INTEGRATIVE PROJECT IN COMPUTER SCIENCE AND MATHEMATIC

420-204-RE

PROJECT

DELIVERABLE-1-

Team

Edelina Alieva
Kyle Bouchard (Team Coordinator)
EnYi Hou
Matthew Leprohon

Team Name

Totally Spies

List of Program Courses and Concepts

List all the program courses that you have already taken or are currently taking and list their key concepts. If there is a discrepancy between team members where a team member did not take a certain course, mention it.

Course	Concepts
Calculus I & II	Derivatives, Integrals …
Mechanics	Kinematics, Newton's Laws …
General Chemistry	Periodic Table, Atomic Structures, Visual Representation of Atoms, Ideal Gas Law, Heat, …
Waves, Optics and Modern Physics	SHM, Sound (Doppler, Harmonics), Interference, Intensity, Quantum Mechanics …
Linear Algebra	Matrix Operations, Vectors, Vector Spaces, Abstract Proofs, Determinants, Linear Combinations, …
Intro to Programming	Java Basics, Primitive Data Types, Arrays, …
Data Structures	Collections (Maps, Trees, Queues, Stacks, Lists), …
Programming in Graphical Environment	JavaFX (Layout Managers, UI Controls), Scene Builder, OpenCSV, …

Project Idea

Each team member must think of and choose a project idea then work with their teammate to select the more convenient ones.

Team Name: Totally Spies	
Team Member's name and Project Idea 1:	Kyle Bouchard: OPUS fare cheapest route calculator.
Team Member's name and Project Idea 2:	Edelina Alieva: 3D Computer aided design app.
Team Member's name and Project Idea 3:	EnYi Hou: Evolution simulation
Team Member's name and Project Idea 4:	Matthew Leprohon: 2D / 3D Physics Engine (allow the user to play with various assets such as boxes or liquids and see how they interact with each other and their environment).
Selected Project Ideas and why:	Plan A: Evolution Simulator Plan B: Computer Design App / Physics Engine

Project Description

Describe you project idea, in brief, all while addressing the following points:

Concept

- Describe the physical and/or mathematical concept(s) behind the project.

Concept Aspects

- Identify and list the main aspects of the concept such as the problem it addresses, the proposed solution, the solution category among other approaches' categories.
- The possible variable parameters that would control the user interface animating the concept.

Typical Input

- Describe the typical input for the solution of the applied concept to work.

Expected Output

- Describe the expected output and how the user interface would look like and what it would allow the user to do.

Feasibility

- List the JavaFX, or similar technology, elements, and implementation components that you expect to use to implement the project.
- Justify the feasibility in terms of timeline and team tasks assignment.

Individual part

- For each team member, describe their individual part and how it would integrate with the whole project with other team members parts.

Description:

Our general idea for the project is to use the concept of evolution to simulate a battle between two species; the prey and the predators. Each entity will be controlled with the help of its own neural network, utilizing 2D matrices as the neurons and 3D matrices, or vectors as the weights.

The intention of the simulator would be to help educate the end user about certain concepts such as population control and to see how modifying certain parameters in nature can affect their survival. The biggest challenges that we'll be facing during development will likely be performance (more entities as time passes + many computations per frame) and optimizing the input variables of the neural network in order to make the learning better.

The user will have control over all of the initial parameters when starting their population. This includes the starting population of each species (within a certain cap) and their rate of multiplication, death, and mutation. We also intend to improve user experience by providing animation controllers and an interface to view the statistics of any given entity. The intended output would be an animation showing the two species reproduce, survive and hunt as well as some data visualizers such as life-cycle graphs and general statistics.

The project will be implemented using JavaFX's provided components (for animation and graphics) as well as Scene Builder to help build the actual structure of the interface. If performance becomes an issue, we may add libraries to offshore expensive tasks to more performant code. The project will be more feasible through proper team management and by dividing the tasks as evenly as possible among the team members and assigning deadlines through GitHub Issues as soon as the timeline is fully planned out.

Our intention is to break up the work as evenly as possible to make sure that as much work as possible can be done synonymously. All current tasks and features required for the project have been broken down into various tasks through GitHub Issues, with each one being assigned a certain point value. To prevent overloading a single person, tasks will be delegated appropriately based on those points to ensure that everyone has equal workloads. These tasks span all the way from our initial build to the beta release to the final project (through Milestones). If needed, we may transfer some people between tasks if any difficulties occur or add extra people to tasks if they prove to be too long. In general, around two people will be assigned to any given part at a time.