Low Level Programming

**Technical Design Document**

A black and red text

Description automatically generated

Created by Claudia Brown, Joe Gamble, Miles Hoyland, Tahir Serali and Razvan Traistrau.

Contents

[*1.1.Platform*: 2](#_Toc39076672)

[*2.1.Programming language:* 2](#_Toc39076673)

[*3.1.Libraries/Tools:* 2](#_Toc39076674)

[*4.1.Features:* 2](#_Toc39076675)

[*5.1.Software*: 2](#_Toc39076676)

[*6.1.Inputs*: 2](#_Toc39076677)

[7.1.Diagrams: 2](#_Toc39076678)

### *1.1.Platform*:

The game will be developed solely for PC’s; however. The game will be published on Itch.io. This is because the platform encourages and endorses indie developers, whilst allowing them to have greater creative freedom and control rather than with a larger developer.

### *2.1.Programming language:*

It will be developed using C++ with in the ASGE 2.0 framework. C++ is the standard for 2D video games. Especially RPG type-dungeon crawler games. This makes it perfect for our project.

C++ is widely used by developers, and therefore has extensive support online and off; Furthermore, the language is versatile, and executions will be quick, which is ideal for a networking game. The libraries within C++ includes tools and functionality that allows us to take full advantage of procedural programming.

### *3.1.Libraries/Tools:*

#### **ASGE 2.0: framework**

The ASGE 2.0 framework will be used to its fullest potential to create and program our network game.

#### **SoLound Engine**

Soloud will be used to add sound into the game; Sound include background music.

**String**

The string library will be used for text-based input for a chatroom between players.

**Font library**

The font library will be used to display text on screen.

### *4.1.Features:*

#### **Interaction**

You will be able to interact with objects and collectables using the ‘E’ button. Along with interacting with the menu and attack system, using your mouse or the appropriate key binds.

#### **Music/Sound**

Music and sound effects have been sourced from free online sources. They will indicate attacks, along with the location of enemies.

#### **Inventory system**

The Inventory system uses a click and drag system to rearrange, pick up and drop certain objects. It can be accessed by pressing the ‘Open Inventory’ Key bind.

### *5.1.Software*:

**CLion IDE**

CLion The IDE from JetBrains will be used to build, debug and run the program from the start to completion of the game.

**GitKraken**

GitKraken, the visual Ul interface for git repositories will be used to control collaboration and management of the game.

**Adobe Photoshop**

Adobe Photoshop CC will be used to rapidly design and iterate the levels if necessary. Along with assisting in design the UI and Wireframes.

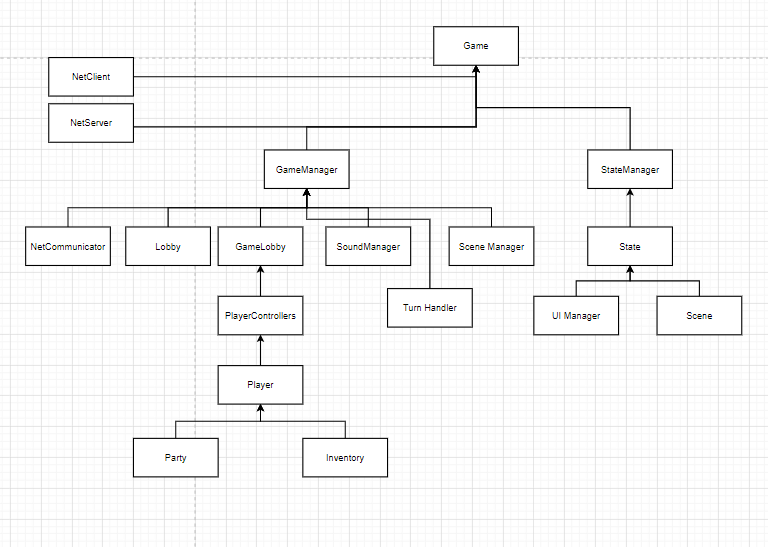
### *6.1.Inputs*:

**Movement Inputs**

|  |  |  |
| --- | --- | --- |
| Command | Input | Alternative Input |
| Move Right | A | → |
| Move Left | D | ← |
| Move Up | W | ↑ |
| Move Down | S | **↓** |

### 7.1.Diagrams:

#### **UMLS**



#### **7.2.Flow chart**

A close up of a logo

Description automatically generated

**7.3.Wireframes**

**Logo**

A black and red text

Description automatically generated

**Menu**

A picture containing drawing

Description automatically generated

**Play Menu**

A picture containing drawing

Description automatically generated

**Options Menu**

A picture containing clock, ball

Description automatically generated

Exit Menu

A close up of a logo

Description automatically generated

**In-game options menu**

A picture containing drawing

Description automatically generated

**Player Inventory**

A star in the background

Description automatically generated

**Player UI**

A close up of a screen

Description automatically generated

**Combat Screen**

A picture containing clock

Description automatically generated