Team 82 – Seng 201 Project Report

Ngoc Le – ID: 46909266

Anh Le

All the class files are split into three main categories. GUI, models, and services. To help with designing the code structure, we started by creating a class diagram. For the services class, we have EnvironmentManager, TowerService, ShopService and InventoryService. For models we created Tower, Cart, and PurchasableItems class. Our preliminary diagram is shown below.

Our EnvironmentManager class holds all the necessary game variables. Examples include the current round number, game and round difficulty, and the amount of points the player has earned. We wanted to track all these important variables inside one class to keep them in one place. The TowerService class interfaces with the InventoryController GUI which controls the inventory screen where you can apply upgrades to your towers. TowerService contains helper methods to upgrade a tower’s stats, like decreasing the recovery time, or upgrading the resource amount that is shot into the cart each time.

The InventoryService class contains methods and variables related to the shop and its GUI. This class interfaces with its corresponding GUI class, InventoryController. It contains methods including setPlayerCoins, sellTower, and upgradeTower.

Initially, we created Round Manager as

• The structure of your application and any design decisions you had to

make. We are particularly interested in communication between classes

and how interfaces and/or inheritance were used. You might want to

reference your UML class diagram.

• An explanation of unit test coverage and why you got a high/low percent-

age coverage.

Include on the second page:

• Your thoughts and feedback on the project.

• A brief retrospective of what went well, what did not go well, and what

improvements you could make for your next project.

• The effort spent (in hours) on the project per student.

• A statement of agreed percentage contribution from both partner