Simplicity
Ryan Siu, Eric Zhang, Tiffany Moi
Period 5
Airline Simulator

Description

Our airline simulator emulates the management of an airline company. Starting with one flight route between two cities and one plane, the user can expand their company to more cities, more flight routes, and more and better airplanes. The goal of this game is to earn as much money as possible; therefore, it's up to the user to choose flight routes for their airplanes to maximize profits (longer routes mean more money, but with the cost of more time spent and transfers). This is where algorithms come in—using path finding (minimizing distance) based on a priority queue of cities (sorted by profitability, accessibility, etc.), the best possible routes are suggested to make it easier for the player. This game is meant to be endless; even when the user has unlocked/purchased all the available planes and flight routes, they can continue to earn money.

Graphics-wise, the user will interact with different GUIs for different tasks. There will be a GUI for each game function, such as buying new airplanes, choosing a flight route, etc. This game should be easy-to-learn and easy-to-play, and simple GUIs can help us achieve that.

Diagram for UI

MVP

- Basic GUI for user interaction
    - Player can select destinations with the distances from a list (rather than a map)
- Path finder, which minimizes the distance traveled and tries to hit cities with the greatest population or profitability
- Basic gameplay, some game progression
    - User starts with minimum resources
    - User can buy different airplanes, go to different cities upon reaching certain milestones

Stretch

- Tutorial for user
- Real life events affecting gameplay/profits
- More game progression
- More airplanes, cities
- Achievements

UX

- GUI to interact with game
- Player can construct the routes for the airline themselves
- Player buys better airplanes to access more cities (game progression)

Class Material Used

- Priority Queues based on population size for cities
- Inheritance for cities and airplanes, which will have different values for instance variables
- Sorting algorithms used for path finding
- Inheritance for GUI elements