Karan Sharma

Dr. Ioana Fleming

CSCI 1300

29 April 2018

Project Report - Reflection Activity

For this project I decided to create a knock off of the popular digital card game, Hearthstone. My version, Chimneyrock, is simplified to have a player up against a bot each with one card in their hand. The game starts with both the player and the bot being dealt a card. Each card has an attack and an HP value randomly assigned ranging from 1-9. The player has the option to redraw his or her card to receive a new card (max 3 times) with random health and attack or lock in his or her card to play it against the opposing bot. Both player's cards are revealed and whichever card has an attack value that is greater than the opposing health value wins giving the card-owning player a point.

I prepared for this project by starting with a drawing that outlined features I wanted to include in my project. From that drawing I determined what class files I needed to implement for each component and how my main method would tie each individual class file to make a cohesive game. I started looking at the minimum requirements and fulfilling them in my outline, then added complexity to my project to ensure I met beyond the bare minimum requirements.

I wrote skeleton code for each class file I decided to create based on my drawing. It was extremely useful as it saved me time during the implementation process trying to figure out what functions I needed to define to execute my project. I also had a very clear cut idea about the purpose of each class file thanks to the fact that I outlined them prior to execution.

I could have done better by adding complexity to my project during the outlining phase rather than after I finished my baseline project. With my approach I completed my base project then started building on top of it to make it more complicated which, after rounds of testing, proved to create more of

a hassle for me as many errors arose due to conflict with my existing class files. I also could have allocated more time for execution than I ended up doing as this project which took me 18 hours total ended up being mostly executed within the span of 3 days. If I gave myself more time I could have added more detail to the game.

Initially, I had a false start as I designed the game to let the user create a deck before playing the game. This, however, meant that I had to hard code several cards for the user to pick from when creating a deck and the bot would end up getting the remainder. On top of this, because in my game some cards are just straight up better than others the deck variety would be extremely limited and it would become very hard for the bot to win. Because of this I scrapped my initial idea and made the generation of cards completely random giving both the player and bot an equal chance of winning the game.