**Jeffrey Igims**

**Project Name:** Super 112 Space Smash

**Project Description:** Super 112 Space Smash is a game involving two characters that spawn in a map. They can be controlled externally by two players who will be able to do action such as jump, double jump, punch, and collect power ups. For example, a power up may increase the players health.

**Competitive Analysis:** Super 112 Space Smash is very similar to the Super Smash Flash series of video games in its gameplay and character used. Both games have the characters and basic movements in common such as the double jump and hand to hand combat movements. Where they differ is in the extra features in Super 112 Space Smash. Players in the new game will possible each have a unique ability of some sort. Also, they are able to jump down from platforms and accelerate downward after their jumps. Extra features that I will possibly add to the game after meeting the minimal viable project stage may include projecting the players’ faces onto their characters in the game. Another feature may include allowing multiplayer gameplay across multiple ports.

**Structural Plan:** I plan to use object-oriented programming and store my code in multiple files. I will have a separate file for the code to create and run the game and a separate file to handle the calls such as when keys and buttons are pressed or when the screen is redrawn. I will also have separate files each for holding classes for the two players, the power ups, and for the platforms in the game. I will have a superclass for players and two subclasses for the two individual players. The players will be sprites and the superclass will have methods such as an update to allow for jumping and jumping by the players. The subclasses will contain methods for their unique abilities. The platforms will also be sprites in another file allowing me to easily check for collisions and landings by the players on the platforms.

**Algorithmic Plan:** The trickiest part of my game will be implementing a character that can be controlled by the computer. To do this, I will create a separate class for the character if the player decides to play against the computer. The player will be a sprite and will take input on the location of the actual player. I will have the computer continuously move toward the player’s horizontal position and then jumping to achieve a closer vertical position to the player if the horizontal positions are close enough. For example, the computer would only attempt a jump if the horizontal positions of the two sprites are within a specified distance. The computer will double jump if the player is higher than one jump away.

**Timeline Plan:** I plan to be able to have two players control their characters in an environment filled with platforms that can be utilized by November 21. I also plan to be able to have the characters switch to different images based on their movements as well as take damage from other players. For example, the image of a character may change if a player jumps. By November 23, I plan to be able to have all of the power ups installed in the game for players to use. By November 25, I plan to have a character be controlled by the computer. By the second deliverable, I hope to meet a minimum viable project by accomplishing the prior statements and allowing character their own special abilities. Two players will be able to battle in map of basic platforms and take damage from one another as well as attain power ups.

**Version Control Plan:** I am backing up my code on google drive. A screenshot of this is included as well.

**Module List:** I am utilizing the module “Pygame” to create my game.

**Term Project #2 Update:** No updates have been made from the original plan of work.

**Term Project #3 Update:** Players can now collect shells to throw and track other opponents. The difficulty level of the computer can also now be adjusted. Instructions and end screens are now included.