Requirements and Analysis Document for NNN

Table of Contents

Version:

Date

Author

This version overrides all previous versions.

1 Introduction

This section gives a brief overview of the project.

1.1 Purpose of application

The projects aims to create a networked multiplayer game in which players control a spaceship with firing capability. The players battle in a restricted zone within space and are supposed to destroy player opponents.

1.2 General characteristics of application

The application will be a desktop, standalone, networked multi-player application
with a graphical user interface for the Windows/Mac/Linux platforms/ using only the keyboard for ensuring optimal laptop experience.

The application will be real-time. An user hosts a game, to which another player can connect to with IP. Directly when two players are inside the game a round starts. In this round the goal is to eliminate other players by maneuvering the ship with thrusters which alters the spaceships velocity and shooting projectiles at other players. There are abandonned space-stations in the space “arena” in which upgrades and/or powerups will be found. Colliding with a space-station will severly damage the hull of the ship. If another players joins the game he will have to wait for next round to start in order to spawn. Last man standing wins.

1.3 Scope of application

There is no meaning with play this game alon, and therefore this will be impossible. The game won’t

support to save a game. There won't be any server application, only direct connection to host. Graphics are entierly 2D and very basic 2D, no special effects. No stats are saved permanently (no database).

1.4 Objectives and success criteria of the project

1.5 Definitions, acronyms and abbreviations

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

Create a list of high level functions here (from the use cases).

2.2 Non-functional requirements

Possible NA (not applicable).

2.2.1 Usability

2.2.2 Reliability

2.2.3 Performance

2.2.4 Supportability

2.2.5 Implementation

2.2.6 Packaging and installation

2.2.7 Legal

2.3 Application models

2.3.1 Use case model

UML and a list of UC names (text for all in appendix)

2.3.2 Use cases priority

A list

2.3.3 Domain model

UML, possible some text.

2.3.4 User interface

Text to motivate a picture.

2.4 References

APPENDIX

GUI

Domain model

Use case texts