

*[Click to Slide 1 - Title Screen]*

Good Evening. My name is Nicholas Farkash and my topic is Video Game Development. Video Game Development, as the name suggests, is the process by which a video game is developed, or created. There are a few words and phrases that you should know before I get started.

*[Click to Slide 2 - Key Phrases]*

The code is the set of words that gets read by the computer. It contains instructions that are carried out to perform tasks. The code is located in a text file known as a script.

C# is the programming language I use for my project. Simply put, it's the language the code is written in. C# was created by Microsoft as part of their .NET initiative with the ultimate goal of creating an easy-to-understand programming language.

A game engine is the software used to combine the elements of a video game together. It takes the graphics, code, and sound and brings them all together. The game engine I am using for my project is Unity.

*[Click to Slide 3 - Video Game Development Team]*

Creating a video game takes a long time and is a lot of work to be done by one person. Most video games are developed by multiple teams of people who focus on one specific task.

Those groups will then put their individual parts of the game together in a game engine to produce the game. Programmers write the code, Graphic Designers create characters, scenery, and objects, Sound engineers create sound effects and music for the game, Level Designers build the levels, and Testers play the game to check for any issues. For a large company, a single team can be composed of hundreds of people. When all the groups are put together, the number of people can easily reach into the thousands, and the game will still take a few years to be fully developed.

*[Click to Slide 4 - Chart]*

Here is a more detailed chart of some of the jobs in video game development.

*[Click to Slide 5 - Process]*

For my presentation, I will be breaking down the process of video game development into six categories. I believe these to be the six most important parts of the process. These are Establishing Controls, Writing the Code, Animating, Creating Graphics, Creating Sounds, and Level Construction.

*[Click to Slide 6 - Controls]*

Establishing controls in Unity is easy because there is very little work involved. By inserting one line of code, “Using UnityEngine”, the code gains access to Unity’s default controls.

From there, I can change the controls to my preference by adjusting the keys on my keyboard or configuring a controller.

*[Click to Slide 7 - Code]*

Code is written in a text file. When imported into Unity, it is most often referred to as a script. It contains information about how objects interact with one another, much like how a play script works.

A Character Controller Script is the foundation for the controls. It uses a line of code, “using UnityEngine;” to access Unity. From there, Unity will refer to the Character Controller Script as the game is being played.

Another major script is the Player Movement Script. This is the code that will move the character.

There are 3 other scripts in my game. 2 of the scripts are for a main menu and a pause menu and the third respawns the character.

*[Click to Slide 8 - Character Controller]*

This is a Character Controller I got from a tutorial. I did not write it myself. It is not recommended that beginners create their

own character controllers because there is too much coding involved. Instead, they should spend their time building the game and writing smaller scripts. With that being said, the Character Controller is responsible for giving the player the ABILITY to move. However, it doesn't actually move the character. That is done in the Player Movement Script.

*[Click to Slide 9 - Main Screen]*

Off to the right, there is a tab called "Inspector". It lists the object that is currently selected, in this case, the Player, and all of the components that are assigned to it. Here is the Character Controller. It contains the script, which is the code on the previous slide. It gives some options, such as how high the character will jump, how fast he will move if he is crouching, and if the character can be controlled mid-air. These "checks" allow the character to perform actions. The Ground Check is used to jump and crouch, the Ceiling Check lets the character get up after crouching, and the Crouch Disable Collider removes his top collider to allow him to move under platforms.

*[Click to Slide 10 - Player Movement Script]*

This is the Player Movement Script. As the name suggests, it is responsible for moving the player. Specifically, it controls horizontal movement, jumping, crouching. It also contains their respective animation codes, which tells Unity to play the

animations. The Movement Script is much smaller than the Character Controller because the Character Controller does most of the work. The Player Movement Script will be referenced by the Character Controller Script, which will then take care of the rest.

*[Click to Slide 11 - Main Screen]*

In the Inspector, you can see that there isn't much under the Player Movement Script component. There is the run speed, which is self-explanatory, and three slots, Script, Controller, and Animator. These are references used by the Character Controller to do most of the work. The script is the Player Movement Script. The Controller is the Character Controller Script, which was previously shown. The Animator holds all of the animations in one place for easy reference. Putting each of these in their respective slots allows them to interact with each other.

*[Click to Slide 12 - Animation]*

So what are Animations? Animations are the visual effects performed by an object in the game. Most often, these animations are performed on the player. These animations make the character run, jump, and crouch. Each animation is comprised of several pictures. The pictures are cycled through quickly to make the character look like it's moving. Animations

are created in the Animation window of Unity. This is not to be confused with the Animator Window.

*[Click to Slide 13 - Animator]*

The Animator controls which animations play depending on what is happening in the game. It creates transitions between animations that switch when certain conditions are met. For example, a walking animation would be played if speed was between 0 and 10, while a running animation would be played if the speed was greater than 10. These are the conditions, which determine what animation is played. In order to switch between the walking and running animations, a transition is needed. Each of these arrows is a transition, and each transition allows the character to switch from one animation to another, following the direction of the arrow. If all conditions of a transition are met, the animations will switch.

*[Click to Slide 14 - Sound and Graphics]*

Now that the gameplay is finished, the next up is sound and graphics. My game music was created in Bosca Ceoil. It is a program that allows the user to create music. It contains many different instruments and allows for innumerable songs to be

composed. To edit the sound clips, I used WavePad Audio Editor. It allowed me to change the sound clip's pitch, timing, and volume.

Most of my personal graphics were created in Microsoft Paint. I created the flowers, grass, rocks, and platforms. However, creating a character is a little more complex, as it needs animation and has to look good because it is the centerpiece of the game. Fortunately, Unity makes things very easy and comes with an Asset Store.

*[Click to Slide 15 - Asset Store]*

For my game, I am using two packages from the Asset Store. The first one was for my character and his animations. The second was an addition to the camera, which allows it to follow the character as he moves and makes the camera smoother.

*[Click to Slide 16 - Level Construction]*

Building levels in Unity is fairly simple. It just involves a bunch of clicking to import assets and add components. After the level has been constructed, it is important to test the level to make sure it works properly.

Testing is an important part of video game development. Programmers are people too, and they often make mistakes that can result in glitches, making a game unplayable. Testers play

through the game to find mistakes and provide feedback, improving the quality of the game.

*[Click to Slide 17 - Asset Screen]*

Importing Assets is very easy in Unity. Among the folders that will appear as the game gets more complex, there will be one labeled Assets. This is where personally made objects will go, but not the ones from the Asset Store. Downloaded assets are placed in individual folders to keep all assets in one place. In the Scene view, you just click, drag, and drop the asset onto the screen and it appears.

*[Click to Slide 18 - Main Screen]*

From there you can move it around and add components. One such component is a collider, which can be added to make two objects hit each other. Another common component is Rigidbody, Unity's physics system. It adds gravity, mass, and other physics-related properties to an object. It allows the character to fall back to the ground after the player has jumped.

When all of these components, Code, Controls, Animation, Sound, Graphics, and Level Construction are put together, the end result is a video game.

My game was available before the presentations began outside the cafeteria and it will be available for you to play [point to where the game] off to the side after all of the



presentations are over.

*[Click to Slide 19 - Future]*

Over the Summer, I will be working at Hofstra University with my mentor, Dr. Edward Currie. Thank you.