**SPECIFICATION TABLE OF THE SOFTWARE ENGINEERING PROBLEM, showing the following elements**

|  |  |
| --- | --- |
| CLIENT | Video Game Company |
| USER | Developers (User) |
| FUNCTIONAL REQUIREMENTS | **R-F1:** Create players.  **R-F2:** Register enemies for a level.  **R-F3:** Register treasures for a level.  **R-F4:** Change a player's score.  **R-F5:** Increase the level for a player. |
| CONTEXT OF THE PROBLEM | * What are the maximum treasures per level? * What is the maximum number of enemies per level? |
| NON-FUNCTIONAL REQUIREMENTS | **R-NF1:** The application must be developed for native web and mobile environments (Android and Apple).  **R-NF2:** The treasure deployment must be done in less than 2 seconds.  **R-NF3:** This version of the game has a resolution of 1280 x 720.  **R-NF4:** This version of the game has a maximum of 20 players, 10 levels, 50 treasures and 25 enemies. |

**Functional Requirements Analysis Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R-F1:** Create players. | | |
| Resume | It allows registering players through their nickname (player's unique id), name, initial score (10) and number of lives (5). | | |
| Inputs | Input Name | Data Type | Choice or Repeat Condition |
| playerNickName | String | If the nickname is available |
| playerName | String |  |
| initialScore | double | Default: 10 |
| initialLives | Int | Default: 5 |
| initialLevel | Int | Default: 0 (First Level) |
| General activities necessary to obtain the results | If the player limit hasn’t yet been reached (20), and the nickname is not repeated. | | |
| Result or postcondition | The player is successfully registered. | | |
| Outputs | Input Name | Data Type | Choice or Repeat Condition |
| newPlayer | Player | All data must be supplied  The Player Limit hasn’t yet been Reached. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R-F2:** Register enemies for a level. | | |
| Resume | It allows registering enemies through their name (enemy's unique id), a type (Enum Type), the score that stays in case he beats the player, the score that is added to him if he is defeated. The X and Y position in pixels (When registering in a level, positions generate randomly) | | |
| Inputs | Input Name | Data Type | Choice or Repeat Condition |
| enemyName | String | If the name is available |
| enType | int |  |
| loseScore | double |  |
| winLives | double |  |
| General activities necessary to obtain the results | If the enemy limit has not yet been reached (25), and the name is not repeated. | | |
| Result or postcondition | The enemy is successfully registered. | | |
| Outputs | Input Name | Data Type | Choice or Repeat Condition |
| newEnemy | Enemy | All data must be supplied  The Enemy Limit has not yet been Reached. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R-F3:** Register treasures for a level. | | |
| Resume | It allows registering treasures through their name, an image URL, the score that is added to player if it is founded. The X and Y position in pixels (When registering in a level, positions generate randomly) | | |
| Inputs | Input Name | Data Type | Choice or Repeat Condition |
| trName | String |  |
| ImageURL | String |  |
| award Score | double |  |
| locatedLevel | Int | The level where the treasure will be found |
| trInLvl | Int | The number of treasures in level |
| General activities necessary to obtain the results | If the treasure limit has not yet been reached (50) | | |
| Result or postcondition | The treasures were successfully registered. | | |
| Outputs | Input Name | Data Type | Choice or Repeat Condition |
| newTreasure | Treasure | All data must be supplied  The Treasure Limit has not yet been Reached. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R-F4:** Change a player's score. | | |
| Resume | It allows registering treasures through their name, an image URL, the score that is added to player if it is founded. The X and Y position in pixels (When registering in a level, positions generate randomly) | | |
| Inputs | Input Name | Data Type | Choice or Repeat Condition |
| modPlayer | String | The nickname of the player exists |
| playerNewScore | Double |  |
| General activities necessary to obtain the results | All data must be supplied. | | |
| Result or postcondition | Score Successfully Changed. | | |
| Outputs | Input Name | Data Type | Choice or Repeat Condition |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R-F5:** Increase the level for a player. | | |
| Resume | Increases a player's current level automatically when the player's score increases. | | |
| Inputs | Input Name | Data Type | Choice or Repeat Condition |
| modPlayer | String | The nickname of the player exists |
| playerNewScore | Double |  |
| General activities necessary to obtain the results | All data must be supplied. It also returns the missing points to reach the next level. | | |
| Result or postcondition | Score Successfully Changed. | | |
| Outputs | Input Name | Data Type | Choice or Repeat Condition |
|  |  |  |