

**Battle!**

**Ken Tjie**

**AP Computer Science ICS 4UP**

**Mrs. Heyes**

**19 January 2015**

Purpose

The purpose of this project is to help me learn more about project management and demonstrate my understanding and mastery of the following programming concepts: data types, self-written classes, array lists, control structures, classes and methods, polymorphism achieved through inheritance, file input and output, sorting algorithms, searching algorithms and event handling.

About

My project was inspired by a game I made with GameMaker in Grade 10. It was pretty bland and boring, so I thought of recreating this game and make it bigger and better. This game is a vertical shooter based on the theme of Pokémon, where the player controls a Pikachu on the right side of the screen and the opponent’s six Pokémon are on the right side of the screen, fought one at a time of course. This game stays true to the formulas used in Pokémon such as the equation that calculates the damage done to the opponent as well as calculation of the stats of the individual characters.

Concepts

Datatypes – These have been achieved throughout all the coding. Depending on what they’re keeping track of, different variables use different data types accordingly. For example, the user name is kept in a String variable, the boss health is kept in a double variable and the player is stored in a self-written Player variable.

Array List – These were created in two different classes. Firstly, the BattleFrame class where an Array List of Boss was created. This stores the bosses and then removes 6 of the forty to be passed into the BattlePanel. Then, the BattlePanel class has two Array Lists, one of Item and one of Projectile. The Array List are cycled through updating each instance as necessary.

Control Structures – Although there are a lot of control structures in the coding such as loops and if/else statements. The most important use for them in my program is the use for collision detection (if/else statements) and cycling though the arrays and Array Lists to update the instances (for loops).

Classes and Methods – Without a doubt, classes and methods are what allow almost any program to function. In my program, there are several classes that contain different things. For example, the BattlePlayer class that runs the game, the BattlePanel class that controls the logic of the game. The other classes contain the methods that help the code look neater and keep it organized. Different methods do different things. In my program I have accessor methods, mutator methods and others that draw graphics on the screen and respond to collision etc.

Polymorphism – I achieved polymorphism through inheritance. I have two abstract classes: Character which is inherited by the Player and Boss classes and the Projectile class which inherited by the BossPorjectile and PlayerProjectile classes. In the character class, the abstract method is the updateHP method which draws the HP in different locations and positioning them correctly when updated. In the Projectile class, the abstract methods are draw, which draws the ball in different locations and the collisionReaction method that changes the damage accordingly with the two different projectiles.

File input/output – The different bosses in my program are read from a text file, it reads from three different files to give the game a little diversity. These text files contain the name, type and stats of the boss to be created. Input is also used to retrieve the data on the scores for the different people which contains the name, points and number of levels they beat. Output is used to print into the text file the bosses and player of the previous game (meant to be looked at outside the program) as well as printing the data of the leaderboard into the text file for future games (not meant to be read outside of program).

Sort and search – Both sorting and searching were used for the leaderboards only. The array of scores is sorted in descending order according to points. The player will then be able to search according to name or according to points. Searching by points will result in binary searching and linear search will be used when the player is searching by name.

Event handling – Without a doubt the use of event handling is what makes my game functional. The ActionListener is used to do the animation, alongside a Timer. The KeyListener allows the user to use the keyboard to control the game, arrow keys to move the player and the space bar to shoot their projectile.

Bugs

The only problem I had doing is incorporating the music into my game. I tried to run the music like I usually do in my other programs, but it didn’t play when I ran my program. I resorted to making the MusicPlayer class that plays the music separate from the BattelPanel/BattelFrame class. Also, I believe that this problem is just the programming but when the user presses one arrow key and another arrow key before letting go of the first one, the player will stop and not move until the user presses an arrow key again.

Acievements

The thing I’m most proud of my project is being able to animate images as well as programing collision detection. My past games, using Java, were just quiz games, using JFrames and JButtons to function. But being able to animate and do collsion detection is something we learned in Computer Science Club (well, just collision detection) that I was able to put into practise for my last Computer Science project in high school. Using concepts like thse makes a game more similar to the other video games and computer games made in this world today.

More Time?

If I had more time, I would have liked to code levels to be more different. Instead of 2 levels having the same AI or attack patterns, just like the randomly selected bosses, I would have liked to make it so that the attack patterns are chosen randomly as well. Also, I would have liked to make the boss move more randomly, without being to shaky. I wanted to make the boss move forward and back, but I needed to make sure that the other components of the game were running and working correctly. I would also have liked to make the collision detection more accurate with the SAT theorem or utilizing circle collision instead of using bounding boxes on circles.

Catholic Expectations

**CGE4f applies effective communication, decision-making, problem-solving, and time and resource management skills**

For the majority, I kept in time with my timeline I had set for myself, minus the time used on coding for aesthetic reasons. I used all the available resources to my advantage to help me in the completion of this project. My game was completed the before presentations began. My game was fully functional and fully playable since the 14th of January this year. I planned out my time from weekdays to weekends to set aside time to work on this project.

**CGE3c thinks reflectively and creatively to evaluate situations and solve problems**

I think that this is by far the best project I have ever created. It’s a lot different than what I did in past years for any major project. I had a lot of problems along the way, but that didn’t stop me from making this game. I used the best resource there is. the Internet, to find out my problem and understand it better and how to fix it.

**User Guide to Battle!**

Here is Pikachu! This is your character that you get to move and control.

**Controls**

Arrow keys: moves Pikachu according to the arrow you press.

Space Bar: shoot a projectile that launches straight forward.

**Items**

* *Leppa Berries* grant you 3 more PP upon contact.



* *Sitrus Berries* restore 50 HP to the player upon contact
* *Coins* grant the player 10 points upon contact, but losing a level will diminish you coins from 25% to 50%!

**Objective**

You must defeat the opposing trainer who has all six of his Pokémon! Unfortunately, you only have your Pikachu on you so you have to fight a 6 vs. 1 battle! Move around the board, dodging your enemy’s while trying to hit them with your projectiles at the same time! Once your HP goes to 0, you’ll lose money (points) running back to the Pokémon Center! If you defeat your opponent by depleting their HP to 0, you move on to the next level with each level getting harder until you defeat all 6 of his Pokémon. If you win, congratulations!

**Notes**

* You start off with 10 PP (Power Points / shots). You must keep track of your own PP!
* Do NOT press another arrow key before letting go of the previous one or else Pikachu will stop moving until you press another arrow key.
* Open up the Previous Battle.txt file to see the Battle! game before your or open it after you play to see the bosses you fought.