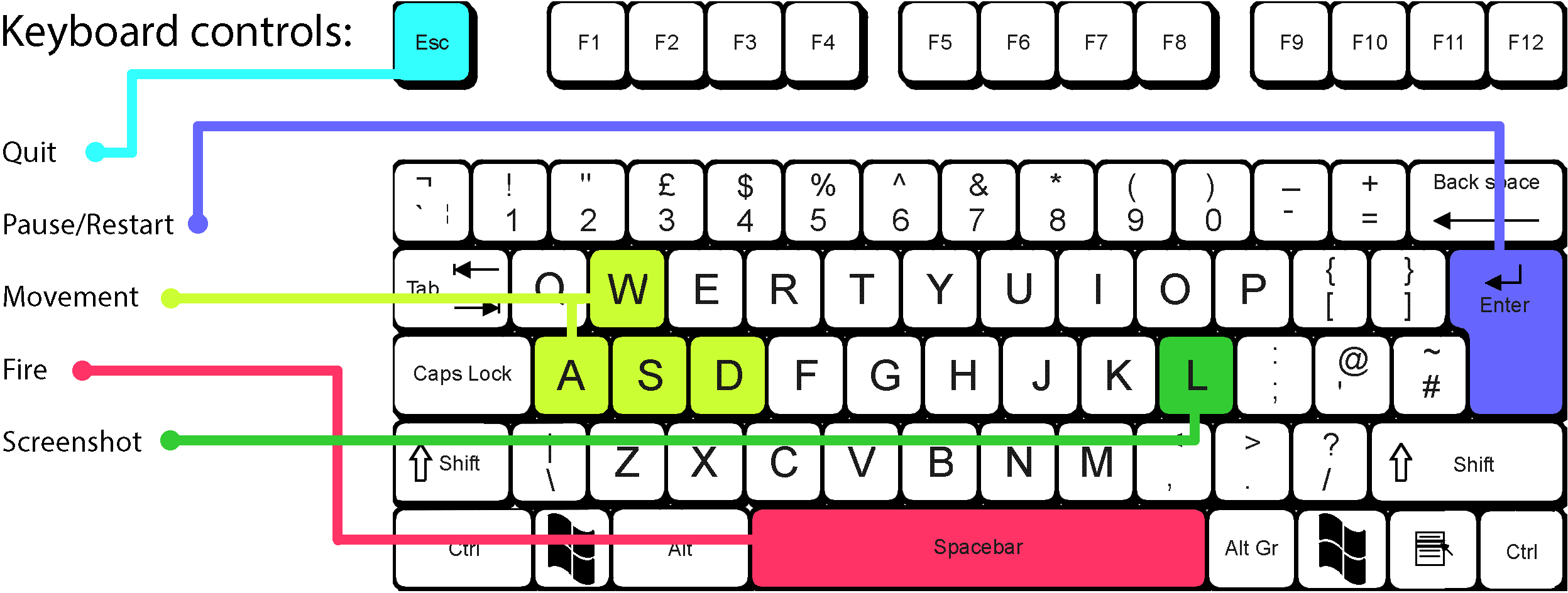
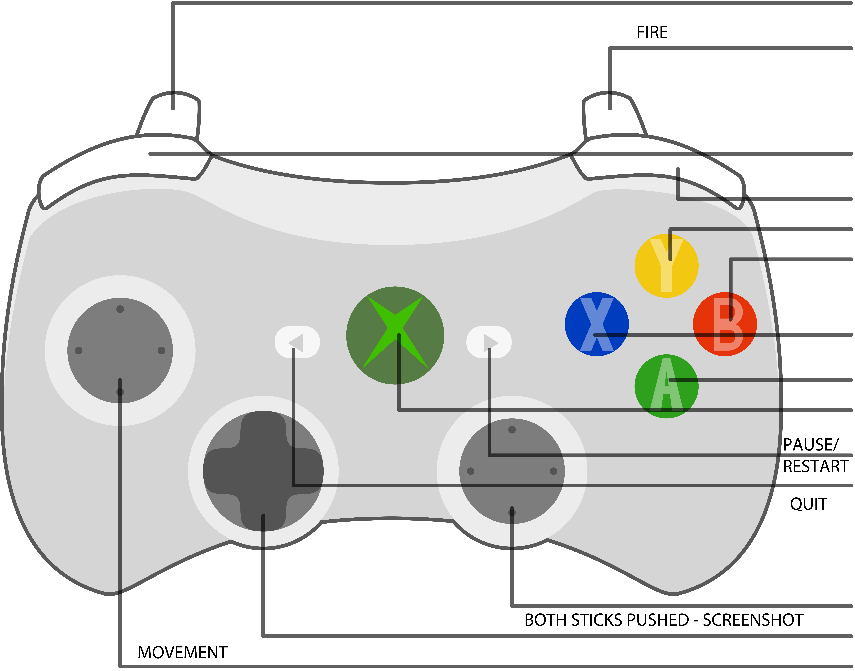
User Guide

**How to play**

This game is a simple 2D sidescrolling shooter, the only major controls for the player would be to move and to fire the cannons. Below are two diagrams for the two ways to play this game:



Keyboard:

 Controller:

Keyboard/Controller diagrams -

<https://www.bristol.ac.uk//it-services/learning/documentation/keyboard-1/keyboard-r1-6.gif>

http://www.clker.com/cliparts/4/b/b/6/1348595413586720353Xbox%20360%20Controller%20Diagram.svg

**Known bugs**

There are some known bugs within the game, most of which do not affect gameplay. Some small issues users may find when playing this demo are:

* When respawning, you may respawn inside of a wall, which will kill the player instantly and give a game over.
* The Pause function sometimes resumes immediately after pausing, due to the corresponding button being pushed for a fraction of a second too long.
* The Screenshot function will pause the game for half a second whilst it records and saves the current screen.

### **Implementation check list**

**Adequate Rated Items (D to C)**

**Graphics**

* A ‘black boxed’ graphic system is in place
* Textures can be efficiently drawn to arbitrary positions on and partially off the screen (clipped)
* Animation is implemented and working correctly

**World State**

* A player entity exists
* Input is recognised and can be used to alter the world state e.g. move the player entity
* The Xbox controller is supported and ‘hot pluggable’

**Code Quality**

* Class interfaces are minimal and complete. Class function and member variable visibility is correct
* Code can be built and executed without compiler errors or warnings in debug and release
* Code is well commented
* There are no memory leaks
* There is good error handling throughout
* You have followed all the submission requirements e.g. made a video, submitted the correct files etc.

**Report**

* All requested sections have been attempted adequately and the report is professionally presented

**Good Rated Items (C to B)**

**World State**

* A world model system is in place. It is separate from other code and black boxed
* There is a game loop handling input, world update and rendering
* Bounding rectangle collisions are detected
* There are multiple world entity types

**Code Quality**

* Good use of object oriented techniques e.g. polymorphism, member variable visibility
* Memory is only allocated / deallocated outside of the game loop
* Const is used correctly

**AI**

* Some AI routines are in place e.g. enemy entities move around the world following paths, use state machines etc.

**Report**

* This report would allow another programmer to work with your code systems

**Other**

* Some sound effects are in place

**Excellent Rated Items (B to A)**

**Graphics**

* Interpolation is used to smooth entity movement

**World State**

* The player entity can shoot projectiles (or equivalent functionality)
* Explosion and bullet management
* Game play is independent of platform capabilities (i.e. uses a model tick approach)
* Game cycling e.g. detection of win / lose conditions and restarting the game
* There is a scoring system with the score shown on screen

**AI**

* Several different enemies with differing behaviours

**Report**

* This report has insightful and balanced reflection

**Extra Marks (Examples)**

* Mapping of world space on to screen space
* Other graphics techniques have been implemented e.g. background scrolling, blending modes etc.
* Level data is loaded from a file
* A difficulty level
* More advanced C++ e.g. use of namespaces, STL, C++ 11 and further patterns
* A\* algorithm
* ‘Intelligent’ enemy behaviour
* There are sound effects for collisions, explosions and firing
* Additional black box systems have been implemented e.g. for AI, Sound
* HUD features beyond simple text e.g. health bars, mini maps etc.
* Other features, please list below:
* Loading a number of the same texture to create a seamless wall, reusable and able to create more walls when required.
* Replayability with difficulty increasing with progression.
* Entities are very easily manageable within the World classes