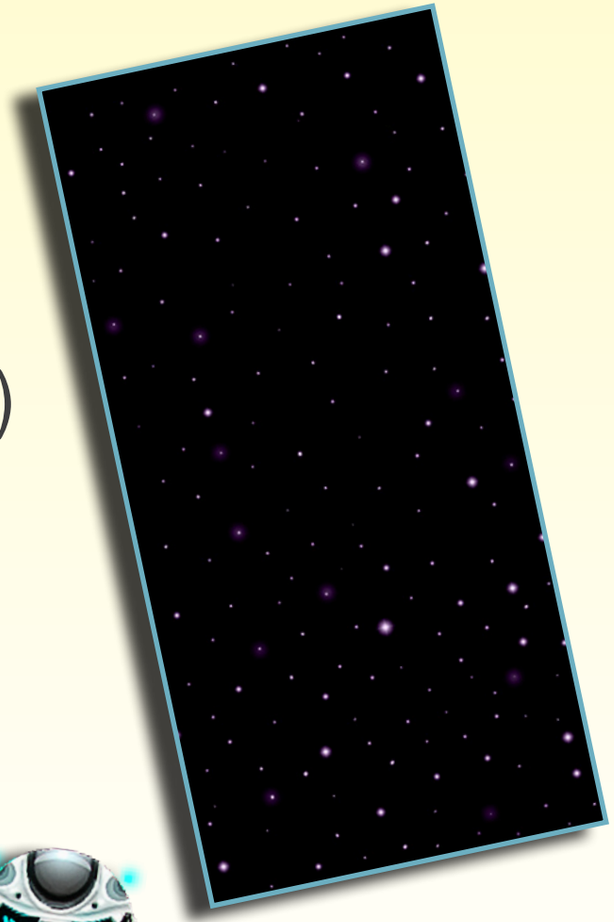
```
scene = GameScene(size: skView.bounds.size)
scene.scaleMode = .aspectFill
```



SKScene Review

5. Inside the **init**() of the scene, we added the **backgroundNode and playerNode** objects in the scene:

```
let backgroundNode = SKSpriteNode(imageNamed: “Background“)
backgroundNode.position = CGPoint(x: size.width / 2.0, y: 0.0)
addChild(backgroundNode)

let playerNode = SKSpriteNode(imageNamed: “Player”)
playerNode.position = cGPoint(x: size.width / 2.0, y: 80.0)
addChild(playerNode)
```



SKScene Review

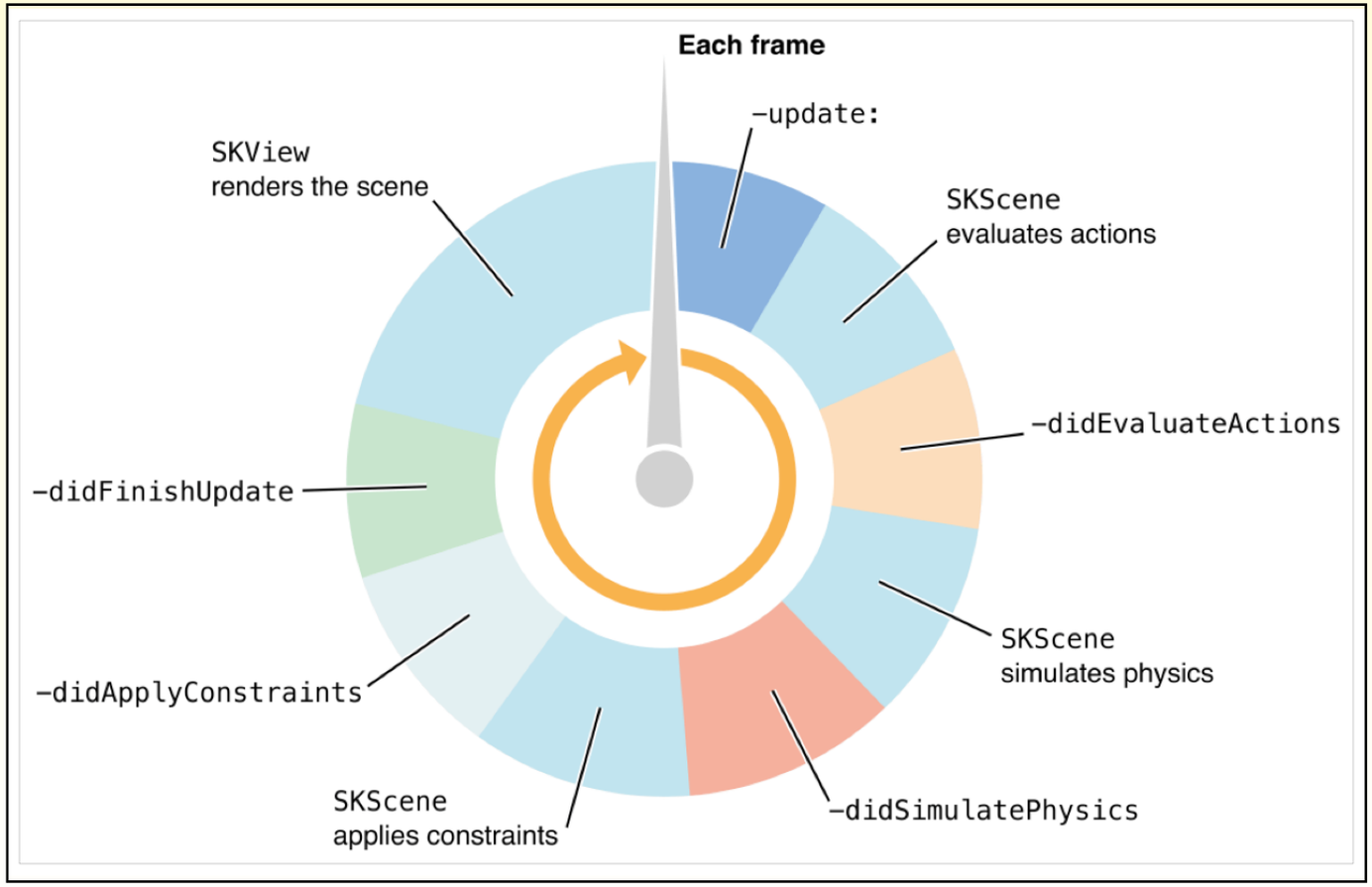
6. Present the complete scene in the **GameViewController** **viewDidLoad** method:

skView.presentScene(**scene**)

SKScene Rendering Loop

Each **iteration** of this **loop** generates the next frame in the scene

The **loop** has several steps that are called for **actions**, **physics**, **and constraints** on the scene



SKScene Rendering Loop

The scene calls its **update**() method

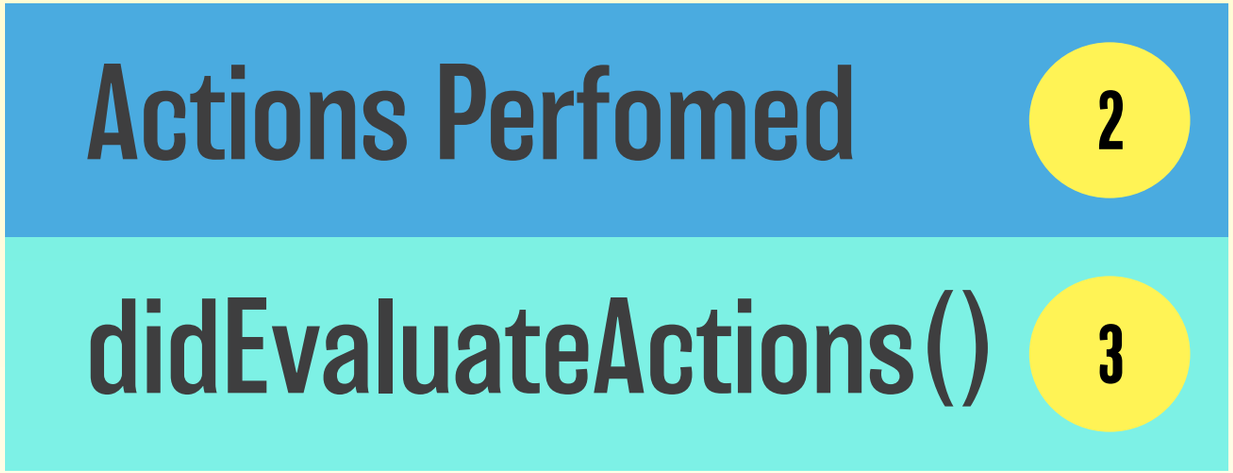
Called **before** actions are evaluated

This is where you will have most of your **game logic**

update() 1

Primary place to handle **AI**, **game scripting**, **input handling**, **actions** or other game logic.

SKScene Rendering Loop



All actions will be performed

Next, the scene calls the **didEvaluateActions**()

Any **post-action game logic** can be put here

Test the position of a node, **after actions** were performed

SKScene Rendering Loop

Physics Simulations 4

didSimulatePhysics() 5

Next the scene **executes any physic simulations** on physics bodies in the scene
The scene then calls the **didSimulatePhysics()** method.

SKScene Rendering Loop

Constraints Applied 6

didApplyConstraints() 7

Next the scene **applies any constraints associated with nodes in a scene**

The scene then calls the **didApplyConstraints()** method

SKScene Rendering Loop

didFinishUpdate() 7

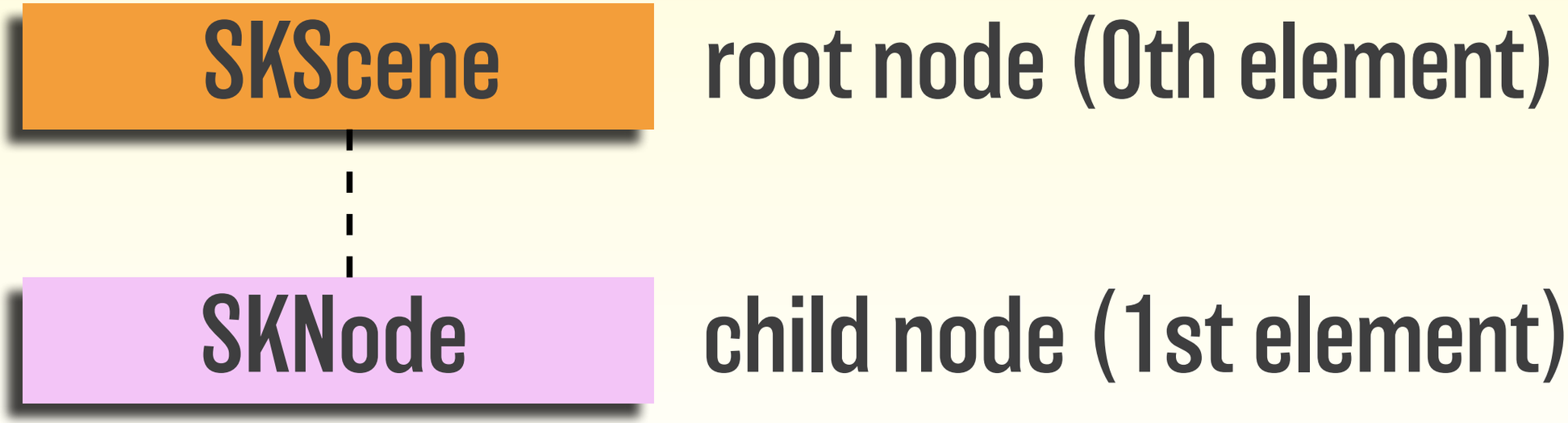
SKView renders scene 8

The last step:

Once **all updates in the frame** are checked, the **SKView renders the scene**

SKScene Node Tree

A SKScene is the **root node** in a **tree** of SpriteKit nodes within a scene



SKScene Node Tree

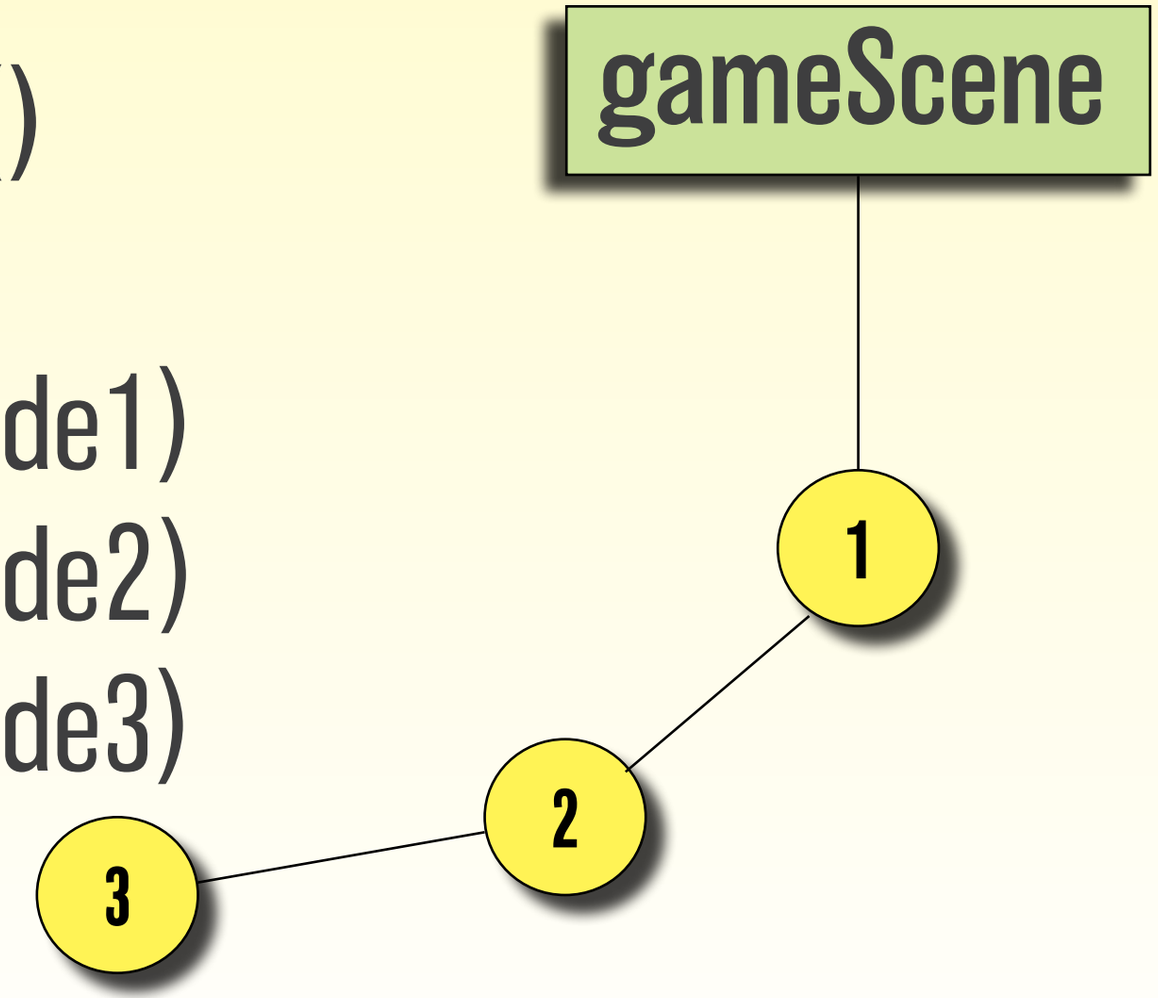
addChild() - adds a node to the end of the receiver's collection of child nodes

insertChild(_:at:) - method inserts a child node at a specific position in the receiver's collection of child nodes

removeFromParent() - removes the receiving node from it's parent

SKScene Node Tree

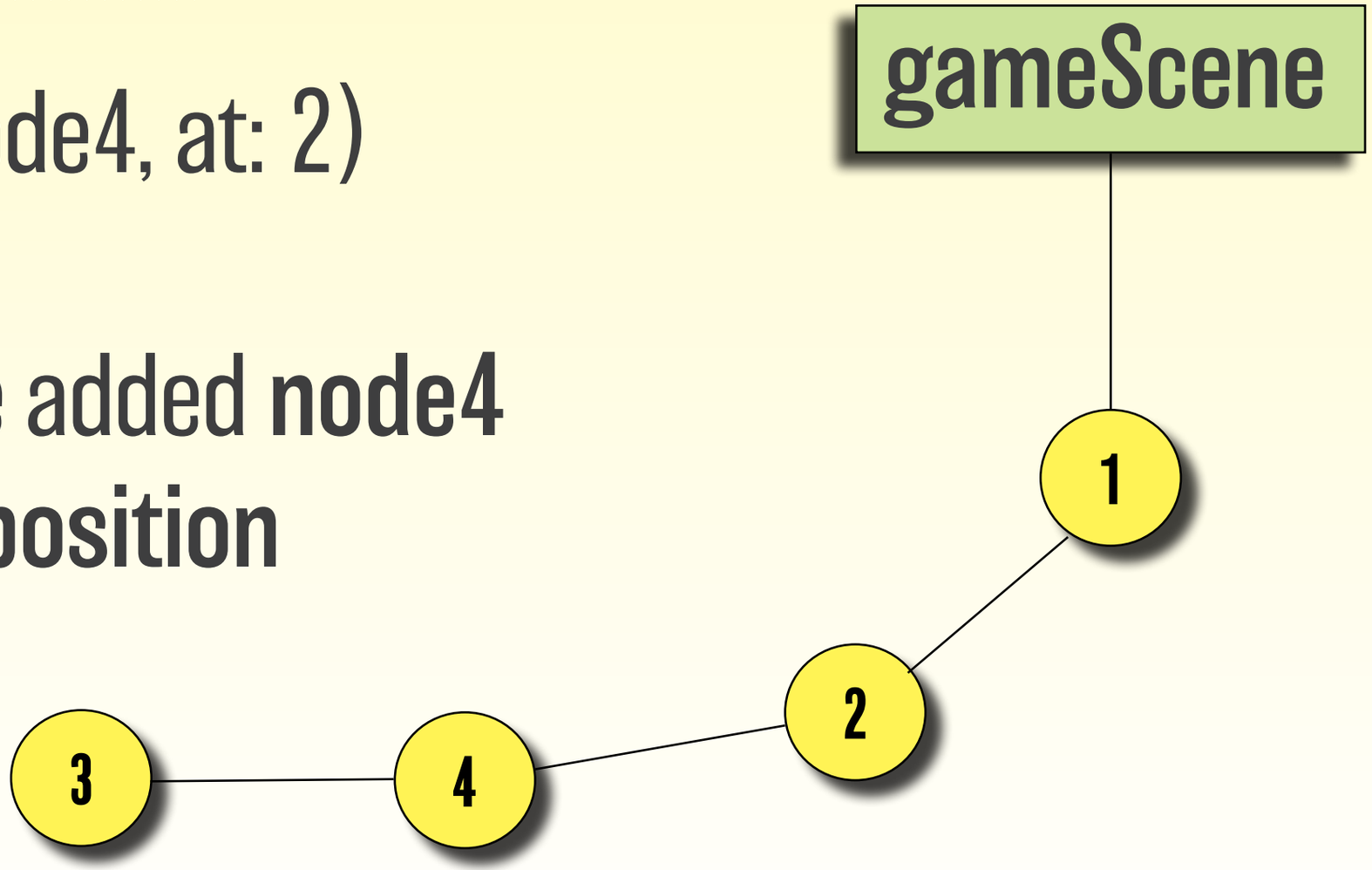
```
gameScene = SKScene()


gameScene.addChild(node1)
gameScene.addChild(node2)
gameScene.addChild(node3)
```



SKScene Node Tree

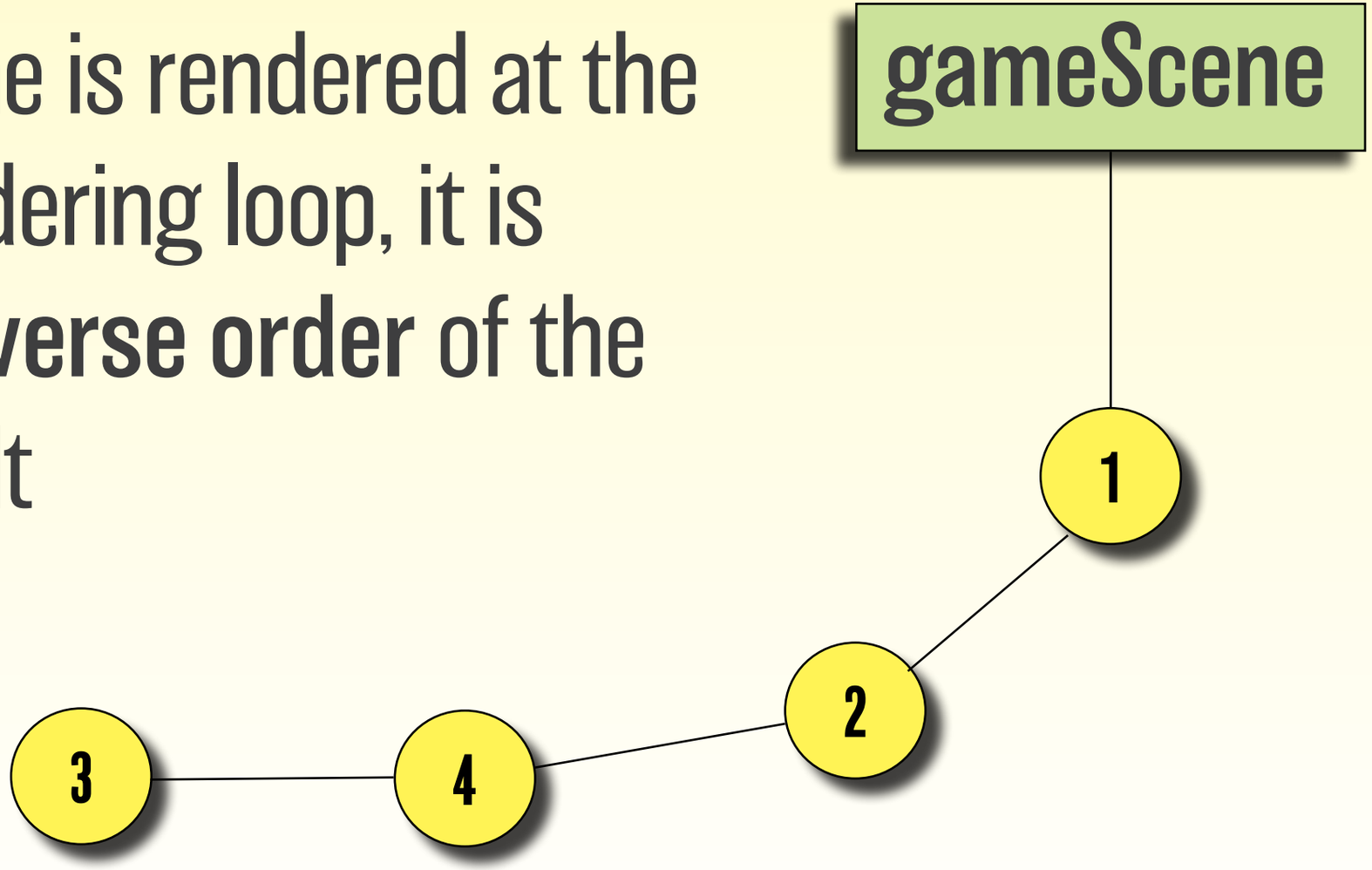
insertChild(node4, at: 2)

In this case we added **node4** into the **third position**



Rendering Nodes

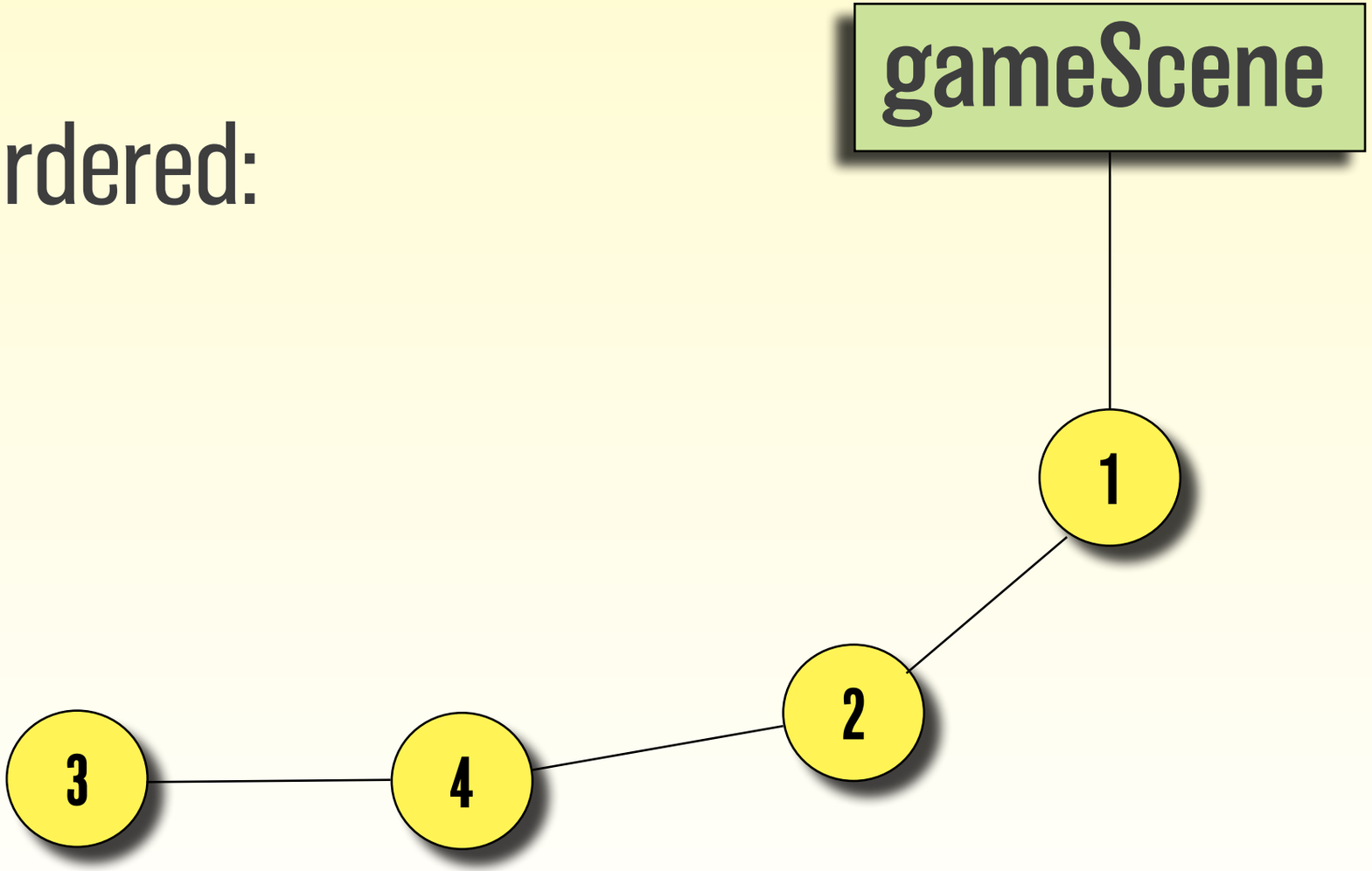
When the scene is rendered at the end of the rendering loop, it is rendered in **reverse order** of the way it was built



Rendering Nodes

Nodes will be ordered:

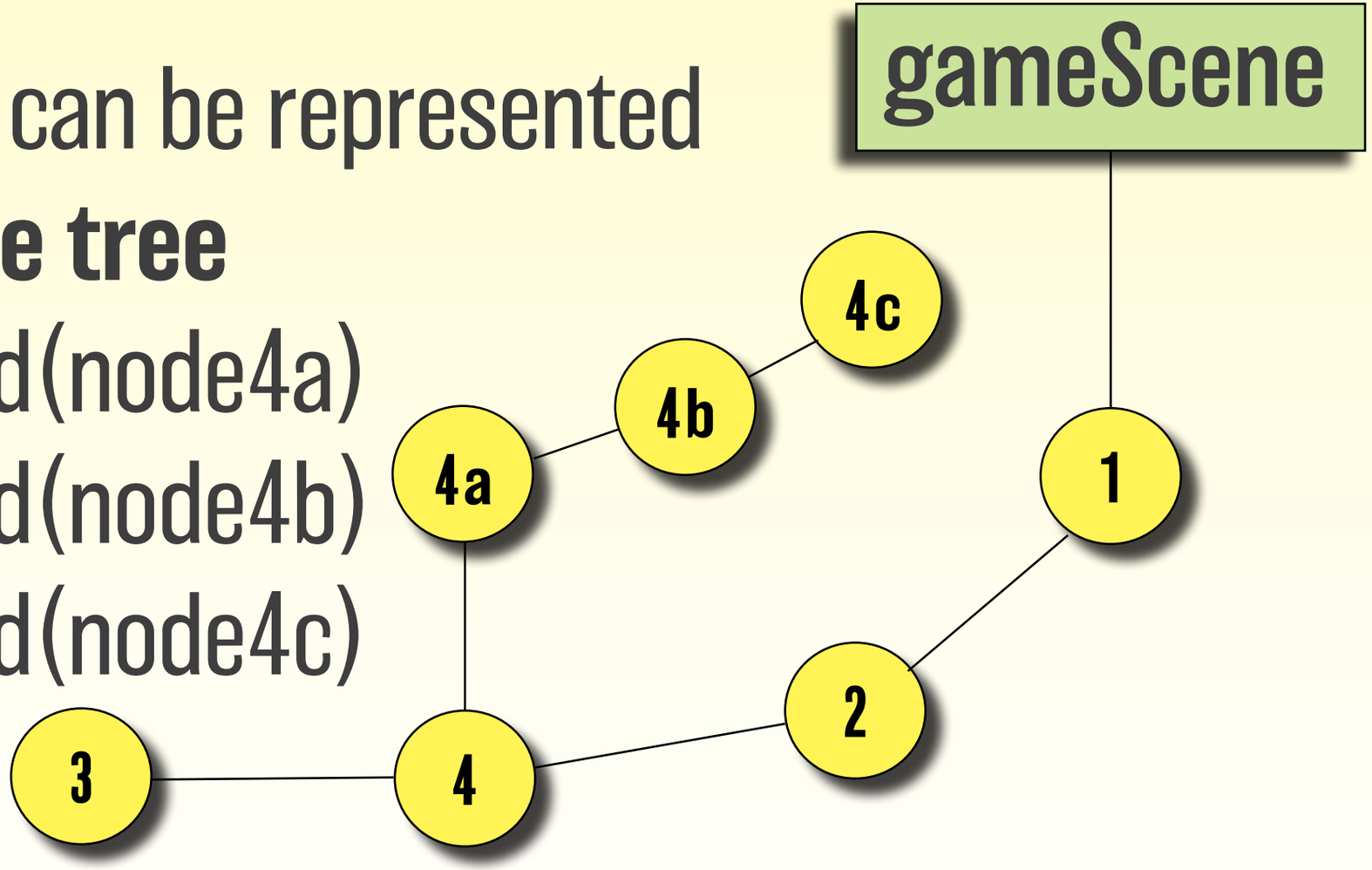
1. **node3**
2. **node4**
3. **node2**
4. **node1**



Rendering Nodes

Nested nodes can be represented within the **node tree**

node4.addChild(node4a)
node4.addChild(node4b)
node4.addChild(node4c)



Rendering Nodes

When doing **hit testing** on the scene, SpriteKit checks the **last rendered node first**.

Searching Nodes

Every **SKNode** object has a **name property** of type **String**

This **name property** can be used to **search for a specific node** within a **scene**

Searching Nodes

When **searching** for a node of name “**Player**” we can use

childNode(withName: “**Player**”)

This will return an optional **SKNode**

Searching Nodes

When **searching** for multiple nodes of the same name within a node tree

```
enumerateChildNodes(withName: “Player”) {
    (node, stop) in
}
```

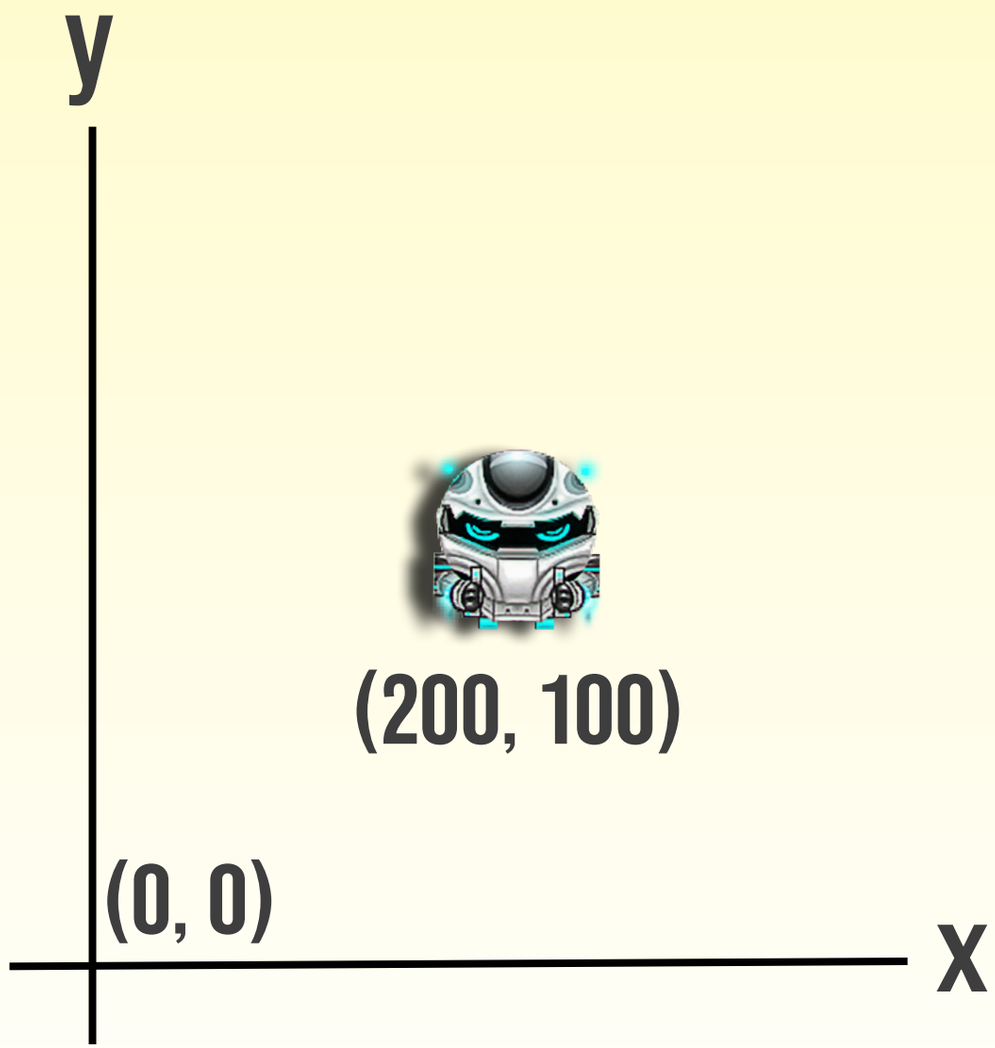
Searching Nodes

```
enumerateChildNodes(withName: “Player”) {
    (node, stop) in

    stop.pointee = true
}
```

1 Used to end searching

Coordinates

The **origin** (0,0) of the **SKScene** is at the **bottom left corner**



Anchor Points

(0.5, 0.5)

Center

(0.0, 0.0)

Bottom Left

(0.5, 0.0)

Bottom Right

(1.0, 1.0)

Top Right