CMPT 276 Assignment 3 Report

Kyle Deliyannides / Bassim Ghasemzadeh
301459316 / 301474075
kkd16@sfu.ca / bga32@sfu.ca

Multithread Cop Car Initialization

When pressing the “start” button on the title screen, the program would freeze for two or three seconds before launching the game. This was indicative of a refactor being needed to improve performance. So, multithreading the cops original breadth-first-search was in order. A new class was created called CopCreatorThread which when instantiated, creates a new cop object and runs the original breadth first search function in its own thread. You can then call getCop() to get the cop that was created. This significantly improved load times as all the cops do a search on their own thread instead of the main thread. The appropriate commits are titled “create cop thread class”, “make cop not public”, and “rename cop thread to cop creator thread”.

High coupling while creating game objects

Creation of the Cop, FoodTruck, and Grid objects were done by the GamePanel object. This did not make sense because the GamePanel object should only be the JPanel and not be responsible for the creation of other objects. Hence, the responsibility should be left to the Game class instead. This was refactored by moving the Grid’s initialization into its own constructor, instead of having the game panel populate it with elements. Then, followed up by moving all the Cop, FoodTruck, and grid creation lines of code into the Game class’s constructor. Later on in a different refactor, the cop and foodtruck creation was even further refactor and encapsulated, explained in an entry below. The relevant commits are “move grid initialization to its own constructor” and “move all object creation from gamepanel to game”.

Game class had too much responsibility

It was identified that the game class was in charge of creation and setup of everything related to the JFrame as well as the actual game logic itself. This was not good as it was asking one class to do the work of two and led to way too long of a class file. So, there was a need to split this up into a GameFrame class that extends JFrame and have the Game class just create and store an instance of GameFrame. The refactor involved moving all related JFrame components into the new GameFrame class and then having the GameFrame class create all the different panels. Additionally, the GameFrame class had to have the ability to return a Scoreboard class to be sent to the GameOver frames when they are created. Additionally, this allowed for the keyboard handler to be attached to the JFrame, instead of the GamePanel JPanel which made it possible to remove lots of redundant code which was used to shift the window’s focus for handling key events. After fixing this redundancy, the key release function could also be removed from the key handler. The relevant commits are titled “seperate game into game and gameframe”, “remove redundant keyboard handler passing around” and “remove key released function”.

Poor encapsulation in Game Class

Poor encapsulation was identified within the Game class that initialized instances of the food truck and cop classes. Each instance of a cop had been hard coded at the top of the Game class which seemed inappropriate since the game class contained logic that managed the game state. Hence, to solve this issue a VehicleSpawner class had been made to better organize the code. The VehicleSpawner class contains a constructor which does the initialization of the vehicle objects so that the vehicles can be gathered from the Game class through the getFoodtruck() and getCops methods. getFoodTruck() returns the instance of the foodTruck and getCops returns an arrayList containing all cops that will be used for the game. By encapsulating the spawning in its own class, the game class will remain the same no matter how many vehicles get instantiated in the VehicleSpawner. Initially only the cops were going to be encapsulated before the realization that the foodTruck should also be included. Hence why the branch for these changes is called "copSpawner". Another slight refactor was later made to simplify the constructor of the VehicleSpawner in the commit "simplified vehicleSpawner constructor".

Observer design pattern for truck tracking

Prior to the code review, each Cop instance had been running a breadth-first search upon initialization and at every movement of the foodTruck. The algorithm would then create a list of directions for the foodTruck to follow. Although functional, this method had been very inefficient as the player moved further from the cops at every move and the breadth first search procedurally became more computationally heavy. Through some testing we noticed that only one breadth first search would have to be done for each Cop once initialized and that each player movement could be appended to each cop's list of directions. This led to the deletion of the trackTruck method that was responsible for running a breadth-first search at every player movement. The solution prompted us to use an observer pattern as we created an attach(cop) and sendUpdate(Direction) to the foodTruck as well as an addDirection(Direction) method for the Cop class. These changes can be found in the commit "Observer pattern for food truck tracking".

Convert custom linked list into hashmap implementation

A derivation of a LinkedList was used to store checked locations during the Cop car's original pathfinding algorithm. This was called "PositionList", and used a for loop to iterate through and check each entry of the list. This list stored instances of Position objects which are custom objects that store a row, column, and previous location for pathfinding use. This runs in O(n) time, and since the list can get pretty big it was in need of a speedup. So, the PositionList was changed into PositionMap, which uses a hashmap to store the position, alongside a key. The keys are generated by a new function in the Position class which returns a string of "row_col". So now calling the contains function runs in O(1) time instead, which speeds up the initial path find of the cop cars, and therefore speeds up game starting. The relevant commit is titled "update position list to use a hashmap".