SpriteKit Scene Editor

Get Games Up and Running Quickly

"Type a quote here."

–Johnny Appleseed

Topics

- Adding sprites to a scene

- Changing sprite properties

- Referencing the sprites in code

- Exploring node types in the object library

  - Fields

  - Lights

  - Particles

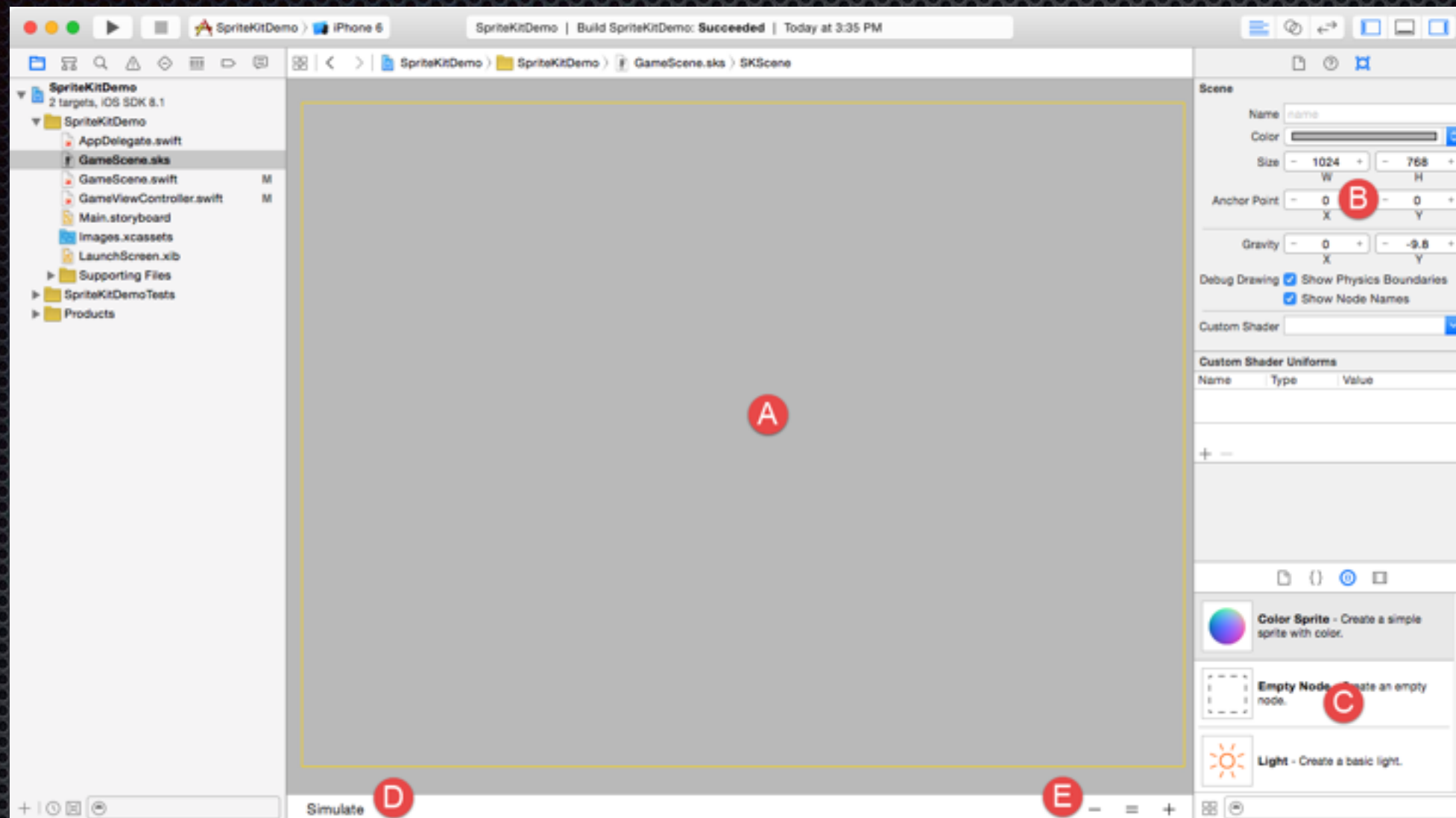- Testing physics interactions

Using the Scene Editor

Benefits

- Visualization

- Faster Development

- Cleaner Code!

Drawbacks

- Incomplete Implementation

- Lacking Documentation

- Wonky Controls

The Scene Editor

A. **Scene Canvas:** Place, position, and configure nodes.

B. **SKNode Inspector Panel:** Set several node properties.

C. **Object Library:** Create nodes of various types from scratch.

D. **Simulate/Edit:** Experiment without building and running.

E. **Zoom:** Useful if controls bother you.

Setting the Scene

Design a level in three steps:

1. Start a new SpriteKit game project.

2. Clear out GameScene.swift methods.

3. Build the level with the scene editor…

Object Library

- Color Sprite (SKSpriteNode): Player, enemies, other objects

- Empty Node (SKNode): Placeholders, end-effectors

- Light (SKLightNode): Static or dynamic light sources

- Emitter (SKEmitterNode): Fire, smoke, explosions, other ambiance

- Label (SKLabelNode): HUD elements, other text

- Shape (SKShapeNode): Complicated shapes, custom drawing

Referencing Nodes

- childNodeWithName(name: String)

  - Good for referencing persistent nodes (ex. a player sprite that is not going away).

- nodeAtPoint(p: CGPoint)

  - Good for getting a node in response to a touch event.

- node.children

  - Returns all children.

Object Library

SKFieldNode

- Linear
- Radial
- Spring
- Drag
- Vortex

- Turbulence
- Noise
- Velocity
- Magnetic?
- Electric?

Object Library

SKLightNode

- Enable / disable

- Position

- Category

- Color

  - Light

  - Shadow

  - Ambience

Object Library

SKEmitterNode

- Texture

- Full Particle Control

- Color / Blend

- No Physics Body

- Only way to controls the particles it creates

Editor as a Learning Tool

* Setting up quickly

* Testing basic interactions in simulator

* Experimenting with object types and node hierarchy

* Creating test environment to learn more