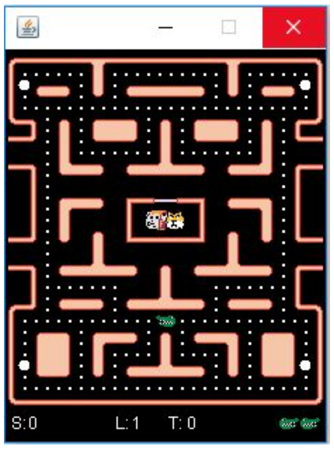
Stephon L. Smith
UFID: 1914-8471
COP 3502
Dr. Blanchard
4/15/2018

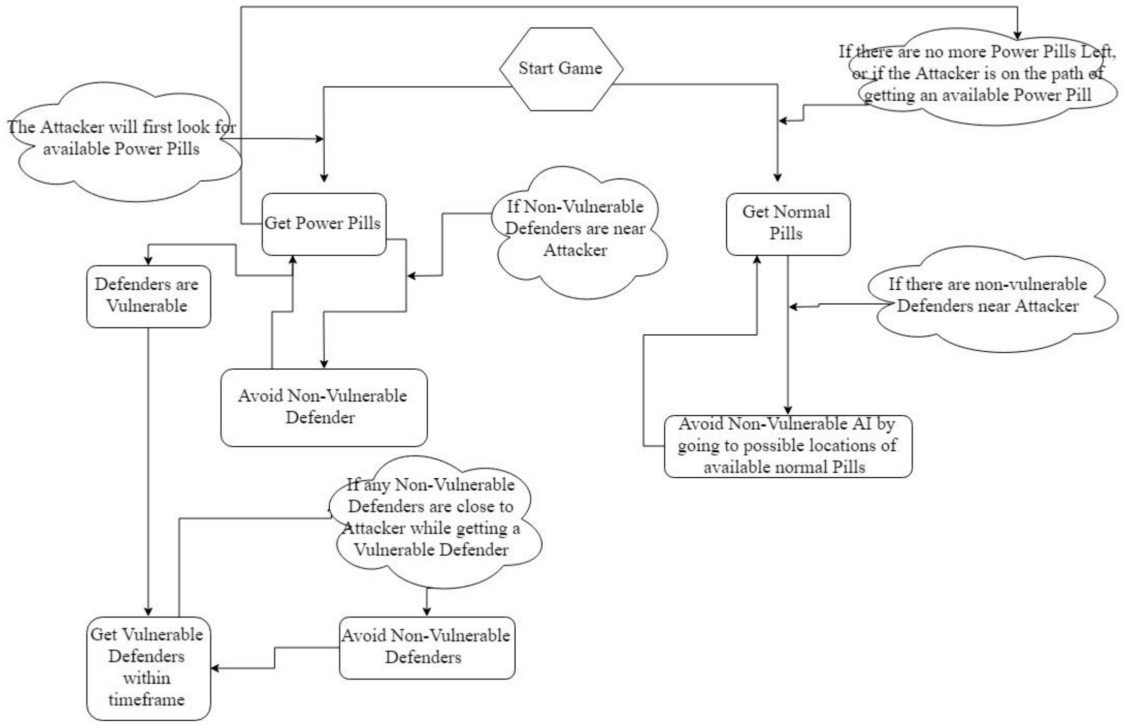
Project 4 (GatorRaider) Post-Mortem

Diagram:

Starting Point of the Game:



Tree Diagram:


Start Game
The Attacker will first look for available Power Pills
If there are no more Power Pills Left, or if the Attacker is on the path of getting an available Power Pill
Get Power Pills
If Non-Vulnerable Defenders are near Attacker
Get Normal Pills
Defenders are Vulnerable
If there are non-vulnerable Defenders near Attacker
Avoid Non-Vulnerable Defender
Avoid Non-Vulnerable AI by going to possible locations of available normal Pills
If any Non-Vulnerable Defenders are close to Attacker while getting a Vulnerable Defender
Get Vulnerable Defenders within timeframe
Avoid Non-Vulnerable Defenders

PakuPaku is a game that focuses on Inheritance, Polymorphism, and Artificial Intelligence. In addition, this game is similar to Pac Man, but with the chance in characters. At the start of the game, the Attacker AI (Gator) will first look for the available Power Pills that are inside of the maze. Once the Attacker has obtained a Power Pill, then the AI is able to attack the vulnerable Defender AI (Other State Universities such as FSU, UCF, ect.). However, once the vulnerable Defender has been eaten, then it will go back into a non-vulnerable state. If the Attacker is able to get a vulnerable Defender, and there is a non-vulnerable Defender that is close to the location of the Attacker, then the Attacker must avoid the non-vulnerable Defender and look for any nearby vulnerable Defenders until the time ends when the Defenders are in a vulnerable state. When that period of time stops, the defenders will switch back into a non-vulnerable state and the attacker will go to another available Power Pill to repeat the process. If there are no more Power Pills left in the maze, or if there are non-vulnerable Defenders that are close to or in the direction of the Attacker, the Attacker will look for the normal Pills that are available in all possible locations of the maze. Once the Attacker has obtained all of the normal pills and has acquired all of the Power Pills, then it will move onto the next level and obtain a high score within a trial until the program has run 100 trials and computes an average score.

Identification:

When coding the StudentAttackerController, the success that resulted in running the program came from utilizing multiple if-else statements in one for-loop in the public int update method in IntelliJ, in addition to leaving both the public void init and shutdown methods empty. The one for-loop is meant for iterating through the size of all the possible directions that the attacker can be able to go through in the maze to retrieve both the Power Pills and the normal Pills. In addition, the for-loop will also iterate to look for the location of the defender AI if they are either in a vulnerable or non-vulnerable state. In addition, another part of what was correct when running the game was for the attacker to determine where there were available Power Pills, and what to do if there are no more Power Pills left, which is to locate all the available normal Pills in all possible directions of the maze. When the attacker obtains a power pill, the AI will successfully navigate to the closest defender that is in a vulnerable state and attacks it while going for the available vulnerable defender within the period of time that the attacker has until the defender turns back into a non-vulnerable state, where the attacker will have to find another Power Pill. Even though there was success into the code for running the game, in addition to Test Agent-Score without any exceptions, there were a majority of failures that resulted in the average score. Part of what went wrong was how the attacker was not able to avoid any defenders that are in a non-vulnerable state while going for the available defenders that are in a vulnerable state, resulting to the point of where the defenders will get the attacker until the game is over.

Reflection:

Looking back at this project, there was both success and failure when creating the code to run the entire program. I can honestly say that I it was both a difficult, yet fun experience to be coding a program that has similar components to Pac-Man. However, the majority of the failure was due to the fact of my Asus laptop crashing to the point of where I was not able to work on the project for 3 days, in which I had to buy a new laptop to finish up the majority of the project. What did help me to obtain the maximum amount of points needed was to start the project as soon as possible before the day of Exam 2. In addition, what also helped me to obtain the maximum amount of points needed was to get help from multiple TA's that were more experienced than me in Java to be able to code the StudentAttackerController.