|  |  |
| --- | --- |
| **Client** | Video Game Company |
| **User** | Players |
| **Functional Requirements** | **R1:** Register and add player to the video game.  **R2:** Register and add level to the video game.  **R3:** Register and add treasure to a level.  **R4:** Register and add enemy to a level.  **R5:** Set difficulty of a level.  **R6:** Modify a player's score.  **R7:** Increase level for a player.  **R8:** Report the treasures and enemies of a level given by the user.  **R9:** Report the amount found of a treasure in all levels.  **R10:** Report the amount found of an enemy in all levels.  **R11:** Report the most repeated treasure in all levels.  **R12:** Inform the enemy that gives the highest score and the level in which it is.  **R13:** Report the number of consonants found in the names of enemies in the video game.  **R14:** Inform the top 5 of the players according to the score. |
| **Problem Context** | A version of a video game is needed that has the functionalities of registration of players, levels, treasures, and enemies. To run the game, the player must choose the resolution he/she wants to start in and choose an option for the game to run. |
| **Non-functional Requirements** | **R15:** The deployment of enemies and treasures in the web application cannot take more than 2 seconds.  **R16:** The system must work in a web application and in a mobile application. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R1:** Register and add player to the video game. | | |
| **Abstract** | The system must allow creating and saving a player within the video game. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Nickname | String | If the player enters the game for the first time and if the array of players within the game is not full. |
| Name of player | String |
| **General activities necessary to obtain the results** | * The program asks for the information to create a player. * The player is created and added to the array of players within the video game. The score, number of lives and level are automatically initialized. | | |
| **Result or postcondition** | The program displays a confirmation message whether or not the player was added. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Confirmation message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R2:** Register and add level to the video game. | | |
| **Abstract** | The system must allow creating and saving a level within the video game. Levels are created automatically when the video game is initialized, i.e. the user cannot create a level. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Level id | int | If it is a new level and the array of levels has not been filled. |
| Score to level up | int |
| **General activities necessary to obtain the results** | * The program asks for the information to create a level. * The level is created and added to the array of levels within the video game. Difficulty, enemy array and treasure array are initialized automatically. | | |
| **Result or postcondition** | The program displays a confirmation message whether or not the level was added. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Confirmation message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R3:**  Register and add treasure to a level. | | |
| **Abstract** | The system must allow creating and saving a treasure within a level. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Name of treasure | String | If it is a new treasure within the level and if the array of treasures has not been filled. |
| Treasure image url | String |
| The score that the treasure will increase. | int |
| **General activities necessary to obtain the results** | * The program asks for the information to create a treasure. * The treasure is created and added to the array of treasures within a level. The position of the treasure within the video game is initialized and generated automatically. | | |
| **Result or postcondition** | The program displays a confirmation message whether or not the treasure was added. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Confirmation message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R4:**  Register and add enemy to a level. | | |
| **Abstract** | The system must allow creating and saving a enemy within a level. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Name of enemy | String | If the enemy does not exist within the level and the array of enemies is not filled. |
| Enemy type | String |
| Enemy decreasing score | int |
| Enemy increasing score | int |
| **General activities necessary to obtain the results** | * The program asks for the information to create a enemy. * The treasure is created and added to the array of enemies within a level. The position of the enemy within the video game is initialized and generated automatically. | | |
| **Result or postcondition** | The program displays a confirmation message whether or not the enemy was added. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Confirmation message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R5:**  Set difficulty of a level. | | |
| **Abstract** | The system should allow to calculate the difficulty of the level based on the score that the enemies decrease and the score that the treasures increase. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Score awarded for treasures. | int | If there are enemies and treasures inside the level. |
| Score lowered by enemies. | int |
| **General activities necessary to obtain the results** | * The program obtains the scores that increase the treasures and decrease the enemies of a level. * The program compares the scores and classify the level in low, medium or high difficulty depending on the comparison of the scores. | | |
| **Result or postcondition** | A message with the difficulty of the level is displayed. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Confirmation message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R6:**  Modify a player's score. | | |
| **Abstract** | The program must allow changing the score of a given player. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Nickname of player | String |  |
| New score of player | int | If the player exists |
| **General activities necessary to obtain the results** | * The program asks for the player's nickname and searches for it within the game. * If the program finds it, it asks for the new score of the player in question and sets the score. | | |
| **Result or postcondition** | The program displays a confirmation message if the player's score has been modified or not. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Confirmation message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R7:**  Increase level for a player. | | |
| **Abstract** | The program must allow a player to level up, if he/she meets the score to do so. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Nickname of player | String |  |
| Id of level | int | If the player exists |
| **General activities necessary to obtain the results** | * The program asks for the player's nickname and searches for it within the game. * If the program finds the player, it asks for the level he/she is going to move up to and if he/she meets the required score, he/she moves up. | | |
| **Result or postcondition** | A confirmation message is displayed to confirm whether or not the player has been leveled up. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Confirmation message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R8:**  Report the treasures and enemies of a level given by the user. | | |
| **Abstract** | The program must be able to display all enemies and treasures within a specific level given by the user. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Id of level | int |  |
| **General activities necessary to obtain the results** | * The program asks for the Id of the level in which you want to display the enemies and treasures. * If the program finds the level, it prints on the screen all the enemies and treasures in the level. | | |
| **Result or postcondition** | A message is displayed with all the enemies and treasures of the level or saying that the searched level does not exist. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R9:**  Report the amount found of a treasure in all levels. | | |
| **Abstract** | The program must be able to display all the treasures with the same name that are inside all the levels. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Name of treasure | String |  |
| **General activities necessary to obtain the results** | * The program asks for the name of the treasure to search for. * If the program finds it, it displays the number of treasures that have the same name at all levels. | | |
| **Result or postcondition** | A message is displayed with all the treasures that have the same name at all levels or shows that the treasure does not exist. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R10:**  Report the amount found of an enemy in all levels. | | |
| **Abstract** | The program must be able to display all the enemies with the same type that are inside all the levels. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
| Type of enemy | String |  |
| **General activities necessary to obtain the results** | * The program searches for enemies with the same type. * If the program finds them, it counts the number of enemies of the same type in all levels and displays them. | | |
| **Result or postcondition** | A message is displayed with all the enemies that have the same type at all levels or shows that the enemy does not exist. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R11:**  Report the most repeated treasure in all levels. | | |
| **Abstract** | The program must be able to show the treasure that is repeated most often at all levels. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
|  |  |  |
| **General activities necessary to obtain the results** | * The program goes through the array of levels, and within each level enters the array of treasures. * Within the array of treasures of each level, the program counts the number of treasures depending on the name. * The amounts of all the treasures are compared and the most repeated one is shown. | | |
| **Result or postcondition** | A message is displayed with the most repeated treasure in all levels or a message is displayed on the screen stating that there are no treasures in the video game, in case there are none. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R12:**  Inform the enemy that gives the highest score and the level in which it is. | | |
| **Abstract** | The program must be able to display the enemy that gives the highest score if defeated and at what level it is. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
|  |  |  |
| **General activities necessary to obtain the results** | * The program goes through the array of levels and within each level it goes through the array of enemies. * The program obtains the score attribute that increases each enemy and compares them, saving the position of the level and the enemy that gives the highest score. | | |
| **Result or postcondition** | A message is displayed with the enemy that gives the highest score and the level where it is located or shows that there is no enemy that gives the highest score in case there are no enemies. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R13:**  Report the number of consonants found in the names of enemies in the video game. | | |
| **Abstract** | The system must allow to count the consonants that have the names of the enemies of the video game. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
|  |  |  |
| **General activities necessary to obtain the results** | * The program enters the array of levels and within each level enters the array of enemies, where it obtains the name attribute of each enemy. * A cycle is made that goes through all the name of the enemy and a condition is made that every time a consonant is found, a counter increases. | | |
| **Result or postcondition** | A message is displayed with the number of consonants of the enemies in the video game or it is shown that there are no enemies in the video game to count the consonants. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name or identifier** | **R14:**  Inform the top 5 of the players according to the score. | | |
| **Abstract** | The system should allow to show the top 5 of the best players according to their score. | | |
| **Inputs** | **Input name** | **Datatype** | **Selection or repetition condition** |
|  |  |  |
| **General activities necessary to obtain the results** | * The system enters the array of players and obtains the attribute score of each player. * Through a cycle that repeats 5 times, the program goes through the array of players, putting, in descending order, the players with the highest score. | | |
| **Result or postcondition** | A message is displayed with the 5 highest scoring players, or a message is displayed stating that there are no players in case there are no players yet. | | |
| **Outputs** | **Output name** | **Datatype** | **Selection or repetition condition** |
| Message | String |  |