Salt N Sepr

Group Members:
Alberto Lee
Archie Godfrey
Jacqueline Eilertsen
Jake Schoonbrood
Joshua Levett
Matteo Barberis

Introduction

Single Statement Of Need: The game, Kroy, is targeted at prospective students and their parents visiting York on open days. The game should showcase the scope of programming and design skills attained by 2nd year students and what the prospective student should learn.
After being given a broad overview of the game, we came together as a group to discuss different aspects of the games and the requirements for each class and general game assets. We broke the requirements into two large sections, visuals and functionality. Within visuals we looked into what assets we would need in order to have a simple yet visually appealing game. As no assets are provided, the easiest way to attain them would be from open source sites. We also looked at whether a menu would be needed and what it would need to allow the user to do.
In the functionality section, we broke down each unit in the game. For each unit we looked at what it should do and how it should interact with other units. We researched the IEEE requirements standard [1] in order to present our findings effectively, illustrated below in each of the tables. We then sent this information to the client
so that we could schedule a meeting and clarify that we had a good understanding of the requirements of the game, and discuss anything we had missed.

After meeting with the client, we learnt that there would be little to no sound on the Open Days the game would be played on. This means that the visuals used should be a large focus as they will be one of the main features that differentiate our game from other student's games. As well as this, prospective students will have a limited amount of time to play the game therefore it should be designed to finish in a short amount of time, 5-10 minutes.

The main focus of the software should be to run on a computer, however considerations should be made so that it can be easily ported to mobile devices. The main difference between mobile and PC is the way the user will interact with the system, using a keyboard rather than a touchscreen. Therefore the controls cannot be overly complex on PC in order to simplify the transition. Another reason for simple controls relates to the target audience. Not all students, and their parents, will be familiar with PC gaming. Therefore the controls must be easy and quick to learn.

User Requirements [2]

ID	Description	Priority
CONTROL_TRUCK	Control the direction the fire truck travels in	SHALL
CONTROL_SPRAY	Control the direction of the water the fire truck sprays	SHALL
RETURN_HOME	Return a firetruck to the fire station to repair and refill it	SHALL
VARIED_TRUCKS	Play as 4 different fire trucks	SHALL
GAIN_INCOME	The user should earn money/points from destroying aliens and/or their fortresses	SHALL

WIN_GAME	Once the user has destroyed all 6 different fortresses they win the game	SHALL
CREATE_MAP	The user should be able to explore a map by controlling the firetruck	SHALL
CREATE_ENTITIES	The user should encounter alien patrols throughout the map	SHALL
DESTROY_ENTITIES	The user should be able to destroy alien patrols and fortresses by spraying them with water	SHALL
NO_VIOLENCE	There will be no violence to appeal to target audience	SHALL
OPEN_SHOP	The user should be able to find out how much different fire trucks cost at any point in the game	MAY
BUY_ITEM	The user should be able to buy different fire trucks from a shop	MAY
MENU	There should be a menu so that the user can access the leaderboard, the minigame, edit settings or play the game	SHALL
LEADERBOARD	See a local leaderboard to compare scores against other players	MAY

Functional Requirements [2]

ID	Description	User Requirements
CONTROL_TRUCK_FUNC	When the user presses the WASD keys the fire truck will move in the appropriate direction	CONTROL_TRUCK
CONTROL_SPRAY_FUNC	The direction of the water cannon will be controlled by the mouse	CONTROL_SPRAY
RETURN_HOME_FUNC	When the firetruck returns to the firestation it will repair and refill over a defined amount of time	RETURN_HOME
FIXED_TIME	After a fixed amount of time the user cannot repair their fire truck at the fire station	RETURN_HOME
NO_HEAL	Aliens should not heal after taking damage	DESTROY_ENTITIES
FORTRESS_HEAL	Alien fortresses should heal over a duration after taking damage	DESTROY_ENTITIES
DESTROY_ENTITIES_FUNC	If the alien patrol or fortress runs out of health it should be removed from view	DESTROY_ENTITIES

VARIED_TRUCKS_FUNC	Able to swap between trucks using a single button	VARIED_TRUCKS
CREATE_MAP_FUNC	A section of the map should be displayed to the user so that they can navigate it	CREATE_MAP
OPEN_SHOP_FUNC	The user should be able to press a button and it then opens a shop GUI that allows the user to upgrade or buy new fire trucks.	OPEN_SHOP
BUY_ITEM_FUNC	When the user tries to buy an item it should compare the value of the item to the balance and if the user has enough, add the item to his inventory for use.	BUY_ITEM

Use Cases [2]

Scenario ID	Destroy fortress	Purchase item from shop	Lose game	Repair and refill fire truck
Primary Actor	Player of the game	Player of the game	Player of the game	Player of the game
Pre-condition	Player has water in their tank and fortress has health	Player is in the shop and has navigated to the item they want to buy	Player has no remaining fire trucks after a fire truck is destroyed	Player has moved their fire truck to the fire station
Trigger	Player sprays water at alien swarm	Player clicks the buy button	Player's fire truck is destroyed	Player's fire truck is on top of the fire station
Main Success Scenario	1) Player sprays at fortress 2) Fortress takes damage 3) Fortress' health reaches 0	1) Player clicks the buy button 2) They have enough money for the item	1) Player takes damage from an alien 2) Player's health reaches 0	1) Player's truck's health is increased over time
Secondary Scenarios	1) The Player stops spraying before the fortress' health reaches 0. The fortress then begins to heal 2) The Player runs out of water before the fortress' health	1) The Player does not have enough money. Purchase is cancelled and the Player is told why	1) Player stops taking damage before their health reaches 0. Game continues	1) Player moves away from fire station so repairing stops

	reaches 0. The fortress then begins to heal			
Success Post-condition	The fortress disappears from the scene	Player receives the item	End game screen is shown to player	Player's fire truck's health reaches its full value

Non-Functional Requirements [2]

ID	Description	User Requirements	Fit criteria
TIME_ACCESSIBILITY	After a fixed amount of time the user will no longer be able to repair fire trucks, therefore the game will always end.	FIXED_TIME	The game should be playable within 5-10 minutes due to limited time on open days
GAME_DOCUMENTATION	It should be easy to understand that the game is won by destroying all 6 alien fortresses. This can be done by a small tutorial	WIN_GAME	The game should be easily understandable for the target audience within a single play through
RESILIENCE	The game should only be won when all 6 alien fortresses are destroyed	WIN_GAME	If the game is won when exactly 6 fortresses are gone
AUDIENCE_ACCESSIBILITY	Instead of showing violence, the enemies will just disappear in order to satisfy the target audience.	NO_VIOLENCE	The game should be appropriate for prospective students and their parents
GAME_ACCESSIBILITY	The system must have a menu so that the user can access the main game and the minigame.	MENU	The game must have a minigame and therefore the user must be able to access it
OPERABILITY	The game should be playable on a PC but considerations should be made for mobile versions in the future.	CONTROL_TRUCK	Users will play the game on a PC on open day
SECURITY	The game should not ask for any sensitive information when displaying scores on the leaderboard. Instead, a nickname should be used	LEADERBOARD	The leaderboard will be displayed to lots of people and sensitive information should not be shared.

A more detailed requirements report can be found on our website

References

[1] "29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering - IEEE Standard", Ieeexplore.ieee.org, 2011. [Online]. Available: https://ieeexplore.ieee.org/document/6146379. [Accessed: 01- Nov- 2019].

[2] "Lecture 2: Requirements Engineering", *York VLE*, 2019. [Online]. Available: https://vle.york.ac.uk/bbcswebdav/pid-3188304-dt-content-rid-8697295_2/courses/Y2019-006404/Requirements%281%29.pdf. [Accessed: 01- Nov- 2019].