Team No. 16

Team Members: Brian Jones, Finn Dobbs, Jacob Swearingen, Joshua Bull, Ryan Boyce

Project Name: Resource Distribution Simulator

Project Synopsis:

- Simulating maximum efficiency in the consumption of resources, reducing the necessary resources needed to maintain a modern society.

Project Description:

- This project aims to anticipate the increases in available surveillance and computer capacity in society to provide better, more efficient ways of living. This project is an introduction to an economic system that predicts individuals' actions and manipulates autonomous vehicles to provide resources at certain locations at certain times. As climate change and other factors which lead to economic uncertainty become more prevalent, information technology can serve as a means of organization and distribution to maintain stability. The end result of this project will be an AI program that can deal with uncertainty in the actions of individuals and make decisions in order to provide for them. Our project will involve us trying to create a central AI that can handle and use this information to optimize the resources of a given society of simulated individuals and "fulfillment centers", a metonym for the full factory-sale-delivery industry. We are aiming to use OpenGL to display in real time resources being delivered by drones to and from individuals.

Project Milestones:

Milestone	**Estimated Completion Date**
Design project architecture	10/12/20
Build object hierarchy	11/09/20
Build graphics interface	02/15/21
Simulate individuals' needs	03/29/21
Simulate all-knowing computer	04/26/21
Simulate drone delivery system	05/10/21

GANTT CHART - Resource Management Simulator

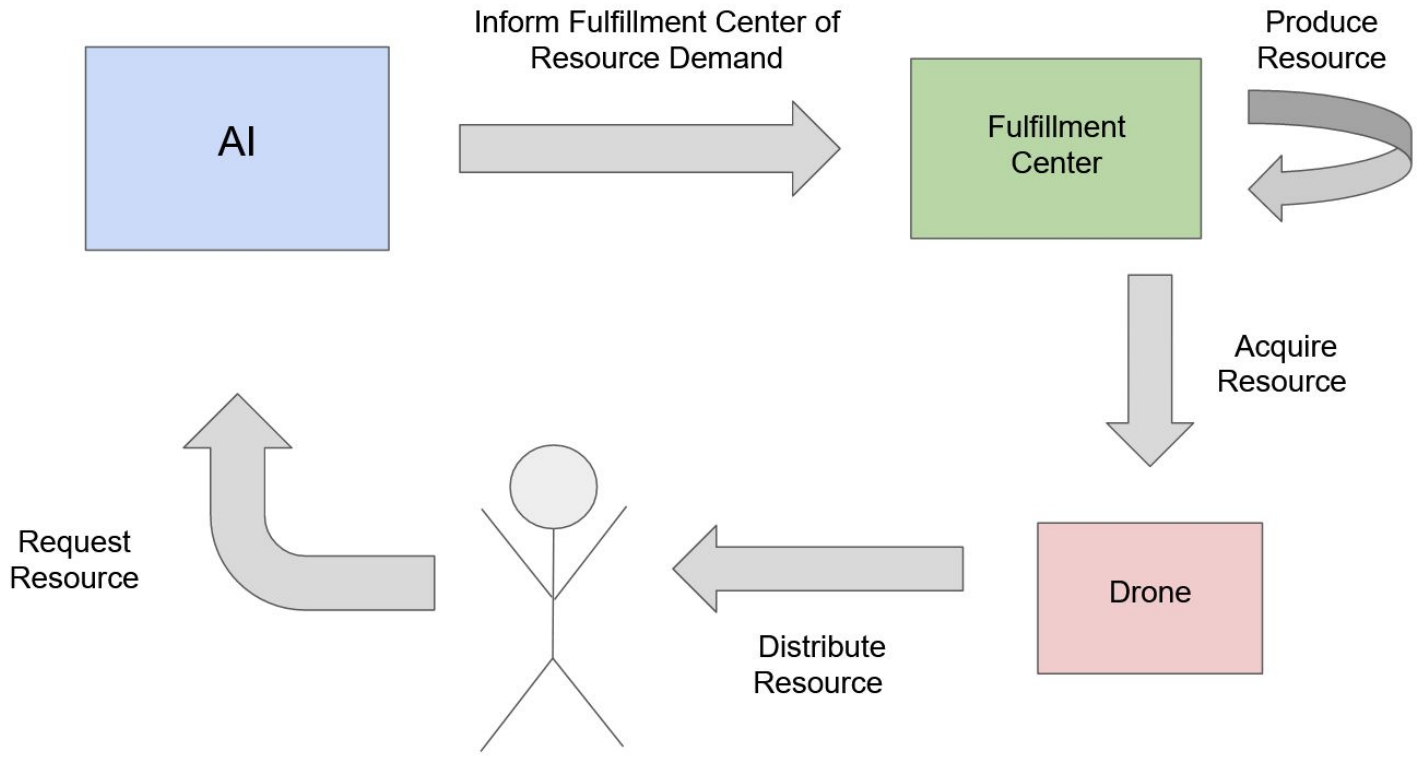
TASK TITLE	PROGRESS	DUE DATE	1/25/21	2/1/21	2/8/21	2/15/21	2/22/21	3/1/21	3/8/20	3/15/21	3/22/21	3/29/21	4/5/21	4/12/21	4/19/21	4/26/21	5/3/21	5/10/21
Initial Architecture	100%	10/19/20																
Inital language/library discussion	100%																	
Skeleton layout	100%																	
Module creation	100%																	
Resource Hierarchy	100%	11/9/20																
Build resource classes	100%																	
Building Implementation	80%	02/15/20																
Building Class	80%																	
Houses	80%																	
Fullfillment Centers	80%																	
Offices	80%																	
Graphics Interface	40%	05/10/21																
Skeleton layout with GLFW library	80%																	
Basic map visualization	60%																	
Map zoom/panning	100%																	
Object selection and display	20%																	
Drone/person movement	0%																	
Map textures	0%																	
Cross platform functionality	20%																	
Simulate All-Knowing Computer	34%	03/22/21																
ID system	80%																	
Simulate passage of time	10%																	
Simulate individuals	60%																	
Task and agenda system	20%																	
Implement prediction algorithms	0%																	
Simulate Drone Delivery System	24%	04/19/21																
Resource production	0%																	
Drone queue system	0%																	
Passenger drones	60%																	
Cargo drones	60%																	
Implement path-finding	0%																	

Project Budget:

- Estimated cost: $0.00
- Special Training:
    - OpenGl needed for graphics integration
    - Machine Learning

Final Project Design:

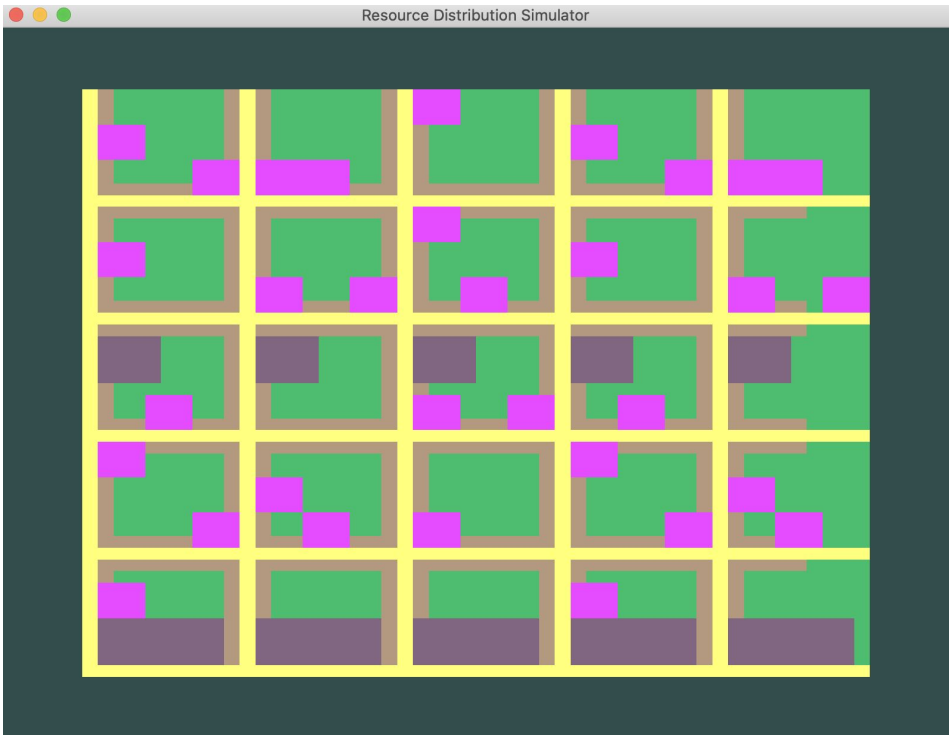
This project is being programmed in C++. We chose C++ given its speed and flexibility when dealing with large programs. We are using Github to collaborate and share code in order to work on the project. As a preliminary, we are running this program on Brian's Apple Computer. We plan to make this program cross compatible as we continue developmeThe program consists of several folders. One houses the computer program which makes the decisions on what to deliver at what time. Another houses the object hierarchy which consists of the objects that can be shipped to designated locations. A third houses the buildings that make up the map. A fourth houses the map classes which are responsible for representing the drones and buildings that make up the map with a 2D array. A fifth folder performs operations which detects groupings of like values on the 2D map array, representing a building with a single point on the map. A sixth folder holds the graphics that display the map onto the screen. The seventh folder holds the information for the drones which ship the resources to designated positions on the map. The eighth folder holds the list of tasks that can be performed which wear the objects used from performing the tasks. The ninth folder holds the characteristics for the individuals whose whims are catered by the overarching computer system. The overarching computer program will use artificial intelligence to predict demands of the individuals at certain times and inform the fulfillment centers to manufacture resources in order to provide the resources necessary to individuals. The drones will then deliver these resources to buildings which contain these individuals.



There will be two modes for assessing the needs of the individuals in the project. The “predetermined” mode will have the agendas of the individuals completed at runtime, which gives the overarching computer program the ability to access needs of these sims ahead of time. The “intelligent” mode will have the sims randomly generate expectations day to day, making the need for AI to predict needs and analyzing trends over time. There are two types of drones that operate in the program. The cargo drone carries only resources. It can carry a maximum payload of 6000 lbs, with a maximum volume of 1000 ft^3. We modeled these numbers off the capabilities of a 20 foot U-Haul truck. The passenger drone is capable of carrying 12 persons. This drone can carry a payload as well, with a payload of 500 lbs and 30 ft^3 of cargo. This cargo is designated as private possessions of the individuals in the drone. The individuals have daily “agendas” which dictate what and where they are going to do during the day. They also wield possessions which follow them along in the passenger drone as they move from location to location. In “predetermined” mode, the individual agendas are composed before runtime of the simulation. In “intelligent”mode, the individual agendas are developed on a day to day basis, making it difficult for the computer program to determine where resources need to be distributed. The object hierarchy is a collection of resources that we expect people use in day to day life. Everything from a grand piano to a toothbrush are implemented to represent resources used in the system. The drones will use simple pathfinding algorithms from going to location to location. There are three types of buildings in our program. The house consists of an occupant capacity of 6 and a volume capacity of 6000 ft^3. The office consists of an occupant capacity of 80 and a volume capacity of 45,000 ft^3. The fulfillment center is responsible for generating drones and resources for the simulation. The map class generates a 2D array that indicates where the streets and buildings are.

Image provided courtesy of Uhaul.com

The graphics end of the project is being implemented using OpenGL with GLFW, an OpenGL C++ library. GLFW is a very basic library, and thus there is a considerable amount of implementation required for a simple map rendering. At the very core, functions are being implemented on top of only being able to render triangles. The graphics C++ file first initializes a screen to be rendered to. The screen will be linked with the map class, which holds the 2D map array. Every "tick" a new screen is drawn to the rendering window with the updated display of the map at a given camera location. The graphics will include functionality to select a given person, drone, or building and display text information about the selected object. A library that adds text rendering will be needed to display the information. The display will be rendered in real time to show movement of people and drones around the map. The graphics will only be retrieving information from the main program. There is no need for the graphics to send information back as it only needs to visually display and not manipulate real time data. Textures will be implemented for different buildings, streets, and people.
The graphics are currently only native to macOS. However, the graphics and subsequently the whole project, will be made available cross platform to masOS, windows, and linux as a desktop application. If there is time at the end of the 2nd semester, this could be extended to mobile applications. The current graphics build displays streets, houses, workplaces, and distribution centers with zoom/pan functionality:



Design Constraints:

We have chosen to write the project in C++ in the backend with OpenGL powering the frontend. The project will be designed to run on macOS initially, then will be expanded to other platforms like Windows and Linux if possible. This is mainly due to the graphics engine. The due date also functions as a time constraint: we have until the end of next semester to finish this. Finding times that work for all members can be a difficult process and can bottleneck productivity until a suitable time is found. This further constrains the amount of time we have to work on the project together.

Ethical Issues:

- This project is intended to model future technological developments for the benefit of society. It models an extreme "what if" scenario which itself is a topic of debate for many futurists, lawmakers, and privacy experts, and in many ways it blurs the line between utopia and dystopia. It is difficult to apply current modern ethical principles to such a technologically revolutionized potential reality. With that in mind, it is still a worthwhile pursuit because this project itself is being produced here and now.

- 1.1 Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.
    - Ideally this program will help society achieve higher levels of productivity and efficiency by revolutionizing the idea of property ownership. We already live in a society where we can ask our next door neighbor for a cup of sugar, and this project attempts to amplify that line of thinking. If you need a chair, and your neighbor isn't using one, and the central computer knows this, then the computer could orchestrate your acquisition of the chair. If the neighbor needs it back, then the computer will help him get it back. This would maximize the wellbeing of the population by satisfying their material desires when an optimization could be made without harming another person.
- 1.6 Respect privacy.
    - The program will have a lot of personal user data that it will need to deal with in an ethical manner. The individuals whom the data tracks will need to know what the program will do with their data, and their data needs to only be used for things that the individuals have agreed to be allowed. Since the AI will need to predict the needs, habits, and intentions of the individuals, they should be aware of this and the data should be protected. If there is ever a breach of private data, the individuals should be made aware so they can take necessary actions to protect themselves.
- 2.9 Design and implement systems that are robustly and usably secure.
    - If this simulation was turned into a real program, we would be dealing with massive amounts of personal data, transportation logistics, and physical goods. It would be very important for the program to be robust and extremely secure. Any breaches or loss of data could result in disruption in peoples' lives who depend on the program to operate in a correct and efficient manner. The integrity of the program would need to be thoroughly and routinely tested and verified in order to mitigate these risks.

Intellectual Property Issues:

- Intellectual property issues for this project will primarily center around our compliance with the licences of the software that we choose to use. Currently, the only library that we are using is GLFW, an open-source library for OpenGL. GLFW uses the zlib/libpng license, which grants free usage, alteration, and distribution of GLFW software, and even allows for use in commercial applications if we pursue a route of commercialization. These freedoms are only granted if we abide by the restrictions provided in the license: we cannot claim ownership of the original GLFW software, nor can we deliberately misrepresent any modifications made to the software as being the original software. It will be critical that we abide by the terms of this license, or any other licenses for software that we use, in order to maintain intellectual integrity.

Change Log:

- In the initial project description, the graphics was set to be worked on last after everything else was finished. Graphics are now being worked on concurrently with other tasks for visualization of the project as it is developed.
- Although a Gantt Chart was included in our initial project description, it was included as a separate document. We have now included screenshots of the chart within this document to make it more accessible.
- The Gantt Chart has now been updated for the new semester.