Untitled Farm Game is a 2D arcade-style maze game set in the distant, dystopian lands of Farming Town, USA. You play as a freedom-loving chicken seeking life, liberty, and the opportunity not to be turned into a four-piece combo. However, ruthless farmers aim to hunt you down and sell your delicious carcass to Big Chicken. To escape, you must ransack human settlements on six playable levels, ranging from large-scale, commercial farms to the bustling, industrial suburbs of the chicken-eaters, to collect your keys to freedom. Avoid farmers and their well-placed traps; otherwise, you'll be seeing the inside of a fryer tonight.

Our final product was similar to the original concept, with one exception. Initially, the playable character was going to be a goose, but we changed it to a chicken instead to more closely match our game's farm aesthetic. Other than that, our game stayed relatively faithful to our prototypes. The general design of the user interface from our mock-ups was replicated almost precisely in our final game, the only differences being cosmetic ones such as higher quality, animated sprites, and assets. Initial use cases also remained valid in our finished product.

As for the code and internal mechanics of the game, we changed quite a bit. We used a structure similar to our original UML diagram with a few changes to make the development process more efficient. The first significant change we made was how we structured our entities. The UML diagram had the entity class and two subclasses for inanimate and animate entities. In our game, we changed this to have inanimate entities as objects and animate entities as the entity class with subclasses for player and farmer. Also, for some reason, we had entity position as a separate class which was changed in our actual game to be two attributes in the entity class. We also had a major change with our collision detection. Our original UML diagram had a separate class for CollisionType, which felt unnecessary. We changed it to a CollisionChecker class and individually checked for each type of collision when appropriate. We also added an extra ability for the chicken (player). This ability freezes the farmers for 1-5 seconds. We added this because we felt that it was possible to run into some frustrating scenarios where you had no way to escape. Adding this ability overall improved the gameplay experience and made our game more enjoyable to play. Most other aspects of the code were quite similar to the UML diagram, with minor changes such as different naming schemes.

During the development process, we learned a lot of lessons that will prove useful in further development projects. The first thing we learned was the importance of using the software development methods we learned about in class. If we had made a lot of our classes singleton from the start, we would have had a much easier time making our code clean and easy to read. We attempted to fix some of it, but some of the code is too coupled for it to be worth rewriting currently. In the future, we will have a better idea of how to structure our project from the start. We also learned the importance of working as a team. This project helped us get experience with working around people's schedules, proper communication, and assigning work to different people. All of these were very useful. By the end of the development process, we all felt much more confident working in a team environment and communicating with each other appropriately. The third important lesson we learned was the importance of testing. Without proper testing, many bugs would not be caught before the project's release. Our test helped us uncover multiple bugs that were difficult to spot previously. This made our game more polished and gave us a lot more experience writing proper tests.

For our tutorial/demo we made a video trailer for our game! The link to our video is here:

https://youtu.be/4MRnaP6zbAo