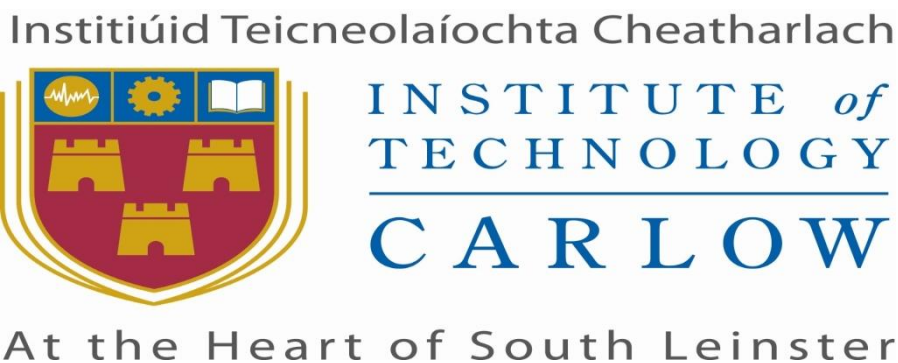# Hashi Puzzle (Islands and Bridges) Functional Specification Document

Created By:

Wojciech Teodorowicz

Technical Document

Student Number: C00193622

Supervisor: Joseph Kehoe

Institute of Technology, Carlow

02nd September 2018

Contents

Abstract

The purpose of this research document is to provide a written account of the research conducted over the course of this project's lifetime. The following document details numerous different technologies and methodologies used in the application that will be changing throughout, as it is a living document.

Introduction

As a part of the 4th year software development student, I was asked to choose a project, I have chosen to implement an Android Application in Java using Android studio.

This document describes the user requirement and functional specification of the Islands and Bridges project.

To describe the procedures involved, a Use case diagram notation is used. The purpose of this documentation phase is to identify requirements and specifications concerning user input, system response, system data and related hardware and logistics matters.

The android application is a puzzle game called Islands and Bridges also known as Hashi the aim of the game is to solve the map by connecting islands using bridges, the user will be able to compete against different players based on their score, the score is calculated using the time it took the user to complete the puzzle and what map they have chosen.

Project Description

As described in my research document he main purpose of this project is to provide users with a puzzle game that will have the ability to let the users to compete with each other. The project will provide the customer the ability to play 3 different levels the user will select the hardness level based on his skills and knowledge of the game.

Goals

A fully functional Map that generates Islands at random

The main goal is to implement a puzzle game where the maps are generated at random everytime the users click new game, this makes every map unique and different ways to solve the puzzle also allow the users to compete by calculating how long it took the user to solve the puzzle.

Get all the documentation and implementations done before deadlines

Getting work done by specific amount of time frame is part of agile software development process. Keeping up with the document and deadline is the way to keep the focus and get the work done and to achieve functional software at each iteration.

Project strategy

The project is consisting of:

· Main menu

· Map

· Cloud AWS.

Main Menu

This is the main menu of my app where the user can choose to either start, continue, read rules of the game or simply close the app.

The start button will provide the user with a alert box to select the mode of the game Easy (7x7), Medium (10x10) , Hard (14x14).

The Rules of The Game button will allow the user to access the rules at anytime they would like to.

Map

The Map will be generated by random every time the user wishes to play the game making every map unique and more options to play the game.

Cloud AWS.

This is where I will store all the scores from different user and the lead board to compare the score of the players with each other. This makes the game a little bit more competitive and could interest the player more.

External Interfaces

Database

The data will be stored in the cloud. Our goal is that whenever a new data is generated for the player this data will be saved in the cloud.

This will be implemented using AWS and java.

We have chosen MongoDB because it allows two types of database structures a non-relational and relational.

Database vs Cloud storage

Local database (few basic and imp things that you own)

- Infrastructure (includes compute, upfront resource allocation)
- Security
- Maintenance
- License management
- You invest in resources today which would be required tomorrow.

Cloud **Database:**

- First of all cloud is more than just a database. In nutshell it's whole infra and engineering.
- You pay for what you use, right from I/O operations.
- Maintenance and managing infrastructure is not your headache.
- Security is precisely controlled because you know you are paying for the best cloud computing in the industry.
- Instant resource/super-cell/cell allocation
- Enabling doors to distributed environment quite easily.
- Scale horizontally(application) as well as vertically(infrastructure).
- Leveraging well tested ready to eat services.

Generating the map

I will implement a algorithm to spawn nodes on the map at random using a two dimensional array the array will produce random integers from 0 to 4 where 0 is an empty Cell and 1,2,3,4 are the islands the numbers will be allocated randomly to the Islands the array will be displayed in a matrix format , the generated map also will need an algorithm that will check if the map is solvable before presenting it to the user:

1. pick a random (or by some other criterium) island that still needs bridges

2. build a bridge between this island and one of its neighbors (obviously a neighbor that also needs a bridge)

3. push the new state of the game (for instance the connectivity matrix of this graph) on a stack

4. if the game contains inconsistencies, pop 1 item from the stack

5. go back to step 1, using the top of the stack as the current state

This will make sure that the generated map is solvable every time the map is spawned.

Solving the map

Whenever the user finds a map to be generated too hard for them, they can simply reset the map by clicking the Reset button.

The map will be solvable by connecting bridges to the Nodes amount e.g. Node (1) needs connection bridges, Node (4) need 4 connection bridges. When ever the user solves the puzzle, they will be prompt with an input box asking them for their nickname or name to place them into the leader board. I will use the AWS cloud to store the scores there is a 12-month free trail, so it is ideal for my project.

Leader board

The leader board will be displayed to the user as an alert box and will present them with their place on the leader board and the time it took them to complete the puzzle.

Deliverables

The project will be user friendly and it will deliver the great time killer for android users. The app will be hosted on play store and available to every one with a android mobile device and it will be free to download.

Languages

My application will be 100% written in java and I am using android studio as my IDE because it's the native android IDE and it supports everything that I need for my project.

The reason I have chosen to program my project in java and not kotlin language is because I have no experience with kotlin and it would take more time to learn kotlin rather than just program in java where I have experience

User Interaction

The risk event can be triggered by the system or the user. The system could lose some data upon the user calls e.g. the user chooses a team and there is no player data.

The system could display an incomplete or incorrect data back to user this could happen by the system being overload by the user calls.

A risk event could happen by lack of experience with specific libraries, software or programming Languages.

Time

Risk of completing the deadline for the project.

The time for the project is a risk upon not finishing the projects our data project in time can lead to wrong data being presented to user like a map cannot be solvable.

The lack of available time to work on this project can lead to risk like falling behind with the deadline not finishing the project in time bad documentation and even bad programming or implementation of the project.

Experience

This is my first time implementing a big project like this, it is also my second time using android studio.

To get the knowledge, I need to implement this project is Time to learn all the technologies and programming languages which this can lead to Time Risks.

Non-Functional Requirements

Functionality

An internet connection will be required to use the application as all user's data will be stored in the cloud.

Usability

A user is not required to login to the game. Users should be informed about the rules of the game before they start to play the game. Users should be have access to the data they provided (their nickname for scoreboard), and be able to view their current position in the scoreboard.

Reliability

The maps must be checked for solvability before the user is presented with a game state.

Performance

The map generation should not take longer than 2 seconds.

Project Plan

One of the most important points with working on an application is project planning. This year's projects were allocated on the 9th of October and the submission date is the 6th of April, that is a total of 26 weeks to complete the project. The time frame will broke up with three Iterations, with each iteration lasting approximately 7 to 8 weeks. The different iterations will focus on different aspects of the project, which include research of existing similar applications, different technologies that will be used, writing documentation of the project and coding.

Iteration 1

This is the first iteration and the beginning of the project. This iteration began on the 27th of November and ended on the 8th of January. In this Iteration I completed my research document that in return gave me a better understanding on the similar applications that were on the internet and also gave me a clear picture of what technologies and programming languages that I was going to use for the project. Knowing that my application is going to be a native based application I found that I will use android studio with java I also worked with how would the algorithm work for my application and I got to know better the logic behind the puzzle. This includes how the islands are connected and that there is only one possible outcome to complete the stage no matter from which

node the user would start. At this stage, I created a very basic grid for testing purposes to make sure that the map is being generated and that all the nodes are in the positions I want them to be in. At the end of this iteration I found that I was ahead of myself and going at a steady progress with having my first document completed and a start of my application. I found that I had spent more time with coding, that doing more research for the Research Document, so it returns it might lead to more work at a late stage updating the document.

Iteration 2

This Iteration began 15th of January and ended on the 19th of February. In this Iteration I had the fingerprint for connecting the bridges completed. I also completed the main functionality of the application by creating a hardcode array of that would represent a solvable map. With this I also have completed other functionalities working that include, users could restart the same state, added a timer for the user playing duration. At the end of the iteration I felt that I was very behind schedule, I spent a lot of time trying to get the fingerprint to draw lines between the islands.

Iteration 3

This Iteration began on the 26th of February and is the final iteration of the project. With iteration not being as productive as I hoped for, it left me in a very pressured situation to get the rest of the work completed. In this integration I have created a 2-dimensional array that generate the map at random, however this is working I ran out time to complete an algorithm that will check if the map is solvable, the algorithm is there but it has few bugs that unfortunately I hadn't had time to fix. I also need to go update any of the previous documentation that I completed in the previous iterations and complete the final report. The final part of this iteration would be testing.

References

In-text: (Anon, 2018)

Your Bibliography: Anon, (2018). [online] Available at: https://www.quora.com/What-is-the-difference-between-a-cloud-and-a-database [Accessed 17 Apr. 2018].