Andrés Felipe Cabezas A00394772

Client	Video game company
User	Developer and administrator
Functional requirements	R1 Initialize Levels R2 Create a player R3 Register treasure at a level R4 Register enemy at a level R5 Modify a player's score R6 Increase a player's level R7 Show the amount of treasure found in all levels R8 Show the amount of an enemy type found in all levels R9 Show the amount found of a treasure at all levels R10 Show the most repeated treasure in all levels. R11 Show the enemy that gives the highest score and the level where it is located. R12 Show the number of consonants found in the names of the enemies in the game. R13 Show the top 5 players according to the score. R14 Generates a random Position X and Y for treasures and enemies
Problem context	The game has a menu with 12 options The user enters an input with the option you want to make The program has 10 levels Maximum of 50 treasures in the game Maximum of 20 players in the game Maximum of 25 enemies in the game
Nonfunctional requirements	RN2 It takes no more than two seconds to deploy the treasures it does not take more than 2 seconds in the web application RN3 The app must be compatible with a web application and a mobile app

Name or identifier	R1 Initialize levels		
Summary	The method initialize 10 levels to the game with all its information(idNumber level, scoreRequired and his difficult		
Inputs	Input name	Data type	Selection or repetition
General activities necessary to obtain the results	1 Init the 10 levels 2 Add the levels to the array of levels of the game 4 Calculates the difficult of the level		
Result or postcondition	Levels initialized and added to the game		
Outputs	Output name	Data type	Selection or repetition condition
	void		

Name or identifier	R2 Create a player		
Summary	Creates a player in the video game with his information with the user inputs (nickName and namePlayer) then initialize the lifes and score of the player		
Inputs	Input name	Data type	Selection or repetition
	nickName	String	
	namePlayer	String	That the entered nickName is not repeated in the video game
General activities necessary to obtain the results	1 Receives the info of the player 2 Check if the nickName is not repeated 3 Create the object person 4 Initialize the lifes and the score of the player 5 Check if there is space to add the person 6 Add the player to the Video game and to the level 7 Return msj		
Result or postcondition	A message that confirms if the player was added successfully		
Outputs	Output name	Type data	Selection or repetition condition
	msj	String	Than inputs and the method processes have worked correctly

<table>
<tr><td>Name or identifier</td><td colspan="3">R3 Register a treasure to a level</td></tr>
<tr><td>Summary</td><td colspan="3">Register a treasure with his info and his quantity into the array of treasures of a level and to the array of treasures of video game</td></tr>
<tr><td rowspan="4">Inputs</td><td>Input name</td><td>Data type</td><td>Selection or repetition</td></tr>
<tr><td>nameTreasure</td><td>String</td><td></td></tr>
<tr><td>urlTreasure</td><td>String</td><td rowspan="2">That there is no problem with the previous entry</td></tr>
<tr><td>scoreTreasure</td><td>String</td></tr>
<tr><td>General activities necessary to obtain the results</td><td colspan="3">1 Receives the info of treasure
2 Receives the quantity of treasures
3 If there is space in the game create an object treasure
4 Add the treasure to the Video Game and to the level
5 Generates a position X and Y (1280 x 720) to the treasure different for each one
6 Return msj</td></tr>
<tr><td>Result or postcondition</td><td colspan="3">A message that confirms if the treasure was added successfully</td></tr>
<tr><td rowspan="2">Outputs</td><td>Output name</td><td>Type data</td><td>Selection or repetition condition</td></tr>
<tr><td>msj</td><td>String</td><td>Than inputs and the method processes have worked correctly</td></tr>
</table>

<table>
<tr><td>Name or identifier</td><td colspan="3">R4 Register a enemy to a level</td></tr>
<tr><td>Summary</td><td colspan="3">Register an Enemy with his info (idEnemy, typeEnemy, damageEnemy, scoreEnemy) and add it to the the video game and the level, if there is space in the game, and if there isn't the same enemy on the level</td></tr>
<tr><td rowspan="5">Inputs</td><td>Input name</td><td>Data type</td><td>Selection or repetition</td></tr>
<tr><td>nameEnemy</td><td>String</td><td></td></tr>
<tr><td>typeEnemy</td><td>String</td><td rowspan="3">That the nameEnemy is not repeated in the level and there is space in the game</td></tr>
<tr><td>damageEnemy</td><td>int</td></tr>
<tr><td>scoreEnemy</td><td>int</td></tr>
<tr><td>General activities necessary to obtain the results</td><td colspan="3">1 Receives the enemy info
2 Check if the enemy is not repeated on the level
3 if there is space on the game creates the object Enemy
4 Add the enemy to the Video game and to the level
5 Generates a position X and Y (1280 x 720) to the enemy different for each one
6 Return msj</td></tr>
<tr><td>Result or postcondition</td><td colspan="3">A message that confirms if the enemy was added successfully</td></tr>
<tr><td rowspan="2">Outputs</td><td>Output</td><td>Type data</td><td>Selection or repetition condition</td></tr>
<tr><td>msj</td><td>String</td><td>Than inputs and the method processes have worked correctly</td></tr>
</table>

Name or identifier	R5 Modify a players score		
Summary	Modify the score of a player with his id and the newScore		
Inputs	Input name	Data type	Selection or repetition
	nickName	String	
	newScore	id	If the idPlayer is found on the game
General activities necessary to obtain the results	1 Receives the nickName of the player 2 Check if the player exist on the game 3 Set the newScore to the player 5 Return msj		
Result or postcondition	A message that confirms if the score was modified successfully		
Outputs	Output name	Type data	Selection or repetition condition
	msj	String	Than inputs and the method processes have worked correctly

Name or identifier	R6 Increase a player level		
Summary	This method check if the player have the score required to pass next level		
Inputs	Input name	Data type	Selection or repetition
	nickName	String	
	idNumberLevel	Int	If the nickName is found on the game
General activities necessary to obtain the results	1 Receives the id of the player 2 Check if the nickName exist on the game 3 Get the score of the player 4 Search the level by his id and get his scoreRequired 5 Check the scoreRequired for the level 6 Compare the scores and If the score of the player is greater than the score required the player level up 7 Return msj		
Result or postcondition	A message that confirms if the player level up or the score required		
Outputs	Output	Type data	Selection or repetition condition
	msj	String	Than inputs and the method processes have worked correctly

Name or identifier	R7 Show the treasures and enemies (separated by comma) of a level given by the user.		
Summary	Show the treasures and enemies of a level that the user input		
Inputs	Input name	Data type	Selection or repetition
	idNumberLevel	int	
General activities necessary to obtain the results	1 Receives the idNumberLevel of the level 2 Check if the level exist 3 Get the info of the treasures and enemies of the level and enumerate them 4 Return msj		
Result or postcondition	A message that shows all the enemies and treasure of the level		
Outputs	Output	Type data	Selection or repetition condition
	msj	String	Than inputs and the method processes have worked correctly

Name or identifier	R8 Show the amount of an enemy type found in all levels		
Summary	Search a type of enemy of the game and show how many there are		
Inputs	Input name	Data type	Selection or repetition
	typeEnemy	String	
General activities necessary to obtain the results	1 Receives the type of the enemy to search 2 Search the enemy in the array of enemies on the game 3 Count the enemies founded 4 Return msj		
Result or postcondition	A message that shows all the type of enemies that user search of the game		
Outputs	Output	Type data	Selection or repetition condition
	msj	String	Than inputs and the method processes have worked correctly

Name or identifier	R9 Show the amount of a treasure found in all levels		
Summary	Search a treasure that the user want and show how many of there are		
Inputs	Input name	Data type	Selection or repetition
	treasureToSearch	String	
General activities necessary to obtain the results	1 Receives the name of the treasure to search 2 Search the treasure in the array of treasures of the game 3 Count the treasures founded 4 Return msj		
Result or postcondition	A message that shows how many of that treasure are on the game		
Outputs	Output	Type data	Selection or repetition condition
	msj	String	Than inputs and the method processes have worked correctly

Name or identifier	R10 Show the most repeated treasure all levels		
Summary	Search the most repeated treasure of the game and shows what it is		
Inputs	Input name	Data type	Selection or repetition
General activities necessary to obtain the results	1 Count the quantity of treasures 2 Return msj		
Result or postcondition	A message that shows the most repeated treasure of the game		
Outputs	Output	Type data	Selection or repetition condition
	msj	String	The method processes have worked correctly

Name or identifier	R11 Show the enemy that gives the highest score and the level where it is located		
Summary	Search a treasure that the user want and show how many of there are		
Inputs	Input name	Data type	Selection or repetition
General activities necessary to obtain the results	1 Search the enemies of the game 2 Get the score of the enemy 3 Compare the highest score and if is highest get his location 4 Return msj		
Result or postcondition	A message that shows the enemy that gives the highest score and where is it		
Outputs	Output	Type data	Selection or repetition condition
	msj	String	The method processes have worked correctly

Name or identifier	R12 Show the number of consonants found in the names of the enemies in the game.		
Summary	Count the number of consonants in the name of all the enemies in the game		
	Input name	Data type	Selection or repetition
General activities necessary to obtain the results	1 Search the enemies of the game 2 Get the name of the enemy 3 Count the letters different to a,e,i,o,u for all the enemies of the game 4 Return msj		
Result or postcondition	A message that shows the count the consonants found in the name of the enemies on the game		
Outputs	Output	Type data	Selection or repetition condition
	msj	String	The method processes have worked correctly

Name or identifier	R13 Show the top 5 players according to the score		
Summary	Search the players on the game and compare the score of the players and shows the top 5		
	Input name	Data type	Selection or repetition
General activities necessary to obtain the results	1 Search the players of the game 2 Get the score of the player 3 If the score is in the top 5 get the name of the person 4 Return msj		
Result or postcondition	A message that shows the top 5 score of the players and his info		
Outputs	Output	Type data	Selection or repetition condition
	msj	String	The method processes have worked correctly

Name or identifier	R14 Generate random position X and Y for treasures and enemies		
Summary	Generate a random position X and Y for treasures and enemies and check if there isn't a repeated position		
	Input name	Data type	Selection or repetition
General activities necessary to obtain the results	1 Generate Random Position X and Y 2 Check if isn't a repeated position 3 set the position X and Y for the treasure or the enemy		
Result or postcondition	The new position of the treasure or enemy		
Outputs	Output	Type data	Selection or repetition condition
	void		

Functional Requirement	Class Name	Method
R1 Initialize Levels	Class VideoGame	initializeLevels()
	Class Level	Level(idNumberLevel: int, scoreRequired: int) sumEnemiesScore(): int sumTreasuresScore(): int calculalteDifficult():int
R2 Create Player	Class VideoGame	createPlayer(nickName: String, namePlayer: String): String addPlayerToGame(newPlayer: Player): String
	ClassPlayer	Player(nickName: String, namePlayer: String)
R3 Register Treasure	Class VideoGame	createTreasure(nameTreasure: String, urlTreasure: String, scoreTreasure: int): String addTreasureToGame(newTreasure: Treasure, quantityTreasures: int):String
	Class Level	addTreasure(newTreasure: Treasure, quantityTreasures: int):String
	Class Treasure	Treasure(nameTreasure : String, urlTreasure : String, scoreTreasure : String) generateRandomPositionX(): int generateRandomPositionY(): int
R4 Register Enemy	Class VideoGame	createEnemy(nameEnemy : String, typeEnemy : String, damageEnemy : int, scoreEnemy : int) : String addEnemyToGame(newEnemy: Enemy): String
	Class Level	addEnemy(newEnemy: Enemy) : String
	Class Enemy	Enemy(nameEnemy : String, typeEnemy : String, damageEnemy : int, scoreEnemy : int) generateRandomPositionX(): int generateRandomPositionY(): int
R5 Modify a player score	Class VideoGame	modifyPlayerScore(nickName: String, newScore: int): String searchPlayer(nickName: String): int
	Class Player	setScorePlayer(aScore: int): int

	Class Level	searchPlayer(nickName: String): int
	Class Player	setScorePlayer(aScore: int): int
R6 Increase a player Level	Class VideoGame	searchPlayerInLevels(nickName: String): int IncreasePlayerLevel(nickName : String, idNumberLevel : int) : String
	Class Level	getScoreRequired(): int
	Class Player	getNickName(): String getNamePlayer(): String getLifesPlayer(): int getScorePlayer(): int
	Class Level	deletePlayer(nickName: String): void addPlayer(newPlayer: Player): String
R7 Show treasures an enemies of a level	Class VideoGame	showLevelTreasuresAndEnemies(idNumberLevel : int) : String
	Class Level	getTreasures(): Treasure[] getEnemies(): Enemy[]
	Class Treasure	getNameTreasure(): String
	Class Enemy	getNameEnemy(): String getTypeEnemy(): String
	Class Level	countTreasures(nameTreasure: String): int
R8 Show the amount of an enemy type in the game	Class VideoGame	showTypeEnemiesInGame(typeEnemy : String) : String
	Class Enemy	getTypeEnemy(): String
R9 Show the amount of a treasure in the game	Class VideoGame	showTreasuresInGame(nameTreasure: String): String
	Class Treasure	getNameTreasure(): String
	Class VideoGame	countTreasures(nameTreasure): int
R10 Show the most repeated treasure in the game	Class VideoGame	showMostRepeatedTreasure(): String
	Class Treasure	getNameTreasure(): String
	Class VideoGame	countTreasures(nameTreasure): int
R11 Shows the with the Highest Score in the game	Class VideoGame	showHighestScoreEnemy() : String

	Class Level	getEnemies(): Enemy[]
	Class Enemy	getNameEnemy(): String getScoreEnemy(): int
R12 Show the amount of consonants in the name of the enemies in the game	Class VideoGame	showConsonantsOfEnemiesName() : String
	Class Enemy	getNameEnemy(): String
R13 Show top 5 players of the game	Class VideoGame	showTop5players(): String
	Class Player	getNamePlayer(): String getScorePlayer(): int
R14 Generate random position X and Y for treasures and enemies	Class Level	checkDifferentPositions(): void
	Class Enemy	getPosisitonXEnemy(): int getPosisitonYEnemy(): int
	Class Treasure	getPosisitonXTreaure(): int getPosisitonYTreasure(): int
	Class Enemy	setPositionXEnemy(aPositionX : int) : int setPositionYEnemy(aPositionY : int) : int
	Class Treasure	setPositionXTreasure(aPositionX : int) : int setPositionYTreasure(aPositionY : int) : int