1. In the doSomething() for bankSquare, I first checked if a hero (my version of the player avatar) passed or landed onto the same coordinates as my bank square. If it did, I then checked if it passed by or if it actually landed on the bank square by counting how many ticks to move it had left. If it had no ticks to move left aka it actually landed on the bank square, I gave the hero the money and played the sounds and did everything else related to transferring the money. If I had ticks left to move I taxed the hero 5 coins or however many coins they had and added them into the bank. Functions called include setting the bank in the StudentWorld.cpp file to the proper amount after collecting and giving out the money from/to the proper heroes. I also used addCoins() and getNumCoins() in the hero class to withdraw and calculate how much money each player had and how much they owned/owed. To prevent overcounting a player landing on the same square twice in one tick, I had a m\_newPeach and an m\_newYoshi private member variable in the square class as a flag that was set to true initially and if it landed on a square the first time, it would be flagged as false so it wouldn’t be activated a second time. If the hero moved off the square, the flag would be counted as true so it could be used again. This was integral in making sure my squares such as bank square didn’t trigger multiple, unnecessary times.
2. So the only error currently that I have found from testing is that sometimes the display will actually show a different amount of rolls than you actually have. Sometimes it shows 0 rolls and it turns out that you have one or it sometimes shows 2 rolls but it turns out you actually have 3. It’s quite a rare error as I only see it happening around 5% of the time but it’s still noticeable when you’re trying to time a square and you end up counting wrong because the display showed an incorrect number of rolls.
3. I didn’t really make many assumptions about the spec because I followed it word for word. One thing that I am particularly confused on though, is if you’re at a corner, why should you prefer up over down or left over right? Because if you think about it, if you’re already at a corner it means there’s only one direction to go or else it would be a fork. So why is there a preference of where to go if there’s only one place to go anyways?