

Intro to CocosSharp

2D Game Development with C#



Building Games with Xamarin

Lots of Options

- MonoGame
- Mono for Unreal Engine
- Sprite Kit
- Scene Kit
- CocosSharp

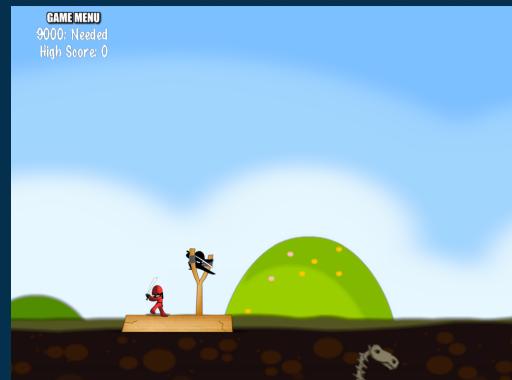


Cross Platform and Open Source



Parts of a CocosSharp Game

- Application
- Scenes
- Layers
- Sprites
- Actions
- Other cool stuff - Particle systems, audio, physics ...



CocosSharp API



CCApplication

- Creates and initializes the graphics device
- Sets the application delegate
- Starts the game

```
var app = new CCApplication ();
app.ApplicationDelegate = new GoneBananasApplicationDelegate ();
app.StartGame ();
```



CCApplicationDelegate

- Handles application lifecycle
- Similar to UIApplicationDelegate in iOS
- Set the application's content folder
- Load the main window's first scene

```
public virtual void ApplicationDidEnterBackground  
public virtual void ApplicationDidFinishLaunching  
public virtual void ApplicationWillEnterForeground
```



Content Folder

Project Folder Containing Resources

- Fonts
- Sounds
- Images
- Animations
- Particle Systems

Set via application's ContentRootDirectory

```
application.ContentRootDirectory = "Content";  
application.ContentSearchPaths.Add ("sounds");
```

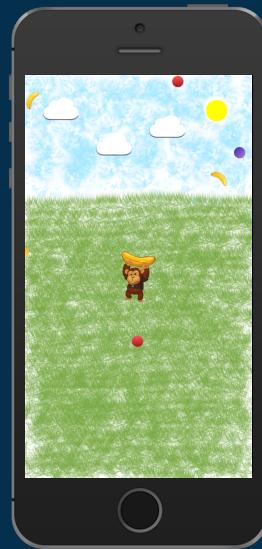


CCDirector

- Available via Window.DefaultDirector
- Manages any additional scene loading

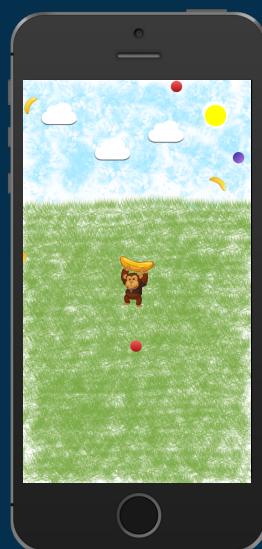
```
Window.DefaultDirector.ReplaceScene (
```

```
GameLayer.GameScene (Window));
```



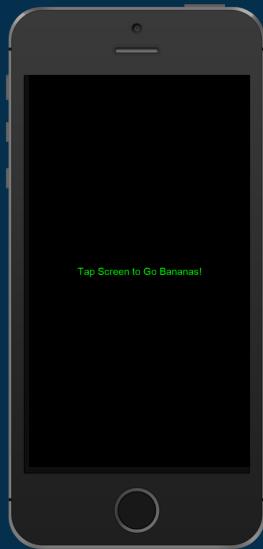
CCScene

- Manages game logic for various portions of the game
- Contains layers (CCLayer)



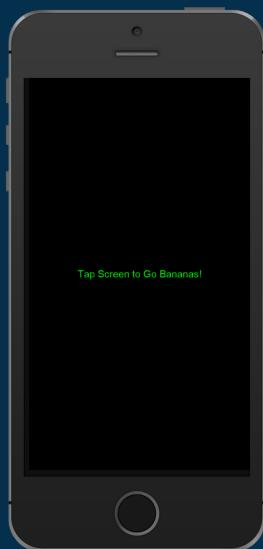
CCLayer

- Added to scene
- Contains node tree (sprites, labels, menus, etc)
- Schedule method to run code at repeated interval



CCLayer

```
var scene = new CCScene (mainWindow);
var layer = new GameStartLayer ();
scene.addChild (layer);
```



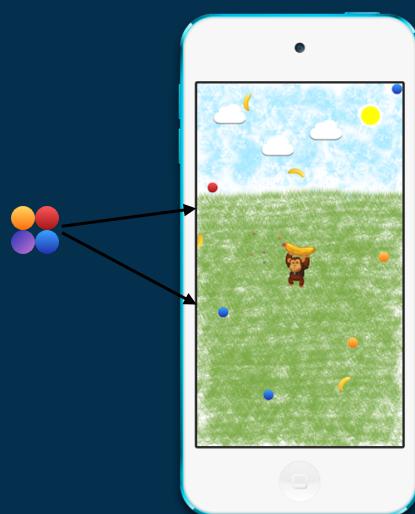
CCSprite

- Sprites are nodes that create images in the game
- Image file must be in Content folder
- DefaultTexelToContentSizeRatio(s)
 - New in version 1.2.1.0
 - To maintain content size across resolutions



CCSpriteBatch

- Efficient sprite loading
- Renders all children



CCAction

- Actions perform tasks on nodes
- For example, animating sprites
- Run multiple actions sequentially using CCSequence
- Run multiple actions in parallel using CCSpawn



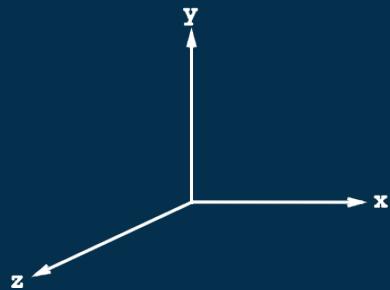
CCTouch

- Encapsulates a touch on the screen
- Override touch methods in layer
 - TouchesBegan, TouchesMoved, TouchesEnded, etc
 - All touches at once
 - Each touch one by one



Accelerometer

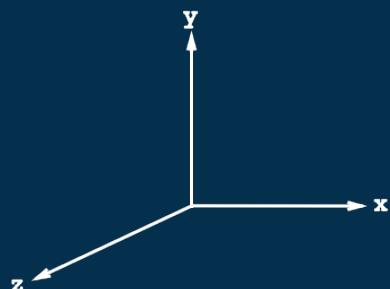
- Available via CCAccelerometer
- Access with Window.Accelerometer



Accelerometer

```
var listener = new CCEventListenerAccelerometer();
listener.OnAccelerate = DidAccelerate;
AddEventListener (listener);

public void DidAccelerate(CCEventAccelerate accelEvent)
{
    accelEvent.Acceleration.X;
    accelEvent.Acceleration.Y;
    accelEvent.Acceleration.Z;
}
```



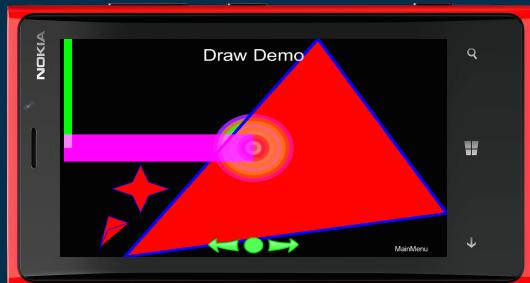
Audio

- SimpleAudioEngine.SharedEngine
 - Sound Effects
 - Background Sound
- Pause/Resume background sound in CCAplicationDelegate
 - Pause when app enters background
 - Resume when app enters foreground



Drawing Primitives

- CCDrawNode
- Draw lines, polygons, circles, etc ...



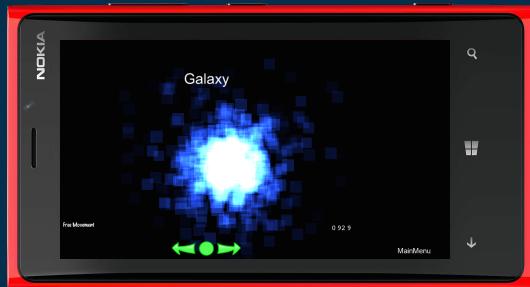
Drawing Primitives

```
var draw = new CCDrawNode();  
  
draw.DrawPolygon(  
    star,  
    star.Length,  
    new CCCColor4F(1, 0, 0, 0.5f), 1,  
    new CCCColor4F(0, 0, 1, 1));
```



Particle Systems

- Graphical effects by rendering a set of particles
- Several built in - smoke, galaxy, rain, ...
- Can add custom particle systems



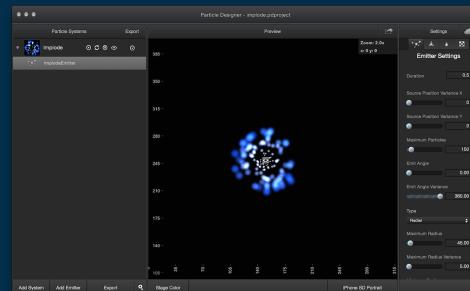
Particle Systems

```
var sun = new CCParticleSun (pt);
sun.StartColor = new CCColor4F (CCColor3B.Red);
sun.EndColor = new CCColor4F (CCColor3B.Yellow);
```



Custom Particle Systems

- Can be designed in popular third party editors
- Import plist into CocosSharp
- Use cross-platform
- CCParticleSystemQuad



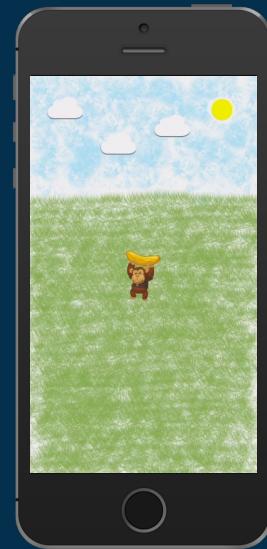
```
var explosion = new CCParticleSystemQuad ("implode.plist");

explosion.Position = pt;
AddChild (explosion);
```



Parallax

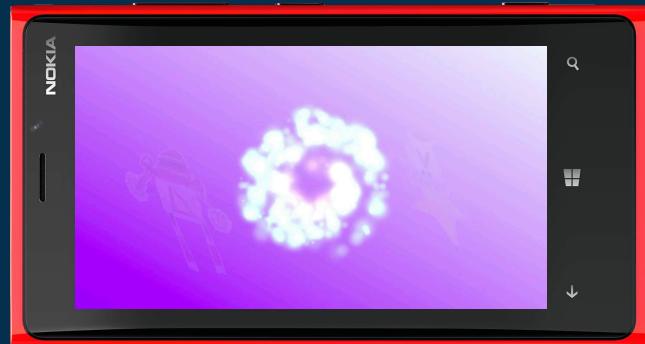
```
var parallaxClouds = new CCParallaxNode {  
    Position = new CCPoint (0, h)  
};  
  
float yRatio1 = 1.0f;  
float yRatio2 = 0.5f;  
  
CCParallaxNode  
  
Children move at different relative speeds  
parallaxClouds.AddChild (cloud1, 0,  
    new CCPoint (1.0f, yRatio1),  
    new CCPoint (100, -100 + h - (h * yRatio1)));  
  
parallaxClouds.AddChild (cloud2, 0,  
    new CCPoint (1.0f, yRatio2),  
    new CCPoint (250, -200 + h - (h * yRatio2)));
```



Effects

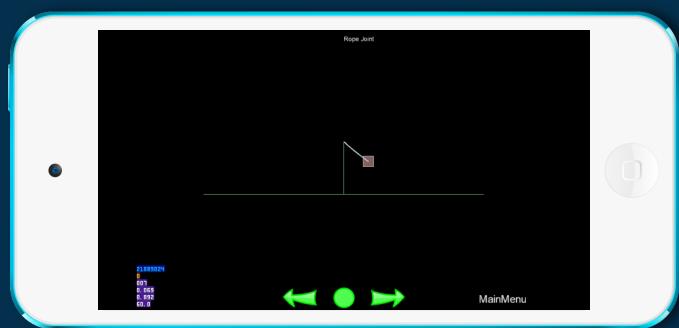
- Lots of visual effects
- Waves, Twirl, Lens3D, Shuffle Tiles, ...
- Implemented using actions

```
var effects = new CCSpawn (  
    new CCFadeIn (5.0f),  
    new CCWaves (5.0f,  
        new CCGridSize (10, 20), 4, 50)  
);
```



Physics

- 2D Rigid Body Physics
- C# port of Box2D
- world, body, shape, fixture
- Chipmunk# also available

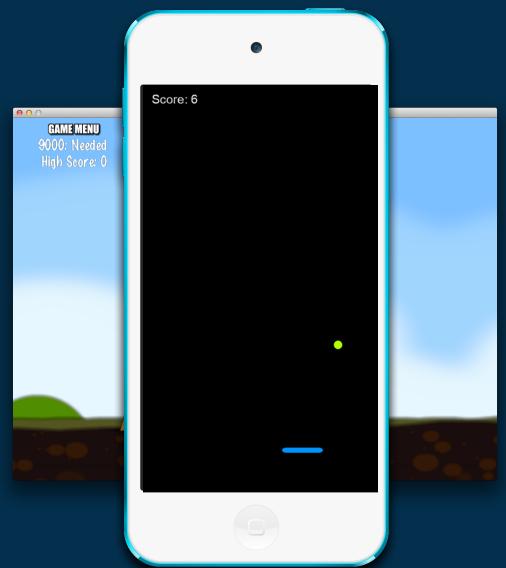


DEMO



CocosSharp Resources

- Source
 - github.com/mono/CocosSharp
- Samples
 - github.com/mono/cocos-sharp-samples
- Gone Bananas Walkthrough
 - github.com/mono/cocos-sharp-samples/blob/master/GoneBananas/Walkthrough.md
- Introduction to Game Development with CocosSharp
 - developer.xamarin.com/guides/cross-platform/game_development/cocosharp/first_game/



Thank You!

Mike Bluestein
@mikebluestein

