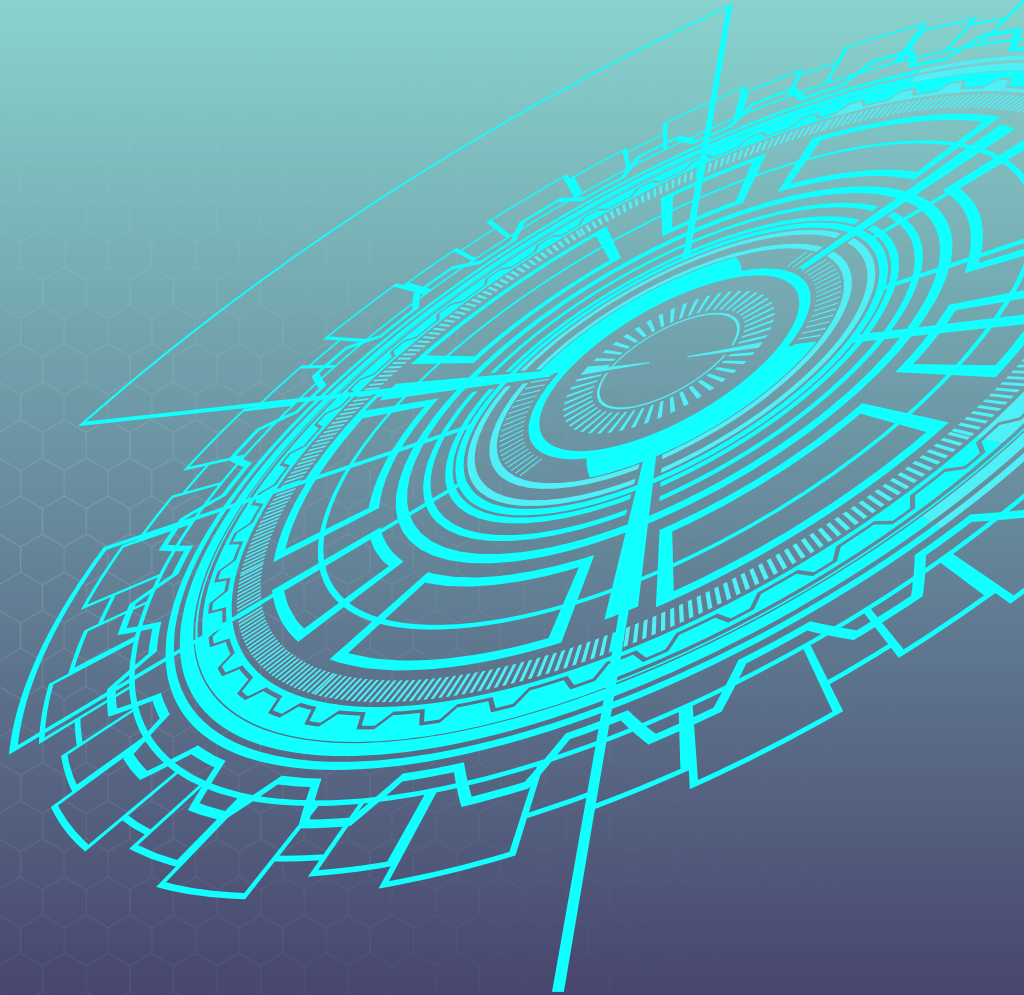


Week 15

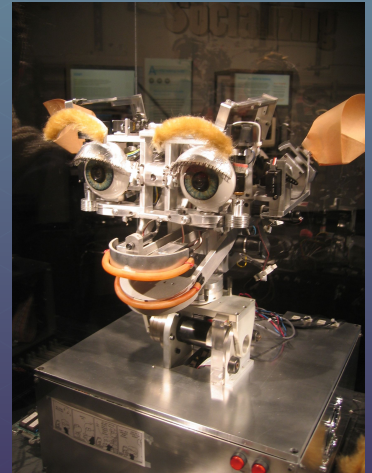
More on the Unity Game
Editor and Workshop 2



Katie Tooher and Carmen Seda

WIT: Cynthia Breazeal

- Born in Albuquerque, New Mexico
- B.S. in electrical and computer engineering from University of California–Santa Barbara in 1989
- S.D. in electrical engineering and computer science from Massachusetts Institute of Technology (MIT) in 2000
- Developed Kismet
 - Highly expressive humanoid robot
 - <https://www.youtube.com/watch?v=8KRZX5KL4fA&feature=youtu.be>
- Currently Director of the Personal Robots Group under the aegis of MIT's famed Media Lab



Warm-Up



- Write a C# program that takes a user input and then prints every odd number starting with 1 until it reaches the user input
- Ex: User input = 5, print "1 3 5"
- Ex: User input = 6, print "1 3 5"



Last Week We Learned..

- **Basics of the Unity Editor**
 - **Hierarchy View**
 - **Project View**
 - **Scene View / Game View**
 - **Inspector View**
- **Terrain and Raw Heightmaps**
 - **How to create and edit terrain**
 - **Adding color and other features / trees / grass**
 - **Added Fog using simple lighting effects**
- **Continued Workshop 1 -- Simple Racing Game**



Finish Workshop I.

- To finish our simple racing game we must now add the c# code to our game to add four things as well as a character!
 1. **Game Manager:** a gatekeeper that controls the different states of the game
 - a. **Example:** If you win or finish a game, the game manager takes a player back to the spawn point to play again
 2. **Water Hazard Detector:** detects if you have fallen into the water
 - a. We create a game object and then add player respawn code to it
 3. **Finish Zone:** Ends the game if reached
 4. **Spawn Point:** where the player will start in your game



Adding Code to your Unity Games

- We use code in our games to add customization to our Game Objects and assets
- Scripts: “are files of codes that are used to define complex or nonstandard behaviors for objects”
- Unity allows you to write scripts in either C# or JavaScript but support for the JavaScript language is being phased out
- The main two libraries we use in Unity Game scripting are
using UnityEngine;
using System.Collections;



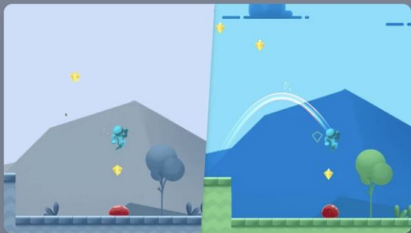
Adding Code to your Unity Games

- When working with scripts make sure to create a folder in your project view assets folder labeled scripts and store them in there before you add them to game objects
- You can add scripts to your project in 2 ways:
 1. Drag existing scripts into the project view
 2. Create new scripts in your project by right-clicking in the project view and selecting **Create > C# Script**
- To add scripts to a Game Object simply drag the script out of project view and onto the game object in the hierarchy view



Workshop 2

- **There are many online tutorials for Unity, today for our 2D game workshop we will be downloading a 2D asset package from the store and following along with a tutorial that shows more basics about the editor and how to create dynamic 2D games using assets from the store**
 - **We will learn more about assets next week**
- **<https://learn.unity.com/project/2d-platformer-template>**
 - **Navigate to this link or search “Unity 2D Platformer Microgame Tutorial”**
 - **Download and Import the package to Unity**
 - **Navigate to the Tutorials folder in the assets folder and it will automatically start the tutorial in your Unity editor**



Platformer Microgame

🔖 1107

This Microgame Template is a classic 2D platform game that you can mod and make your own. Check out the Creative Mods to tweak the project and add your own levels, while learning the basics of Unity. Viewing from the Learn tab in the Unity Hub? Click Download Project > Open Project to automatically open it in Unity. Viewing from the Unity Learn website? Simply go to the Learn tab in the Unity Hub and search for this Microgame, or manually import it via the Asset Store link below.

[🔗 Platformer Microgame](#)

