|  |
| --- |
| East Riding College |
| Block Breaker |
| Introduction to Programming: Assignment 1 |
|  |
| **Tindall, Kai** |
| **10/5/2018** |

|  |
| --- |
| Design documentation for one of my first year assignments, Block Breaker. |

Table of Contents

[Summary 1 2](#_Toc526509603)

[scenario 1.1 2](#_Toc526509604)

[Specification 1.2 2](#_Toc526509605)

[Basic specification 1.2.1 2](#_Toc526509606)

[Extended specification 1.2.2 2](#_Toc526509607)

[Research 2 3](#_Toc526509608)

[Purpose of research 2.1 3](#_Toc526509609)

[Design 3 3](#_Toc526509610)

[Initial design 3.1 3](#_Toc526509611)

[screen mock-ups 3.1.1 3](#_Toc526509612)

[Screen logic 3.1.2 6](#_Toc526509613)

# Summary 1

## scenario 1.1

I have applied for a job at a bespoke industrial software development company that produce programs ranging from web and desktop environments to direct machinery control units. For the interview I have been asked to showcase my skills in C# by creating a “block breaker” game.

## Specification 1.2

Basic specification 1.2.1

1. The game must be a Command Line C# Game.
2. The game must be based around the “*Block Breaker*” game.
3. The game must allow **at least** one human player.
4. The game must be easy to navigate and use.
5. The game must contain a menu system which allows access to scores, new game, exit, etc.
6. The game must keep player scores in the current game.
7. The player should be able to enter their name.
8. The game must include a **minimum of two** layers of blocks in a level.

Extended specification 1.2.2

These are all specification points that are not required but would be nice if I have the time to include them.

1. “Guard mode” – Standard block breaker game but with a twist. Either another player or the computer will try to block your shots from hitting the blocks, if you get your ball through the wall, you win; but if your opponent manages to make you use up all your lives, they win.
2. Power-ups – Random power ups / score multipliers will pop up now and again and if you collect them you will receive a buff of some kind.
3. Score preservation – Scores will be preserved between different instances of the program.
4. Colour – Add random colours to the blocks to make the game more pleasing to look at.

# Research 2

## Purpose of research 2.1

The purpose of this research is to help generate ideas about how I want my game to look and feel, I will also be assessing if there are any parts of the game I enjoy and would like to include, and also if there are any parts I maybe dislike and would prefer not to have in my game.

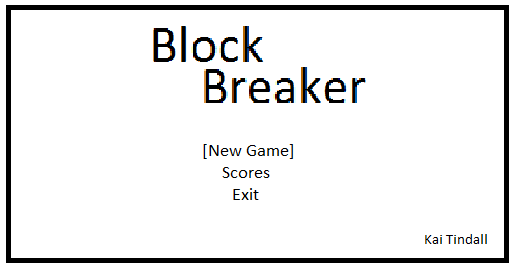
(Research different permutations of the game)

# Design 3

## Initial design 3.1

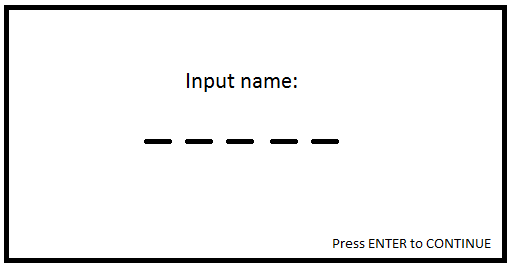
screen mock-ups 3.1.1

Menu 3.1.1.1



Notes: The user will use either W/S or UPARROW/DOWNARROW to traverse the menu and press ENTER to select the option currently highlighted by the square brackets.

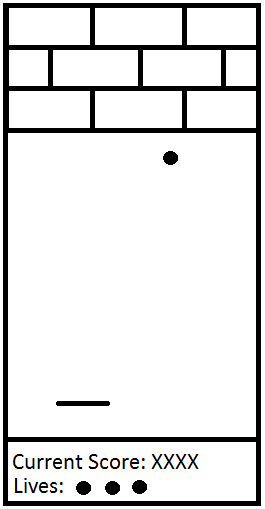
New Game -> Input Name 3.1.1.2



Notes: The name will have a maximum of five letters; this is to make sure the score screen will have no issues formatting the scores later on.

The user will input their name by typing it, each letter they type will replace a dash with their desired letter. They will be able to press the BACKSPACE key to remove their previous letter. To finish inputting their name they will press the ENTER key.

Game Window 3.1.1.3

Notes: The bar along the bottom is the player and will be able to move along the x-axis with either the A/D or LEFTARROW/RIGHTARROW keys.

The ball will be bouncing off of the player, side walls, and bricks. If the ball hits the wall below the player, the player will lose a life and the ball will be eliminated from the game. A new ball will spawn above the player.

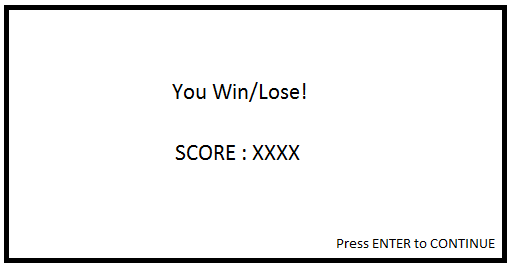
The player’s “paddle” will be cut into different sections; depending on what section the ball hits will change the resulting angle of reflection. Sections closer to the outer edges will have a shallower angle than the middle.

The blocks will be in a brick pattern.

The lives and current score labels at the bottom of the screen will show the current amount of lives and the current score.

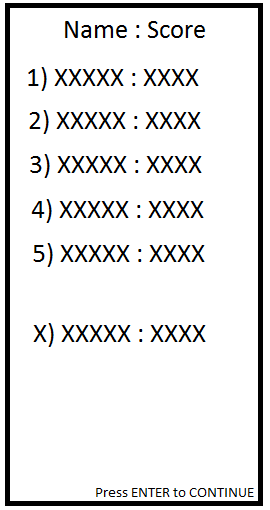
The screen will be portrait.

Win/Lose Screen 3.1.1.4



Notes: Will display whether you won or lost and show your score. If you will your name and score will be added to the scoreboard and you will be taken to the scoreboard on the next screen. If you lose however, your score will not be added to the scoreboard and you will just be taken back to the menu screen.

Scoreboard 3.1.1.5

Notes: The scoreboard always shows the first five high scores, but changes depending on if you’re accessing the screen from the menu or straight out of winning a game. If you have just won a game then the five top scores will be shown, and if your score is in the top five your score will be bold. However if your score isn’t in the top five then your score will appear below the top five. You will be told your ranking and your name and score will be bold

If you are accessing this screen from the menu, all scores will NOT be bold and it will only show the top five scores.

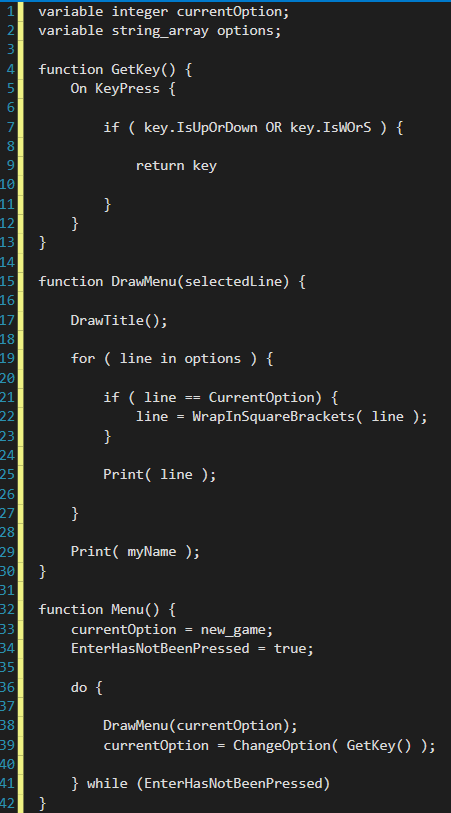
The screen is portrait.

Screen logic 3.1.2

Menu 3.1.2.1

Summary: This screen will draw the menu, and then wait for a key to be pressed. If that key is one of the set keys for up or down, the menu will change accordingly. If the key is the ENTER key, then the highlighted option will be selected. If the key pressed is anything other than those keys, then the menu will do nothing.

Pseudocode:



Input Name 3.1.2.2

Summary: This screen will listen for alphanumeric keys being pressed and present a name from those keys pressed. It will start with five dashes onscreen and as the user types in their name, the dashes will change to be the letters they input. If the user presses the BACKSPACE key then the last letter the user input will be deleted. However, if the user presses the ENTER key, then the name will be entered and they will be taken to the game window.