Requirement Specification for ‘BrickBreak’ Game

Game created for University of Birmingham Computer Science Team Project module.

Prepared by (Group A5):

Hon Yiu Wong

Tamara Herbert

Edward Richards

Xuan Chen

Paul Roskell

Olexiy Dubilet

Date:

26/01/2017

Contents

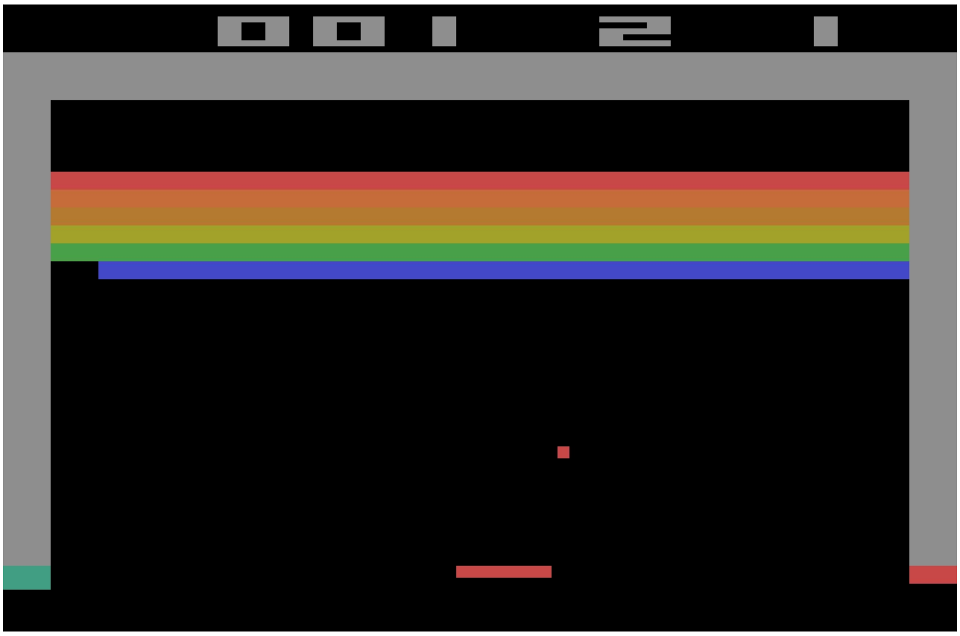
Introduction

Purpose

This document provides the requirement specifications for a multiplayer computer game, BrickBreak, a game influenced by the 1972 game Atari Breakout.
The document specifies user interface attributes, requirements, and long-term ideas for the evolution of the system.

Overview of the original game

Breakout is a 2-D arcade game. The player controls a small paddle that is situated at the bottom of the map, and their main objective is to destroy all blocks present on the map by bouncing the ball off their paddle.
The layout of the game is simple, with a set of blocks towards the top of the screen for the player to destroy, a paddle at the bottom which the player can move in the x-axis, and three walls, at each of the sides and the top from which the ball can bounce off.



An image displaying the original 1-player Atari Breakout.

The ball is always on the move and the player must not let the ball fall below the platform, if that happens the player loses a life. If the ball hits an object in this case either a block, the walls or the paddle, it will bounce off at the corresponding angle.
Player must destroy all the blocks that are located mostly at the upper part of the map.

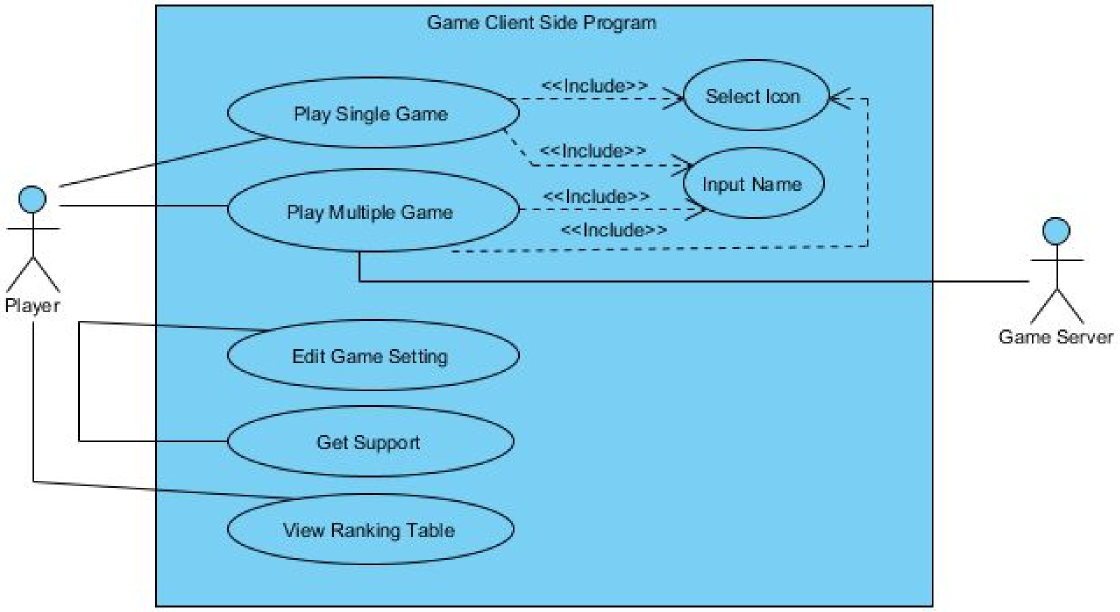
Concept of new game

When comparing the game to be created to the 1972 original there will be many differences, the main one being that the new game is based around the concept of multiplayer interaction. BrickBreak is for 1-6 players, and instead of basing the game around how many blocks a player can break before they run out of lives, the game will be points based, with a winner or loser being decided by how many points they can get before the time runs out, which is when the game ends.

Specific Requirements

Functional Requirements

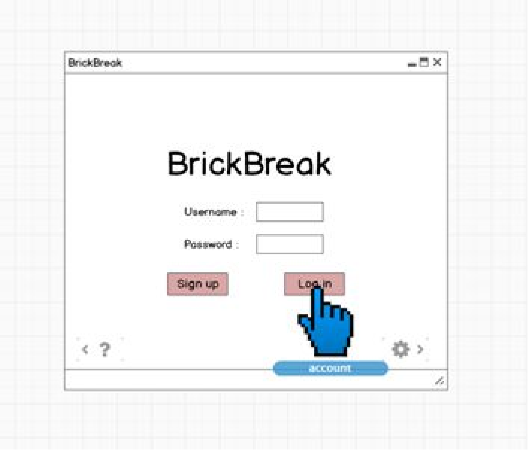
Use Case Diagram

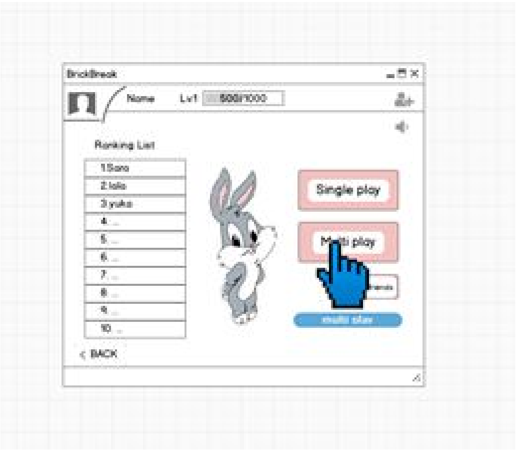


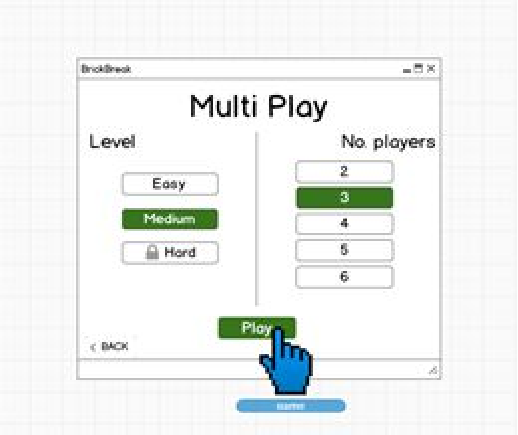
User Interface

There will be 6 states of the program:

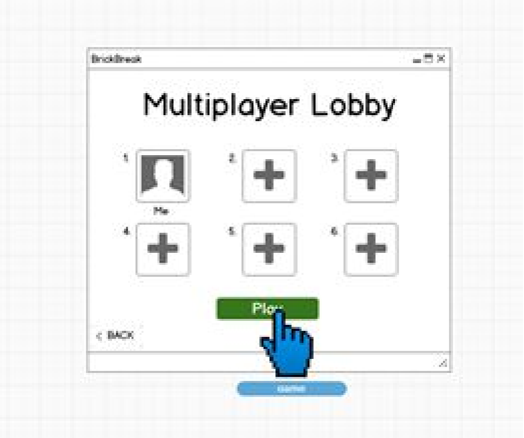
- Main menu - the player is able to decide whether he wants to play Singleplayer or Multiplayer, and has access to “Options” and “Support”.



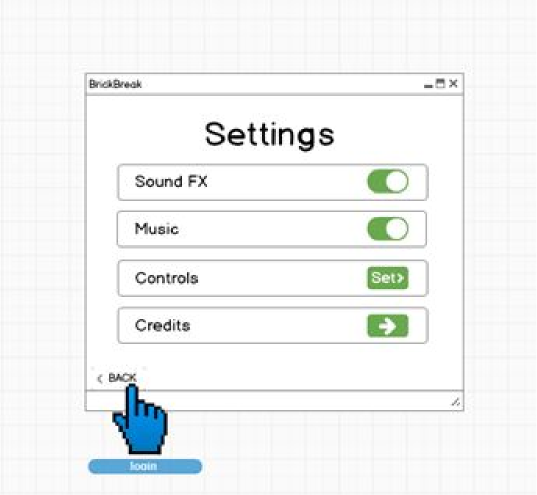




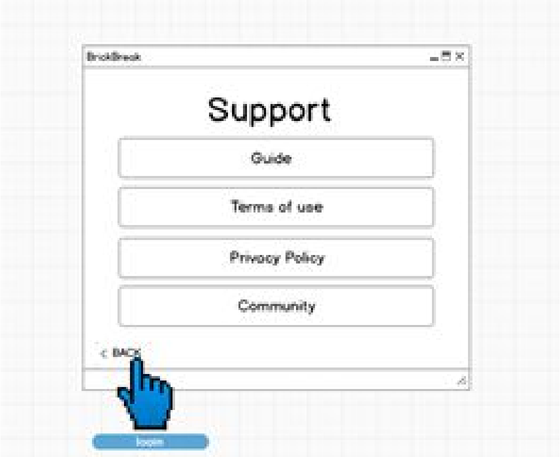
- Multiplayer Lobby - when the game is created, all the players can access the server and wait for the other players to be ready to play.



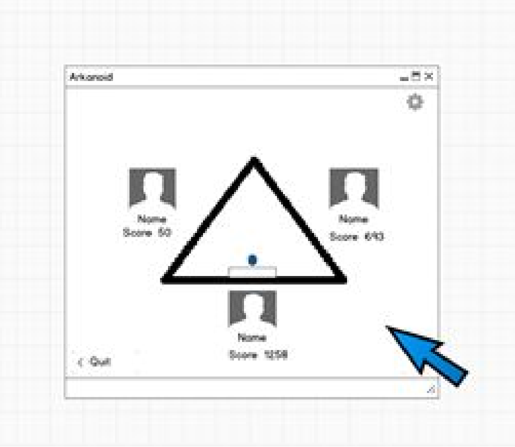
- Options menu - located in the main menu. The player can adjust options such as music volume, sound volume and control mode.



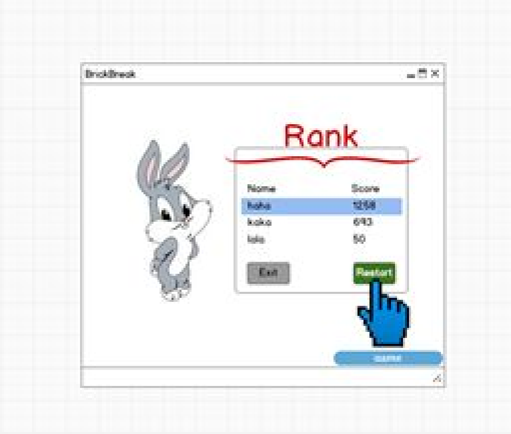
- Support - a state where the player can find the guide of the game.



- Game state - a map where gameplay is happening. There will be only a single element of UI on a map - Menu, that will be located on the top right/left corner.



- Score menu - a tabulated display of scores for all participating players in the prior game, with the winner made clear.

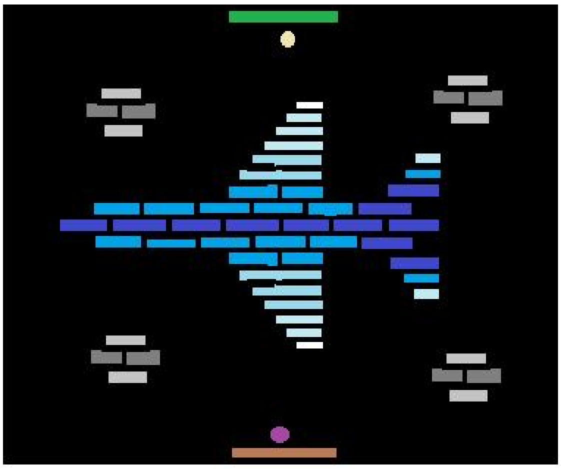


Name	Score
haha	1258
koko	693
lolo	50

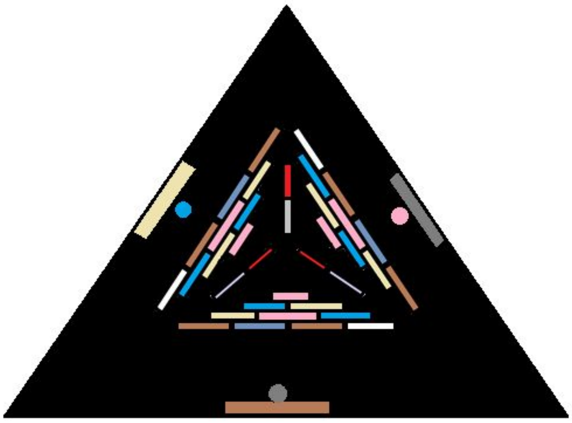
Within the Game State there will also be different maps and levels dependent on the difficulty chosen by the user, and how many users are participating.

In our version we will introduce a competitive gameplay for 2-6 players simultaneously. In addition to a traditional 4-angled shapes there will be 3, 5 and 6 angled shapes to accommodate all players.
These are some examples for block layouts for each number of users:

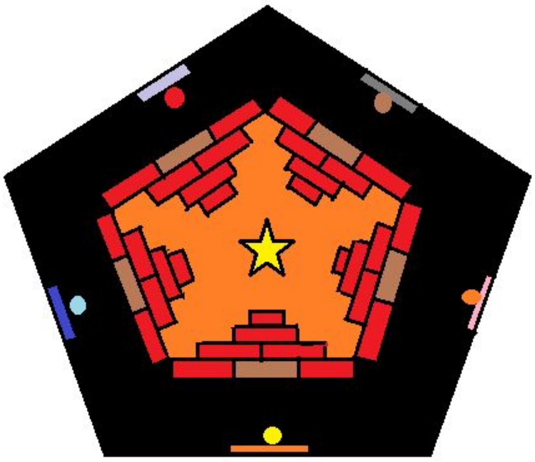
- 2 and 4 players will be playing on a square/rectangle shaped map.



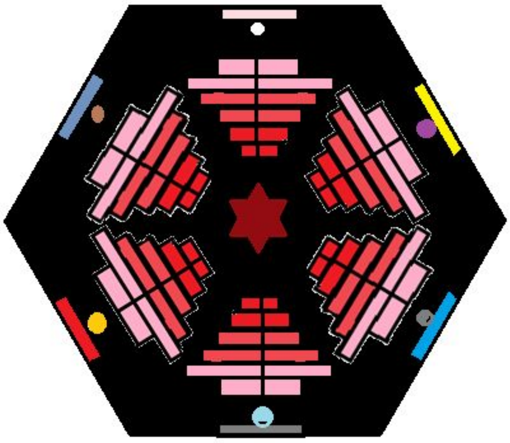
- 3 players will be playing on a triangular map.



- 5 players - regular pentagon.



- 6 players - regular hexagon.



Game Play

The players will share a common map and the same blocks to destroy. There will be a ball for every player. So if there are 5 players - there will be 5 balls present on the map. Balls are not restricted to any particular player, so every player could “bounce” any ball he can reach.

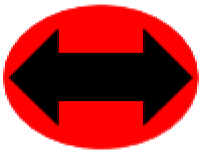
Players will lose points every time a ball passes below their paddle. Every time that happens a new ball will be inserted into the game, since the old ball is lost.
Points are scored by breaking blocks. The points are awarded depending on whose platform the ball last touched. The winner is the player who earned the most points.

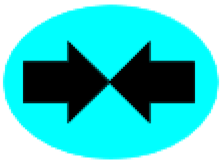
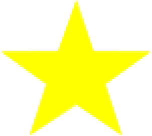
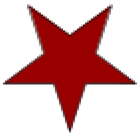
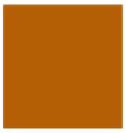
Bonuses

Some blocks if destroyed can “drop” bonuses.

Beneficial for player A	**Harmful for Player A**	**Harmful for B-E**
Increase platform size	Decrease platform size	Decrease platform size of others
Shield on the goals for 3 misses		
Gain extra points	Lose point(s)	Other players lose point(s)
	A new moving indestructible block near Player As goal	A new moving indestructible block near the other players goals
	A random AI player that spawns for 15 seconds near you and tries to block your balls	A random AI player that spawns for 15 seconds near you and tries to block other players

If a player catches the bonus, by the bonus landing on their platform, it will take effect immediately.
Every bonus icon will be unique and be clearly representing what bonus it represents, below are examples:

	Increase players platform size

	Decrease players platform size
	Gain extra points
	Lose points
	A random AI player that spawns for 15 seconds near you and tries to block your balls

There is a chance you will get a bonus depicted by a question mark. If a player catches it - he will receive a random bonus, either beneficial or harmful.
There will be a maximum of one bonus affecting a player at one time. If a player acquires a new bonus while currently affecting them, the new bonus overrides their existing bonus.

Audio

The game will include implemented audio, with both background music whilst the game is running, and spot sound effects for different events, as follows:

Event	**Sound**
Menu Button Action	Click
Ball hits brick and brick is destroyed	‘Break’ sound
Ball hits brick and brick is not destroyed	‘Crack’ sound
Ball hits wall or platform	‘Pop’ sound
Ball is missed by platform/life is lost	‘Sad’ sound
Positive power-up obtained	‘Happy’ sound
Negative power-up obtained	‘Angry’ sound

All audio included should be able to be switched off by use of buttons always on screen, or through an ‘Options’ option in the menu. The background music should be able to be switched off independent of the spot effects, and vice versa.

An important aspect of the audio is that any included should be royalty-free.

Artificial Intelligence

If the player has chosen a mode with more space for players than they have human participants, they will have an option of adding AI bots to the game that will substitute real players.

AI in control of one of the platforms will be aiming to win the game by:

1. Destroying as many bricks as possible.
2. Not missing balls into its goals.

There will also be an AI element through one of the negative bonuses, where an AI player is placed above a player trying to block the player from receiving the ball, and therefore being unable to receive points.

Multiplayer

There will be an option in the game to play against other users as well as AIs, which will be facilitated by use of a central server.
The players will be able to find other users looking for an online multiplayer game by use of the lobby, and create games from there.

Networking

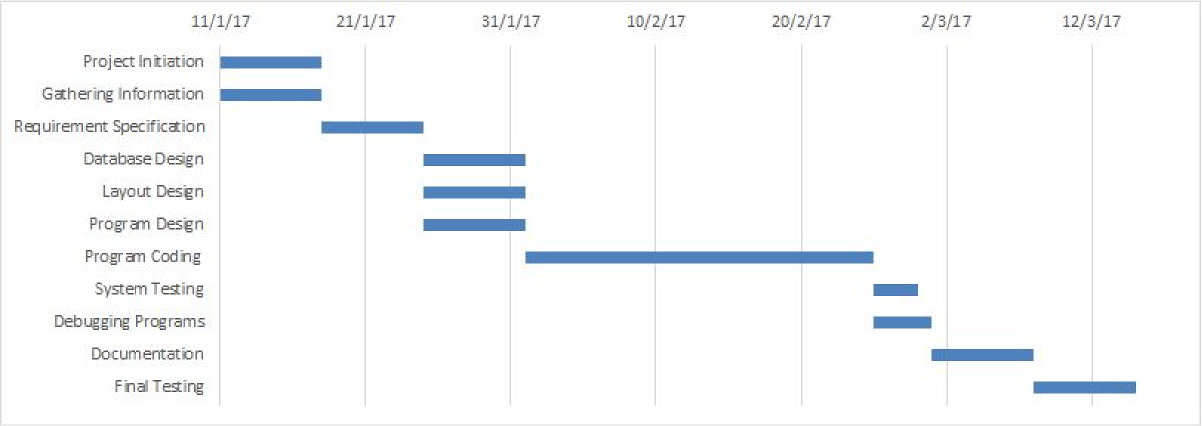
There will be a central server that each client will connect to in order to initiate games, and to communicate with other clients during gameplay. Most of the processing regarding ball and paddle movements will take place client side to reduce bandwidth and lag, but the clients will regularly update each other with their user's data (e.g. paddle coordinates, velocity, power ups, etc) so all clients stay in sync. If this proves to be an issue we will look into running more of the processing server-side. We anticipate the server being used as a 'dummy' gateway for the clients to communicate with each other, and will perform no to little processing of the game itself.

Database Design

There will be a small database aspect to the finished program. It will just contain one table with a list of usernames and the corresponding highest obtained score.

Production Schedule

Description	Start Date	Duration (Days)
Project Initiation	11/01/2017	7
Gathering Information	11/01/2017	7
Requirement Specification	18/01/2017	7
Database Design	25/01/2017	7
Layout Design	25/01/2017	7
Program Design	25/01/2017	7
Program Coding	01/02/2017	24
System Testing	25/02/2017	3
Debugging Programs	25/02/2017	4
Documentation	01/03/2017	7
Final Testing	08/03/2017	7



Risks

- Examinations for half of the team members in Weeks 6 & 11.
- Unforeseen software faults going undiscovered late into the project, potentially leaving little time to refactor the code to remove it.
- Git merge conflicts could delay the development of certain features beyond their allotted deadline, if collaboration is deficient.

To deal with these risks, we aim to finish our prototype early before week 6, so that work does not clash with examinations, and aim to finish our project early, leaving room for final exams and unexpected issues.

Expansion

From the start of the project, the stages below are what we need to aim for sequentially during production to ensure we reach the required standards. To begin with, we will aim to complete stage 1, and then progress.

Stage 1 - Minimum goal

- 2 Player BrickBreak
- Over a networked connection, or vs an AI
- No bonuses
- Limited audio, background music
- One level of difficulty

Stage 2

- Up to 4 players, square map
- If playing with 3 players one is a forced AI
- Spot sound effects
- Two levels of difficulty

Stage 3

- 3 player triangle map included
- Bonuses included
- Leaderboard
- Three levels of difficulty, Easy, Medium and Hard

Stage 4- Maximum Goal

- Multiplayer lobby included
- Up to 6 players

Testing

This section will specify what tests should be run to ensure good working of the program on completion.

Basic Testing:

- Every button displayed on the Menu should provide a related function for player.
- Ensure game can be played as corresponded mode and level.
- Setting options such as music controls should be worked.
- Username and password should be checked with validation.

- (e.g. password should be at least 8 characters, check for empty, weird characters, already existing usernames etc.)

In-Game Testing:

- Check each block is hit the correct amount of times before it breaks.
- Score is incremented for each player correctly.
- Each bonus does what is expected.
- Every player can only have one bonus at the same time.
- AI function should be available for every game modes and levels.

Multiplayers Testing:

- Real-time Synchronization.
- Disconnection of player handle action.

Public Testing:

- Bandwidth of the server.