Triple Action – Growing & Resizing & Maze

Table of Contents

[Triple Action – 3-Phazed Mobile Game 2](#_Toc95312637)

[1. Core Loop && Progress 2](#_Toc95312638)

[2. Code Structure – UML 3](#_Toc95312639)

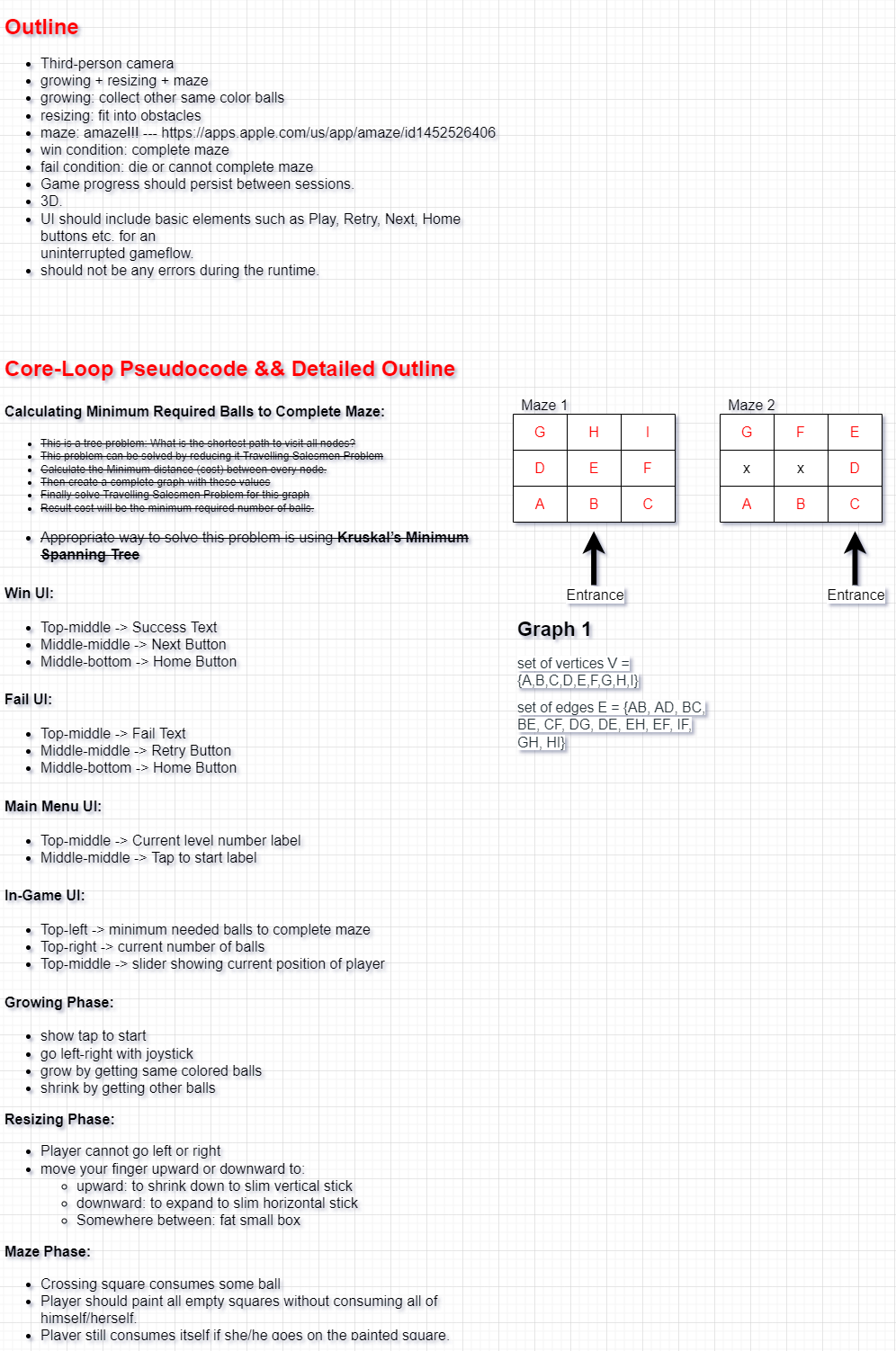
[3. Developer’s Diary 3](#_Toc95312640)

[Initial Maze Approach 5](#_Toc95312641)

[Implementation Resources: 7](#_Toc95312642)

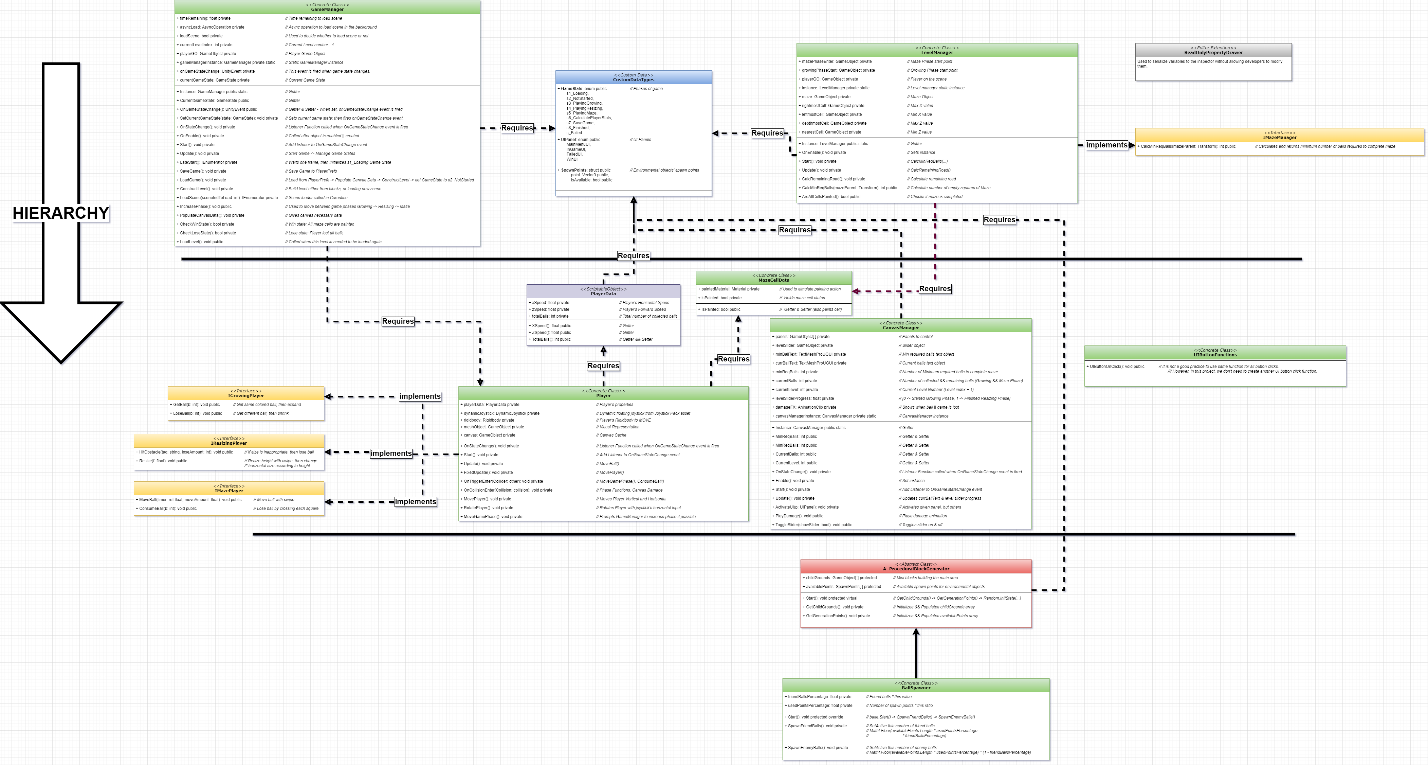
# Triple Action – 3-Phazed Mobile Game

## Core Loop && Progress



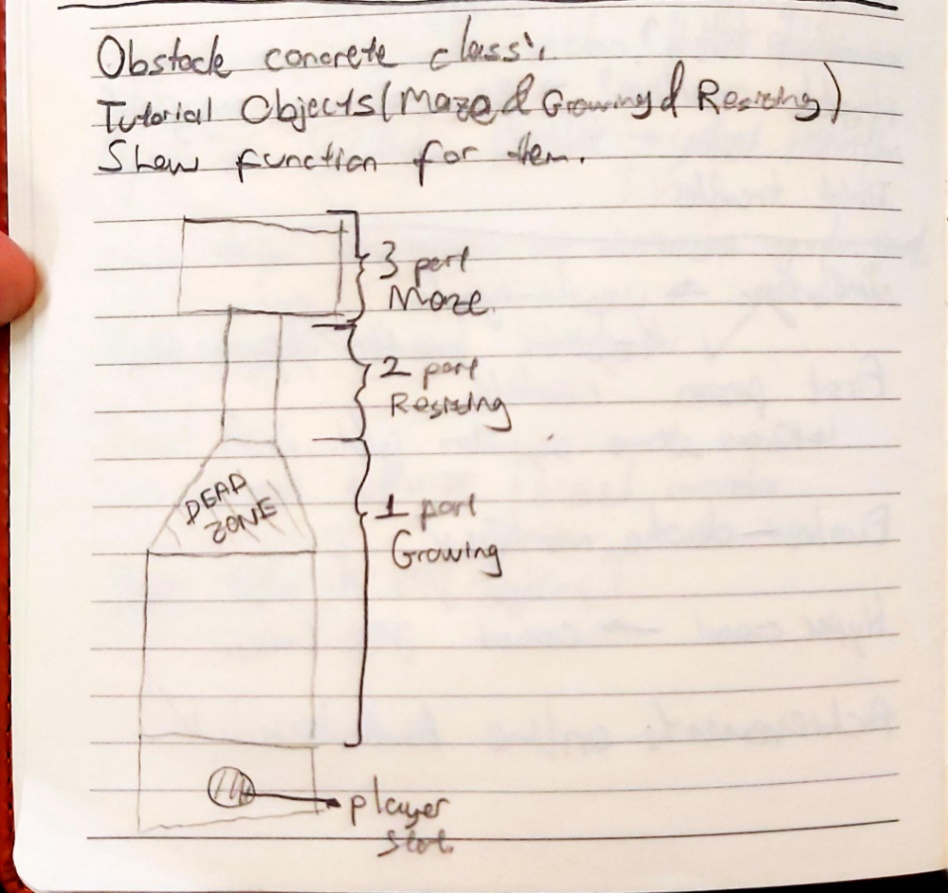
## Code Structure – UML

Link to view easily: [View on Web](https://viewer.diagrams.net/?tags=%7B%7D&highlight=0000ff&layers=1&nav=1&title=RubyGamesGameplayDeveloperTest.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1fBjvGc5j3RzwrOaFrc2A12LD1oJgFPwm%26export%3Ddownload)

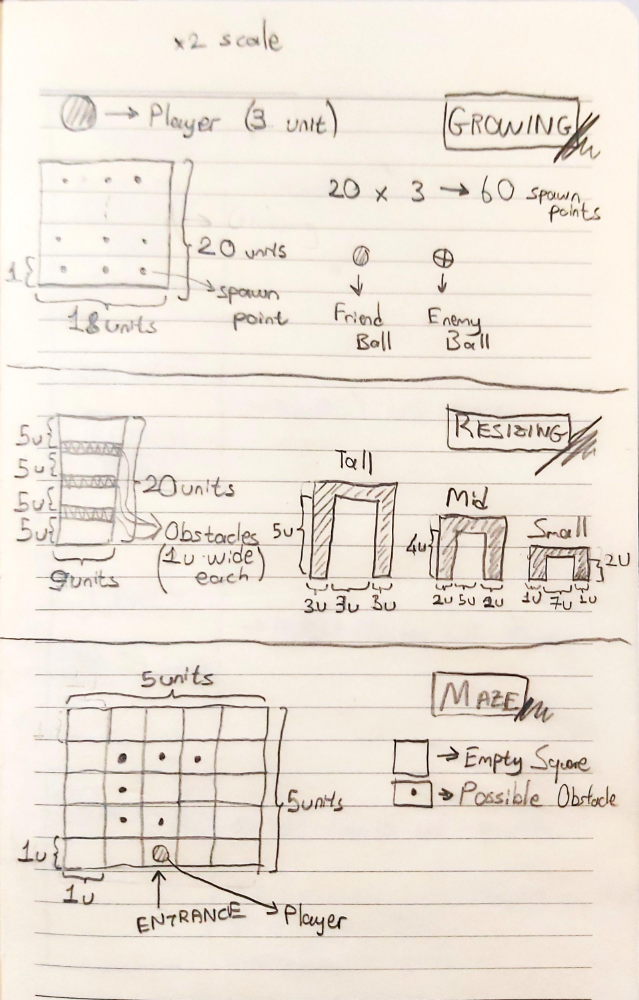


## Developer’s Diary

* First map design was different. Since I have a scale inconsistency between player and scene; I quitted this design:

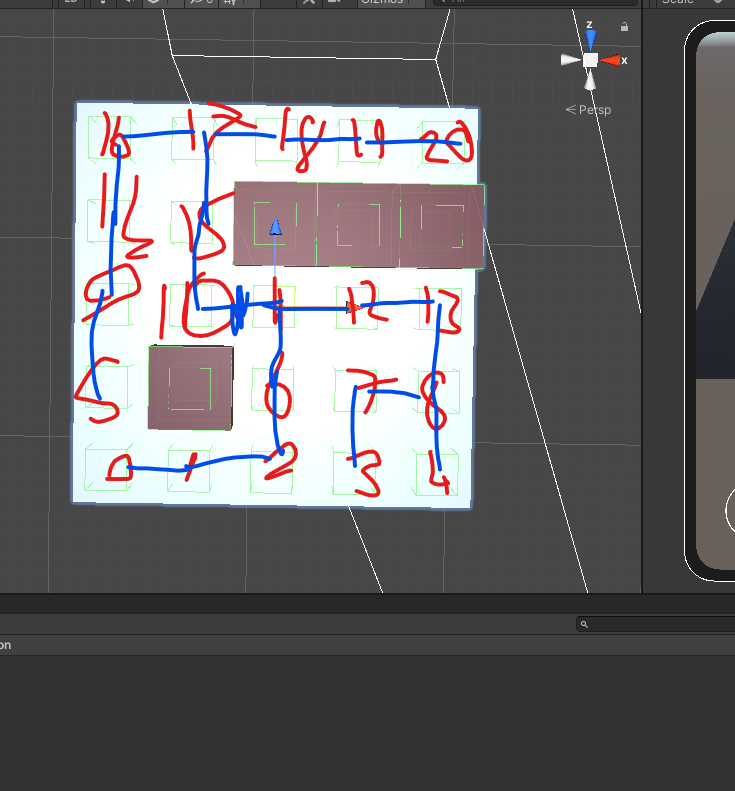


* Scale inconsistency also affected parts of whole level. Phases’ sizes are different from initial design. Following data of Player’s size is pointing wrong value.



### Initial Maze Approach

My initial approach for the Maze Phase was calculating minimum required balls to complete the maze. Ex:



* To solve this, I tried following solutions and encountered following problems:

#### Undirected Weighted Tree – Minimum Spanning Tree in A Graph

* I managed to find an MST in a graph. Maze cells construct a graph having edges between them. But the problem is, it is calculating the minimum cost to cover all nodes of the tree. All connections are evaluated once: which means we don’t traverse graph, rather than summing costs of edges of graph.

A picture containing text, document, receipt

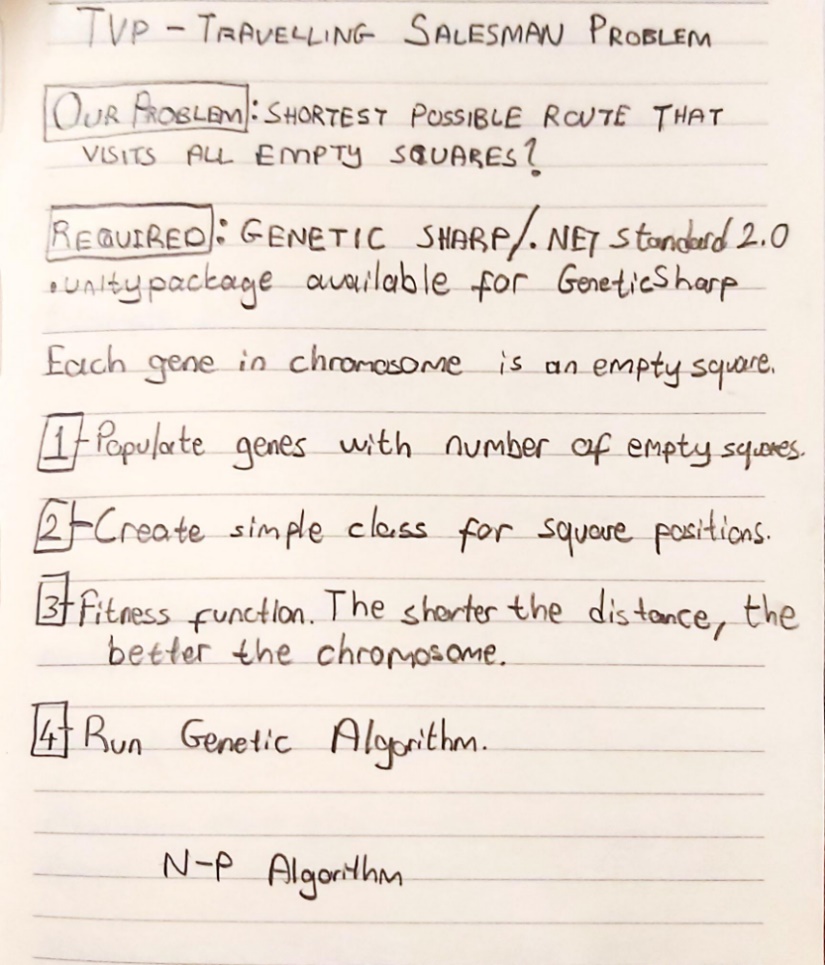
Description automatically generated

#### Travelling Salesman Problem – Pure Implementation

* This was my most hopeless try 😊 Since this is an NP-Hard problem, I couldn’t manage to calculate the maze having 20+ cells.

#### Travelling Salesman Problem – Genetic Algorithm Approach

* NP-hard problems are situations in which there is no known algorithm to solve this problem in O(n) time. These problems need to be solved approximately with the most minimum error. One way to solve TSP is using Genetic Algorithm.
* I also managed to use this algorithm. However, the main problem is that it was calculating direct distance between two points in space. Which means that it considers every distance between two points (points are maze cells) a reachable edge. This leads another failure on my side 😊



### Implementation Resources:

These implementations can be found in this zip file. Since I altered CustomDataTypes, they need some modifications to work again.

Graphical user interface, application

Description automatically generated