

INFT3960 – Game Production

Week 04

Module 4.2

Debugging

Course Overview

Lec	Start Week	Modules	Topics	Assignments
1	3 Aug	Mod 1.1, 1.2	Course Overview, Design Process	
2	10 Aug	Mod 2.1, 2.2, 2.3, 2.4	Unity3D Introduction, Introduction C#, Variables and Components, Hello World	
3	17 Aug	Mod 3.1, 3.2, 3.3	Booleans, Loops, Lists and Arrays	Assign 1 21 Aug, 11:00 pm
4	24 Aug	Mod 4.1, 4.2	Functions and Parameters, Debugging	
5	31 Aug	Mod 5.1, 5.2	Classes, Object Oriented	
6	7 Sep	Mod 6.1, 6.2, 6.3	Agile Processes, Risks and Prototypes, Testing	
7	14 Sep	Mod 7.1, 7.2	Puzzles, Guiding the Player	Assign 2 18 Sep, 11:00 pm
8	21 Sep	Mod 8.1	Game Physics	
9	12 Sep	Mod 9.1	AI for Games	
10	19 Oct	Mod 10.1, 10.2	Game Interface, Storytelling in Games	
11	26 Oct	Mod 11.1, 11.2	Graphics Pipeline, Animation in Games	Assign 3 1 Nov, 11:00pm
12	2 Nov	Mod 12.1, 12.2	Networked Games, Course Review	

Course Details

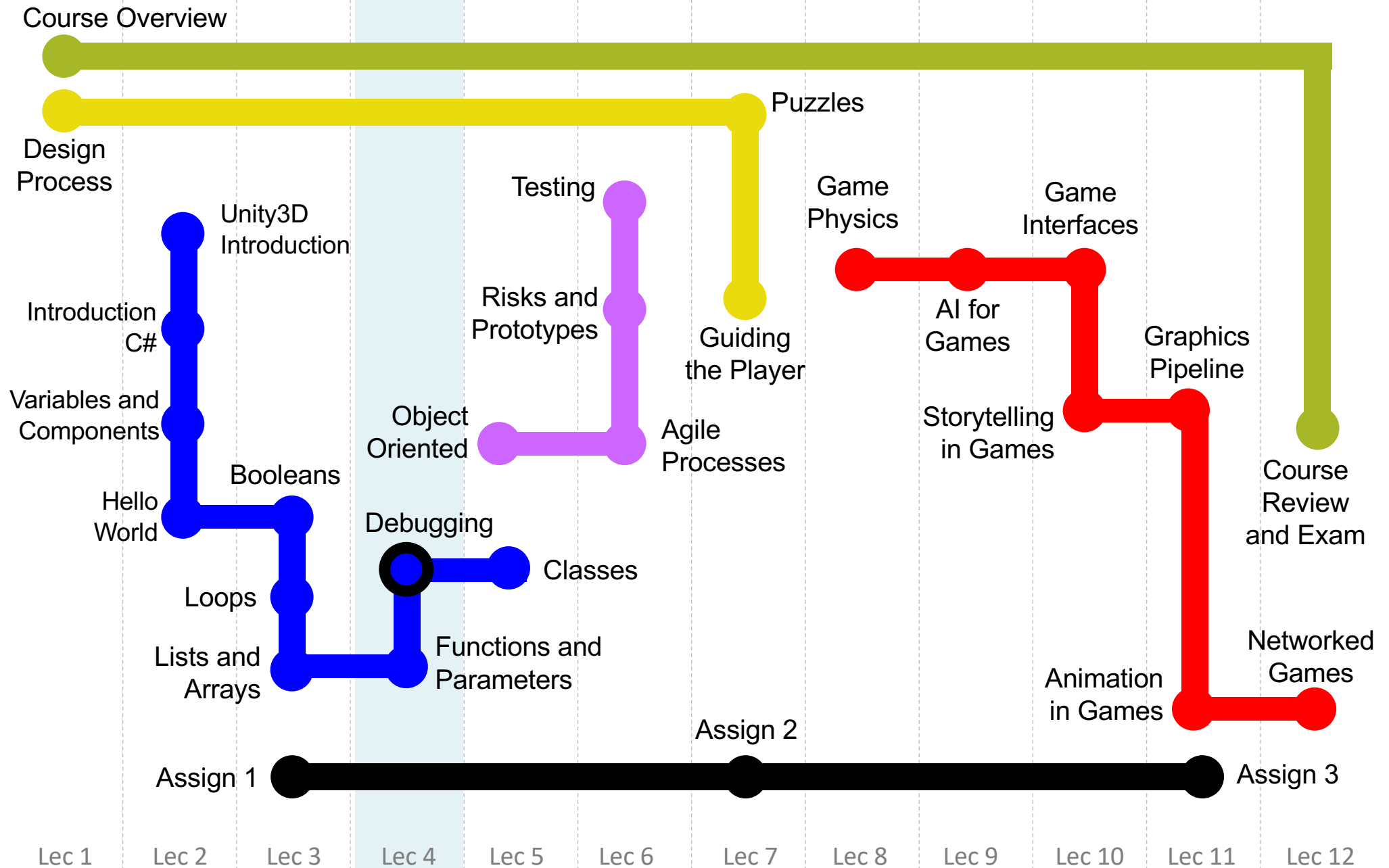
Game Design

Unity 3D and C#

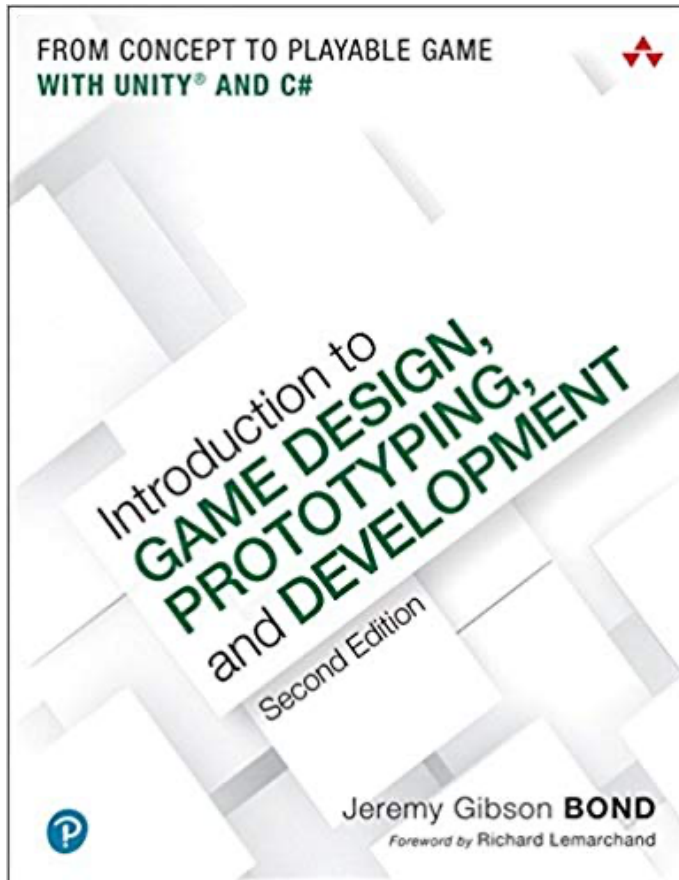
Development Process

Core Game Concepts

Assignments



Debugging – (Chapter 25)



DEBUGGING

Debugging – Topics



Getting Started with Debugging

Types of Bugs



- Compile-Time Bugs
- Bugs Attaching Scripts
- Runtime Errors

Stepping Through Code with the Debugger

- Attaching the Debugger to Unity

Watching Variables in the Debugger

Getting Started

Debugging is a way to step through and watch your code as it is running

This can help you

- Better understand code
- Find errors and bugs
- Track down inefficiencies

Getting Started

Debugging is built into Unity via MonoDevelop

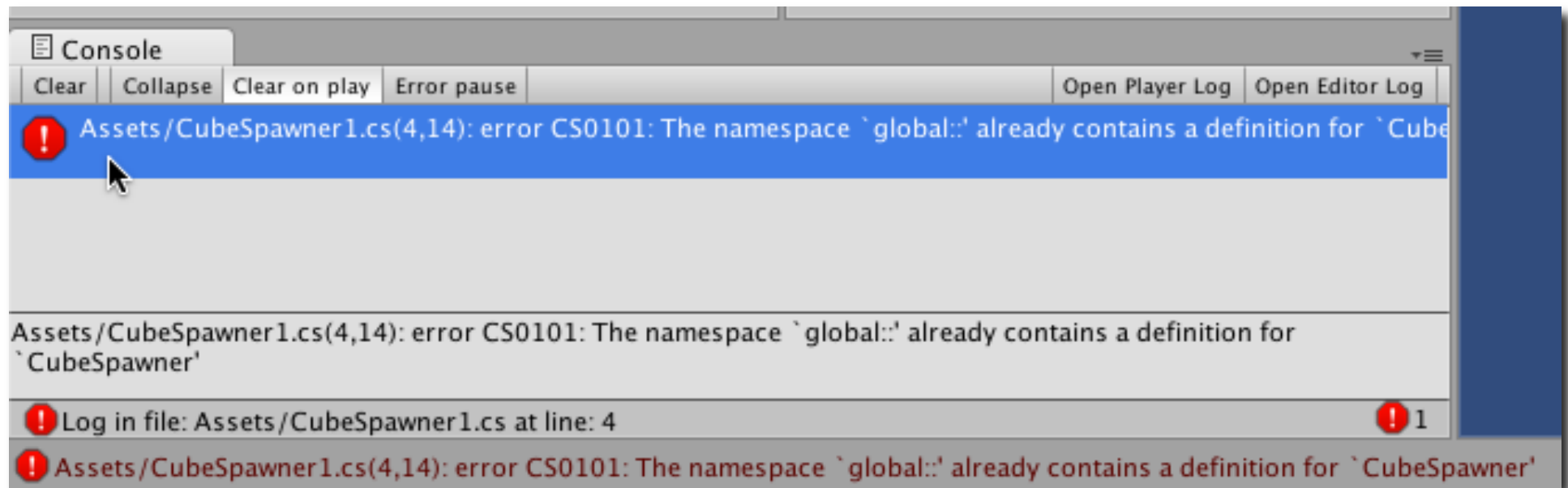
- The MonoDevelop debugger can attach to the Unity process to debug your code
- And, the MonoDevelop debugger can connect to an iOS or Android device and debug code running on the device!!!
- This is very helpful for finding issues with touch interfaces
- Can be done over either a cable or WiFi!
- The book has detailed instructions for using the debugger

Types of Bugs

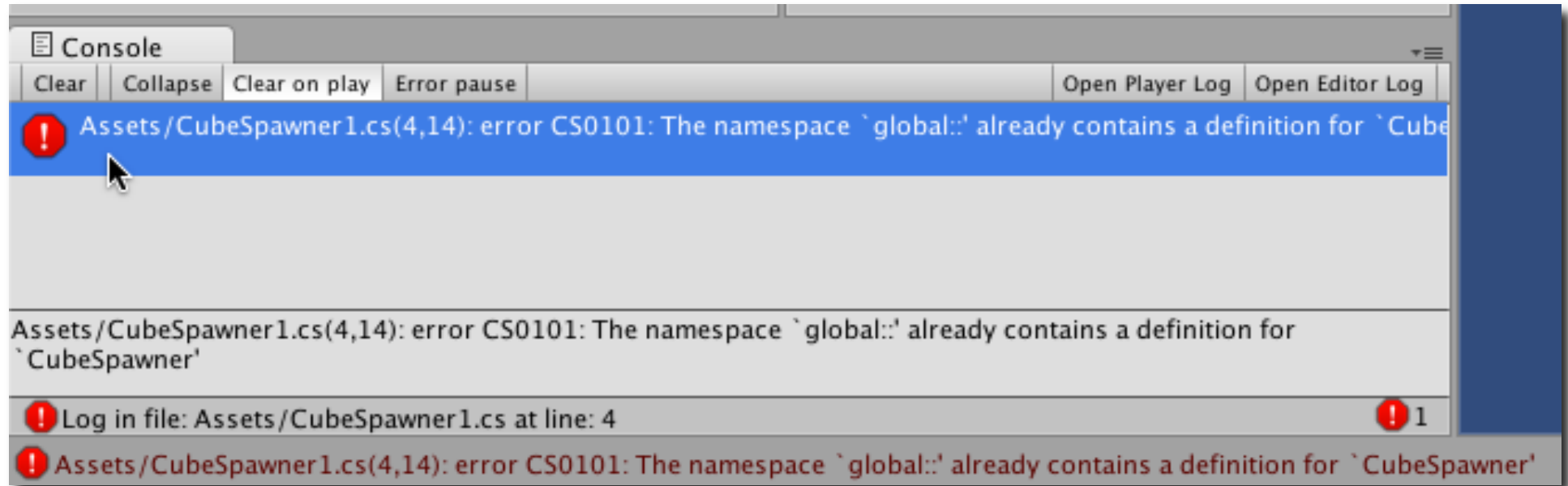
Compile-Time Bugs

- A bug found in the syntax of your code
- Compile-time bugs prevent your code from compiling
- Makes it unusable in Unity until the bug is resolved
- Compile-time bugs usually cause very specific errors

The error below is on line 4, character 14 of CubeSpawner1.cs



Compile-Time Bugs



Click the error message to get more information

The full error text can usually tell you what's wrong

- If not, search the Internet for the error number
- Example: "Unity error CS0101"
- Unity forums and StackOverflow.com have some of the best answers

Some Common Errors

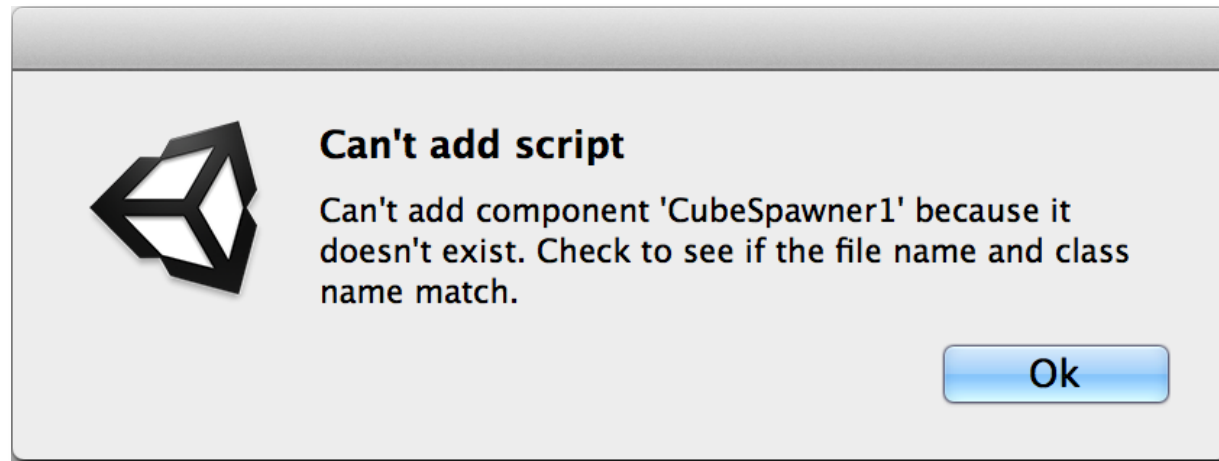
error CS0101: The namespace 'global::' already contains a definition for '_____'

- Two scripts are trying to define the same class
- Change the name of the class in one of the scripts

error CS1525: Unexpected symbol '}'

- Many "Unexpected symbol" errors are caused by a semicolon missing on a previous line or a misplaced brace
- Check line endings for semicolons ;
- Check to make sure all braces have a mate { }

Attaching Scripts to GameObjects



Error occurs when attempting to attach a script to a GameObject

Caused by name of the script not matching name of the defined class

Example

Script filename: CubeSpawner1 (or CubeSpawner1.cs)

Class name: `public class CubeSpawner : MonoBehaviour { ... }`

To Fix: Match the names to each other

Types of Bugs

Runtime Errors

- A bug that occurs when your code is running
- Unity has no way of predicting these

Most common types of Runtime Errors

UnassignedReferenceException

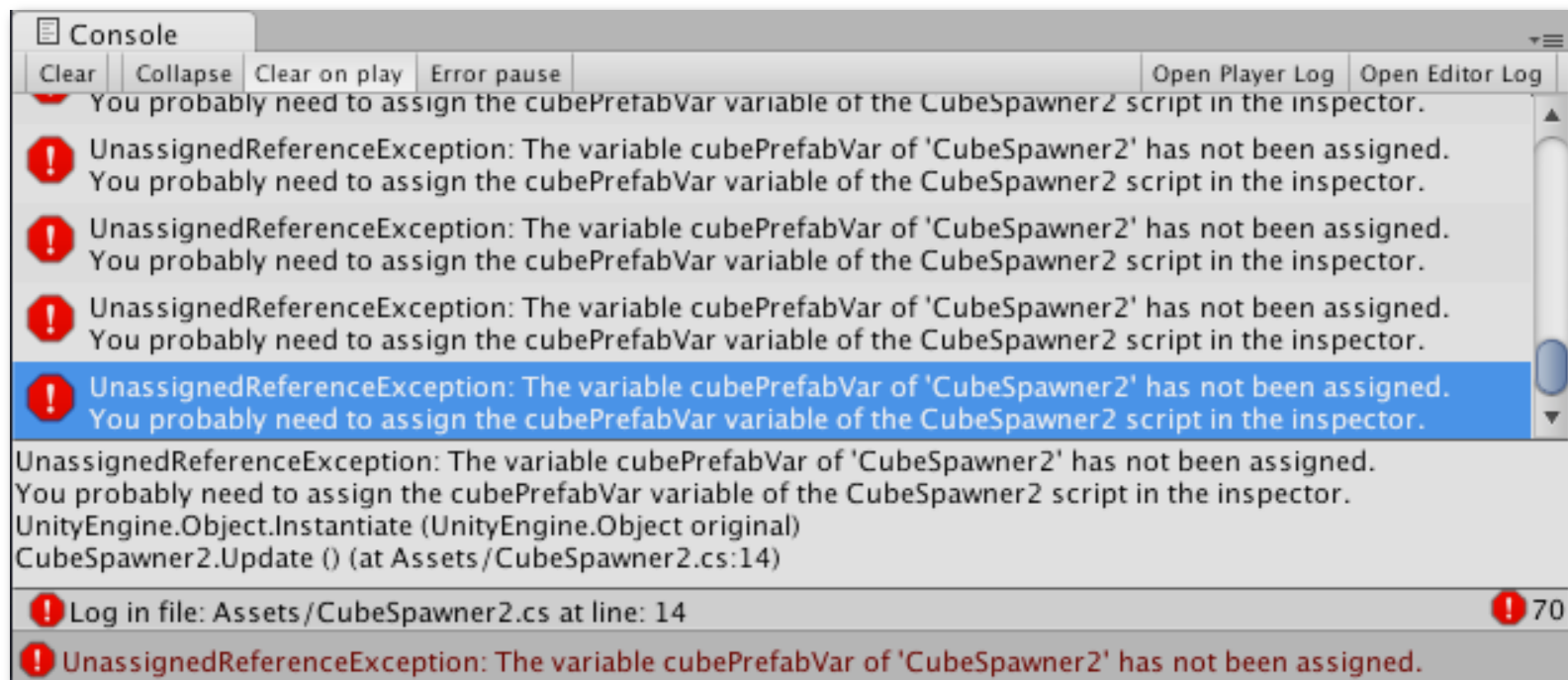
NullReferenceException

UnassignedReferenceException

A variable in the Inspector has not been set

Most commonly GameObject prefabs for Instantiate() calls

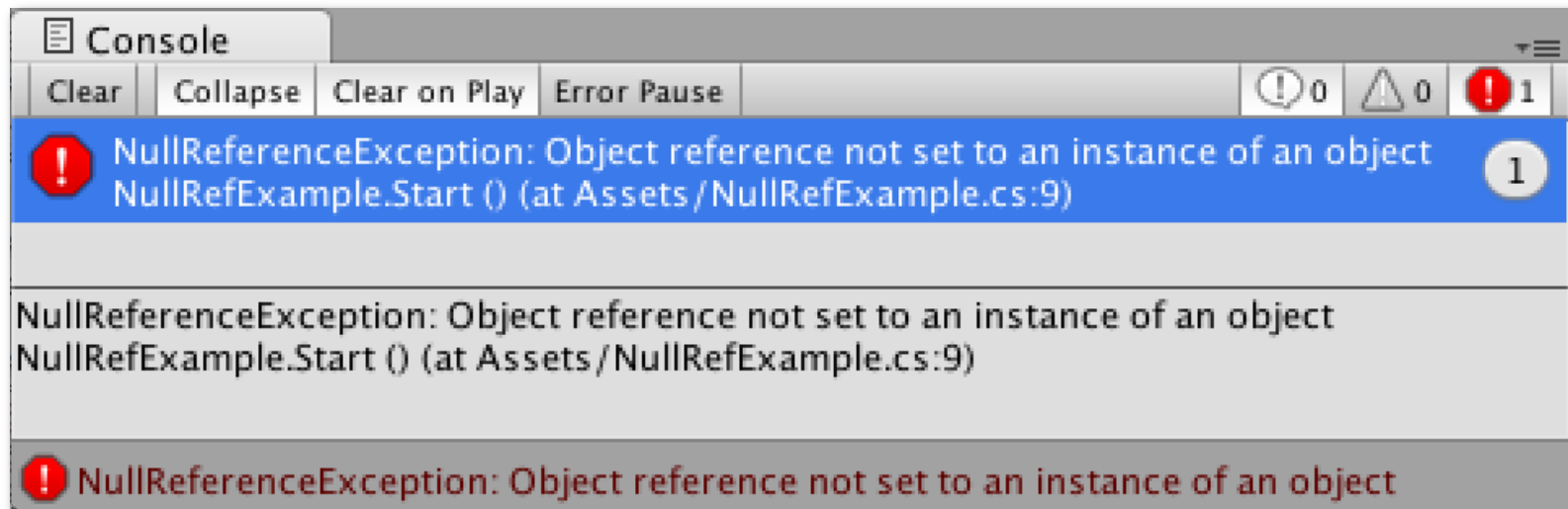
To Fix: Assign the variable in the Inspector



Null Reference Exception

Unity has been asked to access something that doesn't exist

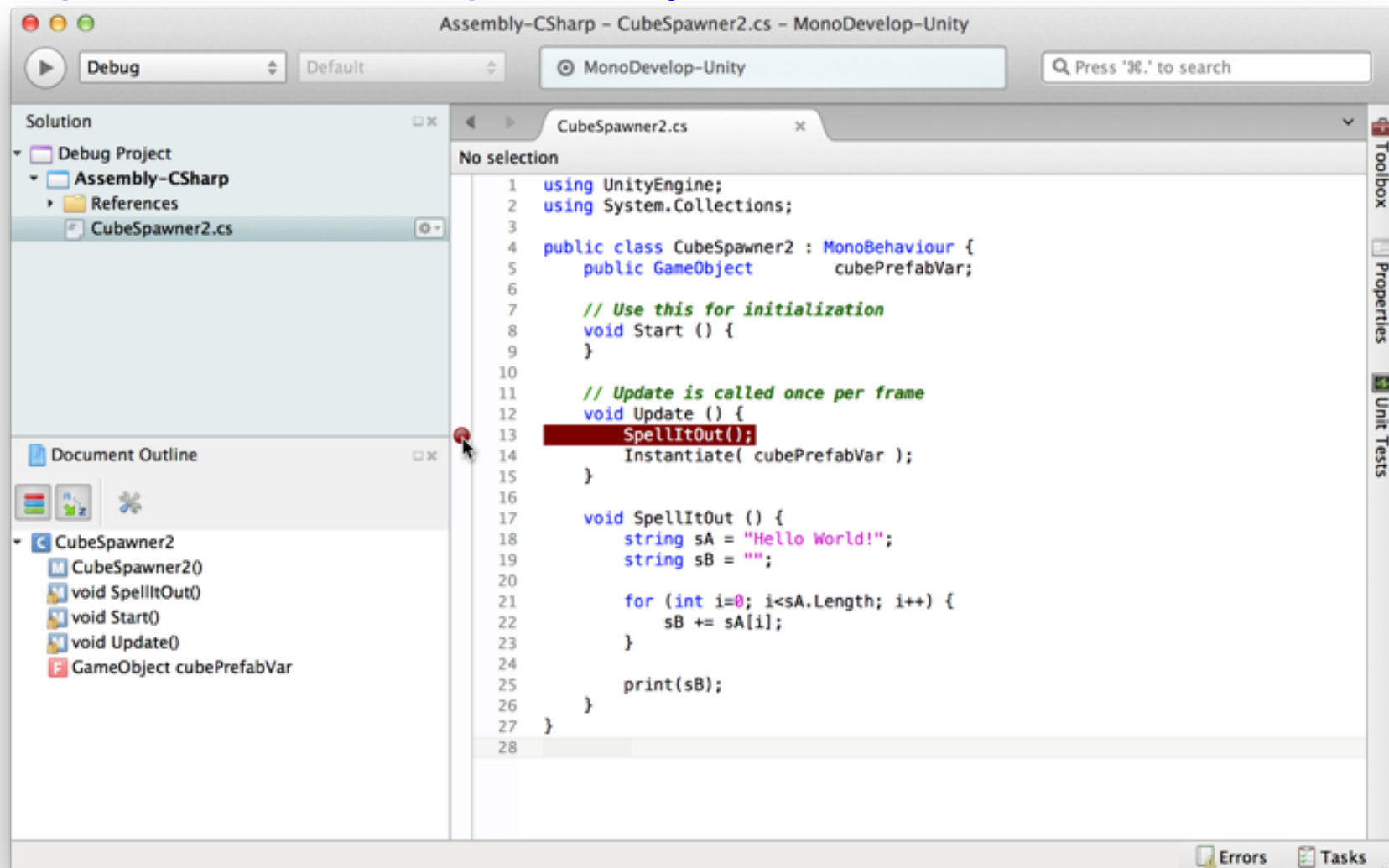
```
7 void Start () {  
8     GameObject[] goArray = new GameObject[10];  
9     print (goArray[5].transform.position);  
10 } // on line 9, goArray[5] is null, so it has no transform
```



- Error can only tell you the line number
- These are difficult to debug!

Stepping Through Code

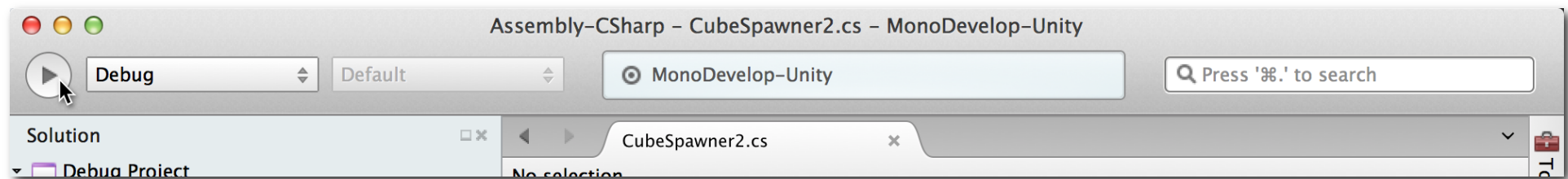
Step 1: Set a Breakpoint in your code



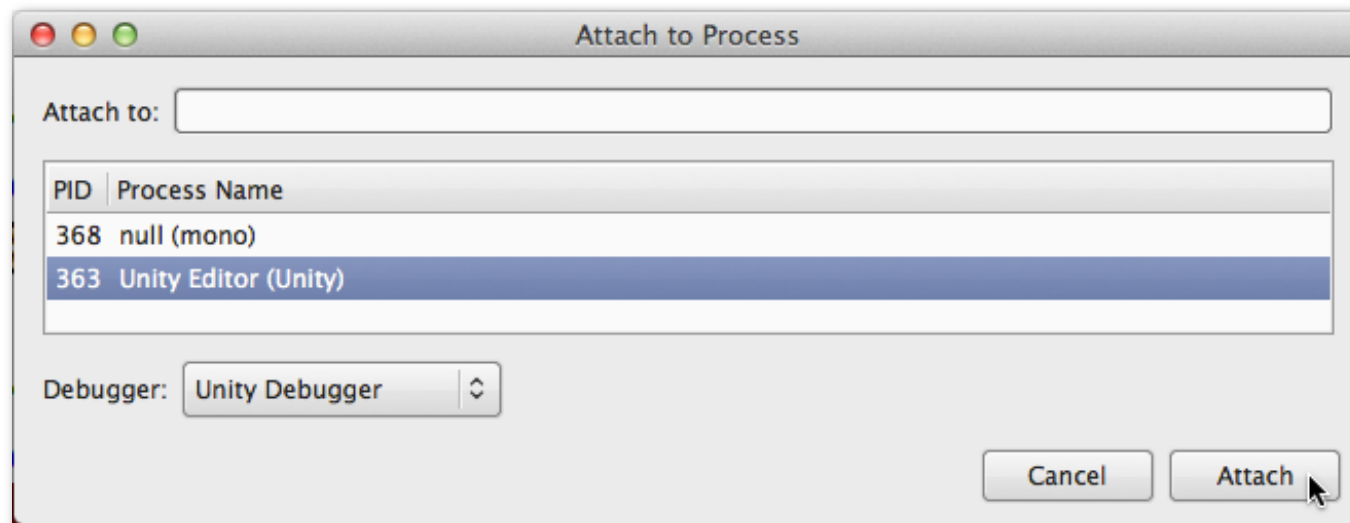
Stepping Through Code

Step 2: Attach the Debugger to the Unity process

- Much more detail in the book (about a potential bug)
- Click the Attach to Process button in MonoDevelop



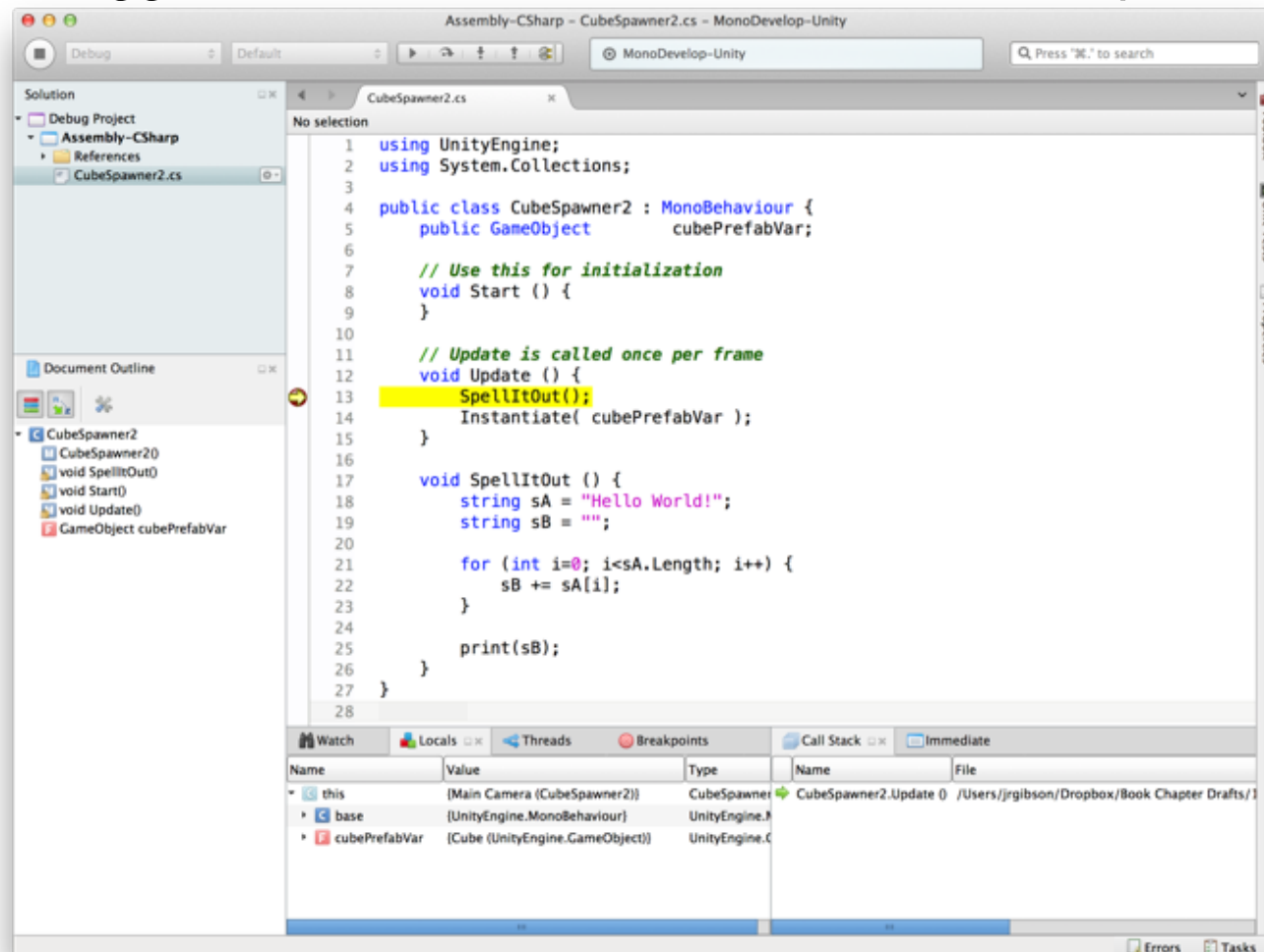
Choose *Unity Editor (Unity)* from the process list & click *Attach*



Stepping Through Code

Step 3: Click Play in Unity

- The Debugger will halt code execution at the Breakpoint



Stepping Through Code

Step 3: Click Play in Unity

- The Debugger will halt code execution at the Breakpoint
- Unity will be completely frozen while the Debugger is halted
 - This means you cannot switch back to the Unity process
- Important buttons at the top of the Debugger window



Stepping Through Code

Each controls the Debugger's execution

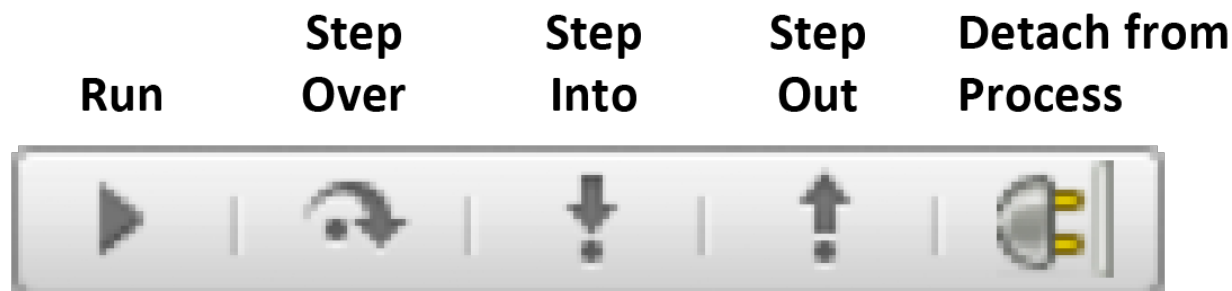
Run – Continues playing the project until another breakpoint is hit (If Run doesn't advance to the next frame, switch back to Unity)

Step Over – Continues to the next line, stepping over function calls

Step In – Continues to the next line, stepping into function calls

Step Out – Exits the current function but continues debugging

Detach Process – Stops debugging altogether



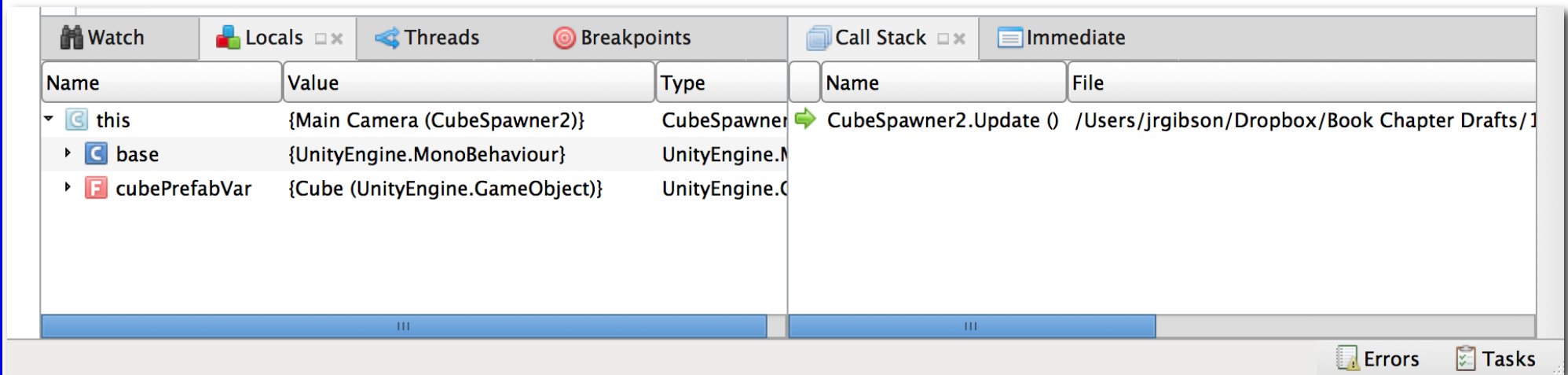
Watching Variables

Panes at the bottom of MonoDevelop have more info

Locals - Allows you to see all local variables (this is a reference to the current class instance)

Watch - Allows you to enter specific variables to watch

Call Stack - Shows you which functions have been called to get to this point in the code - Click a function to jump to its local scope



Summary

Debugging is one of the most important processes in coding

- The MonoDevelop Debugger is one of the most powerful tools
- It's also surprisingly easy to learn to use
- The Debugger can also help you understand complex code
- Use it on the code from the book if you're ever confused

You can also code for Unity using Microsoft Visual Studio, which has its own debugger

Some things to watch ?

<https://www.youtube.com/watch?v=CeXAiaQOzmY>
HOW TO MAKE A 2D CHARACTER CONTROLLER IN
UNITY - EASY TUTORIAL

2D Movement in Unity (Tutorial)

<https://www.youtube.com/watch?v=dwcT-Dch0bA>

2D Animation in Unity (Tutorial)

<https://www.youtube.com/watch?v=hkaysu1Z-N8>