**A Level computer Science**

Component 3

Space Game

Logo

Description automatically generated

By: Louis Pattern

For: A. Issa

**Stoke on Trent Sixth Form College**

Table of Contents

[1.1 Introduction 4](#_Toc112751032)

[1.2 Problem Identification 5](#_Toc112751033)

[1.3 Why the problem is suited to a computational solution 6](#_Toc112751034)

[1.4 Stakeholders analysis 7](#_Toc112751035)

[1.5 Research 8](#_Toc112751036)

[1.6 Features of the proposed solution 10](#_Toc112751037)

[1.7 Limitations of the solution 11](#_Toc112751038)

[1.8 Stakeholder Consultation 12](#_Toc112751039)

[1.9 Hardware and software requirements 13](#_Toc112751040)

[1.10 The requirements of the solution 13](#_Toc112751041)

[1.11 Success Criteria 14](#_Toc112751042)

Chapter One: Analysis of the problem

## 1.1 Introduction

In this project, I aim of entertaining children and teenagers by providing them with a 2D videogame for recreational use. The main goal of the game should be to help teenagers relax in their free time when they are not doing homework or revision. My game will be suitable for college students and even children. Its theme is science fiction and space. The game will have both options for a single player fighting an AI and multiplayer for one player versus another locally, allowing users to play the game how they prefer. The game has the theme of space and science fiction and will involve each player controlling a spaceship on each half of the screen and shooting enemies approaching them. There will be a number of levels in single player, increasing in difficulty as the player progresses. The multiplayer versus mode will be more customisable, allowing the user to change some settings. My stakeholders for this project will include a college student at Stoke Sixthform College, a child and a young adult.

## 1.2 Problem Identification

Many teenagers become stressed or worried about exams and schoolwork. Videogames could help relieve this stress in their free time by providing a means of escapism from their school life. Stress is also a major problem for young adults, so my project would not only be targeting teenagers. Videogames have also been proven to have other benefits. For example, improving focus and reaction time.

Most retro games are fully single player experiences, with no way to interact directly with another human player in the game aside from competing for a spot on a scoreboard after game completion. My project aims to go against this convention by allowing players to directly compete against each over in real time with their scores being tracked and displayed clearly on screen.  
In addition, most 2D videogames that feature a 1vs1 format are usually fighting games, a genre that most people are turned away from because it can be too competitive, forcing players to learn specific ‘combo’ moves reducing the ability for people to play casually and for fun. This could either put people of retro videogames entirely due to frustration or make younger users shy away from multiplayer gameplay entirely. There is also a limited number of retro shooter games with such a format and even fewer with a sci-fi theme.

Furthermore, a lot of 2D shooters feature either just a single large level, a limited number of levels or there is little variation between levels. If there is little change in enemies or combat between levels then the user will become bored due to lack of challenge or stimulation. However, if there is no visual variation between levels then the user can also become bored.   
My game would feature a unique background for each level. This would not only help to keep a user’s interest high (particularly for children) but also make the give the player a sense that they are travelling through different places a galaxy as they progress through the levels.

## 1.3 Why the problem is suited to a computational solution

This problem is amenable by a computational approach because it is a videogame, thus has to involve the user interacting with a computer. My project upon being a videogame rather than a non-computational solution such as a board game has numerous advantages. Firstly, the computer can process the user’s inputs and perform tasks much faster than a human can interact with a non-computational game. In addition, animations and movement for a videogame can be easily displayed by a computer screen whereas in a normal game they cannot. Colours in videogames can also be adjusted, allowing the option to enable a colour-blind mode. Finally, most people with disabilities are still able to play because usual input methods such as mouse and keyboard do not require much movement.

Videogames also allow the user to have more customisable settings, for example: window size and colours used.

## 1.4 Stakeholders analysis

My stakeholders for this project will be, a college student at Stoke Sixth Form College, a high school student and a young adult. I will give each of my stakeholders a demo version of the game and interview them for feedback and criticism. If any of the stakeholders suggest a way to improve the game or a new feature to add to the game, I will try to implement it.

The high school and college students I have selected are gamers, playing a variety of videogame genres, whereas the adult is not a gamer. Doing this will allow me to gather a wider range of feedback and to find out if the game will be intuitive and easy to learn even for someone with little experience in PC gaming.

## 1.5 Research

One existing videogame of a similar format is Defender, an arcade game from 1981. This game is a side scrolling shooter where the player has the objective of shooting aliens on another planet. The player can move in all directions, with up and down moving the ship directly and left and right moving the terrain. However, because of technical limitations of the time, the background is extremely simplistic, with the ground consisting of a single zig-zagging line.



The game is also single player only, something which I aim to improve upon in my game. Different enemy types feature in Defender, with each alien behaving differently and awarding a different amount of points when the player destroys them. These enemies are also vastly different visually, allowing the player to easily differentiate between them. This visual difference also allows new players to easily learn the behaviour and mechanics of each enemy. Another gameplay element of Defender is that the player can rescue captured humans by shooting landers.



## 1.6 Features of the proposed solution

* Single player mode where the player fights against enemies controlled by AI.
* Multiplayer ‘versus mode’ where one player fights another. This will give my project a unique feature compared to other scrolling shooter games. This would also allow for friendly competition that isn’t possible in only single player games.
* High score leader board, featuring the player’s name followed by score. This information will be stored in a database and will be displayed when a player completes the game or they manually select ‘high scores’ from the main menu. Scores should be displayed in descending order and each difficulty level will have a separate leader board. The date when the score was obtained would also be stored in the database. Only the top 5 or 10 scores would be displayed, and these scores would be arranged in descending order with each player’s name clearly next to their corresponding score.
* Difficulty levels: easy, medium and hard. For higher difficulties, enemies will have faster fire rate and their projectiles will travel faster, thus being harder to dodge and more enemies will be present in each level. The AI will also become smarter as you increase the difficulty, with the enemies at the highest difficulties actively attempting to dodge the player’s bullets. Completing the game at the highest difficulty will award the most points and playing on lower difficulties will earn less points.
* The limited number of lives the player has which will be correctly displayed at the top (or bottom) of the UI. The number of lives will depend on the difficulty. For versus mode, the number of lives each player gets could be adjusted directly by the user.
* Some enemies also have multiple lives like the player, but they don’t receive invincibility frames like the player does.
* Invincibility frames: after a player takes damage and loses a life, they will be invulnerable and unable to lose more lives for a short period of time (≈ 0.5s). This feature will be accompanied by a flashing animation on the player’s ship. This is implemented because it prevents the player from losing a life for every frame they are in collision with a dangerous object. In other words, it prevents the user from losing multiple lives in very quick succession.
* A range of projectiles depending on whether they are being fired by an enemy or a player.

## 1.7 Limitations of the solution

* One limitation of the solution is that the game can only use keyboard inputs alone and there is no support for mouse, controllers, or joystick input. This is acceptable because the most likely audience for the game – college students during their free periods, would be unlikely to carry controllers with them.
* Another limitation is that the max frames per second of the game is capped at 60 fps. This is because movement in the pygame loop is based frame by frame rather than real world time so increasing the frame rate beyond 60 increases the speed at which the player can move etc. This limitation is not a major issue since 60 fps has been the standard for gamers for a long time, especially for 2D games, where stability and a constant frame rate is often preferred over a higher, fluctuating frame rate. 60 frames per second appears smooth enough to the human eye. In addition, almost all modern hardware would be able to run my game at a stable 60 fps.
* The graphics for the game are all simple 2D shapes or images.

## 1.8 Stakeholder Consultation

## 1.9 Hardware and software requirements

Hardware requirements:

* 1.5GHz or faster processor
* 4GB RAM
* 1GB free hard drive space – to install the source code, graphics and audio
* Working keyboard and mouse – the game input is all keyboard based and the login menu needs mouse for input
* Working monitor – to display the game’s GUI

Software requirements:

* Operating System: 64-bit Windows 7 or later or OS X 10.11 or later
* Python and pygame would NOT be needed because the game is run through an executable (exe) file.

## 1.10 The requirements of the solution

* Firstly, the user would need to log in with the log-in screen displayed when they launch the program. This would be done by entering their username and password into the boxes and hitting enter. This is done so their username can be saved to the high scores database (along with the score they get)
* The user can navigate the main menu with WASD or arrow keys and select an option by hitting the enter key or the spacebar.
* When in game, the player can move their ship normally with WASD, enable slower movement or ‘focus’ by holding shift while using WASD to move. Spacebar is used to shoot and can be hold down to shoot repeatedly. There is a maximum fire rate.

## 1.11 Success Criteria

* Fully functional game with minimal bugs or exploits.
* User would have to install python or PyCharm in order to run the program. This would make it simpler and easier for players to install and run the game.
* Users find the game enjoyable and fun to play, whether playing single player or multiplayer.