**1. A description of the control flow for the interaction of a player avatar and a bank square. Where in the code is the co-location of the two objects detected, and what happens from that point until the interaction is finished? Which functions of which objects are called and what do they do during the handling of this situation?**

For every square including Baddies, I used two functions called “landedOn” and “passed,” both of which, as their name suggests, determine if a player has landed on a given location or simply passed. This is differentiated using the ticks\_to\_move counter of the player as well as a StudentWorld function I made called checkPlayerOn, which only determines if there is a player at a given coordinate. Before proceeding, I also made an array of two pointers which allows me to access the player class functions in all of my classes, which I used here to determine if any of the players were on or passing the square. I also used “isNew” and “changeStatus” functions so that each time the same player encountered the bank square, they wouldn’t infinitely interact with the square. After each interaction, I set isNew to false, letting the program know that they’ve already interacted once. At the beginning of the function, I test if the player is not on the given bank square or passing the square and if the player has interacted in the past with the square.

Using the two functions from the beginning, I then make sure that the player has landed on the bank square with a waiting to roll state, and added the entire balance of the bank to the player’s coin count using “adjustCoins” with the parameter being how much I wanted to add. Following this, I reset the bank balance to 0. If the player passed but didn’t land (meaning still in the walking state) on the bank square, I then took away 5 coins from the player’s coin count using the same function as before—with the count never breaching below 0. At the end of both these paths, I made sure to set the isNew to false so that, once again, the doSomething() doesn’t execute repeatedly in the same interaction.

**2. A list of all functionality that you failed to finish as well as known bugs in your classes, e.g. “I didn’t implement the Vortex class.” or “My Bowser doesn’t work correctly yet so it behaves like a Boo right now.”**

In some situations, the second possibility for the event square (swapping of players) doesn’t execute correctly. Say if Yoshi just landed on an event square and swapped himself and Peach, who was 2 squares behind, Yoshi won’t be able to activate the event square immediately again even if he rolls a 2 and lands on the event square.

**3. A list of assumptions you made; e.g., “It was not specified what to do in situation X, so this is what I decided to do.”**

It wasn’t too clear if a greater number of stars meant that a player won regardless of their coin counts so I just assumed that star count has greater precedence. I also assumed that the first baddie to be hit by the vortex is the one that is supposed to be impacted.