**TABLA DE ESPECIFICACIÓN DEL PROBLEMA DE INGENIERÍA DE SOFTWARE**

|  |  |
| --- | --- |
| CLIENT | Videogame company |
| USER | Administrator / Developer |
| FUNCTIONAL REQUIREMENTS | RF1: Register Players  RF2: Register treasure at one level  RF3: Register enemy at one level  RF4: Register level  RF5: Calculate complexity level  RF6: Modify a player's score  RF7: Increase the level for a single player  RF8: Report treasures and enemies in a level  RF9: Report the amount found of a treasure at all levels  RF10: Report the quantity encountered of an enemy type at all levels.  RF11: Report the most repeated treasure at all levels.  RF12: Report the highest scoring enemy and the level where it is located.  RF13: Report the number of consonants found in enemy names.  RF14: Report the top 5 players |
| PROBLEM CONTEXT | A video game company requires help for the creation of a new video game, which will consist of ten levels in total, in which the player will fight against enemies such as ogres, bosses, magic and others. In the middle of his fight, he will also have to look for treasures, which will give him the necessary score to pass to the next level. |
| NON-FUNCTIONAL REQUIREMENTS | RNF1: the system must have a very good performance. The display of enemies and treasures must not be longer than two seconds.  RNF2: it must have a web version and a mobile application version. |

**FUNCTIONAL REQUIREMENTS ANALYSIS TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF1: Register Player | | |
| Abstract | In this method as many players as possible are registered, so far, the maximum number of players to be registered is 20. | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| nickName | String | Each player will have a unique nickname, it cannot be repeated. |
| name | String |  |
|  |  |  |
|  |  |  |
|  |  |  |
| General activities necessary to obtain the results | Request this data, store it and verify it | | |
| Result or postcondition | The result will be the player, stored in the array, in case it meets the precondition. | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| true | boolean | If there is still space inside the rest of the players. |
| false | boolean | If an empty position is no longer recorded within the array, which allows saving the player's data. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF2: Register treasure at one level | | |
| Abstract | To fulfill this requirement:  1.Treasures are recorded, which will be those that will give the player the necessary points to advance to the next level.  2.the registered levels and the registered treasures will be show, the user will choose the level to which he wants to add the treasure and the treasure to be added. | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| treasureName | String |  |
| ImageUrl | String |  |
| treasureValue | int |  |
| quantity | int |  |
| levelNumber | int |  |
| treasureNumber | int |  |
| positionX | int | A random number of x is generated between the values of 1 to 1280 |
| PositionY | int | A random number of y is generated between the values of 1 to 720 |
| General activities necessary to obtain the results | \*Register a treasure  \*Register levels  \* Show the recorded levels  \*Show registered treasures  \* Choose the level you want to add the treasure  \* Choose the treasure to be added  \* Randomly generate the x and y position of the treasure once it was added to the level | | |
| Result or postcondition | The treasure added to the level indicated by the user | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| true | boolean | If the treasure could be successfully added to the level |
| false | boolean | If the treasure could not be added to the level |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF3: Register enemy at one level | | |
| Abstract | To add an enemy to a level, there must first be an enemy that can be added.  Therefore, the first step will be to register an enemy, with their respective attributes and then, the levels and enemies already registered will be shown, from which the level and the enemy required will be chosen. | | |
| Inputs | Inputs name | Type of data | Selection or repetition condition |
| identifier | String | This identifier cannot be repeated on the same level |
| enemyType | int |  |
| pointsLost | int |  |
| PointsGained | int |  |
| positionX | int | A random number of x is generated between the values of 1 to 1280 |
| PositionY | int | A random number of y is generated between the values of 1 to 720 |
| numberLevel | int |  |
| enemyPosition | int |  |
| General activities necessary to obtain the results | \*Register enemies  \*Show levels  \*Show registered enemies  \*Choose the level to which the enemy is to be added  \*Choose the enemy to be added to the level  \* Randomly generate the x and y position of the enemy once it was added to the level | | |
| Result or postcondition | The enemy added to the respective level | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| true | boolean | if the enemy could be added to the level |
| false | boolean | if the enemy could not be added to the level |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF4: Register level | | |
| Abstract | Records a level with its respective attributes | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| identifierNumber | int |  |
| passingPoints | int |  |
| complexityLevel | int |  |
|  |  |  |
|  |  |  |
| General activities necessary to obtain the results | Request level data | | |
| Result or postcondition | The level successfully registered | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| true | boolean | If the level meets the requirements to be registered |
| false | boolean | If all ten levels and their respective scores have been recorded. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF5: Calculate complexity level | | |
| Abstract | Determines the complexity of the level | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| levelPosition | int |  |
| enemiesPoints | int |  |
| treasurePoint | int |  |
|  |  |  |
|  |  |  |
| General activities necessary to obtain the results | 1. shows the recorded levels 2. you choose the level for which you want to calculate the complexity 3. The score awarded by the treasures is compared with the score of the enemies when they are not defeated | | |
| Result or postcondition | The complexity of the level set | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| LOW | Complexity | if(treasuresPoints>enemiesPoints) |
| MIDDLE | Complexity | If treasuresPoints == enemiesPoints |
| HIGH | Complexity | If treasuresPoints<  enemiesPoints |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF6: Modify a player's score | | |
| Abstract | The player's current score can be modified and replaced by a new one | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| playerPosition | int |  |
| score | int |  |
| General activities necessary to obtain the results | 1.Display the players  2.Select the player whose score needs to be modified  3.Enter the player's new score | | |
| Result or postcondition | Player with new score | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| true | boolean | If the player's score could be modified |
| false | boolean | if the player's score could not be modified |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF7: Increase the level for a single player | | |
| Abstract | Level must be increased for a player, in case he/she cannot increase the level, he/she must inform the user what score he/she needs to increase | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| playerPosition | int |  |
| LevelPosition | int |  |
| score | int |  |
| General activities necessary to obtain the results | The "assignLevelPlayer" method is called, which is in charge of assigning a player's level.  If this method returns false, the calculation of how much is left to advance to that level will be done. | | |
| Result or postcondition | The message indicating whether or not the player can pass to a higher level | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| msg | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF8: Report treasures and enemies in a level | | |
| Abstract | A report of the treasures and enemies belonging to a specific level will be given. | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| LevelPosition | int |  |
| General activities necessary to obtain the results | Request the level from which the report of enemies and treasures is required | | |
| Result or postcondition | The message indicating the enemies and treasures of the level | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| Msg3 | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF9: Report the amount found of a treasure at all levels | | |
| Abstract | Iit is a report on the number of treasures of the same type of a level. | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| treasureName | String |  |
| General activities necessary to obtain the results | The name of the level on which its quantity is to be obtained is requested | | |
| Result or postcondition | The message indicating the number of treasures with that name | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| msg | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF10: Report the quantity encountered of an enemy type at all levels. | | |
| Abstract | The existing quantity of a type of enemy is searched | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| enemyType | int |  |
| General activities necessary to obtain the results | Request the type of enemy to search for | | |
| Result or postcondition | The message indicating the number of enemies with that type | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| msg | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF11: Report the most repeated treasure at all levels. | | |
| Abstract | we request to know the most repeated treasure at all levels and how often it is found. | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | A tour is made through the different arrangements that will allow us to obtain the name of the treasure and the amount, being within a repetitive structure, it will be executed until the highest result is found, that is to say, until the greatest amount of treasures and the name of the treasure to which they belong are found. | | |
| Result or postcondition | the message with the treasure that is found in the greatest quantity at all levels | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| msg2 | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF12: Report the highest scoring enemy and the level where it is located. | | |
| Abstract | A report will be generated on the enemy that gives the most points when defeated and its location. | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| not required | | |
| General activities necessary to obtain the results | Different arrays within repetitive structures are traversed in order to obtain and compare the "pointsGained" until the largest one is obtained; its structure can be similar to the previous requirement.  Once assigned, its name and its location in the level array will be shown by console | | |
| Result or postcondition | A message indicating the player who scored the most points and his location | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| msg | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF13: Report the number of consonants found in enemy names. | | |
| Abstract | Of all registered enemies, the number of consonants in their names will be evaluated. | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| not required | | |
| General activities necessary to obtain the results | In this case, before going through the repetitive structures, we create an array containing all the consonants, in this way we can check their existence in the names of the enemies and compare them to verify if they are consonants, and each time this process is performed, a count will be started to determine the total number of consonants present in the names of the enemies. | | |
| Result or postcondition | A message indicating the number of consonants present in the enemies' names | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| msg | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | Report the top 5 players | | |
| Abstract | a top five will be generated with the first five players with the highest score. | | |
| Inputs | Input name | Type of data | Selection or repetition condition |
| not required | | |
| General activities necessary to obtain the results | In this requirement, a five-position player array will be created, which will be assigned to the players that are registered, from lowest to highest, until the player with the highest score is found. | | |
| Result or postcondition | A message indicating the top five players | | |
| Outputs | Input name | Type of data | Selection or repetition condition |
| msg | String |  |