[*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 created. Under the mentorship of Dr. Edward Currie, I created a video game that I will present tonight. Before I get started, there are a few words and phrases that you should know.

[*Click to Slide 2 - Key Phrases*]

A **game engine** is the software used to combine the elements of a video game into one location. The game engine I am using for my project is Unity.

A **script** is a text file that contains the code, which is a series of words that gets read by the computer to perform tasks.

A **scene** is a segment of the entire project. Scenes can be changed through a script.

Assets are the foundational elements of a game. They are the graphics, sounds, and scripts of the game.

[*Click to Slide 3 - Process*]

The process I used this year can be divided into the following five categories: Assets, Scenes, Level Construction, Scripts, and Features

[*Click to Slide 4 - Assets]*

I used several assets for my project. **Unity Hub** is a program that, among other things, has tutorials for developers to learn from. It also has creator kits, which are bundles of assets for developers to use. I used the **"Creator Kit: RPG"** asset in addition to my own **personal** graphics and scripts.

[*Click to Slide 5 - Scenes]*

My game contains 6 scenes. The two primary scenes are the **"Overworld"** and the **"BattleSystem"** Scenes. The remaining 4 scenes are the "MainMenu" and 3 additional scenes that serve as transitions. The **Build Index** lists all scenes in a project. It can be interacted with through **functions** located in a script.

[*Click to Slide 6 - Level Construction]*

My game was created using **Tilesets**, which is a group of smaller images placed to create a larger image. The image on the right is the tileset used in my game. There are 4 **Layers** of the tileset, and each layer can have 1 tile active.

The **Camera** must adjust to the layout of a level. In my game, the camera follows the player. However, this can be problematic if the player moves inside a building.

[*Click to Slide 7 - Scripts]*

Scripts can solve this issue. There is a script called "**FadingSprite**" that reduces the opacity of the roof image to

make it transparent. It is the most common script present in my game. Other crucial scripts in the game are the "**SceneManagement**" "**MainMenu**" and "**BattleSystem**" scripts which control their respective parts of the game.

[*Click to Slide 8 - Features]*

Some features of my game include an **Interactive Questline**, an **Inventory** system, and the **Battle System**. These are what make the game unique, so they will be explored more closely.

[*Click to Slide 9 - Inventory]*

The inventory system stores **Inventory Items** the player obtains. These can be **Collected** by walking over them, as seen in the GIF to the right, or by receiving them after completing a quest. When the player obtains the inventory item, the item is removed from the game world and placed in the player's inventory.

[*Click to Slide 10 - Interactive Quest Line]*

The Interactive Quest Line enables the player to converse with and receive quests from an **NPC**, which is a Non-Player Character. These are characters who advance the player's experience and cannot be controlled or manipulated.

The NPCs offer **Quests**, requiring the player to collect items and return them. The player will receive a reward for completing the quest.

When the player interacts with an NPC, they will be presented with dialogue boxes. Each dialogue box leads to a different response, creating **Dialogue Branches.** These create a multitude of different possible conversations the player can have with an NPC, and each conversation can result in a unique outcome.

[*Click to Slide 11 - Battle System]*

The Battle System is a Turn-Based Combat System where the player and an NPC take turns trading damage until one of them is defeated. It's modeled after my favorite video game series, Pokémon.

A **Battle State** is one of five conditions that the battle will be in. These are START, PLAYER TURN, ENEMY TURN, WON, and LOST. When a battle state is switched to, that battle state's functions are called.

The default state is start. After the start function runs, the Battle State switches to PLAYER TURN, allowing the player to select an action.

The **Actions** are the player's choices to attack the enemy, heal, or flee the battle.

Once an action is selected and the function runs its script, the Battle State switches to ENEMY TURN. The Enemy will

select an action, and it will return to PLAYER TURN. The Battle State cycles between PLAYER TURN and ENEMY TURN until one of the parties loses all health or the player leaves the battle. At that point, the Battle State will switch to either WON or LOST, depending on whether the player WON or LOST.

[*Click to Slide 12 - GIF]*

This is a GIF of the Battle System in progress. The **User Interface** consists of what the player can interact with. This includes the Boxes next to each party that displays the name, level, and health, as well as the information on the sides that lists the damage input, maximum health, and held item. There is a text box in the bottom left corner that tells the player what is happening in the battle, providing useful information to the player.

[*Click to Slide 13 - Future Slide]*

Next year, I will be attending Stony Brook University with a declared major in Physics and the goal of pursuing a career in Astrophysics.

I'd like to thank Mr. Burns, Mr. Truesdell, and the Holy Cross staff, as well as Mrs. Kenny and Dr. Paratore for working alongside me over the years. I'd also like to thank my peers for all the fun we had over the last few years. Finally, I'd like to

thank my mentor, Dr. Edward Currie, who was always available if I ever needed help. Thank you all, and stay safe.