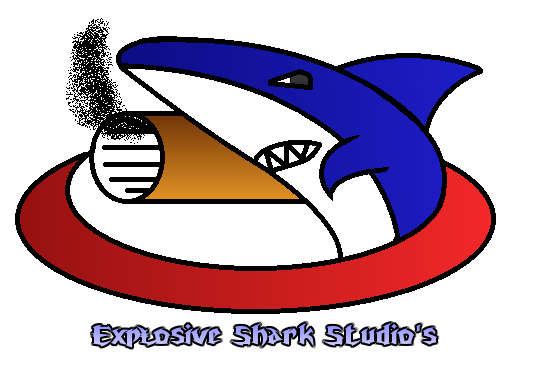
**Technical Document – SpaceRace (working title)**



Sebastiaan van Dijk

Nick Kortleve

Jan Willem Jozic

Lionel Mithoe

Bart Dikmans

Contents

[Code Conventions 3](#_Toc443389970)

[Git Standards: 3](#_Toc443389971)

[Coding Standards: 4](#_Toc443389972)

[Game Architecture 5](#_Toc443389973)

[World Object Hierarchy 5](#_Toc443389974)

[File Structure 5](#_Toc443389975)

[Defined standard class design 5](#_Toc443389976)

# Code Conventions

## Git Standards:

When pushing something on GIT the commit will look like follows:

\*Title\*

insert all the stuff you have done in this commit.

This way we can all read the summary of the commit in the title and can click on the commit to read all of the edits that have been made.

The git will have multiple branches. Every branch will be for a different function in the game.

There will be no pushing to the master branch unless multiple team members have provided proof of the function working.

## Coding Standards:

Every page of code will have a small summary at the top which accurately describes the code.

All functions have to start with a capital letter. For example: Update() or Draw().

All private variables will start with an underscore. For example: unsigned \_example or int \_count

All public variables will use standard camelcase rules.

Every unclear function should have some comments to further describe the function. Rule of thumb to determine an unclear function is:

\*Is the function large (more than 15 lines of code)

\*Are there many variables that look alike/function almost similar

Try to place big comments at the top of the function.

In the case of Egyptian brackets all statements will look like the following example:

if(test)

{

insert stuff here;

}

Try to avoid using cout. Use puts or printf instead (puts automatically puts an /n after the message and printf doesnt)

# Game Architecture

# World Object Hierarchy

# File Structure

# Defined standard class design