

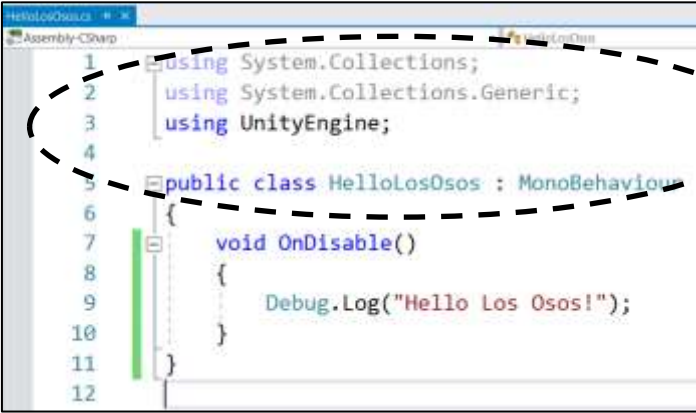


CORNELL NOTES – COMPUTER PROGRAMMING & GAME DESIGN I

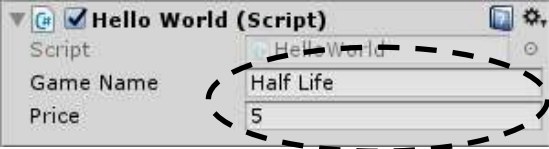
	Topic/Objective:	Name:
	Level 5:	Class/Period:
	Intro to Scripting	Date:

Level Objective:

To understand basic scripting for creation of games in Unity including understanding variables, operators, conditionals, and arrays

Questions:	Notes/Answers/Definitions/Examples/Sentences:
	<p><u>Before We Script...</u></p>  <ul style="list-style-type: none"> • What are scripts? <ul style="list-style-type: none"> • Files of codes that define behaviors for objects • Two ways to create a New Script File: <ol style="list-style-type: none"> 1. Inspector Window > Add Component > New Script 2. Right Click with mouse > Create > C# Script • It's a good idea to create a Scripts folder in the Assets folder of your project • In order to properly run the script, it must be attached to a GameObject <ul style="list-style-type: none"> • Click & Drag the script onto the GameObject or into the Inspector Window • To open Microsoft Visual Studio → Double click the Script file in Projects <p><u>Microsoft Visual Studio Items to Remember</u></p> <ul style="list-style-type: none"> • Anatomy of a Basic Script (circle the area where NOT to edit or delete)  <ul style="list-style-type: none"> • All your code will go in between the curly braces { } <ul style="list-style-type: none"> • Any code between curly braces is known as a block • You must have the same number of opening curly braces as closing ones • Red wavy lines in your code indicates there is an error that needs attention • View the text output & errors in your game with Unity's Console window <p><u>Working with Variables</u></p> <ul style="list-style-type: none"> • What are they? → Placeholders for data that can be used later • You create with the following syntax: <variable type> <name>;

CORNELL NOTES – COMPUTER PROGRAMMING & GAME DESIGN I

Questions:	Notes/Answers/Definitions/Examples/Sentences:																														
	<ul style="list-style-type: none"> Main types of Variables we can assign: <table border="1" style="margin-left: 20px;"> <tr> <td>int</td> <td>Short for integer; stores positive and negative numbers</td> </tr> <tr> <td>float</td> <td>Stores decimal values; always written with an "f" after them</td> </tr> <tr> <td>string</td> <td>Holds entire words or sentences (use double quotations marks)</td> </tr> <tr> <td>bool</td> <td>Short for Boolean; stores true or false</td> </tr> </table> Rules of Naming Variables <ul style="list-style-type: none"> No Spaces camelCasing – writing compound words where each word begins with a capital letter (capitalization is very important in coding!) Making a Variable Public <ul style="list-style-type: none"> Allows you to change values in Inspector Why do this? → allows you to change values easier while testing the game Variables can only be used in the blocks in which they are created 	int	Short for integer; stores positive and negative numbers	float	Stores decimal values; always written with an "f" after them	string	Holds entire words or sentences (use double quotations marks)	bool	Short for Boolean; stores true or false																						
int	Short for integer; stores positive and negative numbers																														
float	Stores decimal values; always written with an "f" after them																														
string	Holds entire words or sentences (use double quotations marks)																														
bool	Short for Boolean; stores true or false																														
																															
	<p>Working with Operators</p> <ul style="list-style-type: none"> What are they? → symbols allowing you to perform changes to data Arithmetic Operators (mathematical operations on operators) <table border="1" style="margin-left: 20px;"> <tr> <td>+</td> <td>Adds 2 numbers; also combines strings together</td> </tr> <tr> <td>-</td> <td>Subtraction</td> </tr> <tr> <td>*</td> <td>Multiplication</td> </tr> <tr> <td>/</td> <td>Division</td> </tr> <tr> <td>%</td> <td>Modulus; divides but returns the remainder value instead</td> </tr> </table> <ul style="list-style-type: none"> Assignment Operators (assigns values to variables) <table border="1" style="margin-left: 20px;"> <tr> <td>=</td> <td>Assigns value</td> </tr> <tr> <td>++</td> <td>Incremental operator (increases a number by 1)</td> </tr> <tr> <td>--</td> <td>Decremental operator (decreases a number by 1)</td> </tr> </table> <ul style="list-style-type: none"> Equality Operators (compares 2 values) <table border="1" style="margin-left: 20px;"> <tr> <td>==</td> <td>Returns true only</td> <td>>= <=</td> <td>Greater than equal to</td> </tr> <tr> <td>> <</td> <td>Greater than; Less than</td> <td>!=</td> <td>Not equal to</td> </tr> </table> <ul style="list-style-type: none"> Logical Operators (combine two or more Boolean values) <table border="1" style="margin-left: 20px;"> <tr> <td>&&</td> <td>AND Operator; determines if both values are true</td> </tr> <tr> <td> </td> <td>OR Operators; determines if either of the values is true</td> </tr> <tr> <td>!</td> <td>NOT Operator; opposite of the Boolean value</td> </tr> </table>	+	Adds 2 numbers; also combines strings together	-	Subtraction	*	Multiplication	/	Division	%	Modulus; divides but returns the remainder value instead	=	Assigns value	++	Incremental operator (increases a number by 1)	--	Decremental operator (decreases a number by 1)	==	Returns true only	>= <=	Greater than equal to	> <	Greater than; Less than	!=	Not equal to	&&	AND Operator; determines if both values are true	 	OR Operators; determines if either of the values is true	!	NOT Operator; opposite of the Boolean value
+	Adds 2 numbers; also combines strings together																														
-	Subtraction																														
*	Multiplication																														
/	Division																														
%	Modulus; divides but returns the remainder value instead																														
=	Assigns value																														
++	Incremental operator (increases a number by 1)																														
--	Decremental operator (decreases a number by 1)																														
==	Returns true only	>= <=	Greater than equal to																												
> <	Greater than; Less than	!=	Not equal to																												
&&	AND Operator; determines if both values are true																														
 	OR Operators; determines if either of the values is true																														
!	NOT Operator; opposite of the Boolean value																														

