Project 3 Report

1. The Bank Square is called every tick by its doSomething() method. If the bank square isn’t alive (after being destroyed when Bowser leaves a dropping), it returns. Then the BankSquare uses the checkNewLanding method, which assigns variables peachVisits and yoshiVisits with a true or false value. The checkNewLanding first calls a method in StudentWorld which asks if there is a player on the square (using the BankSquare’s x and y pixel coordinates). The isPlayerOnSquare method compares Peach’s and Yoshi’s positions to the coordinates of BankSquare. The method returns 0 if neither of them is on the square, 1 if just Peach is on the square, 2 if just Yoshi is on the square, and 3 if both of them are on the square. The method also uses private members yoshiVisited and peachVisited to double-check that their visit isn’t a rapid repeat visit, ensuring only 1 comparison per square. The checkNewLanding method then assigns its parameters a value of true or false depending on if Peach and/or Yoshi properly visited the square. Using the now-initialized peachVisits and yoshiVisits variables, the BankSquare then checks if Peach landed on the square (peach can’t move), or if Peach passes over the square. If Peach lands, then the bank balance is added to Peach's coins, the bank balance is set to zero, and the appropriate sound is played. If Peach passes over the square, the method first checks if Peach’s coins are less than 5, and if they are, it is set to zero, and that value is set as the bankAddition. If Peach has equal or greater than 5 coins, Peach loses 5 coins and the bank balance is increased. The same thing then happens with Yoshi based on the yoshiVisits variable.
2. I implemented all the functionality and have thoroughly debugged my code. The main issue I have had is when a player is at a fork but their roll runs out on a fork square. When on a fork square, you cannot tell if you have to roll again or choose a fork direction. When your roll is not over, you can easily choose a fork direction and move appropriately, but when your roll ends, you must first roll again and then choose a fork direction.
3. One assumption I made was that Vortexes can pass through areas where there are no squares, and you can hit Baddies across gaps. I find that this is a fun addition, as getting a vortex can be difficult sometimes, and finding an opportunity to use it might be even more difficult. Additionally, my code was made with the assumption that there are no future PlayerAvatars, as the addition of more PlayerAvatars would be a bit tedious to implement, as many of the other classes would have to be changed. However, implementing a new Baddie or Square would be much easier to accomplish.