Food Truck Frenzy - A Deliciously Addictive 2D Maze Adventure!

Welcome to "Food Truck Frenzy," an exciting 2D maze adventure where you become a food truck chef on a mission to collect ingredients and recipes while avoiding obstacles and outsmarting the police! Test your skills as you navigate a maze of houses and roads to reach the endpoint after collecting all the ingredients and create the ultimate food truck experience. Will you rise to the challenge and claim victory in this deliciously addictive game?

How to Play:

1. Navigate your food truck through a 2D maze filled with houses and roads by using the WASD keys.
2. Collect ingredients scattered around the maze. Each ingredient is worth 100 points.
3. Optional: Look out for bonus recipes which are worth 200 points. While not necessary to win the game, they can help you rack up an impressive high score!
4. Beware of speed traps and potholes as you make your way through the maze. Getting caught in a speed trap will cost you 196 points, while hitting a pothole will result in a loss of 286 points.
5. Keep an eye on the clock - you are timed, and the faster you complete the game, the better your bragging rights!
6. Watch out for the police! They are following your every move and if they catch up to you, it's game over.
7. Once you have collected all the ingredients, make your way to the endpoint in the bottom right corner of the maze to win.

Do you have what it takes to become the ultimate food truck chef in "Food Truck Frenzy"? Put your skills to the test and see if you can emerge victorious in this thrilling 2D maze adventure! Happy cooking and good luck!

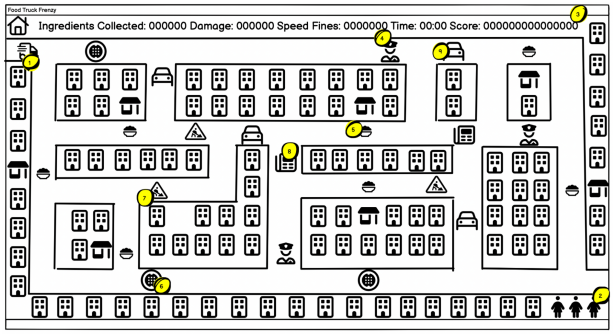
Video Demo: https://youtu.be/VXR-AHahh6Y

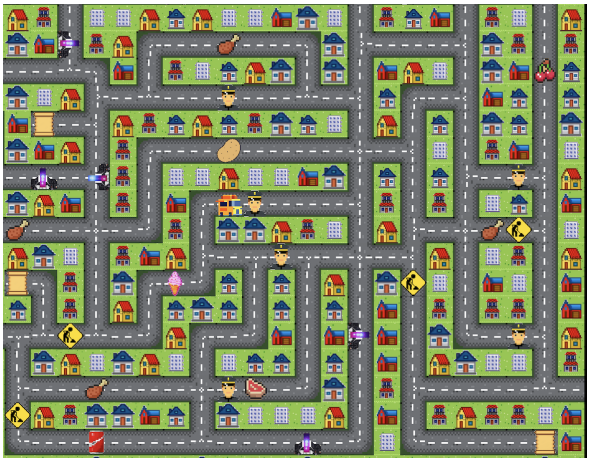
CMPT 276 Team 5: Project Phase 4 Report

Kyle Deliyannides 301459316
Patrick Shaw 301537454
Chris Pinca 301396939
Bassim Ghasemzadeh 301474075

The final version of the game Food Truck Frenzy has remained very similar to the first plan of the game. Initially the idea was to have the player drive a food truck and collect ingredients and recipes before going to drop off meals to various locations. This had to be done without getting caught by the cops that would be chasing the food Truck. However, we soon realized that dropping off meals might overcomplicate the game without bringing any additional value to the gameplay. Hence, the dropoff idea was scrapped in the design face.

With some inspiration from the game Flappy Bird, we wanted our game to be simple yet engaging so that anyone can give it a try and have fun. Throughout the implementation of the game, a valuable lesson learned is that more is not always better. We often found ourselves overcomplicating our approach to certain features that would soon be removed. For instance, when developing the cops chasing feature, the initial idea was that as the player progressed, more cops would appear, increasing the difficulty gradually. The problem with this is that the balance we were seeking could not be accomplished with this approach. If the cops were to spawn at the same set of locations at a certain time increment, the player would know to avoid cops spawning next round and the game would be too easy. If however, there was a type of random spawn system, the difficulty would vary each round depending on how far the cops would spawn from the player. Through trial and error we learned that spawning all cops at a fixed position in the beginning offered a gameplay experience that aligned the most with our goals. This allows the game to be played competitively as there is no sense of luck impacting the level of difficulty for each game. Moreover, while designing our initial UML diagram for our project we had no clue about the scale of the task we were put up to. Our initial UML diagram contained a monolithic GameState class which needed to be split up. Additionally, we didn't know how game loops worked originally, so we had boolean values depending on whether the game was running or not. This proved to be an ineffective way of doing this as we can just create new game timer objects instead, which have all the functionality self contained without the need for another class "supervising" the game loop. Because of this overcomplication of our original design, we learned a valuable lesson in planning ahead and learning about what technologies we are using before diving head first into designs.





As shown in the images above the final version resembles the initial design very much. The left image is a representation of the game layout we had in mind during planning and the image on the right is a screenshot from the actual game. The strong resemblance to the design leads onto another lesson learned during development which is to stick to the plan. While developing certain parts of the game the idea to modify and add certain aspects proved to be counterproductive in the long run. As shown in the mockup above on the left, our design involved a grid layout that restricts movement to be vertical or horizontal. When implementing the movement mechanisms we thought it would be cool to add diagonal movements in an effort to make the game appear more “smooth” or flexible. However, once the grid layout was added, this new feature caused a problem where the player often skipped a cell on the grid when moving diagonally on turns. After several attempts in modifying existing functionality to account for the new movement system, we realized that we actually preferred the gameplay with strict vertical or horizontal movements since diagonal movement did not make much sense for our game. From this moment we learned from this lesson and continued the rest of the development sticking to the plan as much as possible. As for phase 4, the only code changes we made were changing the controls from the arrow keys to the WASD keys as we feel this is more familiar to everyone who plays PC games.