

K.O 3D Game Software Requirement Specification



University of central Punjab

Class: BSCS 7th A

Group Members: Syeda Shajiah Zahra Zaidi(14)
Esha Javaid (34)
Amna Altaf (10)

Supervised By: Prof: Ali Shan

Table of content

1	Introduction	2
1.1	Scope	2
1.2	Glossary	2
1.3	Technologies to be used	2
2	The Process	3
2.1	The process model	3
2.2	General Constraints	3
2.3	Assumptions & Dependencies	3
3	Research	4
3.1	Pygame:	4
3.2	Visual studio code	4
3.3	Python:	4
4	Requirement specification	5
4.1	Functional requirements	5
4.1.1	Menu requirement	5
4.1.2	Pause requirement	5
4.1.3	Other functional requirement	5
4.2	Non Functional Requirements	5
4.2.1	Usability	5
4.2.2	Quality	5
4.3	Hardware requirements	6
4.4	Software requirements	6
5	Visual Example of game	7
6	Risk management	8

1 Introduction

“K.O Game” is a 3D fighting game coded in python. As we know that the gaming world has evolved rapidly and a comparatively new language python is started being used by the gaming programmers in the field. This project is an experience to see the working of python in game development.

Scope

The fundamental goal of this project is to build a game that:

- is enjoyable.
- is ready for commercial release.
- has potential for commercial viability.
- offers 3D graphics.
- offers a 3D environment.
- has sound effects.
- approximates physics in some aspects.
- has a low server load.
- is well documented.

Glossary

Term	Definition
Stakeholder	Developer, a publisher.
User	Developer, any person

Technologies to be used

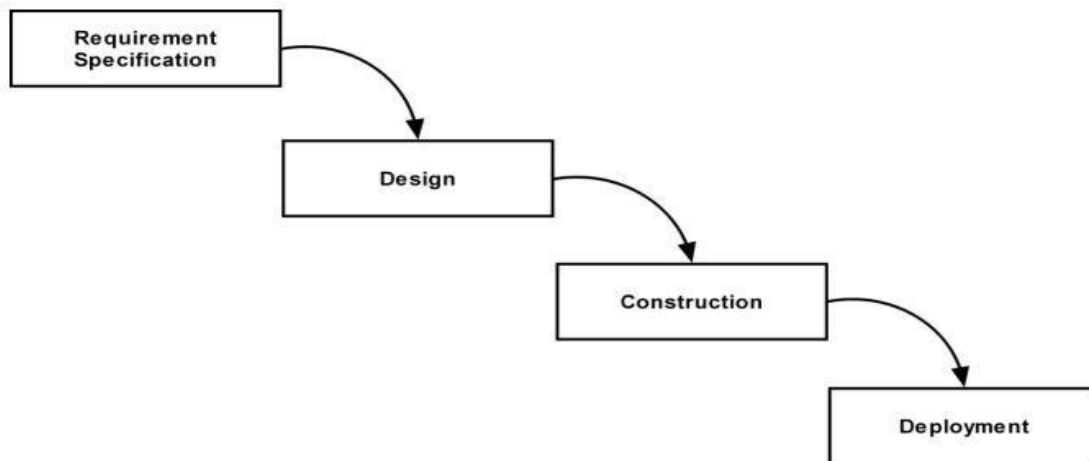
Technology	Description
Development Environment	Being a 3Dgame application of python it has to be done in some gaming engines for the python. Panda3D: It is a gaming engine for python to be used to create 3D environments VS code: Visual studio code is an software that act as host for writing code in many computer languages. GIT bash terminal: To give control commands GIT bash is used in the project. Blender: To model Character and Arena

2 The Process

The process model

Requirements and the design parts of the project will be improved sequentially and they will be completed by the end of first term.

Therefore, we cannot use any incremental or evolutionary model. Also, requirements are well defined - and reasonably stable so there will be limited number of new development efforts in the project. We will try to make a realistic and safe requirements analysis so that we will have a robust design, which will prevent us from returning back to design phase. Taking all of these into consideration, waterfall process model is the most appropriate model for our project. However, there will be some modifications in this model such that coding, testing debugging and integration will be performed - in one phase under the name 'construction', as seen in Figure 1.



General Constraints

Some of the constraints we will face during the project are:

- *Time*
- *Suitable engine to create 3D environment*
- *Detecting errors as a beginner*
- *Scheduling project*

Assumptions & Dependencies

These are the Assumptions & Dependencies:

- User have a device on which the game will run
- The user know the basics of how to deal with the device

3 Research

Panda3D:

Panda3D is a game engine that includes graphics, audio, I/O, collision detection, and other abilities relevant to the creation of 3D games. Panda3D is free software under the revised BSD license.

Panda3D's intended game-development language is Python. The engine itself is written in C++, and utilizes an automatic wrapper-generator to expose the complete functionality of the engine in a Python interface. This approach gives a developer the advantages of Python development, such as rapid development and advanced memory management, but keeps the performance of a compiled language in the engine core. For instance, the engine is integrated with Python's garbage collector, and engine structures are automatically managed.

The manual and the sample programs use Python by default, with C++ available as an alternate. Both languages are fully supported. Python is the most commonly used language by developers, but C++ is also common. The users of Panda3D include the developers of several large commercial games, a few open source projects, and a number of university courses that leverage Panda3D's short learning curve. The community is small but active, and questions on the forum are generally answered quickly.

Visual studio code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter.

Python:

Python is an excellent choice for rapid prototyping of games. But it has limits with performance.

Python provides a built-in library called pygame, which used to develop the game. Once we understand the basic concepts of Python programming, we can use the pygame library to make games with attractive graphics, suitable animation, and sound. Pygame is a cross-platform library that is used to design video games.

Blender:

Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D-printed models, motion graphics, interactive 3D applications, virtual reality, and, formerly, video games. Blender's features include 3D modeling, UV mapping, texturing, digital drawing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animation, match moving, rendering, motion graphics, video editing, and compositing. Blender is popular amongst the FOSS community and beginners due to its availability and free price.

4 Requirement specification

Functional requirements

Menu requirement

Main menu will be displayed to the user at the beginning of the game. Details of the menu are listed below.

Main Menu:

1. Start Game
2. Quit Game
3. Credits

Other functional requirement

Here the functional requirements are presented :

- Players can choose their character
- Players can lose
- Players make different types of sound as a result of their actions.
- Arena, leaf falling, blood effects on hitting, good music and interesting characters are there in the game
- A simple economical model exists to control the flow of currency in the game.

Non Functional Requirements

Usability

Being an education tool, our program will be used by the trainees including both experienced and inexperienced computer users. Taking this into consideration, we try to design human-program interaction as clear as possible. The program is aimed to be easy to learn and satisfying to use. In this respect, we follow a user-centered design paradigm keeping the users' needs in mind all the time.

Quality

Quality of the design and quality of conformance to this design is one of the major requirements of the project's development. The phases should be completed in certain time and the resulting software should be free from the bugs as much as possible. Also it should be user-fault-tolerant to provide a user-friendly environment.

Hardware requirements

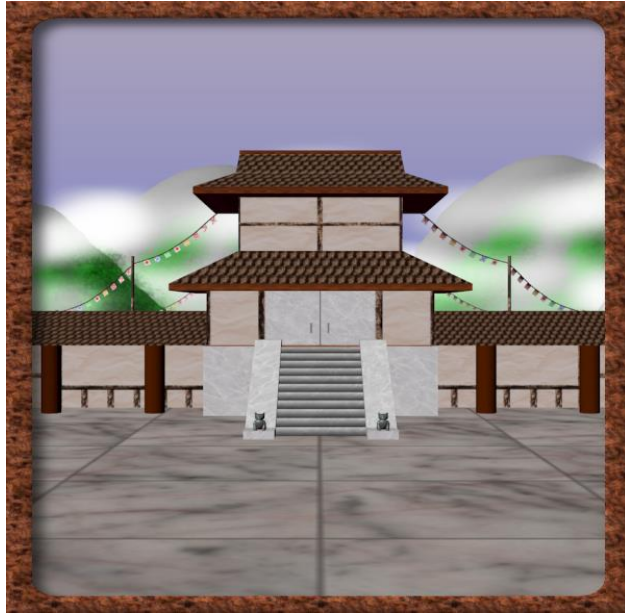
Minimum system requirements are :

- Hard Disk
- Microphone, Speaker, Keyboard, Mouse, Monitor.
- CPU

Software requirements

- Windows operating system
- Linux operating system
- Unix operating system
- Installed python
- Installed code editor
- Installed Panda3D SDK file

5 Visual Example of game Arenas:



Characters:



6 Risk management

Risk management is a key issue for completing a project successfully in time. Having a plan before hand would certainly decrease the impact of a risk. As the subject of the project is mainly about saving human life, it is risky by nature. The people who are using this simulation program are always faced with challenging conditions involving risk. Their each decision may result in many people's death or survival. Therefore, training of them also involves risk. Implementing the project successfully is crucial in this sense. Requirements should be understood very well. The pathways that the user will choose should not cause him to be trained incorrectly and the application should evaluate the user correctly. Especially for this project, not achieving those criteria are possible and hazardous risks. Therefore, we try to do a detailed risk plan and avoid the possible risks as much as possible.