Project Antares

Game Development Document

*Developed & Written by Team JAK*

Contents

[1. Introduction 3](#_Toc117599550)

[2. Scope 3](#_Toc117599551)

[3. Target Platform 3](#_Toc117599552)

[4. Supported Hardware 3](#_Toc117599553)

[5. Development Software 3](#_Toc117599554)

[5.1 Engine 3](#_Toc117599555)

[5.2 Programming 3](#_Toc117599556)

[5.3 Art 3](#_Toc117599557)

[6. Specification 4](#_Toc117599558)

[6.1 Concept 4](#_Toc117599559)

[6.2 Story 4](#_Toc117599560)

[6.3 Setting 4](#_Toc117599561)

[6.4 Level Design 4](#_Toc117599562)

[6.5 Actions 5](#_Toc117599563)

[6.6 Objectives 5](#_Toc117599564)

[7. Graphics 5](#_Toc117599565)

[7.1 Styles 5](#_Toc117599566)

[7.2 Fonts 5](#_Toc117599567)

[7.3 Colours 5](#_Toc117599568)

[7.4 Influences 5](#_Toc117599569)

[8. Data Storage 6](#_Toc117599570)

[8.1 Local Data 6](#_Toc117599571)

[8.2 Online Data 6](#_Toc117599572)

[8.3 Social Data 6](#_Toc117599573)

[8.4 Statistical Data 6](#_Toc117599574)

[8.5 Legal and Ethical Considerations 6](#_Toc117599575)

[9. Gameplay 6](#_Toc117599576)

[9.1 Direct Control 6](#_Toc117599577)

[9.1.1 Interaction System 6](#_Toc117599578)

[9.1.2 Movement 6](#_Toc117599579)

[9.1.3 Combat System 6](#_Toc117599580)

[9.2 Indirect Control 6](#_Toc117599581)

[9.3 Object Types 6](#_Toc117599582)

[9.4 World 6](#_Toc117599583)

[10. User Interface 6](#_Toc117599584)

# 1. Introduction

This document will outline, in detail, specifics for the Design, Implementation and User Testing for Project Antares. Antares will be a 2D platformer set in a Sci-Fi environment with a focus on movement-based melee combat.

# 2. Scope

Project Antares will have several unique levels with specific niches. One level will never play like the other. The player will face parkour and combat challenges, possibly both at once. The player will be able to combo movement and combat moves to perform powerful manoeuvres or attacks against enemies.

# 3. Target Platform

Antares will be initially targeted for a Windows PC release with the potential for porting to Web-based platforms and MacOS. Windows offers support for both Keyboard & Mouse and Controllers of varying types, both of which are viable input devices for the project.

# 4. Supported Hardware

Project Antares will be developed with multiple input methods in mind, using Unity’s highly modular Input Manager system to enable different types of hardware to interact with the game. Early discussion also raised to idea of support old-school arcade input technologies, such as the GDT Arcade Machine in SMB201.

# 5. Development Software

## 5.1 Engine

The Project will be developed in Unity, specifically the Long-Term Support build. Unity LTS uses version 2021.3.11f1 at time of writing

## 5.2 Programming

The code for the project will be developed in Visual Studio Code or Fleet, JetBrains new text editor/IDE. Fleet is a relatively new releases, so VSC is still viable for use as the developer sees fit.

## 5.3 Art

Art for Antares will be made with Paint.net to allow for versatile development of various file types and use of image encoding methods.

# 6. Specification

## 6.1 Concept

The concept of Antares is simply put as a 2D platformer with a focus on melee combat, offering the player the ability to apply combo attacks to the enemy in a multitude of ways, as well as presenting the player with a myriad of encounters, from pure combat levels to puzzles and parkour being mixed in.

The player will have different movement abilities to compliment the melee system, allowing them to close the gap between targets rapidly as well as traverse the environment with ease. These could be powered slides, double jumps, wall jumps, etc.

The melee combo system will play into this system, rewarding the player for utilizing the advanced movement techniques by dealing additional damage to enemies when a melee attack comes soon after a movement ability is used and using a different animation to represent this. For example, a sprint-slide combined with a melee attack would trigger an upper-cut attack animation and deal more damage than a standard sword swing

## 6.2 Story

The player will play as Captain Tyler Elliott, a super soldier from Earth’s advanced special forces unit, Centurions. Tyler is stationed aboard Neptune Station, a Deep Space Black Site in the Epsilon Eridanus System, to oversee security of the station’s many projects. Neptune Station plays host to an assortment of top-secret research projects, ranging from small arms testing to AI development to gene therapy.

The first level will have the player witness a breakout of a large amount of “Xeno-X” Creatures before having to defend themselves against a few of them. These creatures will make up the majority of the enemies faced throughout the game. They were released by accident when an intern from XenoGene Inc aboard the station left their cell unit on low security settings during out of office hours. They then proceeded to form a nest, rapidly reproduce, and release a bunch of other XenoGene abominations.

Tyler’s priority is to find and eliminate the Xeno-X Queen and restore order to the station before an Earth response fleet arrives to destroy the station along with everyone and everything on board in order to prevent the Xeno-X escaping.

## 6.3 Setting

The over-arching setting and style of Antares will be heavily inspired by the “moderate” sci-fi genre (moderate implying some representation of real-world physics in both design and function, while also allowing purely made-up physics to be incorporated into the setting).

## 6.4 Level Design

The level design of Antares will feature terrain layouts promoting the player to be creative with movement abilities and offering multiple avenues to progress through a level. Use of height disparities, cover and slopes will be present in every level.

The levels themselves will be made using Unity’s Tilemap feature. This allows the creation of “tiles” that can be slotted together on a 2D grid to dynamically create 2D levels.

## 6.5 Actions

The player will be able to do a variety of actions during their gameplay. The most basic of these will be the movement system. The player will need to sprint, jump, crouch and slide their way through different levels. This action will be managed via Unity’s Input Manager, which has seamless integration with executing movement mechanics out of the box.

The player will also have access to 2 other actions they have direct control over: interacting with objects and engaging in combat. The way these two features will work under the hood will be effectively identical, using the same Player Interaction script to trigger differing effects.

Interacting with objects can have varying effects. From simply moving an object out of the way or opening a box to reveal a powerup (or enemy) to activating consoles to change the environment or picking up new weapons.

Engaging in combat is to be simple in function. If the player is in range of an enemy when an attack is executed, that enemy is hit by the player’s attack and will have its Hit Points reduced. Enemies can also be stunned by more advanced attacks if they are weaker enemy forms.

Enemies can also interact with the player via use of another interaction where the enemy attempts an attack, and if the player is in a certain radius when the attack is executed, the player’s Hit Points are reduced. In addition, stronger enemies such as bosses may get the ability to stun the player in some way.

## 6.6 Objectives

Boss Fights  
Parkour Areas  
Clearing out areas of minor enemies  
Puzzles

# 7. Graphics

## 7.1 Styles

## 7.2 Fonts

## 7.3 Colours

Light orange for player weapons  
Metallic tones  
Deep blue for enemy weapons

## 7.4 Influences

Strike Force Heroes  
Metal Gear Rising  
Mario  
Star Wars  
Halo

# 8. Data Storage

## 8.1 Local Data

## 8.2 Online Data

## 8.3 Social Data

## 8.4 Statistical Data

## 8.5 Legal and Ethical Considerations

Colour blind settings

Accessible controls

# 9. Gameplay

## 9.1 Direct Control

### 9.1.1 Interaction System

Collider box combined with a sphere gizmo with a set range.  
F will be default keybind

### 9.1.2 Movement

WASD – M&K (rebinding option in future sprint)

### 9.1.3 Combat System

TBC

## 9.2 Indirect Control

## 9.3 Object Types

## 9.4 World

# 10. User Interface