# Project Planning

CS-526: Mobile Games

# Table of Content

# Target Platforms / Devices

- iOS
- Android
- Windows
- MacOS
- WebGL

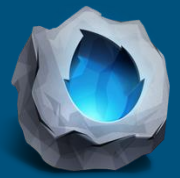

- Handheld
- Augmented Reality (ARKit / ARCore + WMR / HoloLens)
- Virtual Reality (HTC Vive / Oculus / Cardboard)

# Development Tools

# Game Engines

- Unity3D
- Swift
- Cocos
- Corona
- Unreal Engine



Coco Studio


# Game Engines

| | CocosCreator / Cocos2D | Swift (+ SceneKit) | Unity3D | Unreal Engine |
|---|---|---|---|---|
| **Platform** | iOS<br>Android* | iOS | iOS<br>Android<br>AR / VR | iOS<br>Android<br>AR / VR |
| **Language** | Javascript<br>C++ (-x) / Swift (2D) | Swift | C#<br>Javascript | C++ |
| **Difficulty** | ★ | ★ | ★★★ | ★★★★ |
| **2D/3D** | 2D | 2D/3D* | 2D/3D | 2D*/3D |
| **Interface** | basic IDE | library-based | IDE | IDE |
| **Resources** | ★★★ | ★★ | ★★★★ | ★★★★ |

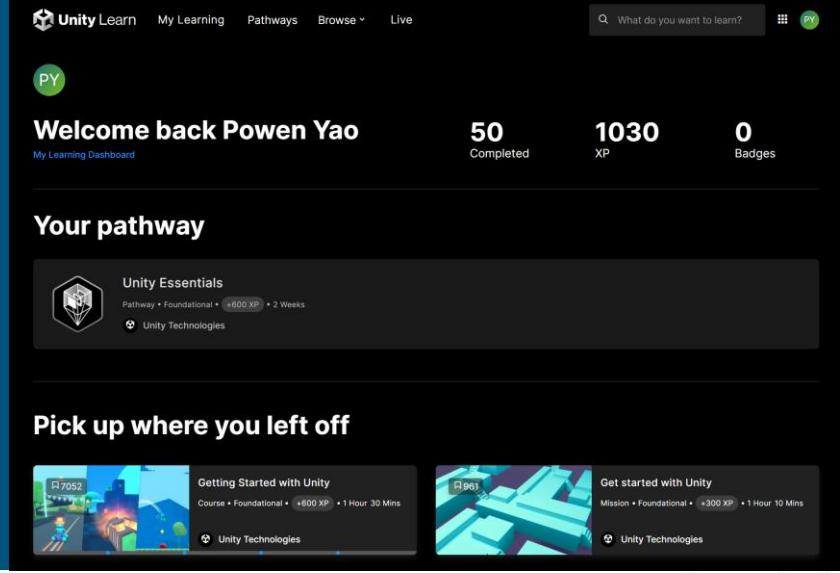
# Game Engines

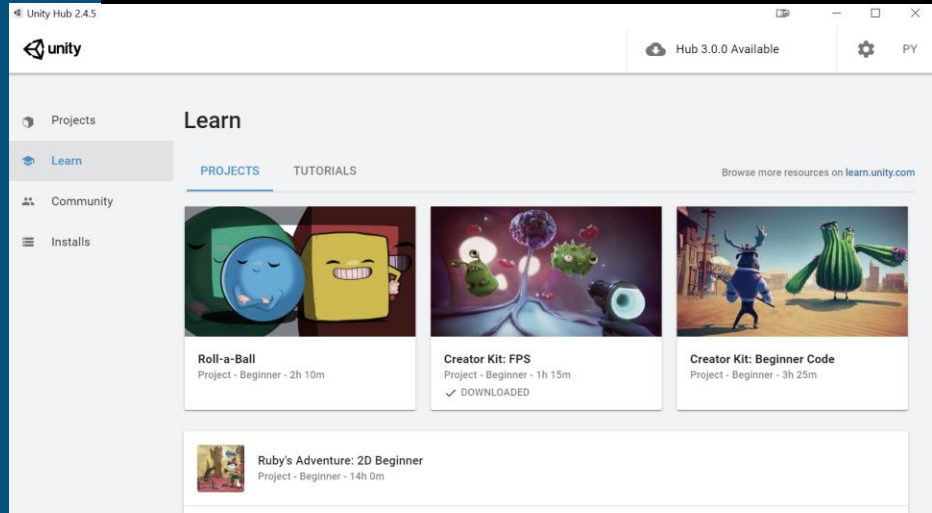
| | CocosCreator / Cocos2D | Swift (+ SceneKit) | Unity3D | Unreal Engine |
|---|---|---|---|---|
| **Pros** | Learning curve<br>Open Source<br>Community* | Learning curve<br>Native<br>(Performance) | Cross-Platform<br>AR / VR<br>Editor<br>Community<br>Asset Store | Cross-Platform<br>Editor<br>Community<br>Graphics |
| **Cons** | Editor<br>Features<br>Updates | iOS only<br>Editor | Collaboration<br>Build size | Learning curve<br>2D |

Other possible solutions: Lumberyard, Corona

# Unity

- Download Unity through Unity Hub
  - https://unity3d.com/get-unity/download
  - It helps maintain different versions of Unity
- Unity Learn
  - https://learn.unity.com/
  - Also Do Projects and Tutorials in Unity Learn through Unity Hub



# Integrated Development Environment IDE

- Visual Studio Code
  - https://code.visualstudio.com/
- Visual Studio 20xx
  - https://visualstudio.microsoft.com/vs/community/
- JetBrain Rider
  - https://www.jetbrains.com/rider/
  - https://education.github.com/pack?sort=popularity&tag=Developer+tools

# IDE - JetBrain Rider

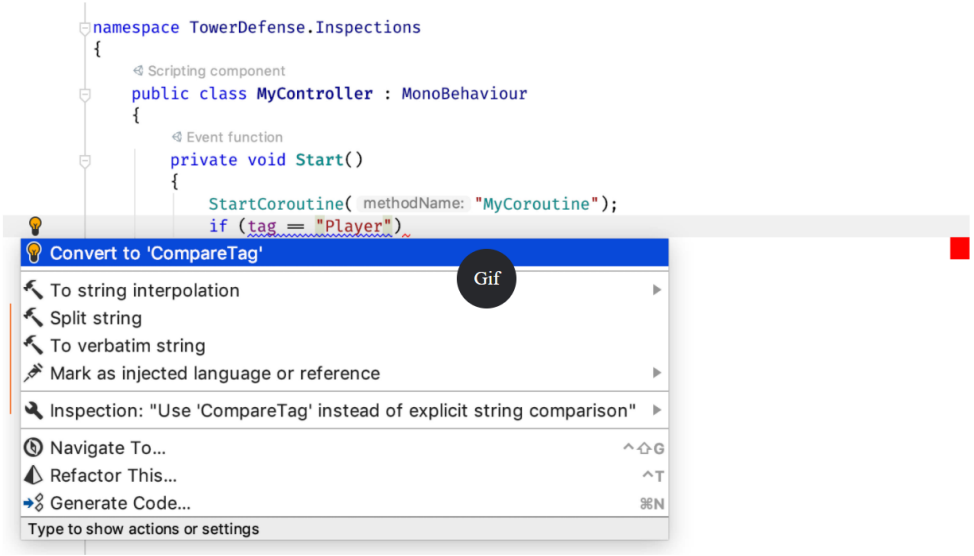
- “Intelligent” IDE
- https://www.jetbrains.com/lp/dotnet-unity/

**Coding hints**

Rider provides top-notch code analysis for C#, and that includes Unity-specific code inspections and quick-fixes for them.

For example, Rider will warn you against using inefficient string literal comparison with the `tag` property, and will provide a quick-fix to rewrite this as a call to `CompareTag`.

Similarly, Rider will warn you if you try to use the `new` keyword to create a new instance of a class deriving from `MonoBehaviour` or `ScriptableObject`. Just press `Alt` `Enter` to have Rider fix the problem for you.



# File Sharing

- Google Drive (unlimited storage for students)
  - Collaborate on documents (i.e. GDD)
  - Share stat sheets (Stats from Analytics, etc.)
  - Share assets (Unity Assets)
- Dropbox
- OneDrive
- Etc (pick what works for you)

# Communication

- Message-based Services
  - Facebook Messenger
  - WeChat
- Channel-based Services
  - Discord
  - Slack
- Meeting-based Services
  - Zoom
- Email

# Ideation / Prototyping

- Miro
- Figma
- Google Drawings
- Etc (pick what works for you)

# Task Tracking

- Class Spreadsheet
    - Required for class
- Trello
- Miro
- Jira
- Git
- Etc (pick what works for you)

# Source Control

| SVN | Git | Perforce | Unity Collaborate |
|---|---|---|---|
| Oldest and Easiest<br>Simple<br>Bigger overhead | Most Commonly used<br>Easier branching | Standard in game industry<br>Better management for big files<br>Complicated | Specific to Unity<br>Free version is limited in team size |

** GitHub (offers Student Developer Pack with 5 private repos)*

- *BitBucket (offers Academic License with unlimited repos)*

- ***Please please PLEASE don't email or use google drive as your version control!***

# GitClients

- Git with Commandline
- SourceTree
- GitKraken
- Etc (pick what works for you)

- For those of you that would like more practice with git and version control, learn it through a game!
    - https://learngitbranching.js.org/

# Recording

- For simple (short) gif, you can use ScreenToGif
  - https://www.screentogif.com/

- For recording, if you are using windows 10 PC, you could do windows+G to bring up Window Game Bar to record.
  - https://support.microsoft.com/en-us/windows/record-a-game-clip-on-your-pc-with-xbox-game-bar-2f477001-54d4-1276-9144-b0416a307f3c
- If you don't want to use Window Game Bar, you can also use the open source software - Open Broadcaster Software
  - https://obsproject.com/

# Documentation

- GDD
  - Required for class

- Wiki
- Google Docs
- Miro
- Git
- Etc (pick what works for you)

# Game Design Document - Purpose

- Direction and vision of project (Start off small!)
- Communication among team members and customers (represented by your instructors)
- Flesh out details for all common game elements
- Sets the big picture for everyone to work towards (Write your goals down.)
- Leads to understanding mappings between game design and technical implementation
- Allows you to have all aspects of your Game’s at one place

# Game Design Document - Outline

- 1st Draft
  - Objective of the game
- Later updates
  - UI concept mockups (ugly, rough, sketches ok)
  - Core mechanics / gameplay
  - How do game objects interact with mechanics and each other?

* This is a living document. You should update it every week

# Work Sessions

- For every 1 unit of class you take, you are expected to spend roughly 1 hour in class and 2-3 hours out of class.
- For a 4-unit class, you should be looking at near 4 hours of class time and 8-12 hours out of class.
- You should coordinate work sessions twice a week, each lasting around 4 hours.
- Utilize slack/discord to work efficiently

# Next Steps

- Decide your target platform and game type (Unity)
- Finalize your game idea
- Decide on game engine
- Setup your repository / communication channels (Discord/GitHub/GitLab)
- Create initial draft of Game Design Document

# Additional Notes

- Games Reference Guide
- *Sample Game Design Documents:*
  - Template
  - Doom: much bigger in scope, but gives you an idea, what a GDD of a full game is like
  - Fallout: Brotherhood of steel 2: final GDD for game
  - Diablo: final GDD for game
- Past Videos:
  - YouTube playlists
  - Various trailers (includes non-mobile games)

# Additional Notes

- Unity tutorials
  - Unity Learning Center
  - Ray Wenderlich Tutorials – Unity
  - Learn Unity – Beginners Game Development Tutorial by freeCodeCamp.org
- Cocos2D
  - Cocos2D-X Developers Guide / Tutorials
  - Cocos2D-X External Tutorials
  - Ray Wenderlich Tutorial - Cocos2D-X
- Swift tutorials
  - Apple developer guide
  - Ray Wenderlich Tutorials - Swift
  - Ray Wenderlich Tutorials - SpriteKit

# Questions