Computer Science Project – A Rough Report

Call of 2-D

**Acknowledgements**

The members of the group behind “Call of 2-D” would like to extend their sincere gratitude to their teachers – in no particular order - Ms. Nirmala Devi and Ms. Harifa Begum for all the support that was given to them during the development of this game, without whom, this project would have been non-existent.

**An Introduction**

Electronic entertainment is a booming industry that boasts revenue of nearly $66 Billion, with projections showing that the $100 Billion mark will be reached before the turn of the decade. Such remarkable success stems from the quality of the sheer quality of the video games that are the industry’s products, which play highly influential roles in the lives of adolescents and adults alike.

As video game fanatics, the members of this group have tried to build, on a modest level, something close to what they have spent a vast portion of their time on, often with gratifying results. The outcome of this endeavor - to design a video game (more specifically, a shooter, which happens to be the members’ favorite genre) – is a 2-D side scrolling, platformer and shooter, which attempts to recreate the addictiveness and the sheer pleasure that such game deliver; what was once a dream, a fascination, now realized.

The group presents to you, Call of 2-D.

**Programming Aspects**

This project was built using the programming language, “C++”, using a variety of techniques, both logical – algorithm formation, problem solving and idea realization – and technical – the use of an advanced programming language and the features of Object Oriented Programming (OOP).

1. **Usage of Files**

Files provide a means of storing data in a more permanent manner, as opposed to “temporary data” that is stored on the computer’s Random Access Memory.

This project employed the use of files to store a number of things, such as a database for highscores and other game-related data, amongst others.

1. **Arrays**

Arrays are defined as a collection of variables of the same data-type, stored in continuous memory locations and most importantly, under the same variable name. Arrays are best utilized when there is a need to store multiple data of the same type and need to be accessed as both a single unit as well as discrete units, as was done in this program.

3. **Structures**

A data structure is a group of data elements grouped together under one name. These data elements, known as members, can have different types and different lengths.   
Where an array has a drawback, in that it can only hold data of the same type, a structure or a class can accommodate data of a variety of types.

1. **Pointers**

Pointers, as the name suggests, is a data-type that directly refers to (or “points” to) the data held in another memory location using its address. These are especially effective in controlling a variable in multiple locations, by simply having a data-type that stores its locations. Pointers also help control data that is stored in the dynamic RAM which is not directly available, thus increasing memory efficiency, and offers the option of de-allocating memory once it is no longer in use, that isn’t possible in the case of variables stored in static memory.