Week 1: Introduction & Overview

Welcome to Week 1

Introduction to the Emerging Analytics Center's Unity 101 course

Slides: Week 1: Introduction and Overview

- 1.1 – Summary of the Unity 101 Course
- 1.2 – Unity 101 Course Goals
- 1.3 – Unity 101 Team Members and Contact Information
- 1.4 – Unity 101 Course Schedule
- 1.5 – Assignment

1.1 – Summary of the Unity 101 Course

Unity 101 is a program made to help students get started in developing Unity applications similar to those developed by the Emerging Analytics Center. The George W. Donaghey Emerging Analytics Center (EAC) is an academic department within UA Little Rock that focuses on research and development of augmented/virtual/mixed reality, immersive visualization, and more! To learn more about the EAC, visit our website here.

This program will walk students through the process of creating an application using the Unity game engine. To see an example of a game application, please see the following example: Flappy Birds game.

Each week, students will visit that week's page. Students will complete all of the reading and assignments for that week. Also, slides for each week can be downloaded from the web site.

1.2 – Unity 101 Course Goals

Course Goals
Learn to use project management tools such as Git
Learn the basics of Unity
Create a 2D game from scratch

1.3 – Unity 101 Team Members and Contact Information

Below is a list of the EAC staff members and students to contact if you have any questions.

EAC Staff Members

Name	Contact info
Tom Coffin	tacoffin@ualr.edu
Jason Zak	jrzak@ualr.edu

EAC Students

Name	Contact info
Kimari Watson	klwatson3@ualr.edu
Willam Nickols	wanickols@ualr.edu
Carlos Ochoa	cjochoa@ualr.edu
J'kyla Miller	jmiller6@ualr.edu
John Baglio	jrbaglio@ualr.edu

1.4 – Unity 101 Course Schedule

Week	Topic
1	**Introduction & Overview** Slides: • Summarize class • Identify class goals • Preview some of the tools that will be used in the class **Assignment** • Play flappybird.io to understand the mechanics of flappy bird • Make Unity Account • Microsoft account for Visual Studio • Install unity hub • Link Unity Account • Install latest version of unity (Version 6)

2	**Unity Basics (Part 1)**
	
	• Go over the Unity version 6 Interface
	• Discuss GameObjects
	• Discuss Components
	
	Assignment
	
	• Microsoft account for Visual Studio
	• Install unity hub
	• Link Unity Account
	• Install latest version of unity (Version 6)
	• Follow along with slides
	
	Outline:
	
	• 3.1 – GameObjects
	  ○ Components
	  ○ Example
	  ○ Scenes and Windows Intro
	
	Slides: link here as slides are made
	
	Original PowerPoint reference
	
	Unity basics – slide 10 – 29
	
	• Components 12– 27
	• scenes/windows 28
	
	While unity is downloading might go through a section of slides

3	**Unity Basics (Part 2)**
	
	Outline:
	
	• Parts of unity
	  ○ scene view
	  ○ game view
	  ○ Hierarchy
	  ○ Inspector
	  ○ Folder structure
	• First code along here
	
	Assignment
	
	• Follow along with slides
	• Unity along
	  ○ Importing bird
	  ○ Adding components to bird
	  ○ Tap screen

4	**Unity Basics (Part 3)** **Flex for the first two parts**

5	**Objects and C# Classes (Part 1)**
	
	• Introduce data types, numbers, and variables
	• Demonstrate how the Inspector shows variables
	  ○ Mention how variables update in Play mode
	• Demonstrate how to manipulate variables in code
	• Introduce the MonoBehaviour functions
	  ○ Awake, Start, Update and OnDestroy
	  ○ GetComponent
	• Demonstrate how to send messages to the Unity Console
	  ○ *Debug.Log*, *Debug.LogWarning*, *Debug.LogError*
	
	Assignment:
	
	• Create a script that displays their name and age in the Unity Console, then add it to a GameObject
	• Create a script that displays a variable's value to the Unity Console, modifies it and then displays the new value

6	**Objects and C# Classes (Part 2)**
	
	• Introduce Prefabs
	  ◦ Drag and drop GameObject onto Assets folder
	• Introduce Object/GameObject functions
	  ◦ Instantiate – creating objects from Prefabs
	  ◦ Destroy – deleting objects with an optional time delay
	  ◦ Find – finding objects by name
	• Introduce logic and conditionals
	• Introduce responding to user input
	  ◦ Use legacy input system, *UnityEngine.Input*
	  ◦ Modern *Input System* package is more modular, but can be difficult to understand
	
	Assignment:
	
	• Create a script that demonstrates usage of the GameObject functions (e.g. spawning an object from a prefab, making an object despawn after X seconds, etc.)
	• Create a script that listens for input and does something (e.g. sending a message to the Unity Console when pressing the 'A' key)

7	**Objects and C# Classes (Part 3)**
	
	• Introduce Rigidbody2D
	  ◦ Mention *gravityScale* and *velocity*
	• Introduce Collider2D
	  ◦ Trigger vs non– trigger colliders
	  ◦ OnCollisionEnter2D, OnCollisionStay2D, OnCollisionExit2D
	  ◦ OnTriggerEnter2D, OnTriggerStay2D, OnTriggerExit2D
	
	Assignment:
	
	• Create a scene that includes an object with a Rigidbody2D and a script which listens for input and moves the object
	• Create a scene that includes a collider and an object which does something when it enters the collider

8	**Canvas** • Introduction to Graphical User Interface (GUI) • Demonstrate adding text to the screen • Demonstrate manipulating text through code **Assignment** • Add text that will be displayed on the screen
9	**Programming (Part 1 – Player)** • Introduce communication between components through code • Add and adjust the Rigidbody through code • Introduce listening for a button press **Assignment** • Make a code where the bird asset is able to fly when the spacebar is pressed.

10	**Programming (Part 2 – Obstacles)**
	
	Outline:
	
	• Make wall prefabs
	  ○ Make Parent
	  ○ make obstacles
	  ○ make gap
	  ○ create prefabs
	  ○ adjust hitboxes
	  ○ create tags
	• Programming the wall part 1
	  ○ Unity functions
	  ▪ Start
	  ▪ Update
	
	Assignment
	
	since he programming part will be a lot here, I'd start with the unity functions and their related variables, then do generate next week

11	**Programming (Part 3 – Obstacles)**
	
	Outline:
	
	• Programming the wall part 2
	  ○ Genrate Obstacle functions
	  ▪ Instatiate
	  ▪ add to list
	  ▪ handle x and y pos
	  ▪ handle speed
	• Make the wall move
	  ○ Create controller in scene
	  ○ link all the values
	  ○ pray it works
	  ○ play with time and speed values

12	**Scoring and Ending the Game**
	
	Outline:
	
	• Create Score counter
	  ○ Make Image
	  ○ Adjust with rect tools
	  ○ Color and font align
	• Create score script
	  ○ Add to score function
	  ○ Edit Player to call addToScore
	• Create end game effect
	  ○ edit MovingObstacle script
	  ▪ Make so can turn off movement
	  ○ edit ObstacleController Script
	  ▪ Make so skips the nonMoving walls
	  ○ edit player script to ontrigger call ongame over
	  ▪ Use tags to confirm collision and end game
	• Create game over screen
	  ○ make image and text, button
	  ○ disable parent
	  ○ link event to the activation of game over
	  ○ link player movement to game over event
	• Adjust score script
	  ○ Add finalscoredisplay
	  ○ link to game over event
	• Game restart script
	  ○ Restart scene function
	  ○ Add to onclick
	
	Assignment
	
	• Change the canvas, so the score counter increases as the player goes through the wall obstacles.

1.5 – Assignment

- Play http://flappybird.io to understand the mechanics of flappy bird
- Make Unity Account