Deliverable 2
420-204-RE sect. 00001

<u>Team Totally Spies:</u>
Edelina Alieva
Kyle Bouchard
EnYi Hou
Matthew Leprohon

Table of Contents

Description:

Our project is an evolution/population simulator that simulates a battle between two species; prey and the predators. Using AI and a neural network built using Linear Algebra, the two species will interact with one another and give a semi-realistic demonstration of population control. Each species has a different goal in order to multiply and survive, the prey being to live as long as possible, and the predators to eat as many prey as possible. The user will have control over various initial parameters to influence the simulation such as modifying the starting populations and various rates such as multiplication, death, and mutation. The simulation will play out as an animation to give a visual representation. Lifecycle graphs, statistics and a representation of the neural network will also be available to include as much information as possible.

Task Breakdown:

Dark greyed boxes are **tracker** issues used to keep track of large components of the project.
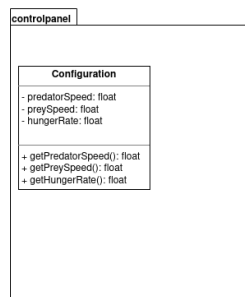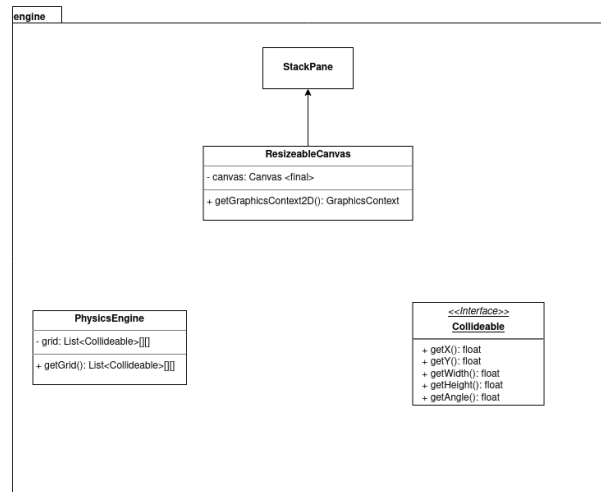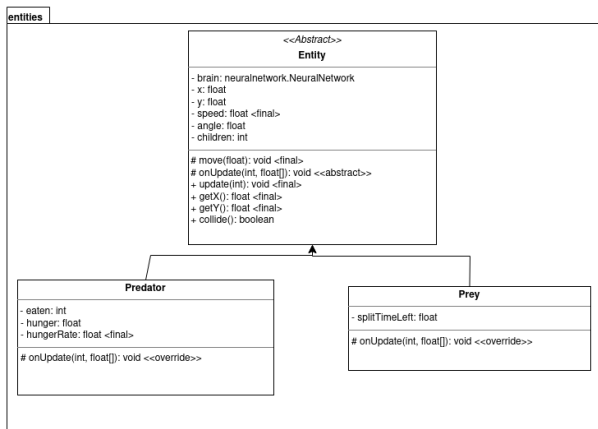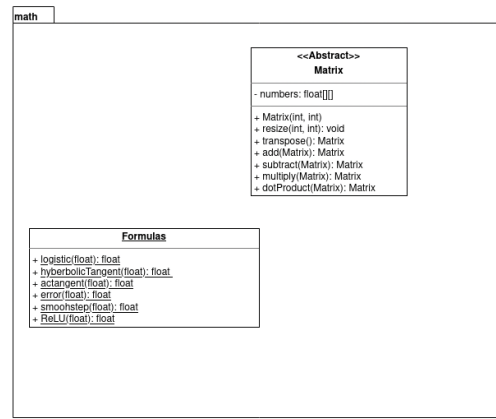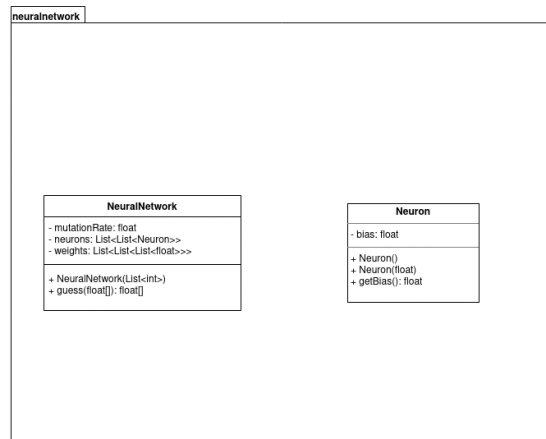*Light greyed boxes* are **unassigned** issues as they haven't been brought to the sprint yet.
Stories are named as TeamName - IsTracker - Issue# - Parent#

Story	Assignees	Due Date
TS-TR-1-P5: Implement Neural Network	Full Team	March 3rd
TS-TR-2-P5: Implement the Entities	Matthew & En Yi	March 3rd
TS-3-P5: Implement Application Control Panel	Kyle & Edelina	March 3rd
TS-4: Follow Entities	**Unassigned**	March 31st
TS-5-P13: Save/Load Files	**Unassigned**	March 31st
TS-6: Video	**Unassigned**	May 12th
TS-7-P8: Graphs	**Unassigned**	April 24th
TS-8: Add Statistics Overlay	**Unassigned**	April 24th
TS-9: View Neural Networks	**Unassigned**	May 12th
TS-10: Entity Customization	**Unassigned**	May 12th
TS-11: Map Customization	**Unassigned**	May 12th
TS-12-P13: Timeline Functionality	**Unassigned**	April 24th
TS-13: Menu Bar	**Unassigned**	April 24th
TS-14: Settings	**Unassigned**	May 12th
TS-15-P13: Help Menu	**Unassigned**	April 24th
TS-21-P1: Implement linear algebra & matrix class	Matthew & En Yi	March 3rd
TS-23-P2: Implement Entity abstract class	Matthew & En Yi	March 3rd
TS-24-P1: Add variable layers and node per layer	Kyle & Edelina	March 3rd
TS-25-P2: Add neural network backend for the Entities	En Yi	March 3rd
TS-26-P1: Implement all properties of the neural network	Kyle & Edelina	March 3rd

|		
TS-27-P1: Create simple API for the neural network	Kyle & Edelina	March 3rd
TS-28-P2: Implement the Entity types and their behaviours	Matthew	March 3rd
TS-TR-29: Complete Deliverable #2	Full Team	February 24th
TS-30-P29: Create the class diagrams	Kyle & Edelina	February 24th
TS-31-P29: Draw out the wireframes	Kyle	February 24th
TS-32-P29: Create sample input & output grid	En Yi	February 24th
TS-33-P29: Create sprint report	Matthew	February 24th

Class Diagrams:

neuralnetwork

NeuralNetwork
- mutationRate: float
- neurons: List<List<Neuron>>
- weights: List<List<List<float>>>

+ NeuralNetwork(List<int>)
+ guess(float[]): float[]

Neuron
- bias: float

+ Neuron()
+ Neuron(float)
+ getBias(): float

math

<<Abstract>>
Matrix
- numbers: float[][]

+ Matrix(int, int)
+ resize(int, int): void
+ transpose(): Matrix
+ add(Matrix): Matrix
+ subtract(Matrix): Matrix
+ multiply(Matrix): Matrix
+ dotProduct(Matrix): Matrix

Formulas
+ logistic(float): float
+ hyberbolicTangent(float): float
+ actangent(float): float
+ error(float): float
+ smoohstep(float): float
+ ReLU(float): float

entities

<<Abstract>>
Entity
- brain: neuralnetwork.NeuralNetwork
- x: float
- y: float
- speed: float <final>
- angle: float
- children: int

move(float): void <final>
onUpdate(int, float[]): void <>
+ update(int): void <final>
+ getX(): float <final>
+ getY(): float <final>
+ collide(): boolean

Predator
- eaten: int
- hunger: float
- hungerRate: float <final>

onUpdate(int, float[]): void <<override>>

Prey
- splitTimeLeft: float

onUpdate(int, float[]): void <<override>>

engine

StackPane

ResizeableCanvas
- canvas: Canvas <final>

+ getGraphicsContext2D(): GraphicsContext

PhysicsEngine
- grid: List<Collideable>[][]

+ getGrid(): List<Collideable>[][]

<<Interface>>
Collideable
+ getX(): float
+ getY(): float
+ getWidth(): float
+ getHeight(): float
+ getAngle(): float

controlpanel

Configuration
- predatorSpeed: float
- preySpeed: float
- hungerRate: float

+ getPredatorSpeed(): float
+ getPreySpeed(): float
+ getHungerRate(): float

Sample Input and Output Data Grid:

A list of inputted parameters and how their output on the simulation

INPUT	OUTPUT
Animation Speed	Increases/decreases the animation speed
Pause/Start	Changes the animation's play state
Speed of the Recorded Animation	Changes the speed of the recorded timelapse
Mutation Rate	Affects the learning speed of the entities
Energy Rate	Affects balancing and the entity behaviours by changing the rate at which their energy decreases
Split Rate	Affects balancing and the entity behaviours by changing the rate at which entities can reproduce
Minimum and Maximum Prey Speed	Affects the balancing and entity interactions by changing the possible speed values an entity can spawn with
Minimum and Maximum Predator Speed	
Angular Velocity	Affects balancing and entity interactions by changing their angular velocities
Initial Prey and Predator count	Affects balancing by spawning a certain amount of entities
Maximum Entity Amount	Affects the balancing and performance of the simulation by capping the entity population
Neural Network Structure	Affects the behaviour and interactions of all the entities
Sensors Amount	Affects performance by changing how many sensors each entity has
Sensors Distance	Affects the interaction and behaviors of the entities by changing their sight
Color	Changes the colours of the entity types
Map Size	Changes the size of the map by adding extra grids

Wireframes:
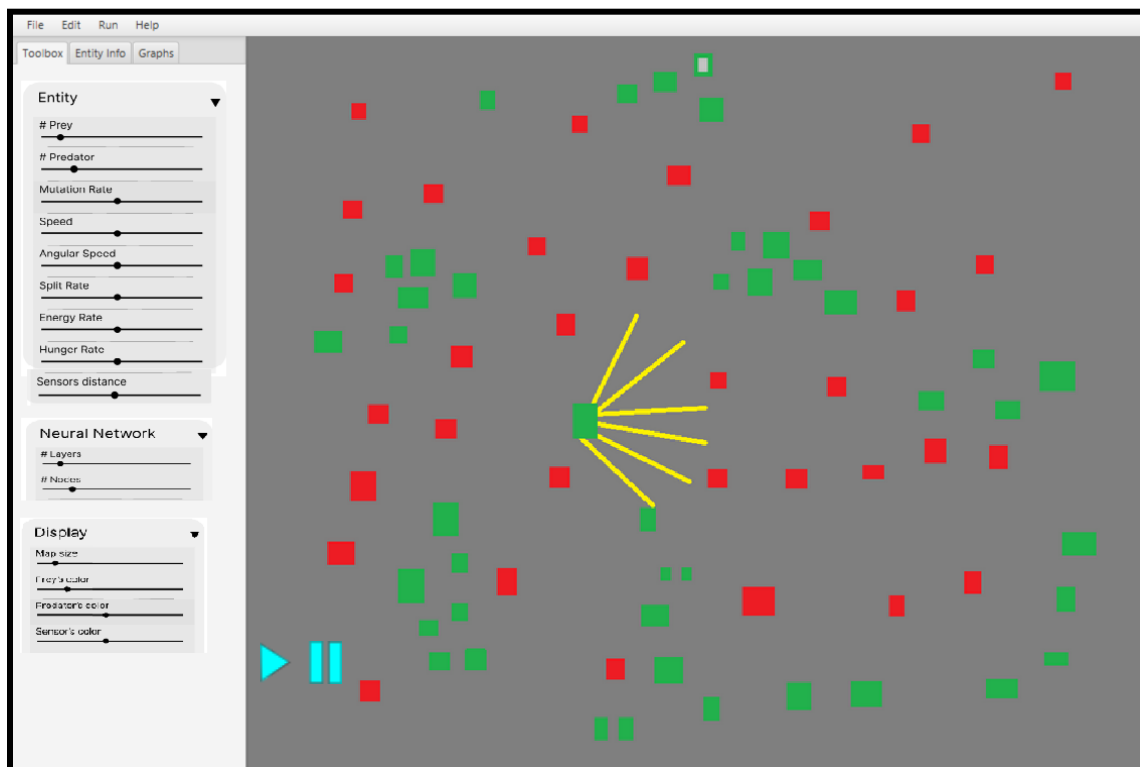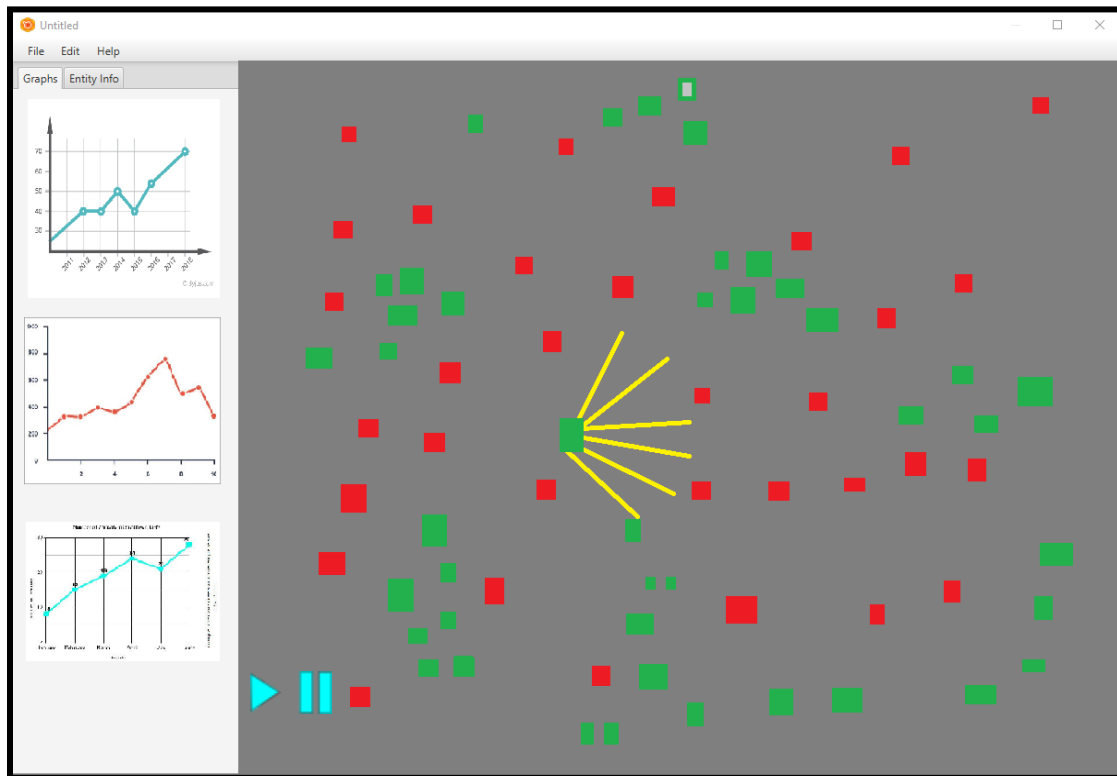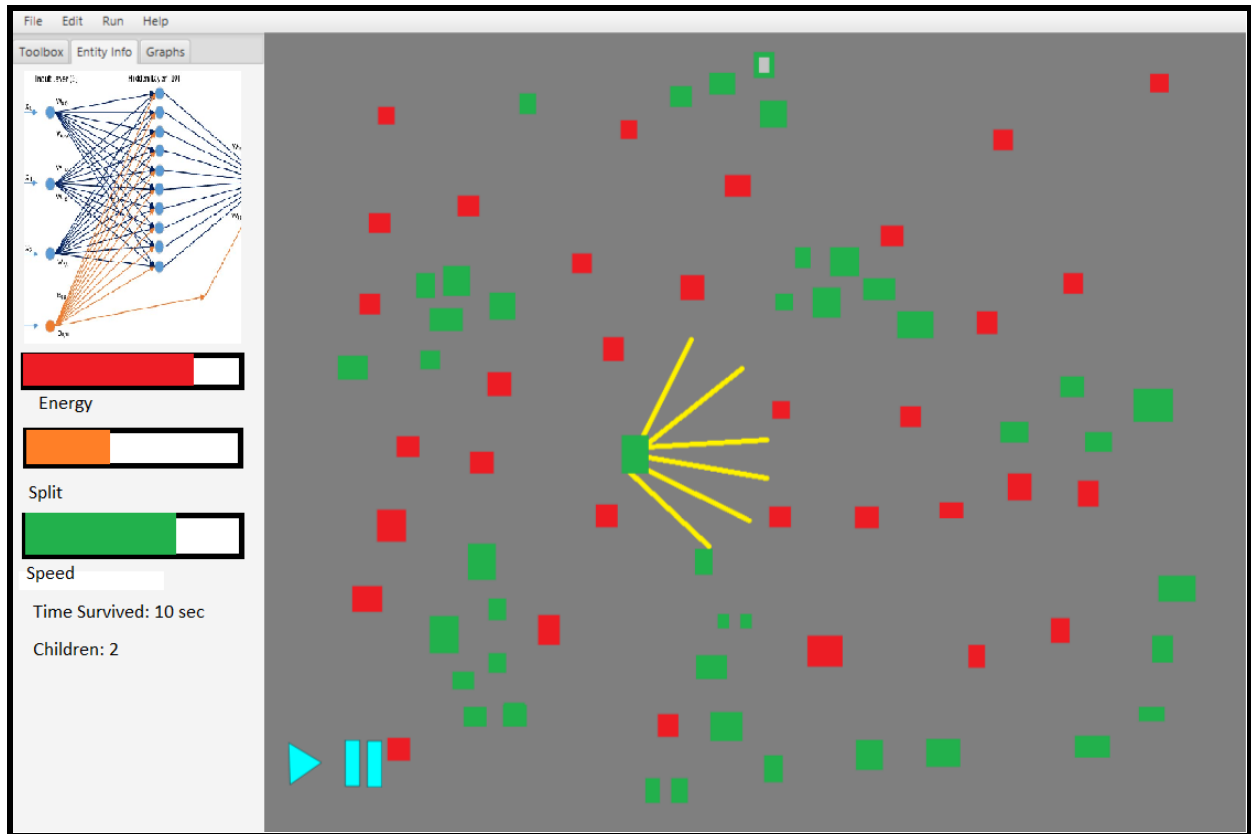
Sprint Report:

Sprint 1 - Full Team
Note that tracker issues are not being listed as they're worth 0 points.

Assigned Stories	Resolved?	Carried Over?
TS-3-P5: Implement the application's control panel (5)	☐	☑
TS-21-P1: Implement linear algebra vector and matrix class (5)	☐	☑
TS-23-P2: Implement Entity abstract class (2)	☐	☑
TS-24-P1: Add variable layers and nodes per layer (3)	☐	☑
TS-25-P2: Add neural network backend for the Entities (3)	☐	☑
TS-26-P1: Implement all properties of the neural network (2)	☐	☑
TS-27-P1: Create simple API for neural network (3)	☐	☑
TS-28-P2: Implement the Entity types and their behaviours (5)	☐	☑
TS-30-P29: Create the class diagrams (2)	☑	☐
TS-31-P29: Draw out the wireframes (1)	☑	☐
TS-32-P29: Create sample input & output grid (1)	☑	☐
TS-33-P29: Create sprint report (1)	☑	☐
Total Points: 33	Pts: 5	Pts: 28

Kyle Bouchard

Assigned Stories	Resolved?	Carried Over?	Blocked?
TS-3-P5: Implement the application's control panel (5)	☑	☑	☐
TS-24-P1: Add variable layers and nodes per layer (3)	☐	☑	☐
TS-26-P1: Implement all properties of the neural network (2)	☐	☑	☐
TS-27-P1: Create simple API for neural network (3)	☐	☑	☐
TS-30-P29: Create the class diagrams (2)	☑	☐	☐
TS-31-P29: Draw out the wireframes (1)	☑	☐	☐
Total Points: 16	Pts: 3	Pts: 13	Pts: 0

En Yi Hou

Assigned Stories	Resolved?	Carried Over?	Blocked?
TS-21-P1: Implement linear algebra vector and matrix class (5)	☐	☑	☐
TS-23-P2: Implement Entity abstract class (2)	☐	☑	☐
TS-25-P2: Add neural network backend for the Entities (3)	☐	☑	☐
TS-32-P29: Create sample input & output grid (1)	☑	☐	☐
Total Points: 11	Pts: 1	Pts: 10	Pts: 0

Edelina Alieva

Started Stories	Resolved?	Carried Over?	Blocked?
TS-3-P5: Implement the application's control panel (5)	☐	☑	☐
TS-24-P1: Add variable layers and nodes per layer (3)	☐	☑	☐
TS-26-P1: Implement all properties of the neural network (2)	☐	☑	☐
TS-27-P1: Create simple API for neural network (3)	☐	☑	☐
TS-30-P29: Create the class diagrams (2)	☑	☐	☐
Total Points: 15	Pts: 2	Pts: 13	Pts: 0

Matthew Leprohon

Assigned Stories	Resolved?	Carried Over?	Blocked?
TS-21-P1: Implement linear algebra vector and matrix class (5)	☐	☑	☐
TS-23-P2: Implement Entity abstract class (2)	☐	☑	☐
TS-28-P2: Implement the Entity types and their behaviours (5)	☐	☑	☐
TS-33-P29: Create sprint report (1)	☑	☐	☐
Total Points: 13	Pts: 1	Pts: 12	Pts: 0

Sprint Backlog · 10
Tasks planned for the current sprint

Vanier-evosim #28
TS-28-P2: Implement the Entity types and their behaviours

Vanier-evosim #27
TS-27-P1: Create simple API for neural network

Vanier-evosim #26
TS-26-P1: Implement all properties of the neural network

Vanier-evosim #25
TS-25-P2: Add neural network backend for the Entities

Vanier-evosim #24
TS-24-P1: Add variable layers and nodes per layer

Vanier-evosim #23
TS-23-P2: Implement Entity abstract class

Vanier-evosim #21

+ Add item

In Progress · 0
Tasks in progress in the current sprint

+ Add item

Under Review · 0
Tasks pending review from members

+ Add item

Sprint 1: 02/24 - Resolved · 5
Completed & reviewed tasks

Vanier-evosim #29
TS-TR-29: Complete Deliverable #2

Vanier-evosim #33
TS-33-P29: Create sprint report

Vanier-evosim #31
TS-31-P29: Draw out the wireframes

Vanier-evosim #30
TS-30-P29: Create the class diagrams

Vanier-evosim #32
TS-32-P29: Create sample input & output grid

+ Add item