***Final Project: Pokenom***

In this program, we will redesign a game that has been a classic in many of our lives. For a large portion of our lives, the Nintendo Classic Pokemon has been created and recreated to fit the many changes and ideas that permeated through the times. The original style of the game that many have played throughout our childhood, however, has changed very little. From the regions of Kanto to the Islands of Alola, the core idea behind the Pokemon has always remained the same, keeping players engrossed throughout many years. The idea, as simple as it was, was able to keep players glued to their small screens. The role-playing-game, with quests and all sorts of interactions, remained the same for a very long time, until recently, as graphics improved.

In our project, we plan on recreating this coveted childhood game, but with its own little quirks. The original games’ simple design made it rather easy to recreate. However, as the gameplay was almost exactly the same, only spruced up with new characters and pokemon, we decided to add other function and restrictions to allow more interactive and thoughtful gameplay. It was only until the last two generations of games that the games began to include better graphics and a more realistic setting. However, we’ve decided to go back to basics, and begin again with the simple two dimensional maps and interactions with objects.

Our project, appropriated named Pokemon, combines the style and characters of pokemon with a unsatiated need to consume food. The gameplay is simple, similar to pokemon in several ways, yet also unique in ways that this game much more interesting that a simple Pokemon game.

The entire game is run a map, that will be as large are we dictate needed. Fruits and enemies spawn randomly on the map, and the player cannot move off the map.

The player, a munchlax, will run around collecting fruit while fighting enemies, eventually gaining enough experience to level up and become a snorlax, and fight the final boss. The player system keep track of several things, the player health, experience, level, and moves. Each one has different effects. The Health represents the Health points left within the player. If that bar hits zero, the player dies, and it's game over. Therefore, the player needs to keep a close eye on his health while fighting. After each battle, the health is completely restored, and the munchlax can continue his adventure.

Experience is also another section that needs to be focused on. After each battle, if the munchlax wins,he gains a set amount of experience. Eating berries will also provide a set amount of experience. After hitting a certain amount experience, the Munchlax will level up, gaining more health, and will learn a new move at certain intervals. As the Munchlax levels, the experience bar will get larger and larger, requiring more and tougher monsters to beat to fill up the bar.

Levels are a set of numbers that describe the Munchlax. With each level, his health will increase, and at certain marks, Munchlax will unlock a new ability that can be used.

Moves are the set of abilities that Munchlax and Snorlax will be able to use during battle. As the player levels up, they will unlock new abilities that will be able to used. Each one has a certain amount of power, and a certain amount of damage.

Interactable objects in the game are characterized into two types: fruits and enemies. Each will give certain benefits and perhaps problems if run into. Once the player hits or interacts with the object, the object will disappear, and the effect will be activated.

Fruits are all sweet without any drawbacks. Interacting with a fruit will give the munchlax some experience, and allow the munchlax to continue his journey. Each berry will give a different amount of experience.

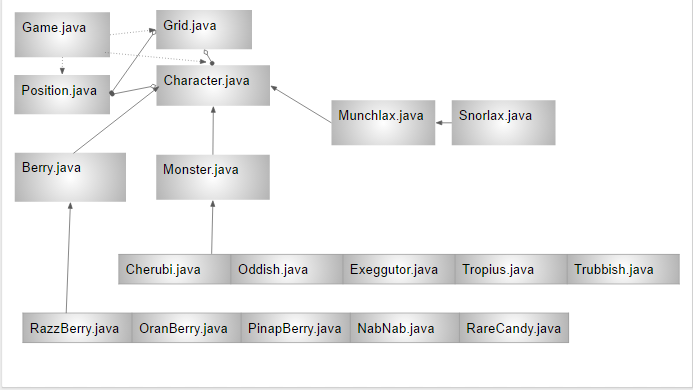
Enemies trigger the beginning of the RPG battling style in the game. These enemies are disguised as fruits, so the player has no idea if the fruit he is interacting with is a fruit or an enemy. After hitting an enemy, the player must fight until one faints. If the player wins, he continues the game, and gains a substantial amount of experience. If he loses, it's a game over.

Although this project seems simple enough to code and play, the amount of thought needed to fully make this program work smoothly is larger than expected. The Code itself may not be difficult, but working everything to a project that runs well, and is able to look half decent is a pretty large challenge.

(As the Coding process continues, there will be additions and changes made to this Preliminary Specification)

Object Oriented Design

The figure below shows the class diagram for Pokenom. The project involves 17 classes, each with its own use and need.



As seen in this figure, the Character class is the superclass, and all the characters, including the player extend this class. Berries, Monsters, and Munchlax are characters, just each with its own extra portions. The Game classes runs everything in a continuous loop that begins at the beginning and continues until Game Over. The Position and Grid are there to show the player where he is, and where he can find berries/monsters and progress in the game.

Testings

There needs to be large and extensive testing required to ensure a quality product. It cannot simply be a Junit test that makes sure the methods works as it should. We as a group must also continually beta test the game and pull every single runtime error out of the program. As with every other game, this is a vital step in ensuring that the game runs as it should. Characters should be tested separately from the Grid.

3 arraylists-one of berries, one for monsters, one for position.