**Final Project Documentation**

**Game Design Document**

SCENES:

* **Main**: first scene that shows up upon running the program. It has 3 buttons – play, which takes the user to the load\_game screen; how to play, which takes the user to the how\_to\_play screen; and quit, which closes the program if running through godot.
* **How\_to\_play**: explains the controls. It has 2 buttons – play, which takes the user to the load\_game screen; and back, which takes the user back to the main screen.
* **Load\_game**: allows the user to load the game if a save file exists. It has 3 buttons – new game, which creates a new save file and silently clears the old save file if one exists, as well as take the user to lvl\_screen; load game, which loads the existing save file, as well as take the user to lvl\_screen; and back, which takes the user back to the main screen.
* **Lvl\_screen**: allows the user to pick a level out of 3. The user can only advance to the next level if they’ve cleared the previous level. It has 4 buttons – lvl 1, which takes the user to the first level; lvl 2, which takes the user to the second level; lvl 3, which takes the user to the third level; and back, which takes the user to the load\_game screen.
* **Lvl1**: this is the first level of the game. The user must navigate to the end of the level w/o dying and find the star to be able to get to the next level, or lvl 2. Upon grabbing the star, the user is taken to the lvl\_screen.
* **Lvl2**: this is the second level of the game. The user must navigate to the end of the level w/o dying and find the star to be able to get to the next level, or lvl 3. Upon grabbing the star, the user is taken to the lvl\_screen.
* **Lvl3**: this is the third and last level of the game. The user must navigate to the end of the level w/o dying and find the star to finish the game. Upon grabbing the star, the user is taken to the lvl\_screen.

SPRITES:

* **Player**: the sprite is automatically in each level. The sprite can stand there in its idle animation, move around with the left/right arrow keys and play its run animation, jump with the spacebar and play its jump/fall animation, get hurt by colliding with an enemy sprite and play its hurt animation, or die and play its death animation (this takes the user back to lvl\_screen)
* **Enemies**: automatically in each level. the sprite can stand there in its idle animation, walk towards the player and play its walk animation, or die and play its death animation.
* **Coins**: these exist in every level. it has only one animation (idle), but it also can disappear when walked into by the player.
* **Stars**: these exist in every level and must be collected to proceed to the next level and eventually finish the game. Like the coins, it has only one animation (idle), but it also can disappear when walked into by the player. After being grabbed by the player and a second passes, the player will be taken back to lvl\_screen.

**Software Engineering Plan**

I created this on my own. I gave myself a week to work on it. This was the general timeline:

DAY 1: research game engines

DAY 2: research and install godot

DAY 3: find sprites for player, enemies, and collectables. create majority of godot scenes

DAY 4: animate player + work on player script (code inputs and play animations when necessary)

DAY 5: animate enemies (and enemy script), collectables (coin/star scripts)

DAY 6: add in tilemaps and create level designs

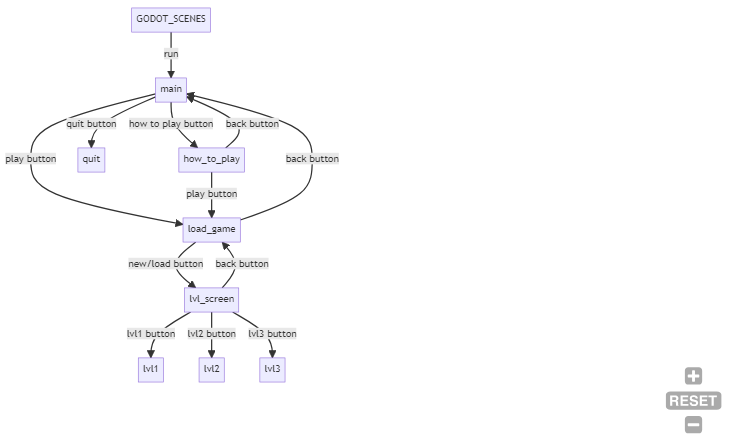
DAY 7: play test and turn in

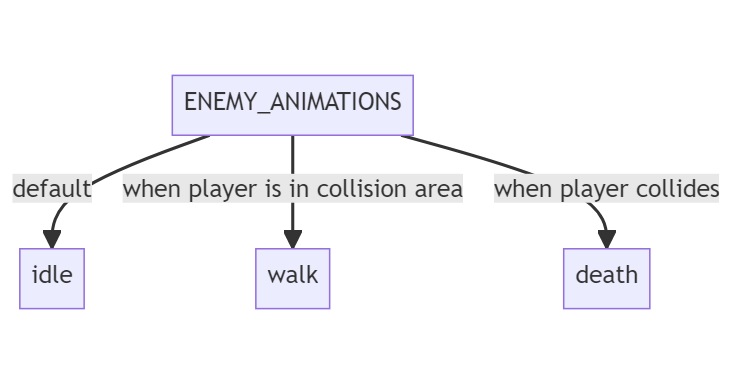
**State Transition Diagrams**

**A diagram of a game

Description automatically generated with low confidenceA picture containing text, line, diagram, font

Description automatically generated**

****

****

**User Instructions**

To run the game, all you have to do is go here: <https://nwaziz.itch.io/csci-43700-platformer> and click on run game. No installation is necessary. Users must run the game on a laptop; it will not work on a phone.

**Godot Strengths and Weaknesses**

Strengths: It was very easy to learn! I was quickly able to make the game on my own. The documentation for the varying properties and methods was also easy to understand; I found myself looking through it a lot to discover new ways to improve my program.

Weaknesses: Honestly, I don’t really have any complaints about this engine so far. It’s great!