Transitioning Between Two Scenes

Overview

Typically, you transition to a new scene based on gameplay or user input. For example, if the user presses a button in your main menu scene, you might transition to a new scene to configure the match the player wants to play.

Code below shows how you might implement the event handler in a sprite. The handler first runs an animation on itself to highlight the button (not described here). Then, it creates a transition object and the new scene. Finally, it calls the view to present the new scene. The transition means that this change is animated.

```swift
override func mouseUp(with event: NSEvent) {
        run(buttonPressAnimation)
        let reveal = SKTransition.reveal(with: .down,
                                          duration: 1)
        let newScene = GameConfigScene(size: CGSize(width: 1024, height: 768))

        scene.view.presentScene(newScene,
                                transition: reveal)
    }
```

When the transition occurs, the scene property is immediately updated to point to the new scene. Then, the animation occurs. Finally, the strong reference to the old scene is removed. If you need to keep the scene around after the transition occurs, your app needs to keep its own strong reference to the old scene.

When organizing your game, it can be helpful to create a diagram that shows all the scenes in a game, the transitions that occur between scenes, and the data that must be passed to the new scene when a transition occurs. Unlike view controllers in iOS, SpriteKit does not provide a built-in mechanism for passing data between scenes. If you need to provide data during a scene transition, you need to implement your own mechanism to configure the new scene. Typically, this means defining custom methods and properties on each scene.