# Game Design Documentation

## Game Description

NetWalk is a 2D puzzle game for the Android where the object is to connect every terminal to the main server. In this version, every terminal and piece of cable must be connected to the main server.

The program will generate a randomised version of Prim's Algorithm to find a minimum spanning tree. A spanning tree of an edge-weighted graph is a spanning tree whose weight is no larger than the weight of any other spanning tree.

The maze will be automatically generated before the game starts by rotating the elements. The players will be able to rotate each tile clockwise or anticlockwise using touch inputs; I could do this by giving clockwise and anticlockwise rotations different touch inputs. Clockwise rotations could be a simple tap whereas anticlockwise rotations could be a hold input.

## Design Goals

The game aims to achieve the following targets:

1. Contain no dead elements such as unused buttons and menus
2. Integrate the algorithm to generate a random maze every time.
3. Gameplay to run smoothly, allowing the user to be able to rotate the cells without any faults occurring.
4. Game should be able to know when the player has solved the puzzle.
5. Game to contain multiple difficulties.
6. Displaying a high-score in the game

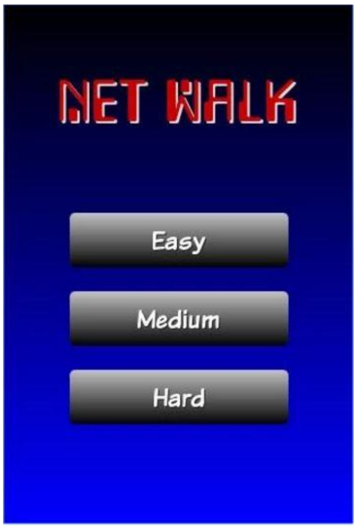
I want the interface of the game to be simple and compelling for players of all ages. I also want the game to be accessible so that it caters for gamers with disabilities and other impairments. We can do that by making the aesthetic of each icon easy to differentiate between one another instead of relying on colour.

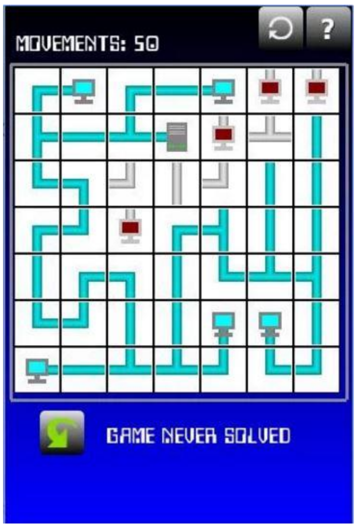
## Influences & Sources

We have researched multiple similar Android games extensively in order to get a better idea of the right look and feel for the game, as well as the right features to include. We paid particular attention to the three Android games; Net Walk by Arousa Games, Scrambled Net by Silent Services UG and Network Puzzle by Android-AZ.

### Net Walk

Net Walk, created by Arousa Games has a very simple design as the main menu for this program is the same as the difficulty section.





The design of each tile is very easy to read and understand apart from the logo of the game.

The program also contains a well written tutorial of the game for newcomers in the "How to Play" section, this is a good idea to help players that don't know the rules of the game. The "How to Play" section could improve by making it multiple paragraphs instead on one giant piece of text.

The game contains three difficulty levels which are:

- Beginner
- Medium
- Hard

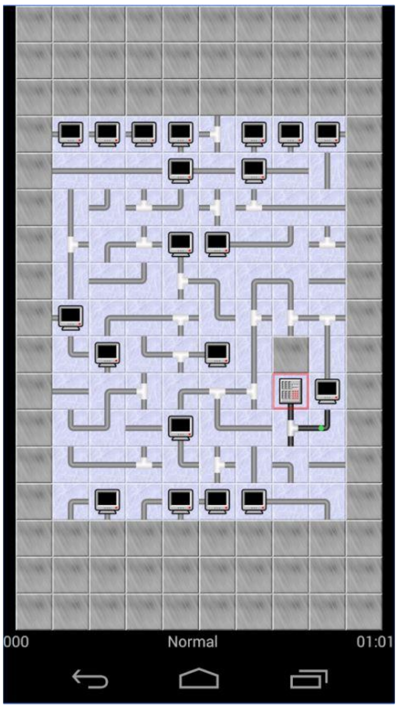
Different difficulties correspond to different size mazes, with Beginner being 5x5, Medium being 7x7 and Expert being 9x9 tiles. On the expert level, connections can wrap from left to right and top to bottom to make it more difficult for the players. Players can rotate each tile clockwise and anticlockwise using the touch input on the Android phone.

I like that they provided the users a "How to Play" button at the top right of the screen during the game at all times because it can be very useful for newcomers.

I also think that the size of the buttons is well placed and sized. However, I think that the background on the program can be improved by using brighter colours

I believe the design is very simple, but effective. However I feel that the design could have improved because the game isn't very eye-catching and may not attract a lot of new players.

## Scrambled Net

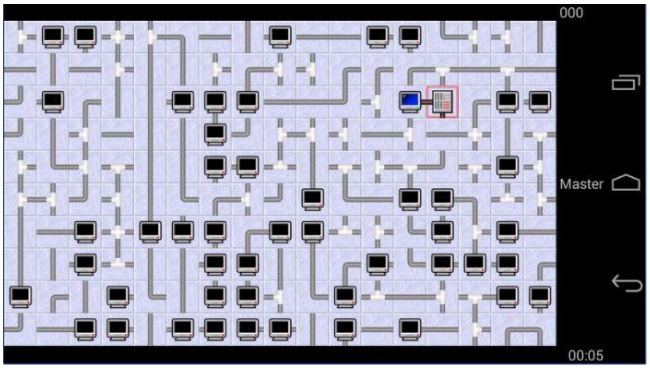


In this version that was created by Silent Services OG, the design for the game is less minimalistic as the previous version. The design of the tiles has a similar aesthetic as such games as Minesweeper and Bomberman.

This game contains more options for the players compared to Arousa Games' version, additional options such as being able to undo a move if the player made a mistake or to solve the game if the player is unable to solve the puzzle, both with a click of a button. I think that those features are very useful and it makes it easier for newcomers to learn and get better at the game.

The game, like the previous version has a written explanation of how to play the game for newcomers. I believe a written explanation of the game rules should always be in a game such as this.

This game contains more grid sizes for players; this is good because it increases the replay value of the game. The game is also playable in landscape and portrait.

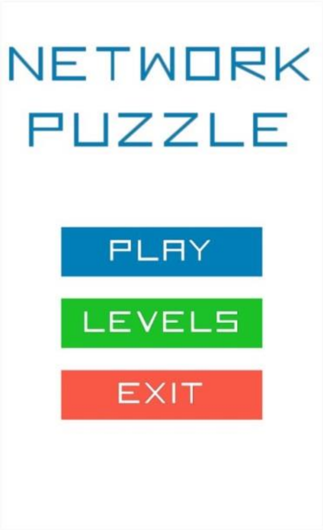


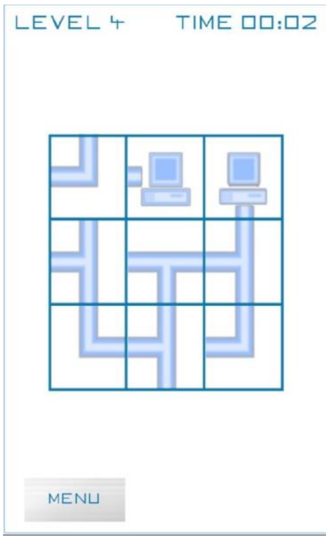
Like Arousa Games' version, the controls use touch inputs on the Android phone to control each tile rotation. The simplicity of the controls is essential to make it easy to use for players of all ages.

I also believe that the additional options that were included in this version is very useful for new players. However, I think that Silent Services OG could have improved on making it more accessible for players that may have disabilities and other impairments

I believe this game is a big improvement to Net Walk in terms of options for the players. I prefer the design of the tiles, terminals and server in Net Walk though because it's more eye-catching.

## Network Puzzle





Finally this version created by Android-AZ is much different to Net Walk and Scrambled Net. This is because Network Puzzle contains pre-created levels instead of using the Prim's Algorithm to generate random puzzles for the players.

Each level is a puzzle that gradually gets more difficult for the player to solve.

The game contains a very simple main menu containing three buttons, play, levels and exit.

There are no other options for the user in this version. I think that including buttons such as "Help" and "Options" would have been useful for players.

When the player starts the game, on the top right will contain the time remaining to solve the puzzle. On the top left of the game, it shows the player what level he is on. On the bottom left, there is a menu button which the player can press if they want to restart or quit the game.

I think that this game has the best design. However, this version is the blandest one out of the three because it has very little options available for the users.

## Conclusion

After analysing three applications that are similar to the "NetWalk" game I am planning to create, I have concluded that making the game look simple and accessible for newcomers as well as providing additional options for to increase the replay value is very important. Android-AZ's version of the game had the best aesthetic; however Silent Services OG's version had the most additional options for users. I plan to make my game look and feel easy and simple like Android-AZ while providing users additional options such providing a help section.

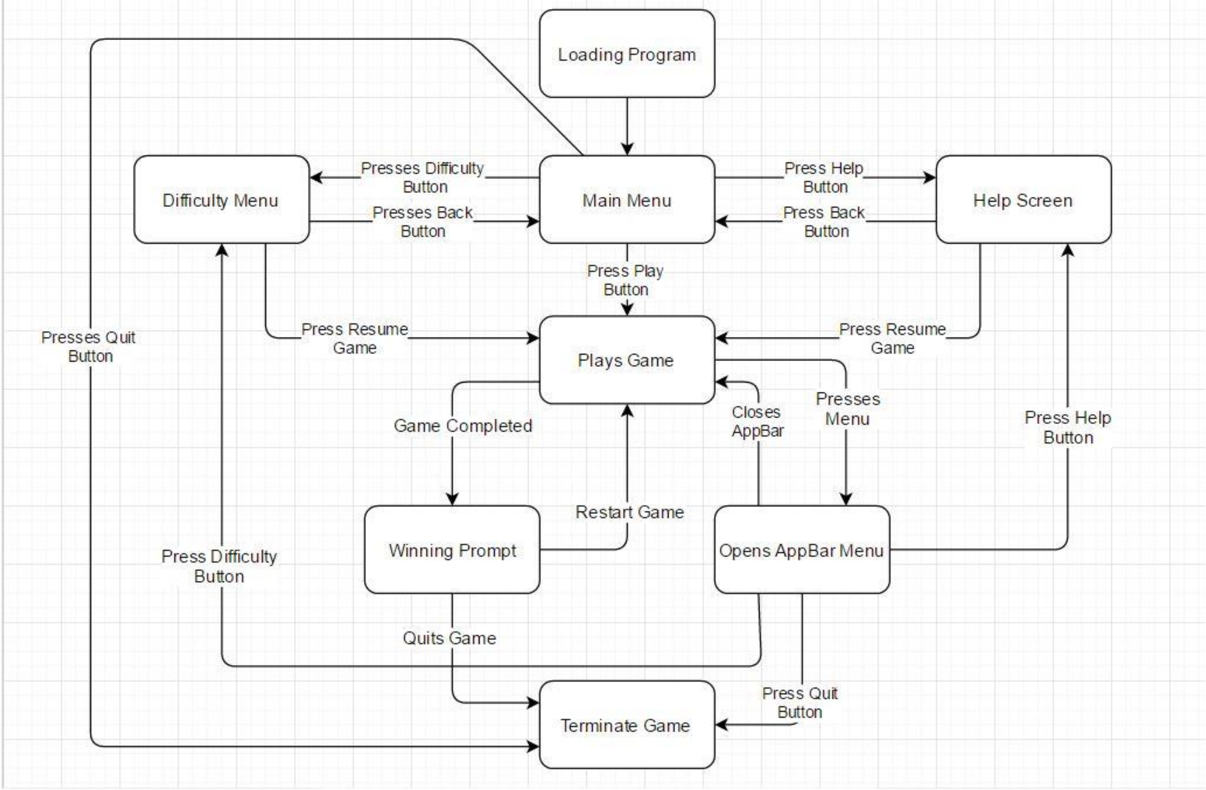
## Target Market

Our ideal user has the following characteristics:

- Male or Female
- Ages 10+
- Has an Android Device
- Likes Puzzle Games
- Plays Mobile Games

# User Interface

## Flowchart



Here we have the flow diagram of the puzzle game. This is used to represent the physical route or flow of the application.

The first thing that occurs when the user runs the application, the main menu will appear and will give you multiple options. The user can open the difficulty menu, open the help menu, play the game or terminate the program.

If the difficulty menu is opened, the user can change the difficulty of the game and go back to the main menu. The user has to go back to the main menu in order to play the game.

Similar to the difficulty menu, if the help screen is opened, the user will have to go back to the main menu in order to play the game. In the help screen, there will have a written description on how to play the game effectively.

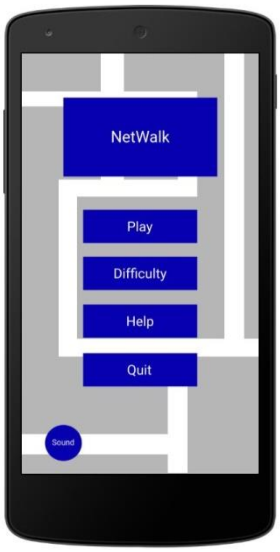
Pressing the “Play” button will start the puzzle game. While playing the game, the user can open the AppBar at the top left of the screen to show additional options. With this AppBar, the user will be able to go straight to the help screen and the difficulty screen. The user can also terminate the program from the AppBar. If the user goes to the help or difficulty screen from the AppBar, they are able to go back straight to the game without losing any progress.

If the user solves the puzzle, a winning prompt will appear. The user has the decision to replay the game, make option changes or terminate the program.

# Mock-ups

## Main Menu Screen

Firstly, we have the Main Menu Screen. This will be the first thing that players will see when they start the application. Here we have the logo of the game at the top with 4 buttons below it.



The main menu screen contains a background image of terminals and servers. I decided to add this instead of making the background plain like in Net Walk because it makes the title screen more exciting for the player.

*Play*: Starts the game. The program will use Prim’s Algorithm to generate a random puzzle for the user. The game is on the difficulty the user chose or Easy by default.

*Difficulty*: Page in which the user can change the difficulty of the game. There are three difficulties which are easy, medium and hard. The harder the difficulty is, the larger the maze will be.

*Help*: Gives the player a written tutorial on how to play the game for newcomers.

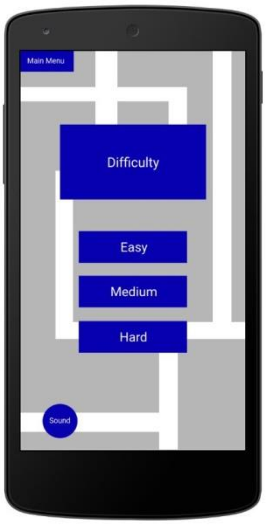
*Quit*: Terminates the program.

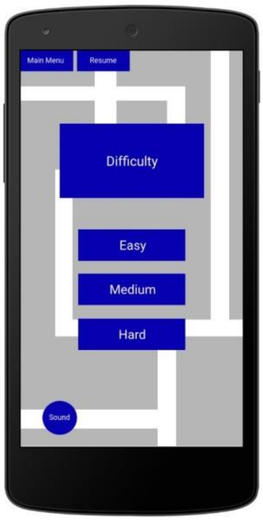
*Sound*: Toggles the background music.

## Difficulty Screen

If the user presses the ‘Difficulty’ button while on the Main Menu, it will take them to the difficulty screen. In this screen, the user can change the difficulty of the game.

The Difficulty Screen will have the same background as the Main Menu Screen. When the user presses any of the three buttons, a notification will appear at the bottom of the screen confirming





it. The user has to return to the main menu in order to play the game. If the user used the AppBar to go to the Difficulty Screen while playing the game, there will have a ‘Resume’ button next to the ‘Main Menu’ button that can return him/her straight back to the game.

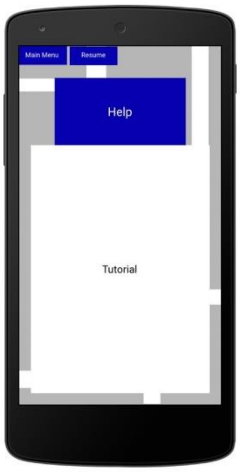
*Easy*: Sets the game difficulty to easy. The maze will be 5x5 tiles.

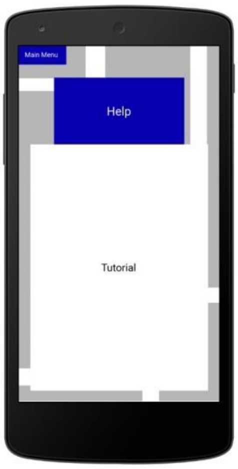
*Medium*: Sets the game difficulty to medium. The maze will be 7x7 tiles.

*Hard*: Sets the game difficulty to hard. The maze will be 9x9 tiles.

*Back*: Takes the user to the Main Menu Screen.

*Resume*: Takes the user back to the game.





## Help Screen

If the user is new to the game, they can choose to click the ‘Help’ button in the Main Menu Section that will take them to the Help Screen.

In this section, it contains a well written description of the rules of the game and how to change the difficulty.

This section, like the Difficulty Section has a Main Menu button at the top left corner of the screen to take them back to the Main Menu Section.

If the user entered this section using the AppBar while playing the game, there will have a ‘Resume’ button to take them right back to the game as quickly as possible.

*Main Menu*: Takes the user to the Main Menu Screen.

*Resume*: Takes the user back to the game.

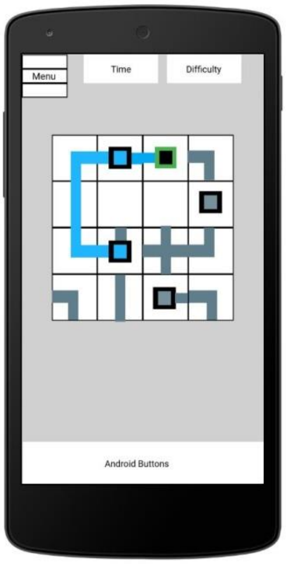
## Game Screen

Here we have the screen which will be displayed when the player starts the game. In the middle of the screen will have the puzzle, the size of the maze will differ depending what difficulty the player chooses.

At the top of the screen will show the user the difficulty he/she is playing on and the time he has taken to complete the puzzle.

Bottom of the screen will have the standard Android buttons which are to go back, leave the application or use the multitasking. The user will need to swipe up from the bottom of the phone first for these Android buttons to appear though.

*Menu*: Opens the AppBar containing Difficulty, Help and Quit.



*Difficulty*: Takes the user to the Difficulty page.

*Help*: Takes the user to the Help page.

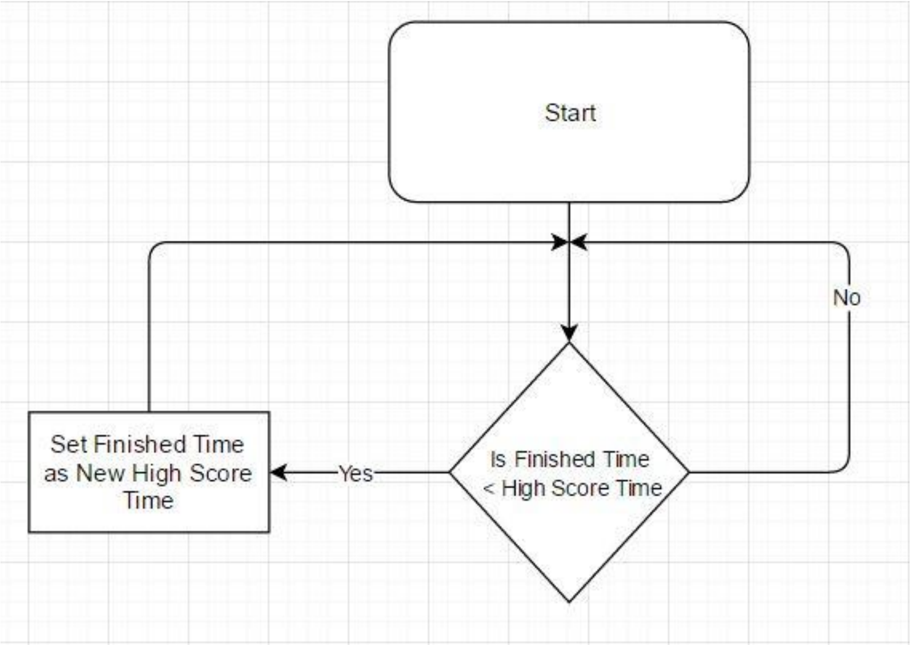
*Quit*: Terminates the program.

# Technical Specifications

## Platform and OS

This game is specifically developed for Android devices, an operating system developed by Google Inc. The game is developed for Android 6.0 or above. The game can easily be ported to iOS and Windows with a few small adjustments.

## High Score


Start
No
Set Finished Time as New High Score Time
Yes
Is Finished Time < High Score Time

Here is a flowchart of the High Score system to explain how it works. If the player completes the puzzle quicker than the current high score timer, then the high score will update.

The player completes the puzzle. If the player didn’t beat the high score time, then the high score doesn’t change and will loop back to the beginning. If the player however did beat the high score time, the time the player got will equal the new high score time and then loops back to the beginning.

## Code Design

In this application, we’re planning to develop 5 main sections, ‘Controller’, ‘Events’, ‘Model’, ‘View’ and ‘Main Game’. Each section is responsible for specific main tasks, it is best to separate them to 5 separate segments because it makes it much easier for other programmers to read the code and make changes if necessary.

The 'controller' class will contain an event queue loop, which waits for the new event in the queue and takes it and starts processing the method. The method should be responsible for processing the event from the queue. The class connects with the other classes, 'Model' and 'View' if necessary.

In the 'Events' section, it should contain the following classes:

- Generate Maze Event
- Rotate Event
- Show Help Event
- Show Win Event

The Event classes are the root class from which all event state objects shall be derived. All Events are constructed with a reference to the object. 'Generate Maze Event' is there to develop the events of the maze when the user starts playing the game. 'Rotate Event' is used to set up the events of the tiles rotating during the game. 'Show Help Event' is there for the user when they need help playing the game or navigating themselves through the application by generating the event for the help section. Finally, if the player completes a puzzle successfully, the 'Win Event' will appear indicating that the user has won.

In the 'Model' section, it should contain the following classes:

- Maze Generator
- Model

The 'Model' classes are the class that displays all the models in the application. This class should have a relationship with Events and View. 'Maze Generator' is the algorithm used to generate a random maze every time a new game has started. 'Model' is used to generate the item models for the game.

In the 'View' section, it should contain the following classes:

- Help
- Menu
- Maze
- View

The 'View' classes are representing a view of the text model, or a piece of the text model. It is this class that is responsible for the look of the component. There are three main sections in the application, Help, Menu and the game. This means that we should have 3 separate classes dedicated to each section. This is important to make it simpler to make changes to the application if necessary.

# Sources

https://play.google.com/store/apps/details?id=com.arousa.games.netwalk&hl=en_GB

https://play.google.com/store/apps/details?id=com.silentservices.netscramble&hl=en_GB

https://play.google.com/store/apps/details?id=games.networkpuzzle&hl=en_GB