

PROJECT ICEWEASEL

Design Document

Murray, Alex (TheComet)
alex.murray@gmx.ch

Niedermann, Ferdinand (nerdinand)
ferdinand.niedermann@gmail.com

Liam Carter (Leyanor)
zymbiont@hotmail.ch

Julie Pelckmans (PelckiJ)
pelckij@hotmail.com

Markus Lochbrunner (Cyberdrace)
cyberdrace@gmail.com

Simeon (PulchraSD)

Rebecca (Drakoni)

June 28, 2016

Contents

1	Introduction	3
1.1	Scope	3
1.2	Target Audience	3
2	Target System	4
3	Development Software	5
3.1	Game Engine and Programming Language	5
3.2	3D Modelling Programs	5
3.3	3D Animation Programs	6
3.4	Image Editing Programs	6
3.5	Sound and Music Editing Programs	6
4	Specification	7
4.1	Concept	7
4.2	Story	7
4.3	Game Structure	7
4.4	Characters	7
4.5	Game Mechanics	7
4.6	NPCs	7
4.7	Items and Pickups	7
4.8	Graphics	7
4.8.1	Characters	7
4.8.2	Props	7
4.9	Environment	7
4.10	Items/Pickups	7
4.11	Shaders	8
4.12	Data storage	8
4.13	GUI	8
5	Front-End	9
6	Development Tools	10
6.1	GitHub and Git	10
6.2	Google Drive	10
7	Team	11
8	Time Plan	12

1 Introduction

This game will be a fun and addictive on-line multi-player first person shooter. There will be a selection of characters to choose from, each with their own special abilities.

None of this concept is permanent.

1.1 Scope

This document is intended to be read by programmers, artists and producers involved in the design, implementation and testing of Project IceWeasel.

1.2 Target Audience

Our target audience are players that have enjoyed games like Overwatch or Team Fortress 2 and like playing first person shooters on-line.

2 Target System

This game is intended to run on all three major desktop platforms.

- **Windows** Versions 7 and upwards.
- **Mac OSX**
- **Linux**

3 Development Software

The tools to create this game are outlined.

3.1 Game Engine and Programming Language

The game will be developed using either Unreal Engine 4 or the Urho3D game engine. It is not yet clear which engine is most suitable.

Choose an engine

The programming language will primarily be C++.

Below we compare the two engines' capabilities.

Urho3D

- **Supported Platforms** Urho3D supports all 3 platforms mentioned, including mobile platforms.
- **Level Editor** Urho3D ships with a level editor designed around the engine. It can be used by artists or programmers to create the worlds the game plays in.
- **On-line Multi-player** Urho3D supports automatic network synchronization of scene nodes. However, it does not support client-side prediction, which is important for a first person shooter. This would have to be programmed manually.
- **3D Model Work-flow** An Urho3D exporter plugin exists for **blender**. The models that are exported from blender can directly be imported into Urho3D's level editor, along with materials. If other 3D modelling software besides blender is used, then Urho3D is capable of importing any format supported by the importer library **AssImp**.

Unreal Engine 4

- **Supported Platforms** UE4 seems to support every platform under the sun, including Windows, Linux, MacOSX, Android & iOS as well as consoles.
- **Level Editor** UE4 ships with a powerful level editor designed around the engine. It can be used by artists or programmers to create the worlds the game plays in.
- **On-line Multi-player** Apparently the engine supports multiplayer out of the box, including client-side prediction
- **3D Model Work-flow** UE4 uses .fbx for meshes, which is supported in blender by default. Materials are handled almost entirely within the engine; Blender is used only for UV-Mapping.

3.2 3D Modelling Programs

Any modelling program capable of exporting to formats supported by **AssImp** can be used.

Blender is the preferred modelling program, as it is capable of directly exporting to Urho3D's format (if Urho3D is to be chosen as the engine).

3.3 3D Animation Programs

Any modelling program capable of skinned animation can be used.

Blender is the preferred modelling program, as it is capable of directly exporting to Urho3D's format (if Urho3D is to be chosen as the engine).

3.4 Image Editing Programs

Any program capable of exporting to common texture formats, such as .PNG or .DDS.

3.5 Sound and Music Editing Programs

Any program capable of exporting to common sound/music formats, such as .WAV, .OGG or .MP3.

4 Specification

4.1 Concept

Lium: Describe the basic concept of this game. What is it? What's it about? What style is the game? What genre is the game? What mechanics (in general) are in this game?

4.2 Story

Write about the story (if there is one?) What's the backstory of this game?

4.3 Game Structure

What modes of gameplay are there? What does each mode do?

4.4 Characters

How many characters are in this game? What are their names? What do they look like? What are their abilities?

4.5 Game Mechanics

How does the game work? What abilities *in general* are there? What is the point of the game?

4.6 NPCs

Are there NPCs? If so, what are their names? What does each one do? How do they behave? Do they have a backstory?

4.7 Items and Pickups

What items exist in the game? How do they affect the player and/or game?

4.8 Graphics

What style of graphics are we going for? High detail? Low detail? Cartoon? Realism? What technologies will be incorporated to achieve these graphics? What shaders will we use, what special rendering techniques (if any) will be used?

4.8.1 Characters

What is the poly budget for 3D artists? What are the texture resolution limits? What technology/rendering technique are we using for textures? What textures must exist on characters (diffuse, normal, specular, emissive, etc.)? What is the skeleton bone limit? What are the skinning limits? What world size (in units) should characters have? What precautions should be taken when modelling a character?

4.8.2 Props

What is the poly budget for 3D artists? What are the texture resolution limits? What technology/rendering technique are we using for textures? What textures must exist (diffuse, normal, specular, emissive, etc.)? What size (in units) should maps have? What other precautions should be taken when modelling worlds? What is the skeleton bone limit?

4.9 Environment

Water? Lava? How are these things implemented?

4.10 Items/Pickups

What is the poly budget? Texture resolution limits? etc.

4.11 Shaders

What shaders will be used in detail?

4.12 Data storage

How are assets organized? What do artists have to look out for?

4.13 GUI

How is the graphical user interface structured? What technologies should we use? (Animated buttons or static? etc.)

5 Front-End

Anything related to high level things that have to do with the player. Cutscenes, etc.

6 Development Tools

These are the tools to be used to coordinate and develop this game.

6.1 GitHub and Git

The project's code, documentation and read-only assets are hosted on github.com under <https://github.com/TheComet93/iceweasel>. Team members directly related with the game engine shall download and install **git** from <https://git-scm.org>.

6.2 Google Drive

Google drive is used to synchronize assets (sound, music, 3D models, textures, concept art). The folder can be accessed by following the link on our Discord server in the channel **#announcements**.

7 Team

Title	Name	Handle	Contact
Project Manager, Programmer	Alex Murray	TheComet	alex.murray@gmx.ch
3D Modelling, 2D Artist	Liam Carter	Leyanor	zymbiont@hotmail.ch
Programmer	Ferdinand Niedermann	nerdinand	ferdinand.niedermann@gmail.com
Graphics Design, Storyboard	Julie Pelckmans	PelckiJ	pelckij@hotmail.com
3D Modelling, Level Design	Markus Lochbrunner	Cyberdrace	cyberdrace@gmail.com
Character Design	Simeon	PulchraSD	
Character Design	Rebecca	Drakoni	

8 Time Plan

Create time plan