

CORNELL NOTES – COMPUTER PROGRAMMING & GAME DESIGN I



Topic/Objective:
Level 6:
Scripting Control
Flow

Name:
 Class/Period:
 Date:

Level Objective:
How to take data that has been collected (through variables and arrays) and to do something with it including through the use of loops, conditional statements, and SWITCH statements.

Questions:	Notes/Answers/Definitions/Examples/Sentences:
	<p>Introduction to Loops</p> <ul style="list-style-type: none"> What is it? → The ability to repeat a block of code X amount of times until some condition is met Why use these? → If a programmer needs to use the same lines of code over and over again, using loops would save time and effort This concept is also known as iteration. <p>WHILE Loops</p> <ul style="list-style-type: none"> What is it? → A loop that is repeated as long as the expression is TRUE <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Initialization</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Testing the condition</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Increment/Decrement (or "break")</div> <div style="border: 1px solid black; padding: 10px;"> <pre>void Enable () { int i = 10 while (i < 10) { Debug.Log (i); i++; } }</pre> </div> </div> <p>DO WHILE Loops</p> <ul style="list-style-type: none"> What is it? → A loop that repeats until an expression becomes FALSE Difference between WHILE and DO WHILE? <ul style="list-style-type: none"> Statement will execute at least one time (before possibly terminating) <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <pre>void Update () { int i = 0 do { Debug.Log (i); i++; } while (i < 10) }</pre> </div> <p>FOR Loops</p> <ul style="list-style-type: none"> What is it? → A loop that runs for a preset number of times <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;"> <p><u>value</u></p> <p><u>condition</u></p> <p><u>increment or decrement</u></p> </div> <div style="border: 1px solid black; padding: 10px;"> <pre>void Update () { for (int i = 0; i <= 10; i++) { Debug.Log (i); } }</pre> </div> </div>

CORNELL NOTES – COMPUTER PROGRAMMING & GAME DESIGN I

Questions:	Notes/Answers/Definitions/Examples/Sentences:
	Conditional Statements
	<ul style="list-style-type: none">What is it? Statement based on certain conditions where if "true", a certain piece of code is executed
	➔ if (condition goes here)
	Use this when you first start your conditional statement
	else if (condition goes here)
	Use this when multiple conditions exist
	else (condition goes here)
	Use this when you have one last default condition to add to "IF"
	<ul style="list-style-type: none">What can your conditions include?
	<ul style="list-style-type: none">○ A == B means "If A is equal to B"
	<ul style="list-style-type: none">○ A != B means "If A is different than B"
	<ul style="list-style-type: none">○ A <= B means "If A is less than or equal to B"
	<ul style="list-style-type: none">○ A >= B means "If A is greater than or equal to B"
	<ul style="list-style-type: none">○ A = B && B = C means "If A is equal to B and B is equal to C"
	<ul style="list-style-type: none">○ A = B B = C means "If A is equal to B or B is equal to C"
	Methods
	<ul style="list-style-type: none">Also known as functions
	<ul style="list-style-type: none">What are they? A script that performs an operation or function by giving your game objects custom behaviors.
	<ul style="list-style-type: none">You've already worked with methods and didn't even know it. They include void Start, void Update, void Enable, void Disable, etc.
	<ul style="list-style-type: none">Once a method is written, you can then use the method later by calling it
	<ul style="list-style-type: none">○ Ex. someMethod ()
	<ul style="list-style-type: none">Structure:
	<pre><visibility><return type><name>(<parameters list> { <Inside the method's block> }</pre>
	1. Visibility – public or private (just like a variable)
	2. Return Type:
	a. If you do not want a value returned, use void
	b. If you want to return a variable back to whatever code called it, use string, int, bool, etc.
	3. Name – name of your method
	a. Must start every method with a capital letter with no spaces in the name
	4. Parameter List – list of variables that's part of the methods declaration
	a. Multiple variables are separated by commas
	5. Method Block – where the code of the method goes

