|  |  |
| --- | --- |
| Client | A videogame company |
| User | Players |
| Functional requirements | **R1**: Register of players  **R2:** Register of levels  **R3:** Register of treasures  **R4:** Register of enemies  **R5:** Screen resolution  **R6:** Know player level  **R7:** Add treasures to level  **R8:** Add enemy to level  **R9:** Modify player score  **R10:** Increase level of a player  **R11:** Show treasures and enemies by level  **R12:** Show the same treasure in every level  **R13:** Show the same type of enemy in all the levels  **R14:** Show the most repeated treasure  **R15:** Enemy that gives the highest score  **R16:** Consonants in enemy’s name  **R17:** Top 5 |
| Problem context | * The videogame company needs a video game, the game consists of 10 levels in which the player collects treasures and fights with enemies. |
| No functional requirements | **RN1:** the display of the treasures and enemies of a level in the web application should not take more than 2 seconds  **RN2:** the system must work on websites and mobile applications. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R1**: Register of players | | |
| Resume | The system must allow the register of players, they have a nickname that identifies them, a name, the initial score (the player starts with 10) and have a few lives (starts with 5). | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| nickname | String | It cannot be repeated, must be unique for each player |
| name | String | Every player has a name |
| General activities necessary to obtain the results | Save the inputs and all player data | | |
| Result or postcondition | Register a player | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| creationMessage | String | Mention if the player was created or if there was an error in the process |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R2**: Register of levels | | |
| Resume | The system must allow automatically register the different levels of the game. A level has a number that identifies it and the points that are required to go to the next level. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| idNum | String | It cannot be repeated, must be unique for each level |
| pntsNd | String | Each level has a minimum amount to allow the player to go to the next level |
| General activities necessary to obtain the results | Save the inputs and all level data | | |
| Result or postcondition | Register a level | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| creationMessage | String | Mention if the level was created or if there was an error in the process |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R3**: Register of treasures | | |
| Resume | The treasures are the mechanism that the players have to reach the points to pass a level. Of the treasures of a level, there is the name, a URL to the image that represents it, the score it gives the player when it is found and the position X and Y in pixels in which it is located (these positions X and Y are generated randomly, according to the resolution of the screen used). | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| name | String | It can be repeated for each level |
| uRL | String | Leads to a page with an image of the treasure being register |
| givSc | int | Score given by the treasure and added to the player’s points |
| General activities necessary to obtain the results | Save the inputs and all treasure data | | |
| Result or postcondition | Register a treasure | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| creationMessage | String | Mention if the treasure was created or if there was an error in the process |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R4**: Register of enemies | | |
| Resume | Enemies decrease the player’s score. They have a name (Identifier), a type, the score that remains in case the player beats the player, the score that is added if he is defeated and the position X and Y in pixels in which he is (these X and Y positions are generated randomly, according to the resolution of the screen used (see annex)). In a level the enemies cannot be repeated because once defeated, you would already know how to defeat the others of the same level. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| name | String | It cannot be repeated, must be unique for each level |
| type | Type | Determines the class of enemy, be it ogres, abstract, boss, or magical |
| subtrScr | int | Score that the enemy subtracted to the player’s points |
| sumScr | int | Score added to the player’s points when the enemy is defeated |
| General activities necessary to obtain the results | Save the inputs and all enemy data | | |
| Result or postcondition | Register an enemy | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| creationMessage | String | Mention if the enemy was created or if there was an error in the process |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R5**: Screen resolution | | |
| Resume | The system must ask the user what is the resolution on which the game will be executed, and this information will be used when randomly generating the positions on the screen. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| resolScr | String | Save user choice |
| General activities necessary to obtain the results | Save the inputs and all enemy data | | |
| Result or postcondition | Once the user’s choice of screen resolution is saved, this information should be used to create the position of enemies and treasures. | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| creationMessage | String | Mention if enemies and treasures have been created from the chosen resolution or if an error occurred |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R6**: Know player level | | |
| Resume | The system must allow establishing the level of the player. Considering the player’s score and the score required to pass the level. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| selPlayer | String | The user enters the nickname of the player whose wants to know the level |
| General activities necessary to obtain the results | The system compares if what the user typed is equal to the player’s nickname, if it is, then the player’s score is extracted and compared with the score of each level | | |
| Result or postcondition | It is known in which level the player is located | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| lvlMessage | String | Shows the player’s level |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R7**: Add treasures to level | | |
| Resume | The system must add treasures for each level. In a level the same treasure can be found in different positions, that is, a diamond can be found in two different positions in the same level, so when entering the treasures into the game, the user will receive the question of how many treasures to register for a same level. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| selTre | String | The user enters the name of the treasure that wants to add to the level |
| selLevel | String | The user chooses the level. |
| rptTimes | int | Times that a treasure is added to a specific level |
| General activities necessary to obtain the results | The system compares what the user typed with the name of the treasure and if they are the same, adds the treasure the number of times typed for the user to the level chose. | | |
| Result or postcondition | One or more treasures are added to a specific level. | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| tTolvlMes | String | Mentions if the treasure is added to the level or not |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R8**: Add enemy to level | | |
| Resume | The system must add an enemy for each level. In a level the enemies cannot be repeated because once defeated, you would already know how to defeat the others of the same level. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| selEne | String | The user enters the name of the enemy that wants to add to the level |
| selLvl | String | The user chooses the level. |
| General activities necessary to obtain the results | The system compares what the user typed with the name of the enemy and if they are the same, adds the to the level chose. | | |
| Result or postcondition | One enemy is added to a specific level. | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| eTolvlMes | String | Mentions if the enemy is added to the level or not |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R9**: Modify player score | | |
| Resume | The system must be able to modify the player score when a treasure is added, for example. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | The player’s initial score (10 points) points will be added or subtracted depending on what the player does, like find a treasure or fight with an enemy | | |
| Result or postcondition | Player’s score increases or decreases | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| eTolvlMes | String | Shows the score the player had before and after the changes |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R10**: Increase level for a player | | |
| Resume | The system must be able to level up a player when the player has the necessary points. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | The system compares the required score of each level with the player’s score. If the player’s score is within the range of the level, this will be the player’s new level, if he continues at the same level, it will be reported how many points he needs to go up to the next level. | | |
| Result or postcondition | The player’s level is increased and/or a message is generated. | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| incLMes | String | Mentions if the player’s level increase or not |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R11**: show treasures and enemies by level | | |
| Resume | The system must be able to report the treasures and enemies (separated by commas) of a level given by the user | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| SelLev | String | The user enters what level wants to know the enemies and treasures |
| General activities necessary to obtain the results | The system search for all the enemies and treasures that have been added to the same level and show them | | |
| Result or postcondition | Show enemies and treasures | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| eTMes | String | Show enemies and treasures in the level chosen by the user |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R12**: Show the same treasure in every level | | |
| Resume | The system must be able to level up a player when the player has the necessary points. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| selTrsr | String | The user types the treasure that wish to know the quantity |
| General activities necessary to obtain the results | The system compares if what the user typed is equal to the treasure name, if it is, then look for this treasure in the ten levels. | | |
| Result or postcondition | Amount of the same treasure in all levels | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| taLMes | String | Mentions the amount of the same treasure in all levels |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R13**: Show the same type of enemy in all the levels | | |
| Resume | Inform the amount found of a type of enemy in all levels, that is, if the user wants to know how many ogres exist in all levels. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| selTnm | String | The user enters the type of enemy that wish to know the quantity |
| General activities necessary to obtain the results | The system compares if what the user typed is equal to the enemy’s type, if it is, then look for this enemies in the ten levels. | | |
| Result or postcondition | Amount of the same enemy’s type in all levels | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| eaLMes | String | Mentions the amount of the same type of enemy in all levels |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R14**: Show the most repeated treasure | | |
| Resume | Inform the most repeated treasure in all levels. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | The system checks all the levels and counts how many times it finds the same treasure, then compares these sums to know which is the treasure that is more times in all levels | | |
| Result or postcondition | It’s known the treasure that more times is in the game | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| rptMes | String | Mentions the most repeated treasure |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R15**: Enemy that gives the highest score | | |
| Resume | Inform the enemy that gives the highest score and the level where it is located. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | The system checks all the levels and search for the one who gives the player the highest score when it is defeated comparing the sumScr of the differences enemies. | | |
| Result or postcondition | It’s known the enemy that more score gives to the player | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| eHMes | String | Mentions the enemy that gives more score |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R16**: Consonants in enemy’s name | | |
| Resume | Inform the number of consonants found in the names of enemies in the game. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
| selEnem | String | The user enters the name of the enemy that wants to know how many consonants has in its name |
| General activities necessary to obtain the results | The system converts the entire String of name to lowercase and compares if the letters are equal to the vowels, if so, adds to a counter variable. Otherwise, it adds one to another variable that will be the one that keeps track of how many consonants there are in the name of the enemy. | | |
| Result or postcondition | It’s known the number of consonants in the enemy’s name. | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| eCMes | String | Show the quantify of consonants that the enemy has in its name |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R17**: Top 5 | | |
| Resume | Inform the top 5 of the players according to the score. | | |
| Inputs | Input name | Datatype | Selection or repetition condition |
|  |  |  |
| General activities necessary to obtain the results | The system compares the score of each player and keeps them to find out who are the five best players among all. | | |
| Result or postcondition | Five best players among the twenty that exist | | |
| Outputs | Output name | Datatype | Selection or repetition condition |
| tfMes | String | Show the top five players |

|  |  |  |
| --- | --- | --- |
| **Functional requeriments** | **Name of the Class** | **Name of the Method** |
| R1. Register of players | VideoGame | registerPlayer(String nickname, String name) |
| VideoGameController | showPlayer() |
| Player | registerPlayer() |
| R2. Register of levels | VideoGame | createdLvl(int num, int score) |
| VideoGameController | createdLvl() |
| Level |  |
| R3. Register of treasures | VideoGame | addTreasureToLevel(String name, String url, int givSc, int numLvL, int times) |
| VideoGameController | showTreasure() |
| Treasure | addTreasuretolvl() |
| R4. Register of enemies | VideoGame | registerEnemy(String name, int type, int subtrScr, int sumScr, int numLvL) |
| VideoGameController | showEnemy() |
| Enemy | registerEnemy() |
| R5. Screen resolution |  |  |
|  |  |
| R6. Know player level | VideoGameController | showPlayer() |
| R7. Add treasures to level | VideoGame | addTreasureToLevel(String name, String url, int givSc, int numLvL, int times) |
| VideoGameController | addTreasuretolvl() |
| Treasure |  |
| R8. Add enemy to level | VideoGame | registerEnemy(String name, int type, int subtrScr, int sumScr, int numLvL) |
| VideoGameController | registerEnemy() |
| Enemy |  |
| R9. Modify player score | modifyPlayerScore() | searchPlayer(String nickName) |
|  | modifyScore(String nickname, int newScore) |
| R10. Increase level of a player | VideoGame | changePlayerLvL() |
| VideoGameController | changePlayerLevel(nickname) |
| Player |  |
| R11. Show treasures and enemies by level | VideoGame | enemiesAndTreasuresOneLevel() |
| VideoGameController | trAndEneLevel(int level) |
| R12. Show the same treasure in every level | VideoGame | amountTrsGame() |
| VideoGameController | treasuresAllLevels(String name) |
| R13. Show the same type of enemy in all the levels | VideoGame | amountEnmGame() |
| VideoGameController | enemiesAllLevels(int type) |
| R14. Show the most repeated treasure | VideoGame | mostRepeatTreasure() |
| VideoGameController | mostRepeatTrsr() |
| R15. Enemy that gives the highest score | VideoGame | mostScoreEnemy() |
| VideoGameController | mostScoreEnm() |
| R16. Consonants in enemies’ names | VideoGame | consonantsInEnemyName() |
| VideoGameController | countCnsEnmName() |
| R17. Top 5 | VideoGameController | getTop(int[] array) |
|  | getAllScores() |
|  | getOrderedScores(int[] scores) |
|  | showTop5(int[] orderedArray) |