

SE 3XA3: Software Requirements  
Specification  
BlockBuilder

Team 28, OAC  
Owen McNeil, mcneilo  
Christopher DiBussolo, dibussoc  
Andrew Lucentini, lucenta

December 5, 2018

# Contents

<b>1</b>	<b>Project Drivers</b>	<b>1</b>
1.1	The Purpose of the Project . . . . .	1
1.2	The Stakeholders . . . . .	1
1.2.1	The Client . . . . .	1
1.2.2	The Customers . . . . .	1
1.2.3	Other Stakeholders . . . . .	1
1.3	Mandated Constraints . . . . .	2
1.3.1	Solution Constraints . . . . .	2
1.3.2	Implementation Environment of the Current System . . . . .	2
1.3.3	Partner or Collaborative Applications . . . . .	2
1.3.4	Off-the-Shelf Software . . . . .	2
1.3.5	Anticipated Workplace Environment . . . . .	2
1.3.6	Schedule Constraints . . . . .	2
1.3.7	Budget Constraints . . . . .	3
1.4	Naming Conventions and Terminology . . . . .	3
1.5	Relevant Facts and Assumptions . . . . .	3
1.5.1	Relevant Facts . . . . .	3
1.5.2	Relevant Assumptions . . . . .	3
<b>2</b>	<b>Functional Requirements</b>	<b>4</b>
2.1	The Scope of the Work and the Product . . . . .	4
2.1.1	The Context of the Work . . . . .	4
2.1.2	Work Partitioning . . . . .	5
2.1.3	Individual Product Use Cases . . . . .	5
2.2	Functional Requirements . . . . .	5
<b>3</b>	<b>Non-functional Requirements</b>	<b>7</b>
3.1	Look and Feel Requirements . . . . .	7
3.2	Usability and Humanity Requirements . . . . .	7
3.2.1	Ease of Use Requirements . . . . .	8
3.2.2	Personalization and Internationalization Requirements . . . . .	8
3.2.3	Learning Requirements . . . . .	8
3.2.4	Understandability and Politeness Requirements . . . . .	9
3.2.5	Accessibility Requirements . . . . .	9
3.3	Performance Requirements . . . . .	9
3.3.1	Speed and Latency Requirements . . . . .	10

3.3.2	Safety-Critical Requirements . . . . .	10
3.3.3	Precision or Accuracy Requirements . . . . .	10
3.3.4	Reliability and Availability Requirements . . . . .	10
3.3.5	Robustness or Fault-Tolerance Requirements . . . . .	11
3.3.6	Capacity Requirements . . . . .	11
3.3.7	Scalability or Extensibility Requirements . . . . .	11
3.3.8	Longevity Requirements . . . . .	11
3.4	Operational and Environmental Requirements . . . . .	12
3.4.1	Expected Physical Requirements . . . . .	12
3.4.2	Requirements for Interfacing with Adjacent Systems . . . . .	12
3.5	Maintainability and Support Requirements . . . . .	12
3.6	Security Requirements . . . . .	13
3.6.1	Access Requirements . . . . .	13
3.6.2	Integrity Requirements . . . . .	13
3.6.3	Privacy Requirements . . . . .	13
3.6.4	Audit Requirements . . . . .	13
3.6.5	Immunity Requirements . . . . .	13
3.7	Cultural and Political Requirements . . . . .	14
3.8	Legal Requirements . . . . .	14
3.9	Health and Safety Requirements . . . . .	14
<b>4</b>	<b>Project Issues</b>	<b>15</b>
4.1	Open Issues . . . . .	15
4.2	Off-the-Shelf Solutions . . . . .	15
4.3	New Problems . . . . .	15
4.3.1	Effects on Current Environment . . . . .	15
4.3.2	Effects on Installed Systems . . . . .	15
4.3.3	Potential User Problems . . . . .	15
4.3.4	Implementation Environment Limits . . . . .	16
4.3.5	Follow-Up Problems . . . . .	16
4.4	Tasks . . . . .	16
4.4.1	Project Planning/Development Process . . . . .	16
4.5	Migration to the New Product . . . . .	16
4.5.1	Requirements for Migration to the new Product . . . . .	16
4.5.2	Data that has to be modified or Translated to the new System . . . . .	16
4.6	Risks . . . . .	16
4.6.1	Excessive Schedule Pressure . . . . .	16

4.6.2	Management Malpractice . . . . .	17
4.6.3	Low Quality . . . . .	17
4.7	Costs . . . . .	17
4.8	User Documentation and Training . . . . .	17
4.8.1	User Documentation Requirements . . . . .	17
4.8.2	Training . . . . .	17
4.9	Waiting Room . . . . .	18
4.10	Ideas for Solutions . . . . .	18
<b>5</b>	<b>Appendix</b>	<b>19</b>
5.1	Symbolic Parameters . . . . .	19

## List of Tables

1	<b>Revision History</b> . . . . .	i
2	Work Partitioning . . . . .	5

## List of Figures

1	BlockBuilder Context Diagram . . . . .	4
---	--	---

Table 1: **Revision History**

Date	Version	Notes
Oct 5, 2018	1.0	Initial creation of the SRS pushed to repo
Dec 1, 2018	2.0	Begin making changes for rev1
Dec 5, 2018	2.1	Final changes made for rev1

This document describes the requirements for BlockBuilder. The template for the Software Requirements Specification (SRS) is a subset of the Volere template (?).

# 1 Project Drivers

## 1.1 The Purpose of the Project

The purpose of this project is to provide people a fun and free open world video game for people to unleash their creativity.

## 1.2 The Stakeholders

### 1.2.1 The Client

~~The clients of BlockBuilder are potential users of all age above 4 years who will play the game itself. The users must be satisfied with the product upon delivery.~~ The client for the BlockBuilder project is the 3XA3 professor, Dr. Asghar Bokhari.

### 1.2.2 The Customers

In this situation, the customers are the same as the client. The customers of this product are individuals of all age above 4 years who own a personal or desktop computer. Since BlockBuilder will be free, anyone has the ability to download the game and use it in their own time.

### 1.2.3 Other Stakeholders

Additional stakeholders of the product include the following:

- ~~The TA's of SE3XA3~~
- The devlopement team, consisting of Owen McNeil, Chris DiBussolo and Andrew Lucentini
- ~~The professor of SE3XA3~~

## 1.3 Mandated Constraints

This section describes the constraints on the eventual design of the software.

### 1.3.1 Solution Constraints

**Description:** The product shall operate using a Windows OS (ideally Windows 7 or higher) and/or MacOSX.

**Rationale:** The average user utilizes either one of these operating systems.

**Fit criterion:** The product shall be approved on both Mac and Windows operating systems.

### 1.3.2 Implementation Environment of the Current System

N/A

### 1.3.3 Partner or Collaborative Applications

The software shall run with the Pyglet library for graphic generation. The software will not run correctly if the Pyglet package is not installed on the user's computer.

### 1.3.4 Off-the-Shelf Software

**Description:** The product shall utilize the Pyglet library for 3D graphics support.

**Rationale:** Development costs and development time will be greatly reduced. Performance will likely be better than developing an engine in-house.

**Fit criterion:** The product will be built on top of Pyglet and integration tests will be performed.

### 1.3.5 Anticipated Workplace Environment

N/A

### 1.3.6 Schedule Constraints

- The documentation of the software shall be delivered by each deadline provided by the SE3XA3 outline.

- The final implementation of BlockBuilder shall be complete by December 5, 2018.

### 1.3.7 Budget Constraints

N/A

## 1.4 Naming Conventions and Terminology

**FPS** - Frames per second. A measurement for how many unique consecutive images a monitor can handle each second.

**GUI** - Graphical user interface. It includes graphical elements, such as windows, icons and buttons.

**3D** - Rendering on a 3-dimensional plane consisting of (x,y,z) coordinates.

**Minecraft** - 3D first-person sandbox game developed by Mojang.

**Pyglet** - Python multimedia library used for 2D and 3D graphics rendering.

## 1.5 Relevant Facts and Assumptions

### 1.5.1 Relevant Facts

- Pyglet has been proven to be capable of effectively powering a 'Minecraft' style 3D game.
- The original project has approximately 900 lines of code.

### 1.5.2 Relevant Assumptions

The target user of this product is assumed to be:

- Physically and mentally capable of enjoying a video game.
- Interested in open-world video games.
- Accepting of "blocky" and low-resolution style graphics.
- Creative.

It is assumed that the average user will have:

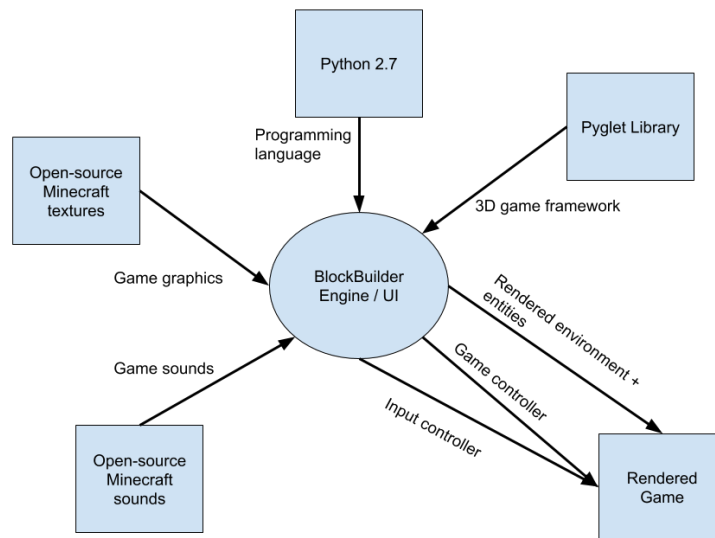
- A modern computer capable of running a simple 3D Python game.
- A keyboard and mouse.
- The game will also allow the use of a track-pad if being played on a laptop.
- A monitor capable of refreshing at REFRESH\_RATE

## 2 Functional Requirements

### 2.1 The Scope of the Work and the Product

#### 2.1.1 The Context of the Work

Figure 1: BlockBuilder Context Diagram





### 2.1.2 Work Partitioning

Table 2: Work Partitioning

Event Name	Input and Output	Summary
1. Player performs action	Player action (in)	Parse the action in game controller and modify game state as necessary.
2. Chunk boundary reached	Player coordinates (in)	Generate required chunks and render them.
3. New entity generated	New Entity (in)	Create new entity object, add it to render and controller cycles.

### 2.1.3 Individual Product Use Cases

The primary use case for this product is to be enjoyed as a game that users play in their free time. Given it's open-world, creative nature, and lack of clear objectives, it isn't possible to determine all possible use cases, or play styles.

## 2.2 Functional Requirements

- ~~REQ1: The game must feature textures similar to those present in 'Minecraft'.~~
- ~~REQ2: The player should be able to break and place blocks as done in 'Minecraft'.~~
- ~~REQ3: The game environment, including the "biomes", entities and overall input/output should mirror that of 'Minecraft'.~~
- REQ4: The game must be implemented using a ~~Python 2.7~~ **Python 3.7**/Pyglet stack.
- ~~REQ5: The game must be three dimensional.~~ **The user must be able to move their character in 3-dimensions.**
- ~~REQ6: The player must be able to control their character.~~

- REQ6: The user must be able to move their character in the forward direction while the 'W' key is pressed.
- REQ7: The user must be able to move their character in the left direction while the 'A' key is pressed.
- REQ8: The user must be able to move their character in the right direction while the 'D' key is pressed.
- REQ9: The user must be able to move their character in the backward direction while the 'S' key is pressed.
- REQ10: The user must be able look around in the game by moving the mouse or trackpad.
- REQ11: The user must be able to simulate a jump in the world when the space bar is pressed.
- REQ12: The user must be able to enter 'Flying mode' when the 'Tab' key is pressed.
- REQ13: The user must be able to place blocks in the world with a mouse right-click provided they are looking at a previously existing block in the world at a minimum distance.
- REQ14: The user must be able to remove blocks in the world with a mouse left-click provided they are looking at a block at a minimum distance.
- REQ15: The user must be able to place Grass blocks after they hit the '1' num key.
- REQ16: The user must be able to place Brick blocks after they hit the '2' num key.
- REQ17: The user must be able to place Sand blocks after they hit the '3' num key.
- REQ18: The user must be able to place Stone blocks after they hit the '4' num key.

- REQ19: The user must be able to remove all blocks (when possible) except Stone blocks.
- REQ20: The game must be robust in it's ability to handle a large variety of user inputs.
- REQ21: The player should not be able to place blocks in a position that would result in a collision between the player and the newly placed block.
- REQ22: The player should not be able to move through blocks and attempting to should result in a "collision" that makes the player unable to move through the block instead makes it look like they are walking into it.

### 3 Non-functional Requirements

#### 3.1 Look and Feel Requirements

The look and feel requirements of the product are listed below.

- **LF1:** The product shall be attractive to users of all age with a direct focus on individuals MAX\_FOCUS\_AGE and younger.  
Fit Criterion: After 2.5 months after release, a survey shall be administered to users MAX\_FOCUS\_AGE and younger to reveal that 90% of the users liked the product.
- **LF2:** The GUI of the product should resemble that of Minecraft.  
Fit Criterion: After 2 months after release, a survey shall be administered to users to reveal that 98% of the users thought that the product resembled Minecraft.

#### 3.2 Usability and Humanity Requirements

This section is concerned with requirements that make the product usable and acceptable to the users.

### 3.2.1 Ease of Use Requirements

- **UH1:** Users of all ages (MIN\_AGE+) should be able to play the game.  
Fit Criterion: 95% of the users from a test group aged MIN\_AGE+ shall be able to play the game on their own after 10 minutes of playing the game.
- **UH2:** The users shall remember how to use the product after 1 session of playing the game.  
Fit Criterion: After playing the game once, 90% of the users from a test group shall correctly answer a set of questions asking which inputs correspond to their respective in-game actions.
- **UH3:** The product shall make the users want to use it.  
Fit Criterion: After 2.5 months after release, a questionnaire shall be administered to all users to reveal that 90% of the users wanted to use the product after their first use.
- **UH4:** The product shall be playable by users of all languages.  
Fit Criterion: 100% of users from a test group consisting of 15 users, each speaking different languages, shall be able to play the game on their own after 10 minutes.
- **UH5:** The user shall interact with the game through the typical "WASD" keyboard control scheme set up.  
Fit Criterion: A tester shall play the game to ensure that they move their player using the "WASD" keys.

### 3.2.2 Personalization and Internationalization Requirements

N/A

### 3.2.3 Learning Requirements

- **UH6:** The users shall be able to learn the product within LEARNING\_TIME minutes of playing.  
Fit Criterion: A team of 10 users shall perform a test after LEARNING\_TIME minutes to reveal that 95% of the users can teach the game to another user.

- **UH7:** The software shall be learnable without a tutorial.  
Fit Criterion: A team of 20 users shall perform a test after LEARNING\_TIME minutes to reveal that 98% of the users can play the game without instructions.
- **UH8:** The software shall be learnable by any user above the age of MIN\_AGE regardless of educational background.  
Fit Criterion: A team of 15 users with various educational backgrounds (including none) shall perform a test after LEARNING\_TIME minutes to reveal that 98% of the users can play the game without instructions.

#### 3.2.4 Understandability and Politeness Requirements

- **UH9:** The software shall not use any complicated text that may confuse the user.  
Fit Criterion: After 2 weeks upon release, a questionnaire shall be administered to all users to reveal that 100% of users reported no complicated text on the screen.
- **UH10:** The construction of the software should be hidden from the users.  
Fit Criterion: After 2 weeks upon release, a questionnaire shall be administered to all users to reveal that 95% of users do not understand how the game works.

#### 3.2.5 Accessibility Requirements

- **UH11:** The software shall be usable by individuals who suffer from hearing impairment.  
Fit Criterion: Test subjects who suffer from hearing impairment will participate in a survey that reveals all of the users can play the game on their own.

### 3.3 Performance Requirements

This section is concerned with requirements regarding the performance of the system.

### 3.3.1 Speed and Latency Requirements

- **PP1:** The software shall respond to user input with less than a INPUT\_DELAY second delay.  
Fit Criterion: Specific test will reveal that the average software response time was less than INPUT\_DELAY seconds for 100 cases.
- **PP2:** The software shall operate at a AVG\_FPS fps average.  
Fit Criterion: The game will be run 20 times to reveal that in each instance, the software operated at AVG\_FPS fps.

### 3.3.2 Safety-Critical Requirements

- **PP3:** The software shall not encourage violence or destructive behaviour.  
Fit Criterion: A questionnaire will be administered to the testing group to reveal that each member agrees that the software does not encourage violent or destructive behaviour.

### 3.3.3 Precision or Accuracy Requirements

- **PP4:** The input controls to the software should always produce the correct output on the GUI.  
Fit Criterion: 200 input test cases will reveal that 100% of the inputs in each test case produced the correct outputs.
- **PP5:** Any displayed values on the GUI should be accurate to two decimal places.  
Fit Criterion: 10 running instances of BlockBuilder will reveal that values displayed on the GUI are accurate to two decimal places.

### 3.3.4 Reliability and Availability Requirements

- **PP6:** The product shall be available for use of up to AVAIL\_TIME a day.  
Fit Criterion: 10 test users will reveal that they were all successfully able to run the software for AVAIL\_TIME.
- **PP7:** The product shall achieve a UPTIME up-time. Fit Criterion: User surveys after 3 weeks shall verify that 98% of users were able to achieve UPTIME up-time on their respective systems.

### 3.3.5 Robustness or Fault-Tolerance Requirements

- **PP8:** The product shall remain in tact when the user administers a high volume of input commands at any given time.  
Fit Criterion: A test group consisting of 5 members will each administer approximately 5 inputs a second over 20 seconds to reveal that the product is working correctly after 20 seconds. The product works correctly if all inputs produce the correct outputs.

### 3.3.6 Capacity Requirements

- **PP9:** The product shall be able to handle the loading produced by a single individual. Fit Criterion: User surveys after 4 weeks of launch will show that at least 90% of users had no issues running the game in any load environment.

### 3.3.7 Scalability or Extensibility Requirements

- **PP10:** The product shall be easily modifiable to allow future growth of the software.  
Fit Criterion: Guest programmers will verify the modularity of the design by attempting to modify the game and state how easy it is to modify on a scale from 1-10.
- **PP11:** The product shall allow additional features to be easily implemented.  
Fit Criterion: After the product is released, a team of 10 software engineers will study the code and all agree that additional features can be easily implemented.

### 3.3.8 Longevity Requirements

- **PP12:** The product is expected to operate until the next major python update.  
Fit Criterion: When the next major python update arrives, a questionnaire will be administered to all users to reveal that 90% of users were able to use the product during the time period between launch up until the day that the questionnaire was administered.

## 3.4 Operational and Environmental Requirements

This section discusses the requirements regarding the operation and environment in which the system will be used in.

### 3.4.1 Expected Physical Requirements

- **OE1:** The product shall be usable in any physical environment that the system it is running on is subject to.  
**Fit Criterion:** The product should be able to run on 95% of PC or Mac computers supporting the appropriate operating systems mentioned above.

### 3.4.2 Requirements for Interfacing with Adjacent Systems

- **OE2:** The product shall run on the most recent version of the Mac and Windows Operating systems.  
**Fit Criterion:** The game shall be updated within 2 weeks time of a new OS update to address any necessary changes.
- **OE3:** The product shall run on the most recent version of python (python 3.7).  
**Fit Criterion:** N/A

## 3.5 Maintainability and Support Requirements

- **MS1:** The product will receive updates whenever bugs are identified.  
**Fit Criterion:** The developers will release new patches every 3 weeks in response to any bugs brought fourth by the community. Such bugs may also be addressed by other programmers working on the open-source projects before said patch.
- **MS2:** Given that the product is open source, other programmers will be welcomed to catch and fix bugs as they see fit.  
**Fit Criterion:** N/A



## 3.6 Security Requirements

### 3.6.1 Access Requirements

- **SS1:** The product will be a free software download for anyone to use.  
Fit Criterion: N/A

### 3.6.2 Integrity Requirements

- **SS2:** All data shall be stored on the user's local machine, not on an external database.  
Fit Criterion: N/A
- **SS3:** The product will ensure the precision of data being introduced.  
Fit Criterion: The team will ensure that any new data introduced in future updates is 98% safe, fixing any 2% discrepancies in consecutive updates.
- **SS4:** The product will take measure to ensure the data cannot be abused for reasons such as cheating.  
Fit Criterion: The developers will ensure that 99% of data introduced to the game is safe from tampering that would give players an unfair advantage.

### 3.6.3 Privacy Requirements

- **SS5:** The product will not collect or store any data about the user.  
Fit Criterion: At any time, the developers will retain 0 instances of personal information about its users.

### 3.6.4 Audit Requirements

- N/A

### 3.6.5 Immunity Requirements

- **SS6:** The team will take measures to ensure any download links provided by the team do not contain viruses, Trojans horses etc.  
Fit Criterion: The team will ensure that 100% of download links issued by the development team are 100% safe. The development team

is not responsible for any illegal redistribution of the product and any downloads or potential viruses associated with them.

- **SS7:** The team will provide a disclaimer that they are not responsible for download links from outside sources.

Fit Criterion: N/A

### 3.7 Cultural and Political Requirements

- **CP1:** BlockBuilder will not exhibit any racially or culturally inappropriate images or texts.

Fit Criterion: After 2.5 months of availability, the team will use survey results to get feedback on any controversial subjects that the community wants addressed. It is at the discretion of the developers to determine whether or not such changes are necessary.

- **CP2:** The BlockBuilder team will ensure that anything considered remotely resembling an offensive or inappropriate symbol is merely a coincidence and that the development team does not support such ideas.

Fit Criterion: N/A

### 3.8 Legal Requirements

- **LE1:** The product will not violate any copyright laws or internet download laws.

Fit Criterion: Two copyright lawyers will review the product to ensure that the game does not violate any copyright laws or internet download laws.

### 3.9 Health and Safety Requirements

- **HS1:** The product shall not endanger the safety of the users.

Fit Criterion: A test team will perform a risk analysis of the software to determine that the product is safe for all users.

- **HS2:** The product shall be downloaded with a disclaimer stating that the game should not be played for long periods of time.

Fit Criterion: The disclaimer shall state that the recommended use time is no greater than 8 hours per session.

- **HS3:** The disclaimer should state the risks being exposed to video games for long periods of time.  
**Fit Criterion:** The disclaimer issued by the developers should state the healthy amount of digital exposure time for users as determined by reliable sources such as [The Canadian Paediatric Society](#).

## 4 Project Issues

### 4.1 Open Issues

The issue of whether Python is the best fit for what the project is trying to accomplish is still under investigation.

### 4.2 Off-the-Shelf Solutions

There are many other free, fan made versions of Minecraft similar to what this project is attempting to develop and improve upon.

Many of the other personally developed versions of Minecraft can be used as templates to creating key features about the game as well as provide ideas on how to optimize the source material to make it more readable and efficient.

There are many resources on the internet that can assist the team in learning how to create 3D games using multimedia libraries like Pyglet.

### 4.3 New Problems

#### 4.3.1 Effects on Current Environment

N/A

#### 4.3.2 Effects on Installed Systems

N/A

#### 4.3.3 Potential User Problems

Any drastic changes or remodelling to the games features such as the way that the user's character mines may have an ill effect on the user and make them shy away from the game because, "it doesn't feel like Minecraft". The

team should prioritize making the game feel as faithful to it's inspiration as possible.

#### **4.3.4 Implementation Environment Limits**

The teams unfamiliarity with the Pyglet Python library may cause some issues in implementing more detailed parts of the game. The team should prioritize consistency and reliability over fancy features.

#### **4.3.5 Follow-Up Problems**

- Will this project violate and copyright laws should it be published or spread widely?

### **4.4 Tasks**

#### **4.4.1 Project Planning/Development Process**

Please refer to ProjectSchedule/ProjectSchedule.pdf in the repository for a detailed Gantt Chart outlining the tasks for this project.

### **4.5 Migration to the New Product**

#### **4.5.1 Requirements for Migration to the new Product**

N/A

#### **4.5.2 Data that has to be modified or Translated to the new System**

N/A

### **4.6 Risks**

#### **4.6.1 Excessive Schedule Pressure**

Should the project become more ambitious than a simple 3D remodelling of Minecraft, it may become quite difficult for the team to meet their deadlines and produce a working prototype on time.

Should the team fall behind and fail to properly learn the environment they will be working in, in this case learning to use Pyglet to implement 3D objects, it will become very difficult to learn on top of meeting deadlines and they will have an overwhelming schedule.

#### **4.6.2 Management Malpractice**

The team must ensure that they stay organized on their management of time and resources while working on the project as falling behind make them unable to deliver.

#### **4.6.3 Low Quality**

Excessive focus on fancy features beyond the scope of the project will lead to the team being overwhelmed and producing a low quality game. The team should avoid this risk by focusing on consistency and reliability, making sure the simple features work properly, instead of producing unneeded features that create more problems.

### **4.7 Costs**

Since the game is being written in Python on Pyglet, there is no cost for the implementation environment. The only cost this project will produce is the developers' time.

### **4.8 User Documentation and Training**

#### **4.8.1 User Documentation Requirements**

The development team will include a user manual for the game inside the game from the start menu. The manual will detail character controls and descriptions of important game features.

Given that the project is a game, some features may be left to player discovery at the discretion of the team.

#### **4.8.2 Training**

There is no safety or other training required to use the product.

## **4.9 Waiting Room**

- Any detailed features not part of the simple 3D modelling and game play demo will be released at a later date, after the demonstration of the prototype.

## **4.10 Ideas for Solutions**

The game could also be rendered in one of the free game engines such as a Unity or Unreal Engine 3.

## 5 Appendix

N/A

### 5.1 Symbolic Parameters

The following are definitions of symbolic constants that are used throughout this document:

- LEARNING\_TIME: 10
- REFRESH\_RATE: 60Hz
- MAX\_FOCUS\_AGE: 25
- MIN\_AGE: 4
- AVG\_FPS: 50
- AVAIL\_TIME: 20 hours
- UPTIME: 99 percent
- INPUT\_DELAY: 0.2