Carolina Pasuy Pinilla A00358427

Juan Manuel Palta Cortes A00369611

PROJECT: VIRTUAL SCHOOL GAME

The project is about a video game where you can be a university student and your objective is to pass the semester defeating the homeworks (PDF'S FILES) you have. Is a game with different stages, each stage has a different quantity of enemies (PDF'S).When you defeat all PDF'S in the stage, you will pass to the next stage with more enemies. The game does not have an end. It is a competitive game with scores, try to be the best and be the number one in the ranking.

The game will have a main menu that allows you to start a new game, load a saved game, check game scores, information on the sessions started and exit the game. When starting a game you can choose between the two characters, male or female, and as the enemies are eliminated, the total moodleCoins will increase.

Why does the project need two members?

We can distribute the different responsibilities and achieve a good quality game in a short time, in addition to the fact that the two members have a great love for video games, which is why we will show our commitment and performance not to abandon the project.

FUNCTIONAL REQUIREMENTS:

RF 1. Menu: The program needs to have a menu with the options play,score and exit.

RF 2. Save score: The program needs to save the score of the player when he loses or he leaves the game.

RF 3. Show ScoreTable: The program needs to show the score of all players of the game.

RF 4. Exit: The program has a button to close the application.

RF 5. Choose character: Show a screen to choose between two characters, **MALE OR FEMALE.**

RF 6. Attack: the player could attack with a gun the PDF'S.

RF 7. Buy guns: the player could buy guns with **MoodleCoins.**

RF 8. Save game: the player could save the game to return later.

RF 9. Show ScoreInScreen: the score will be updated and displayed on the screen.

RF 10. Shop: The player can access a shop to buy different powers with MoodleCoins, the button shop will be on the screen of the game.

RF 11. Pause game: The player can pause the current game.

RF 12. Move player: the player can move in the map of the game.

RF 13. Enemies:there will be several enemies, the enemies will chase and kill the player.

RF 14. ChangeStage: when the player kills all the enemies, the stage will change and will generate more quantities of enemies.(the advice of change of scenery will be displayed on the screen).

RF 15. SearchPlayerScoreTable: the program can search the scores associated with the nickname to search.

RF 16. Collisions: the player, enemies and the powers will have collisions to know when the power impacts the enemy or when the enemy attacks us.

NO FUNCTIONAL REQUIREMENTS:

RNF 1. Serialize all the information of all players and their own scores

RNF 2. Design and implement automated unit tests for all project methods

RNF 3. Save both the score of each player and the damage generated to enemies in binary trees

RNF 4. Implement the binary search in the search for a user and the position of the game score

RNF 5. Use the bubble sorting algorithm to sort the levels, the insertion algorithm to sort the amount of enemies and the selection algorithm to sort the amount of money generated by the user.

RNF 6. Implement motion interface for characters.

RNF 7. Create custom exceptions for:

- Creating a new game the user enters a required nickname
- The user does not buy a gun without having enough money
- Search a user and position existing

RNF 8. Create java api-own exceptions for all persistent files

RNF 9. Apply inheritance to the Characters, Weapons, and Enemies classes

RFN 10. Load a player's saved games using plain text file persistence

RNF 11. Implement threads for the movement and attack of the players and enemies

RNF 12. Implement the binary three for the players and for the data session

RNF 13. Use naturally recursive methods in the two binary tree of players and the data session