SOFTWARE ENGINEERING PROBLEM SPECIFICATION TABLE

Video Game Integrative Task 2

CLIENT	Video game company
USER	Video game player
FUNCTIONAL REQUIREMENTS	RF001 - Set game resolution RF002 - Player registration RF003 - Level registration RF004 - Treasure registration RF005 - Enemy registration RF006 - Register Enemy to Level RF007 - Register Treasure to Level RF008 - Change player score RF009 - Increase player level RF010 - Print level treasures and enemies RF011 - Count treasure name RF012 - Count enemy type RF013 - Check most repeated treasure RF014 - Check most valuable enemy RF015 - Count enemy consonants RF016 - Print player top 5
CONTEXT OF THE PROBLEM	The program will be used by an administrator. Each level contains treasures and levels. The user is related to a level depending on his score.
NON-FUNCTIONAL REQUIREMENTS	- The game will have 10 levels and a maximum of: ○ 20 players ○ 50 treasures ○ 25 enemies - The game will calculate the positions of treasures and enemies depending on the

	resolution. - The application must display in no more than 2 seconds. - The application will work on the web and mobile devices as well

Functional requirements analysis

Name or identifier	RF001 - Set game resolution		
Summary	The system allows the user to select a game resolution which will be used to calculate the positions of the elements.		
Input	Input name	Data type	Selection or repetition condition
	resolution	int	It must be between 1 and 7
General activities necessary to obtain the results	- Display the resolution options - Get user input for resolution - Assign the given resolution for the game		
Result or post condition	- The resolution of the game is changed to the new value		
Output	Output name	Data type	Selection or repetition condition
	---	---	---

Name or identifier	RF002 - Player registration

Summary	The system allows the user to register a player based on user input.		
Input	Input name	Data type	Selection or repetition condition
	id	String	it must be unique
	name	String	
General activities necessary to obtain the results	- Get user id and name from input - Create a new Player object with given values (score and lives are assigned automatically) - check if the user id is already in the videogame; if so, display error message and stop the process. - Find the first empty place in player list and assign the new Player to that position. - If the player list was full, return an error message; otherwise, return the success message and a player list.		
Result or post condition	If the player is added: - Player list is modified with a new player defined in it.		
Output	Output name	Data type	Selection or repetition condition
	statusMessage	String	It will be either a success or error message
	playerList	String	

Name or identifier	RF003 - Level registration		
Summary	The system creates automatically the 10 levels, assigning them a score limit.		
Input	Input name	Data type	Selection or repetition condition
	---	---	---

General activities necessary to obtain the results	- Create a new Level object with automatic values for the i^{th} level score (default is i*100) - Find the first empty place of level list and assign the new level to it - Set the level id to the list position + 1, this way, level id will be enumerated from 1 to 10. - Set the difficulty to medium by default (there are 0 enemies and 0 treasures, so the difference of their score additions will be 0) - Repeat the whole process 10 times		
Result or post condition	The level list will be full with the new levels.		
Output	Output name	Data type	Selection or repetition condition
	---	---	---

Name or identifier	RF004 - Treasure registration		
Summary	The system allows the user to create a template of a treasure.		
Input	Input name	Data type	Selection or repetition condition
	name	String	
	pictureUrl	String	
	scoreAddition	double	
General activities necessary to obtain the results	- Get treasure name, picture url and score addition - Create a new Treasure object with given values (position is set to (-1, -1) since it is only a template)		

	- check if the treasure name is already in the videogame; if so, display error message and stop the process. - Find the first empty place in treasure list and assign the new Treasure to that position. - If the treasure list was full, return an error message; otherwise, return the success message and a treasure list.		
Result or post condition	If the treasure is added: - Treasure list is modified with a new treasure defined in it.		
Output	Output name	Data type	Selection or repetition condition
	statusMessage	String	It will be either a success or error message
	treasureList	String	

Name or identifier	RF005 - Enemy registration		
Summary	The system allows the user to create and add an enemy to a specified level		
Input	Input name	Data type	Selection or repetition condition
	id	String	
	type	int	it must be between 1 and 4
	scoreSubstraction	double	
	scoreAddition	double	

General activities necessary to obtain the results	- Get enemy id, type, score addition and score substraction - Create a new Enemy object with given values (position is set to (-1, -1) since it is not assigned to a level yet) - check if the enemy id is already in the videogame; if so, display error message and stop the process. - Find the first empty place in enemy list and assign the new Enemy to that position. - If the enemy list was full, return an error message; otherwise, return the success message and a enemy list.		
Result or post condition	If the enemy is added: - Enemy list is modified with a new enemy defined in it.		
Output	Output name	Data type	Selection or repetition condition
	statusMessage	String	It will be either a success or error message
	enemyList	String	

Name or identifier	RF006 - Register enemy to a level		
Summary	The system allows the user to select an enemy and assign it to a level		
Input	Input name	Data type	Selection or repetition condition
	enemyId	String	
	levelId	int	it must be between 1 and 10

<table>
<tr><td>General activities necessary to obtain the results</td><td colspan="3">- If there is not any enemy in the enemy list, stop the process and show an error message, else, show the available enemies to add.
- Get the enemy id and level id
- Search both enemy and level from their ids and assign them to variables.
- If enemy or level is not found, return an error message, and stop the process
- Generate a new position for the enemy and add it to the specified level.
- If the enemy is not in the level and the level’s enemy list is not full, add it to the first empty place.
- set the position to the new random one.
- Calculate and set the difficulty for the level.
- Return an error or success message and the statistics of the level (difficulty, enemies and treasures), including the new enemy and its position.</td></tr>
<tr><td>Result or post condition</td><td colspan="3">If the enemy is added:
- Enemy list is of the specified modified with a new enemy defined in it.</td></tr>
<tr><td rowspan="3">Output</td><td>Output name</td><td>Data type</td><td>Selection or repetition condition</td></tr>
<tr><td>statusMessage</td><td>String</td><td>It will be either a success or error message</td></tr>
<tr><td>objectList</td><td>String</td><td></td></tr>
</table>

Name or identifier	RF007 - Register treasure to a level
Summary	The system allows the user to select a treasure template and assign it to a level

Input	Input name	Data type	Selection or repetition condition
	treasureName	String	
	levelId	int	it must be between 1 and 10
General activities necessary to obtain the results	- If there is not any treasure in the treasure list, stop the process and show an error message, else, show the available treasures to add. - Get the treasure name, level id and the number of treasures to add. - Repeat the following the number of times given by the user: ○ Search both treasure and level from their ids and assign them to variables. ○ If treasure or level is not found, return an error message, and stop the process ○ Generate a new position for the treasure and add it to the specified level. ○ Create a copy of the template of the treasure so that the position of each treasure is different. ○ If the treasure is not in the level and the level's treasure list is not full, add it to the first empty place. ○ set the position to the new random one. ○ Calculate and set the difficulty for the level. - Return an error or success message and the statistics of the level (difficulty, enemies, and treasures), including the new treasure and its position.		
Result or post condition	If the treasure is added: - Treasure list of the specified level is modified with a new treasure defined in it.		
Output	Output name	Data type	Selection or repetition condition

	statusMessage	String	It will be either a success or error message
	objectList	String	

Name or identifier	RF008 - Change player score		
Summary	The system allows the user to modify the score of a given player		
Input	Input name	Data type	Selection or repetition condition
	playerId	String	
	newScore	double	it must be greater than the previous score
General activities necessary to obtain the results	- Search the player by his id and ensure that it is on the player list - check if the new score is greater than the current one. - Update the score with the new value.		
Result or post condition	If the score meets the condition: - Player score is updated with the new value.		
Output	Output name	Data type	Selection or repetition condition
	statusMessage	String	It will be either a success or error message

Name or identifier	RF009 - Increase player level		
Summary	The system allows the user to increase the level of a player based on his score		
Input	Input name	Data type	Selection or repetition condition
	playerId	String	
	levelId	int	it must be between 2 and 10
General activities necessary to obtain the results	- Search the player by his id and ensure that it is on the player list - Search the preceded level of the level given by the user. - if the player or the level does not exist in the video game, return an error message. - If the selected level is lower than the current, return an error message. - Compare the score limit of the searched level, if the player score is greater the player level is updated to the new one, else return an error message indicating how many points the user needs to get to that level.		
Result or post condition	If the score meets the condition: - User score is updated with the new value.		
Output	Output name	Data type	Selection or repetition condition
	statusMessage	String	It will be either a success or error message

Name or identifier	RF010 - Print treasures and enemies of a level

Summary	The system allows the user to select a level and print its enemies and treasures		
Input	Input name	Data type	Selection or repetition condition
	levelId	int	it must be between 1 and 10
General activities necessary to obtain the results	- Search the level by its id, and ensure that it exists in the level list - Get the enemies of that level, and add it to a text - Get the treasures of that level and add them to a text - Return the list of objects		
Result or post condition	---		
Output	Output name	Data type	Selection or repetition condition
	objectList	String	

Name or identifier	RF011 - Count treasure name		
Summary	The system prints the number of times that a given treasure appears in the whole game		
Input	Input name	Data type	Selection or repetition condition
	treasureName	String	
General activities necessary to obtain the results	- Declare a cumulative variable and initialize it to 0 - For each level repeat this: ○ For each treasure do: ▪ Compare the name given by the user to the name of the current treasure		

	▪ If they are the same, add 1 to the summation - Return the count		
Result or post condition	---		
Output	Output name	Data type	Selection or repetition condition
	Number of occurrences	int	

Name or identifier	RF012 - Count enemy type		
Summary	The system prints the number of times that a given enemy type appears in the whole game		
Input	Input name	Data type	Selection or repetition condition
	enemyType	int	It must be between 1 and 4
General activities necessary to obtain the results	- Declare a cumulative variable and initialize it to 0 - For each level repeat this: ○ For each enemy of the level do: ▪ Compare the enemy type of the level, with the type that corresponds to the number entered by the user. ▪ If they are the same, add 1 to the summation - Return the count		
Result or post condition	---		
Output	Output name	Data type	Selection or repetition condition

	Number of occurrences	int	

Name or identifier	RF013 - Check most repeated treasure		
Summary	The system prints the treasure that appears most times in the game and the number of occurrences.		
Input	Input name	Data type	Selection or repetition condition
	---	---	---
General activities necessary to obtain the results	- If the list of treasures is empty, return a message indicating that there are not treasures yet. - Get the max of treasures by doing: ○ Assign the most repeated treasure to the first position of the treasure list ○ Assign the max value to the count of that treasure. ○ For each treasure in the list, starting with the second: ▪ Check if the count of the current treasure is greater than the max, if so, make it the most repeated treasure. - If the max is 0, there are not treasures assigned yet, so, return message indicating that; else, return the name of the most repeated treasure		
Result or post condition	---		
Output	Output name	Data type	Selection or repetition condition
	Most repeated Treasure	String	

Name or identifier	RF014 - Check most valuable enemy		
Summary	The system prints the enemy that has the greatest score addition in the game, its value, and the level in which it appears		
Input	Input name	Data type	Selection or repetition condition
	---	---	---
General activities necessary to obtain the results	- Start with a null Enemy, and a max score of -1, which will be used to compare at first. - For each level, do: ○ Make the first enemy the most valuable ○ For each enemy starting at the second, do: ▪ Compare it with the most valuable enemy, only if the current enemy is not null ▪ If the score addition of the current enemy is greater, make it the most valuable of the level. ○ If the score addition of the most valuable enemy of the level is greater than the max, update the info (level, max, Enemy) of the most valuable enemy of the game. - If the max is -1, there is not any enemy assigned to a level (since all score additions are positive), in that case, return a message indicating that situation - Else, return the most valuable enemy (it is guaranteed that it is not null if all score additions are positive).		
Result or post condition	---		
Output	Output name	Data type	Selection or repetition condition

	Most valuable enemy	String	

Name or identifier	RF015 - Count enemy name consonants		
Summary	The system counts the number of consonants that are in all enemy names		
Input	Input name	Data type	Selection or repetition condition
	---	---	---
General activities necessary to obtain the results	- If there is not any enemy in the game return an error messsage - Initialize a cumulative variable for the consonants. - For each enemy in enemy list, do: ○ Get the name of the enemy ○ For each character in the name of the enemy: ▪ If the character converted to lowercase is a consonant (is not a vowel) add 1 to the sum of consonants. - Return the number of consonants		
Result or post condition	---		
Output	Output name	Data type	Selection or repetition condition
	Consonant number	int	

Name or identifier	RF016 - Print player top 5		
Summary	The system prints the name and score of the players that have the greatest score in the game.		

Input	Input name	Data type	Selection or repetition condition
	---	---	---
General activities necessary to obtain the results	- If there is not any player in the game. return an error message - get the number of real players that are in the game ○ Iterate the player list until finding an empty place - Initialize a list (number of players size) that will be the players ordered by score. - for each place of the new list, iterate the player list and get the max score, and add it to the new list ○ if the player is already in the top, the iterator will not evaluate it - Add the elements to the list to readable text, until 5. - Show the top to the user.		
Result or post condition	---		
Output	Output name	Data type	Selection or repetition condition
	User top 5	String	