REQUIREMENTS AND ANALYSIS DOCUMENT FOR VOXEL GALAXY: ARENA

Version:1.0
Date:29/05-16
Author: Simon Mare, David Iliefski Janols, Hannes Thell, Lukas Wallner

This version overrides all previous versions.

CONTENTS

1 INTRODUCTION

1.1 PURPOSE OF APPLICATION

The main goal of the project is to implement a simple but entertaining desktop game where two players can compete against each other. We aim to give the game a high replay value through both level variety and good design, and an interesting competitive aspect.

1.2 GENERAL CHARACTERISTICS OF APPLICATION

The game will be a desktop, single- or multi-player application with a graphical user interface for the Windows/Mac/Linux platforms.

The game will be in real-time, which means that the application handles input from both players simultaneously. There are no time constraints to the application; a round of the game will be finished when either player has brought the other players health to zero, or when cancelled manually.

The game will be set on a flat stage where the players can move around freely in two dimensions and shoot their weapons at each other while also picking up power-ups.

1.3 SCOPE OF APPLICATION

In addition to playing the game against another human player, there is also the possibility of playing against an AI-controlled computer-player. The application will not record any data or statistics of previously completed game rounds.

1.4 OBJECTIVES AND SUCCESS CRITERIA OF THE PROJECT

It should be possible for one or two players to play and finish a round of the game that is both entertaining and runs smoothly. The game does not need to be highly advanced or plastered with functions, but rather characterized by smart design and a minimalist, but appealing graphical appearance.

1.5 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

- GUI, graphical User Interface
- Power-Up, a pick-up able object providing a boost of some kind to the player
- AI, artificial intelligence

2 REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS

The player(s) should be able to:

- Start a new game.
    1. Select player vs. player or player vs. computer.
- Control their character in the following ways
    1. Move their character in every direction.
    2. Aim and shoot their weapon independently of which direction they are moving in.
    3. Perform a dashing move, giving the player a quick speed burst.
- Pick up power-ups by running into them.
- Exit the application. This will end the ongoing round.

2.2 NON-FUNCTIONAL REQUIREMENTS

2.2.1 USABILITY

The game should be easy to learn, but difficult to master. More specifically, normal users should be able to pick up the game and complete a game-round within a short period of time, while advanced users who are more used to computer gaming should understand the game right from the get-go, but still be required to play several rounds to figure out the more advanced features and start forming their own strategies.

Tests with multiple users, both normal and advanced, should be performed to verify the usability. Test results should be part of the final documentation.

2.2.2 RELIABILITY

Use cases will be tested alongside the implementation phase to ensure that they are functional.

Final, integrated project code will be tested by third-party software to ensure that at least 80% of the code is covered at runtime, and is functioning properly.

2.2.3 PERFORMANCE

Since the game will be very fast paced, a low input response time will be required. The aim is to achieve a response time as low as possible while keeping a threshold at maximum 50ms.

2.2.4 SUPPORTABILITY

The application must be implemented so that the GUI can easily be switched out to suit other platforms than computers, including Web, Mobile Apps etc.

2.2.5 IMPLEMENTATION

To achieve platform independence the application will use the Java environment. All hosts must have the JRE installed and configured. The application needs to be installed on all hosts where it will run.

2.2.6 Packaging and installation

The application will be delivered as a zip-archive containing:

1. A file for the application code
2. All needed resources (models, textures etc.)
3. Start programs to start the game on the different platforms
4. A README-file documenting installation and start of application.

2.2.7 Legal

The game will be designed from scratch, including player and stage models. Therefore, no legal issues will be present.

2.3 Application models

2.3.1 Use case model

See APPENDIX for UML diagram and textual descriptions.

2.3.2 Use cases priority

1. Start Game
2. Move
3. Shoot
4. Taking Damage
5. Game Over
6. Rotate Gun
7. Dash

2.3.3 Domain model

See APPENDIX.

2.3.4 User interface

The application will use a 3D graphical user interface that will be able to take into account almost all kinds of screen sizes. See APPENDIX for screenshots.

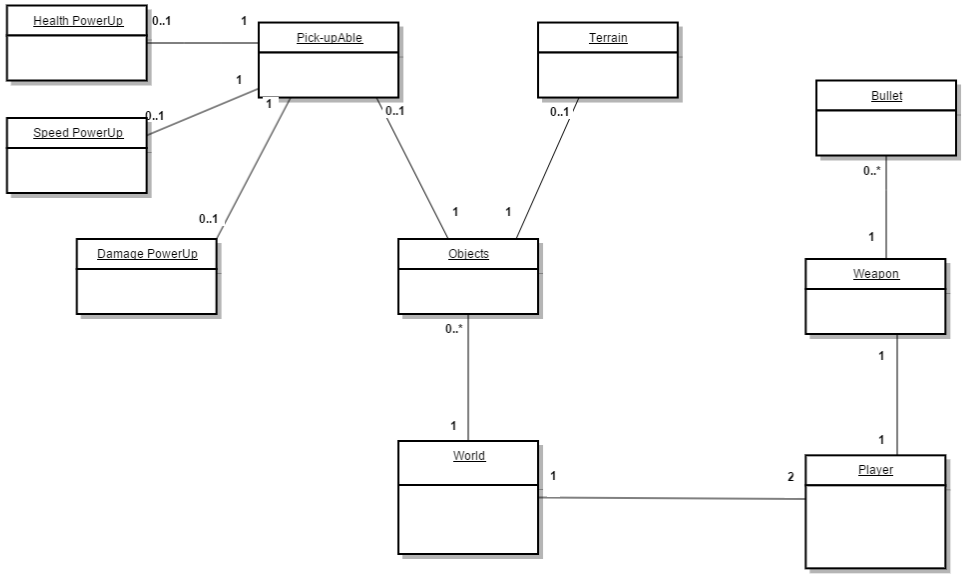
2.4 References

N/A

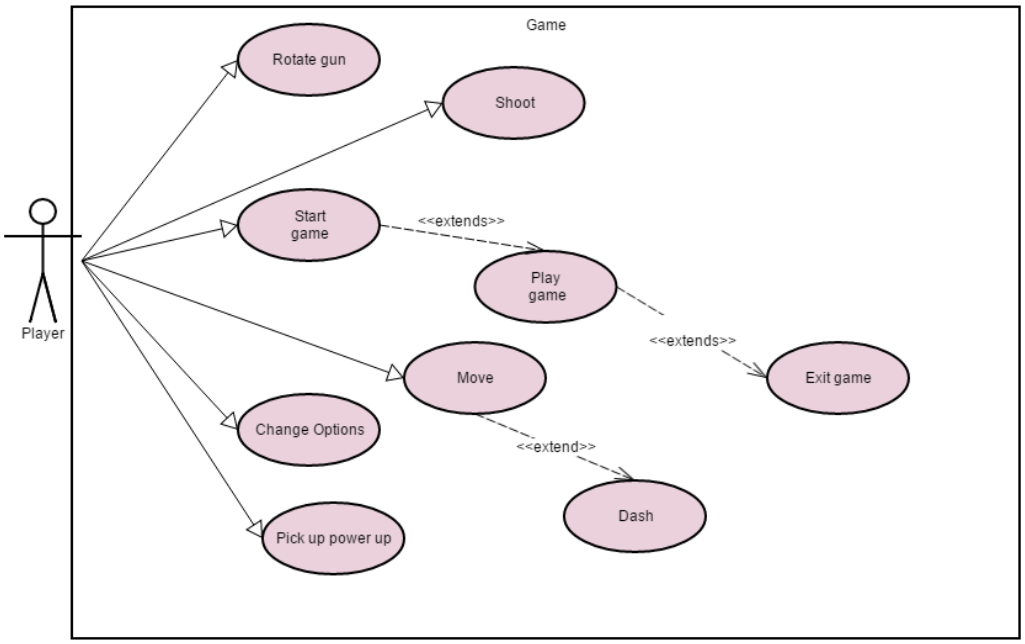
APPENDIX

GUI

DOMAIN MODEL



USE CASE MODEL



USE CASES

Use case	"Move"
Summary	How the players move their characters.
Priority	High
Participators	The player
Normal flow of events	Movement of the player with no errors

Actor	***System***
Holds down WASD	
	If not physically constrained, character starts moving
	Keeps moving while pressed down
	Stops on release

Alternate Flows

Flow	Movement is blocked
Actor	***System***
Holds down WASD	
	Turns in the desired direction but does not move.
	Keeps facing the direction until player changes it
	Stops on release

Use case	Pick up power up
Summary	Player model walks over an animation for a power up
Priority	Medium - High
Participators	Player, power up box.
Normal flow of events	Player gets a boost depending on the powerup type

Actor	
Actor moves over a power up animation.	
	Gains the power of the power up.
	The powerup box is removed

Alternate flows

Flow	The player has already reached the threshold for this powerup
Actor	***System***
Actor moves over a power up animation while having already reached the threshold	
	Does not gain the power of the "power up"
	The power up box is removed.

Use case	"Starts game"
Summary	The player starting up a new round/session of the game
Priority	High
Participators	One of the players.
Normal flow of events	All settings are valid and the game is able to start

Actor	
Player presses start game button	
	The game starts with the chosen properties active

Alternate flows

Flow	Player tries to start game with illegal settings active

Actor	***System***
Player presses start game button	
	The game can't be started due to illegal settings
	The game prompts the player to change these settings.

Use case	"Rotate Gun"
Summary	Rotating the gun independently of direction of player movement
Priority	medium
Participators	Player
Normal flow of events	The gun is rotated

Actor	
Presses rotation key	
	Gun is rotated

Use case	"Shoot"
Summary	Shooting the weapon
Priority	High
Participators	The player
Normal flow of events	A shot from the chars weapon

Actor	***System***
Clicks shoot-button	
	fires a shot

Alternate flows

Flow	Shooting automatic
Actor	***System***
Holds down shoot-button	
	Fires automatically
	Keeps shooting while button is pressed
	Stops on release

Use case	"Taking damage"
Summary	The player gets hit by a bullet
Priority	High
Participators	Player and source of damage
Normal flow of events	Takes damage, loses HP

Actor	***System***
Takes Damage	
	Loses appropriate amount of healthpoints

Use case	"Player wins"
Summary	One player dies and a new round starts.
Priority	medium
Participators	All players
Normal flow of events	2P game, Someone wins and the game restarts, resetting both players and saves the win for the winning player.

Actor	
Reaches a target goal	
	Resets both players' stats. Can play again.

Use case	"Dash"
Summary	Dash quickly in the direction of the player
Priority	medium
Participators	Player
Normal flow of events	The player dashes

Actor	
Presses dash key	
	Player dashes forward