Nolan Chung
3/11/14
AP Computer Science

Independent Java Project Narrative

From the beginning I was off to a late start regarding this project. My project idea as a whole was not set for a while and I wasted a lot of time getting started as I finally arrived at my final project idea and partnership with Jaz.

My initial thought of creating a virtual black jack card game was something I had considered since the independent project was announced. (See record of thinking #1) As such, I automatically selected it as my project when the time came, as can be seen in my repository titled with the pun: HitMe. I spent the earlier classes planning the classes and overall design and structure of the code. How the interface would work and how different classes would play the virtual deck with the player.

However, upon starting and planning things in class I realized there were already two other students who planned to pursue the same topic and I questioned my choice and idea. Mr. Kiang brought up in class the potential for group projects a little earlier and I was immediately interested in Jaz's idea in creating a virtual chess game. (See record of thinking #1) I was also leaning towards a gridworld-based project since we had recently spent most of our time learning about grid world and practicing questions and labs in it. I felt I would be comfortable enough working in gridworld and all the resources we had been given on it, plus I thought the practice would further familiarize with subject matter that would be on the AP exam later.

I joined Jaz on her project and spent most of that early time helping her plan out the code for the game, things like strategy on how to make it object oriented and efficient from a design standpoint. (See record of thinking #2 Fortunately and also unfortunately, I did not get too much coding during this period and spent a lot of time working and planning along side her and her computer. I helped plan how different classes might work and how we might have one for each different chess piece. How each would know how to move in its own unique way and interact with other actors on the board.) I worked alongside her as we planned out different pieces and constructed a basic grid to act as the game board. At this point in time we figured we would divide up work between the unique piece classes and continue from there. (See initial java files in repository)

However, my project direction underwent a significant change as Mr. Kiang returned and conferenced with us about our project. After discussion the decision was made that the two-player virtual chess game did just justify having both of us collaborate together. I was then assigned to work on an AI to compliment Jaz's chess game. I was at first very intimidated by this daunting task and had no idea to begin. I was reassured by Mr. Kiang though that I would not be expected to create and optimal AI program. Still though, I was quite lost and intimidated so I spent the few classes researching the concept of chess AI programs and trying to figure out how I would do this. In the end it was of little help. The programing was far beyond me and used complex algorithms that iterated moves and moves ahead in planning.

I continued planning and strategizing on this when another big direction change came about. It was decided that our game would be switched from chess to checkers. It was

something Jaz and me had thought about but didn't consider too seriously until Mr. Kiang talked with us again. We figured it would be best if we stopped making big changes but with his support we decided to change game types. The change was definitely for the best. It would simplify much of our work by limiting the variation of pieces to just that of kings and normal pieces as oppose to all of the different chess pieces. This would reduce having to figure out how to code each pieces potential of moves, eating other pieces, weight and value for prioritizing an AI, and other aspects of chess such as castling. Overall, the game structure would be very similar. The interface would still include using the grid as a game board as well as other actors as the pieces. Jaz's work on player controls to move pieces around would be the same, and I would similarly need to prioritize pieces on the game board to and select an educated move. I still needed to plan more though so I decided to investigate how checkers may be different for my purposes. Through my research I discovered that checkers, unlike chess, was actually a solved game. Unfortunately, similarly to my chess research, I discovered such methods were still much too complex for me and required a lot of computing power for years to solve the game.

I decided to work more of a defensive strategy to at least implement an educated AI. I did work on a method to determine what pieces were in danger so I could prioritize my pieces and subsequently select an advantageous move to nullify threats. (See record of thinking #3) That went along without too much trouble and I set out to figuring how I would actually select a piece after that. I received strategic aid from Zach in class as we talked and he drew diagrams on the board and such portraying his idea. It pretty much to use my inDanger() method to determine those pieces, find the location that needed to be filled to block a threat, iterate through neighboring pieces to that location, and see if any were suitable options that could move to said location and defensively protect pieces. I then worked further on implementing this idea and coded it into the program. Wyatt Smith also helped me solve some important issues. I was unable to call the getGrid() method until he suggested I just have my code extend Bug as bug was a class that could use that method. It was a valuable solution as the getGrid() method was used throughout my code and required only a few changed. An instance of my AI would just need to be implemented as an actor on the grid so it could run its methods during the game. Throughout this process however I went through a number of scope issues, as variables used in certain methods would need to be implemented in others to work efficiently. As my AI program was also already an actor bug, I consolidated the code into that one file as it had been previously spread. I continued working on the program, creating different outputs if my other strategies failed, primarily the random move method I created to execute a random move should there be no other options.

The main difficulty in undergoing this project, however, was having my project be so dependent on Jaz's. Although I did my best to aid her in implementing the overall game there were still just many problems we did not for see running into. One of the main one's was how she struggled trying to get the game to recognized consecutive clicks on the grid to execute a move. She spent a ton of effort on it but to no avail. We eventually settled for a solution I created, which was merely to program move methods into the piece class and access them through the Grid's interface by clicking on the actor, then the direction of the move. This served as a much simpler solution to the formerly complicated problem. However, there were still many difficulties Jaz needed to encounter on her end after going through the set back of working on the click recognition for so long for nothing. The lack of

completion of the overall structure of the game meant I couldn't test my code on her interface with out it being functional from the player's end as well. Even at the end of the project, we struggled together figuring out how to implement the turn-based nature of the game. Mr. Kiang helped us but we still couldn't implement getter and setter methods to control a turn Boolean or successfully use a global variable. Wyatt Smith was very helpful here again as he suggested using World.step() to have the grid run through the AI's code by putting it all in my actor's act method. This would allow the player to make their move and have the game automatically cycle through the AI's methods by implementing World.step(). However, in the end, we still struggled to finalize a functioning system. It made it very difficult to work along side and code my AI program.

Hopefully overall though, despite the many set backs I had in constantly having my project change and having to work without a functioning game interface. The overall process and work done were adequate. I was very intimidated by the daunting task of a complex AI program, but am happy I worked up a means and strategy of doing so with the limitations I had.