

# Gravity

*Group 10*

Daniel Walz

Jesper Ekhall

Lukas Kalinski

Fredrik Nordh

Alexander Nyberg

# Project Overview Document

## 1. Target demographic

Our target demographic would typically be male in about our age bracket, which is 18-40. This group might include a middle aged office worker that has some spare time now and then. We don't want to focus on the lower ages because they often demand better graphics and special effects. This game will be simple and easy to start up and that's why we're trying to focus on casual gamers that have some time to kill or just want to do something fun and relaxing for a shorter period of time. Another potential user would be a nostalgic player starting up the game at home either to just kill some time or to play a fun game with a friend. The main problem our product will solve, or at least alleviate, is boredom. Whether it's a dreary day at the office or to brighten a social occasion.

## 2. The game concept

In general, Gravity is a game consisting of planets with their corresponding gravities and space ships with the ability to fire at least one weapon. Both space ships and some weapon projectiles will be affected by gravity. The key idea is to combine classic game shooting with gravity constraints, forcing the player to think about more factors than just where to shoot. Keeping the game simple, the view is in 2D and seen from above.

The main usage may be divided into three stages, namely: *game setup*, *game play* and *game exit*.

Game setup will be done using a simple menu, allowing the player(s) to choose which map she would like to play, what game mode (single player or multiplayer) she is interested in and the nickname she would like to use. Choosing multiplayer mode would also involve choosing what nickname to use for the second player. Finally, a "getting started" option will also be available, explaining what the game is about and how to control it.

Game play is simply about the game being played, according to the specification mentioned above. The game play may be paused/unpaused by using the pause toggle key. Leaving this stage is done by pressing the exit key and confirming the exit request.

Game exit stage will present statistics about the most recently played session, letting the player choose whether to go to the game menu or to exit the whole game application.

**Use case #1:** The two brothers, Peter and Kevin (10 and 11 years old respectively), have a dispute about who is going to use the only computer available. While trying to resolve the situation their father, who recently have heard about a game called Gravity, decides to install it on the computer, remembering that it allows two users to play simultaneously on the same computer. When installed, he quickly sets up a game session by choosing a map and selecting split-screen mode. After a short negotiation Peter and Kevin agree to give it a try. With Peter using alphanumeric steering (i.e. the letters on the keyboard) and Kevin the arrows the game begins. The classic “shoot your opponent” approach, combined with gravities affecting both ships and shots fired, contributes to their mutual satisfaction. After a while a new dispute occurs, this time because of their father exiting the game, telling them to go and eat their dinner.

**Use case #2:** The office employee Bob gets bored by waiting for his boss to give him the green light for a solution he proposed. Therefore he feels free to start playing this game called Gravity, which he recently installed. Bob likes it for its simplicity and for not giving anyone the impression that those who play it are addicted gamers. After starting the game, Bob selects single-player mode, chooses a map, takes a quick look at which the game controls are and what they do, and finally makes the game start by selecting the corresponding option in the menu. During the game session Bob's boss comes by, asking him some details about his proposal. Bob hits the pause button and explains what's needed to explain, and when the boss leaves, Bob hits the same button again and continues his game session. A couple of minutes later Bob fails to avoid a planet, as a result of an asteroid coming towards his ship, and crashes. Bob feels that, it was enough playing for this time, and exits the game.

### **3. System context and environment**

Since the game will be for PC computers it will be available in any environment where such systems are in place, for example as mentioned in the use cases at work or at the home computer. With the event of multimedia computers and laptops together with the broad availability of cheap projectors and capable TV-sets one could also imagine the game being used in a living room setting. One might expect that we would make the game available to mobile platforms, due to the relatively low complexity of the project, but we decided against this since the controls on such systems are not yet sophisticated enough. Our system will be used on breaks or whenever a user has some spare time or in a more social context, for example in a one to one battle with a friend.

#### 4. System scope

The game is in a 2D environment depicting space, with a few surrounding planets that attract every other object according to Newton's physical laws. Each player controls a space ship that can steer left, right and accelerate forward. The ship is affected by the gravity of the surrounding planets and its steering is controlled by the keyboard. There may be more than one player, in which case the players share the keyboard for controlling their respective ships. The ships fire weapons that destroy asteroids or the opponent ship. If any ship collides with an asteroid or a planet the ship is destroyed. If an asteroid collides with a planet the asteroid is destroyed, planets cannot be destroyed.

<b>Topic</b>	<b>In</b>	<b>Out</b>
Mobile platform		X
Multiplayer	X	
Single player	X	
Mouse control		X
Opponent ships	X	
Planets	X	
Asteroids	X	
Simple weapon	X	
Gravity towards planets	X	
2D view	X	
3D view		X
High score system	X	
Moving planets		X
Object collision	X	

## **5. Design and building factors**

There will be an option for a desktop icon so you can start up the game fast and easy without going to the command prompt.

The game will be user friendly by having a menu that pops up in the start up of the game. It will show all the control keys and a short description of the game.

If for example your boss comes in or you get interrupted in one way or the other while you're playing the game there's a feature available so that you can pause the game and continue from the same place.

Because of its low requirements it will not affect the processing speed of your computer and can be in the background when you for example work.

## **6. Technologies and risks**

OpenGL – We choose it because we have some background knowledge of it and that it's well tested and available on many platforms. Risks include debugging in graphical systems is difficult, errors are not easy to find. Also not everyone in the development team is experienced in OpenGL programming.

C++ – We choose it because of object orientation, speed, library availability and because it's easy to use OpenGL from it. C++ is also good to use in environments where garbage collectors might be unwanted because it creates delays. Risks are that it's a large and complex language. It's also difficult to debug partly due to pointers and no memory checking. There is also the need to manage memory manually because of no garbage collector.

Windows XP – We choose windows because it's available virtually everywhere and the team has experience with it. Many useful libraries are available such as OpenGL. Also there is support for I/O with gaming in mind with the DirectX interfaces. Another thing is that it's easy to use, especially compared to UNIX systems such as Linux. Risks are that it's a complex system and it requires skill to be stable. The complexity risk is mostly about our users and it's more of a challenge because we're targeting casual players rather than experienced gamers.

DirectInput – It's a system used to take input from game input devices including keyboards and joystick. We intend to use it for keyboard input only. Risk is that we have no background knowledge at all about this. As an alternative we are considering the built in support for input handling in SDL (a platform independent multimedia library), the plus being that there is more experience within the group.

DirectSound – Used for outputting sound effects and music. We have no background knowledge about this system, but knowledge about similar systems. We are also considering using FMOD, although easy to use and portable we are a bit unsure of the licensing agreements.