PIP: THE GAME

Name: Kim Karjalainen & Juho Vaihoja

Group: TTV15SP

Date: 15.4.2016, edited 14.9.2016

[1 INTRODUCTION 3](#_Toc445981955)

[1.1 Team members and responsibilities 3](#_Toc445981956)

[1.2 Project schedule 3](#_Toc445981957)

[2 Game mechanics 3](#_Toc445981958)

[3 technical execution of the project 3](#_Toc445981959)

[3.1 Class diagram (uml) 3](#_Toc445981960)

[3.2 Software and external libraries used in the development of the project 3](#_Toc445981961)

[3.3 Terms of programming used in the project 3](#_Toc445981962)

[3.4 Implementation of version control 3](#_Toc445981963)

[4 ideas for further development 3](#_Toc445981964)

# Introduction

PIP: The Game is (supposed to become) a shoot-em-up space rocket game. The player controls a space rocket and tries to survive a continuous onslaught of enemy ships and big asteroids.

## Team members and responsibilities.

Kim Karjalainen: Coding & sound system

Juho Vaihoja: Coding & graphics

## Project schedule.

24.3. starting date

22.5. finishing date

# GAME MECHANICS

The player moves inside the game window in eight different directions. The player can use Focus Time to move slower if there’s too many bullets on the screen. The main point of the game is to kill enemies by shooting projectiles and avoiding enemy projectiles and other obstacles. There may be some bullet hell boss fights as well.

# TECHNICAL EXECUTION OF THE PROJECT

## Class diagram (uml)

You can find the class diagram in ClassDiagram.cd

## Software and external libraries used in the development of the project

We are using SFML

## Terms of programming used in the project

Terms of programming for Pip: The Game:

If anyone is foolish enough to reuse code or assets from Pip: The Game, they are free to do so under any and all copyright laws. This is an open source community project that anyone can contribute to. The assets used in Pip: The Game are available to use by anyone, at any time, for any reason. We take no responsibility for the idiots who actually end up using our code.

## Implementation of version control

We’re using GitHub through Git BASH for version control.

# Ideas for further development

* possible bullet hell bosses
* multiple levels
* score system
* cool graphics

# Project Assesment

Co-operation worked really well between both of us. The end result, while mechanically questionable, works well and looks (and sounds) good. We are both satisfied with the end result.