**Student:** Maria Alejandra Mantilla Coral

**Code:** A00395792

**VIDEOGAME**

**SOFTWARE ENGINEERING PROBLEM SPECIFICATION TABLE**

|  |  |
| --- | --- |
| **CLIENT** | Videogames Company |
| **USER** | Players |
| **FUNCTIONAL REQUIREMENTS** | **R1:** Register new players.  **R2:** Register the different game levels.  **R3:** Register the treasures on the levels.  **R4:** Register the enemies on the levels.  **R5:** Modify a player's score.  **R6**: Increase the level for a specific player and inform the user the score he/she needs to increase in case he/she doesn’t have enough points.  **R7:** Report the treasures and enemies of a given level by the user.  **R8:** Report the amount found of a treasure at all levels.  **R9:** Report the quantity encountered of an enemy type at all levels.  **R10:** Report the most repeated treasure at all levels.  **R11:** Report the enemy that awards the highest score and the level where it is located.  **R12:** Report the number of consonants found in the names of enemies in the game.  **R13:** Report the top 5 players according to the score. |
| **CONTEXT OF THE PROBLEM** | To solve the software requirements of this video game we must pay attention to the following aspects:  - Generate screen resolutions in the init (first time we run the program).  - Use OOP concepts and good coding practices.  - Create the necessary authentication methods (to create a player verify the existence of the nickname and to create an enemy, verify if the enemy does not exist in the level).  - Generate random positions based on the screen resolution.  - Create the necessary methods to verify that a user has the necessary requirements to level up.  - The number of players must be a maximum of 20.  - The number of enemies must be a maximum of 25.  - The number of levels should be a maximum of 10. |
| **NON-FUNCTIONAL REQUIREMENTS** | **Performance:** The display of treasures and enemies in a level should not take more than 2 seconds in the web application.  **Multi-platform:** The system must work on both web applications and mobile devices. |

**FUNCTIONAL REQUIREMENTS ANALYSIS TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R1:** Register new players. | | |
| Summary | It allows the user to register as a player. The system asks them for their nickname and name, and automatically assigns them a score of 10, and 5 lives initially. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| pNickname | String | Only if the nickname is already in use. |
| pName | String |  |
|  |  |  |
| General activities necessary to obtain the results | - Read pNickname  - Check if nickname doesn’t exist  - Read pName  - Read pScore (preset by the system).  - Read pLives (preset by the system).  - Read pLevel (preset by the system, level 1 initially).  - Check if there are available positions in the players array. | | |
| Result or postcondition | Player added to the game or error in the registration. | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_register | String | Only if the data entered above does not contain any errors or problems. |
| msj\_error | String | Only if all required information fields are not filled in and/or the nickname is already registered. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R2:** Register the different game levels. | | |
| Summary | It allows, at a given level, to record the information of the levels (the points required to advance, the types of enemies to defeat and their level of complexity, which will be calculated from the previous information). | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| levelId | int | Only if the value of the level entered is less than 0 or greater than 10. |
| scoreRequired | int |  |
| enemyType | int | Only if the user entered a value outside the range of those in the displayed menu. |
|  |  |  |
| General activities necessary to obtain the results | - Read the level.  - Read the required points.  - Read the types of enemy to register. | | |
| Result or postcondition | If all data entered do not contain any errors, display a successful registration message. Otherwise, display an error message. | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_register | String | Only if the data entered above does not contain any errors or problems. |
| msj\_error | String | Only if not all required information fields have been filled in and/or the enemy type has been previously registered. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R3:** Register the treasures on the levels. | | |
| Summary | Allows the user to register the treasures in the levels. For this operation, the following treasure data is requested: the name, the amount, the level, and the score it gives to the player. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| tName | String |  |
| tUnits | Int |  |
| tLevel | Int |  |
| tScore | Int |  |
|  |  |  |
| General activities necessary to obtain the results | - Read tName  - Read tUnits  - Read tLevel  - Read tScore | | |
| Result or postcondition | Treasures successfully added to the level or error in the treasures registration. | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_register | String | Only if the data entered above does not contain any errors or problems. |
| msj\_error | String | Only if not all required information fields have been filled in. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R4:** Register the enemies of the players. | | |
| Summary | Allows the user to register an enemy in a specific level. The system asks for the level where the enemy is going to be added, the name (id), the type, the score it adds, the score it subtracts, and automatically generates a position in X and a position in Y. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| eLevel | Int |  |
| eNameId | String |  |
| eNameType | String |  |
| eType | Int |  |
| eScoreDown | Int |  |
| eScoreUp | Int |  |
|  |  |  |
|  |  |  |
| General activities necessary to obtain the results | - Read eLevel  - Read eNameId  - Read eNameType  - Check if the enemy doesn’t exist at that level.  - Read eType  - Read eScoreDown  - Read eScoreUp  - Read ePositionX  - Read ePositionY | | |
| Result or postcondition | Enemy added to the level or error in the enemies registration. | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_register | String | Only if all the information requested above has been entered. |
| msj\_error | String | Only if not all required information fields have been filled in. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R5:** Modify a player's score. | | |
| Summary | Allows the user to modify a player's score. This operation requires the player's nickname to be entered. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| pSearch | String | Only if the nickname of the entered player is not found in the list of registered players. |
| pNewScore | Int |  |
|  |  |  |
| General activities necessary to obtain the results | - read pSearch  - Check if the nickname (pSearch) entered by the user exists.  - Get the current score.  - Read the new score to assign. | | |
| Result or postcondition | Score successfully modified. | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_new\_score | String | Only if the data entered above does not contain any errors or problems. |
| msj\_error | String | Only if all required information fields are not filled in and/or the nickname is already registered. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R6**: Increase the level for a specific player and inform the user the score he/she needs to increase in case he/she doesn’t have enough points. | | |
| Summary | Allows the user to increase the level of a specific player. For this operation the user is prompted to enter the player's nickname. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| pNicknameLevel | String | Only if the nickname of the entered player is not found in the list of registered players. |
|  |  |  |
| General activities necessary to obtain the results | - Read pNicknameLevel.  - Check if the nickname (pNickName) entered by the user exists.  - Get the player's current score.  - Get the necessary scores for each level.  - Check what level the player can go up to with his current score. | | |
| Result or postcondition | Successfully raised level or it was not possible to upgrade the player's level, the required score is: (Score). | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_level\_up | String | Only if the player has the necessary score to level up. |
| msj\_score\_required | String | Only if the player doesn’t have the necessary score to level up or was not found in the list of registered players. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R7:** Report the treasures and enemies of a given level by the user. | | |
| Summary | It allows the user to know the enemies and treasures found in a specific level. For this operation, the user is asked to enter the level from which he/she wishes to obtain the information. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| levelSearch | Int | Only if the value of the level entered is less than 0 or greater than 10. |
|  |  |  |
| General activities necessary to obtain the results | - Read levelSearch.  - Verify if the entered level exists.  - Get the treasures contained in that level.  - Get the enemies contained in that level. | | |
| Result or postcondition | Level: (Level), Treasures: (List of treasures), Enemies: (List of enemies). | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj | String | Only if all the information requested above has been entered correctly. |
| msj\_error | String | Only if not all required information fields have been filled in or the level was not found. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R8:** Report the amount found of a treasure at all levels. | | |
| Summary | It allows the user to know the amount that exists in all levels of a specific treasure. For this operation the user is asked to type the name of the treasure. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| tNameSearch | String | Only if the name of the treasure was not found. |
|  |  |  |
| General activities necessary to obtain the results | - Read tNameSearch.  - Check (with the name) if the treasure exists.  - Get the amount of this treasure at all levels. | | |
| Result or postcondition | Name of the treasure: (name), total amount of the treasure: (amount), or, it was not possible to find the treasure. | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_all\_treasures | String | Only if the data entered above does not contain any errors or problems. |
| msj\_error | String | Only if all required information fields are not filled in and/or the name of the treasure was not found.. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R9:** Report the quantity encountered of an enemy type at all levels. | | |
| Summary | It allows the user to know the amount that exists in all levels of a specific enemy. For this operation the user is asked to type the enemy type. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| eTypeSearch | Int | Only if the type of enemy entered is less than 0 or greater than 4. |
|  |  |  |
| General activities necessary to obtain the results | - Read eTypeSearch.  - Check (with the enemy type) if the enemy exists.  - Get the amount of this enemy at all levels. | | |
| Result or postcondition | Enemy type: (type), name: (name), amount: (amount) or, it was not possible to find the enemy. | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_all\_enemies | String | Only if the data entered above does not contain any errors or problems. |
| msj\_error | String | Only if all required information fields are not filled in and/or the enemy type was not found. |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R10:** Report the most repeated treasure at all levels. | | |
| Summary | It allows the user to know the most repeated treasure in all levels. No user input is required for this operation. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | - Count (for each treasure) its quantity in each level by performing a complete run through the arrangements. | | |
| Result or postcondition | The most repeated treasure in all levels is: (name of the treasure). Amount of the treasure: (amount) | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_most\_treasure | String |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R11:** Report the enemy that awards the highest score and the level where it is located. | | |
| Summary | It allows the user to know which enemy gives the highest score when defeated. No user input is required for this operation. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | - Perform a full run (for-cycle) on the arrays of each enemy with their associated scores to calculate which has the highest score value. | | |
| Result or postcondition | The highest scoring enemy is: (name of enemy). Score value: (score). | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_most\_enemy\_score | String |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R12:** Report the number of consonants found in the names of enemies in the game. | | |
| Summary | Allows the user to know the total number of consonants in the names of the enemies in the game. No user input is required for this operation. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | - Obtain the names of the enemies.  - Calculate the number of consonants in each name.  - Add up the total number of consonants found in each name. | | |
| Result or postcondition | Total number of consonants found: (number). | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_consonants | String |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R13:** Report the top 5 players according to the score. | | |
| Summary | Allows the user to know the top 5 players with the highest score in the game. No user input is required for this operation. | | |
| Inputs | Input name | Data type | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | - Scroll through the arrangement of the players, while saving the score of each one.  - Compare all players' scores.  - Rank the 5 highest scores. | | |
| Result or postcondition | Top 5 players:  - Nickname: (nickname). Score (score).  ...  ...  ...  ... | | |
| Outputs | Output name | Data type | Selection or repetition condition |
| msj\_top\_five | String |  |
|  |  |  |