Modeling and simulation of complex systems

Project Report

Nguyen Xuan Tung (M22.ICT.006)

1 Introduction

1.1 Subject

The project aims to investigate the evolution of butterfly wing colorations from prey-predator interactions and camouflage mechanisms. The study focuses on how the color patterns of the butterflies' wings have adapted through time. There are multiple factors that influence the colors, such as the dominant colors in the butterfly's habitat, the evasion from predators through camouflage, the reproduction of different types of butterflies, and the change in environmental colors. The methodology involves creating an agent-based model to simulate butterfly populations of 3 types of butterflies: gray, black, and white, their reproduction, coloration transmission, and the impact of predation pressure and environmental color transitions.

1.2 Hypothesis

The project hypothesis suggests that dynamic environmental transitions and predation intensities significantly affect the development of butterfly coloration. In order to test the hypothesis, we need to calibrate the model and run batch simulations to find out which factor is the most influential to the number of butterflies.

2 Base Model

2.1 Overview

The purpose of the base model of the project is to simulate how butterfly colors change over time and how prey-predator interaction and camouflage influence these colors. The model is an agent-based representation that describes the butterfly population, its active time, its fixed-rate reproduction, and its coloration based on a simple genetic transmission system: white butterflies transmit only white coloration, black butterflies transmit only black, and gray butterflies have a 25% chance of transmitting either black or white and 50% of transmitting gray. The model incorporates predation pressure, representing the predator population that hunts butterflies during their active time but can be deceived by their camouflage abilities. The model's environment has a range of colors from black to white, which measures how well the butterflies can hide themselves. The butterfly population at the beginning of each simulation will be equal to each other to ensure no bias.

2.2 Formulation

The formulation can be defined as:

- Set the time duration of a time step to 1 day.
- Define a grid cell that has a gradient of color ranging from black to white (horizontally).
- Define the butterfly species with random movement to a distance of 8 cells and put it into cells.
- Define the reflex `reproduction` that allows the butterflies to randomly find mates and create offspring
- Define the active time of the butterfly from 9:00 am to 17:00 pm.
- Define predator species with random movement to a distance of 8 cells and put it into cells.

- Define the reflex `choose_target` that allows predators to choose any butterfly within a neighborhood of 24.
- Define the reflex `move` that enables predators to chase the target butterflies. Along the path, if the predator meets another butterfly, it can eat that butterfly.
- Define the active time of the predator from 18:00 pm to 6:00 am and the low probability that the predator will be active from 6:00 am to 18:00 pm.
- Display species on the environment grid.

2.3 UML

2.3.1 Environment

The environment agent has several attributes and methods:

- `color`: An RGB color of the cell.
- `red_color`: An integer that tracks the red component of the cell's color.
- `is_occupied`: A boolean indicating whether the cell is occupied by a predator or butterfly.
- `neighbors`: A list of adjacent cells, represented as a list of `Cell` objects.
- `init()`: Initializes the cell with its color.

The UML of the environment agent is represented as:

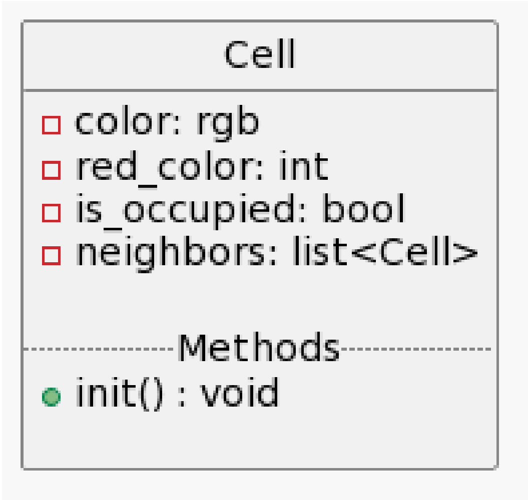

Figure 1: UML of agent environment

2.3.2 Butterfly

The butterfly agent has several attributes and methods:

- `my_cell`: A `Cell` object that the butterfly currently occupies.
- `color`: An RGB color of the butterfly.
- `my_cycle`: An integer tracking the current life cycle of the butterfly (in hours).
- `red_color`: An integer that tracks the red component of the butterfly's color.
- `my_icon`: An image visually representing the butterfly on the display.
- `butterfly_life_cycle`: An integer representing the length of the butterfly's life cycle (in hours).
- `init()`: Initializes the butterfly with its color and location in the environment.
- `reflex_count_cycle()`: Increments the `my_cycle` attribute to track the butterfly's life cycle.
- `reflex_move()`: Enables the movement of the butterfly to a new `Cell`.

- `reflex_butterfly_die()`: Determines the conditions for the butterfly's death.
- `reflex_reproduce()`: Represents the reproduction of the butterfly and its genetic transmission to offspring.

The UML of the butterfly agent is represented as:

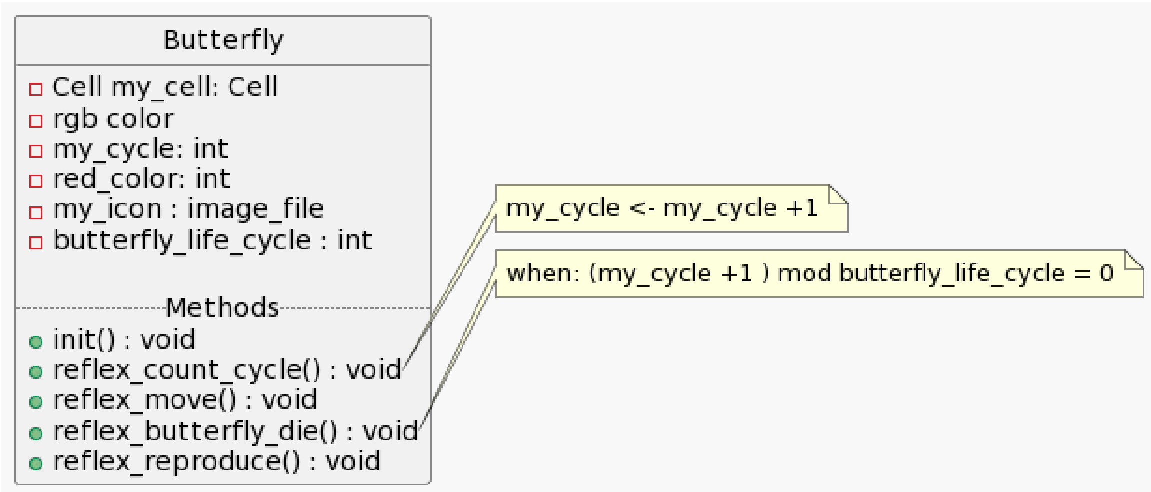

Figure 2: UML of agent butterfly

We define the reflex `move` that allows the butterflies to randomly choose a new cell in the neighborhood without being occupied and to move to this cell.

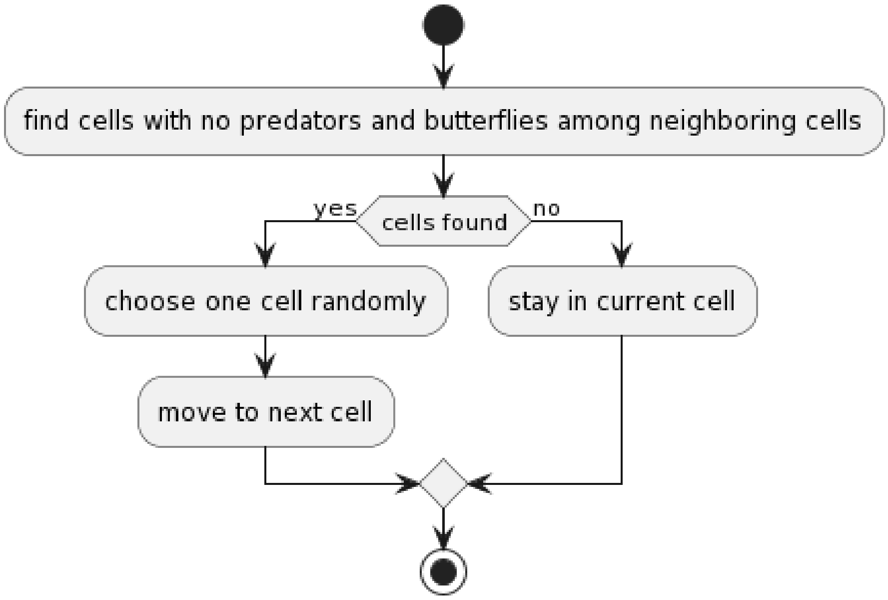

Figure 3: Activity diagram of a butterfly's movement

We define the reflex called `reproduction` that will be activated when there is another butterfly in the neighborhood and, according to the probability `butterfly_proba_reproduce`. The parents will create `nb_offsprings` (random number from 1 to `nb_max_offsprings`). The color of the offspring is based on their parents, and the offspring is placed into any free cell in their parents' neighborhood.

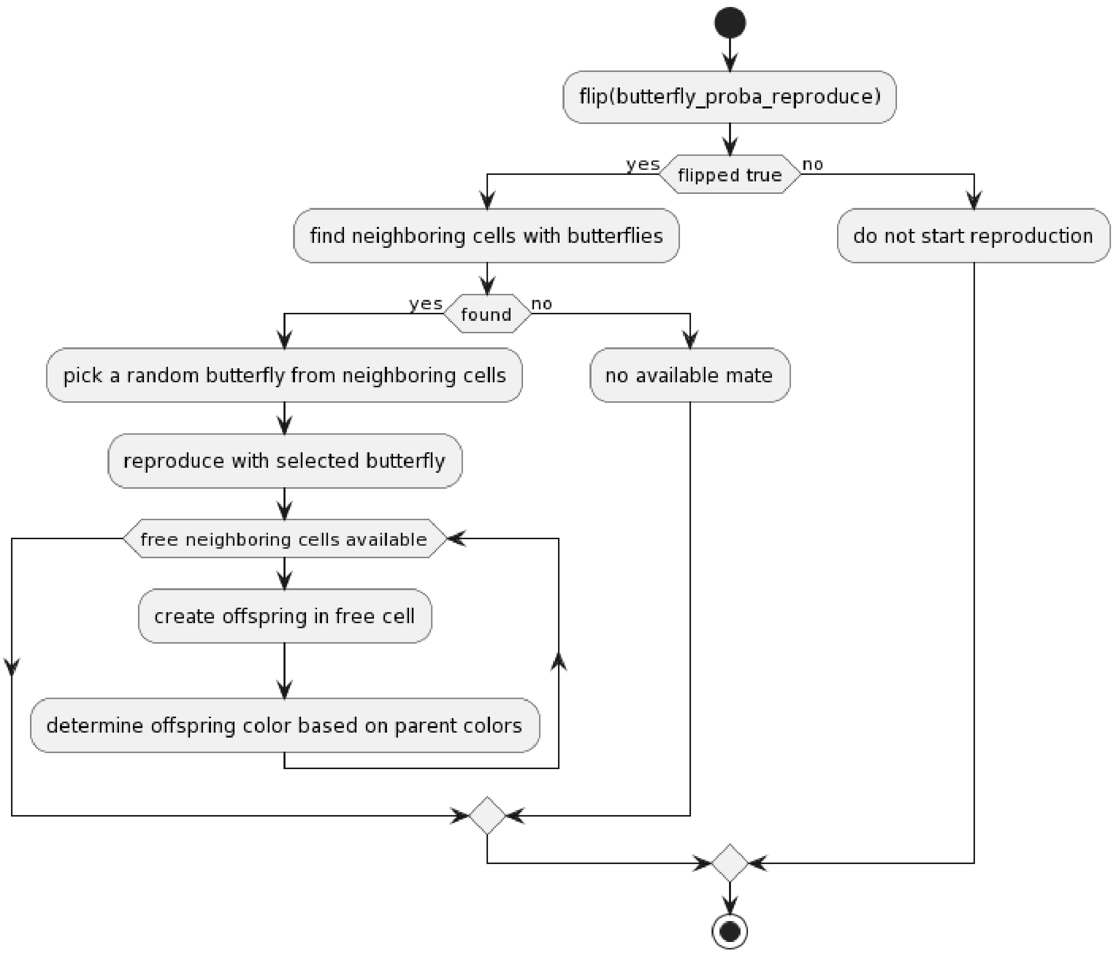

Figure 4: Activity diagram of a butterfly's reproduction

2.3.3 Predator

The predator agent has several attributes and methods:

- `my_cell`: A `Cell` object that the predator currently occupies.
- `color`: An RGB color of the predator.
- `red_color`: An integer that tracks the red component of the predator's color.
- `my_icon`: An image visually representing the predator on the display.

- `goal`: The target point the predator aims to reach.
- `source`: The current point of the predator.
- `target_path`: The calculated path to the target.
- `target_butterfly`: A butterfly that is currently being targeted.
- `init()`: Initializes the predator ly with its color and location in the environment.
- `reflex_choose_target()`: Select a new target butterfly based on certain criteria
- `reflex_move()`: Allows the predator to move towards its target.
- `action_set_next_cell(Cell)`: Sets the next cell as the predator's current location and updates `is_occupied` status.

The UML of the predator agent is represented as:

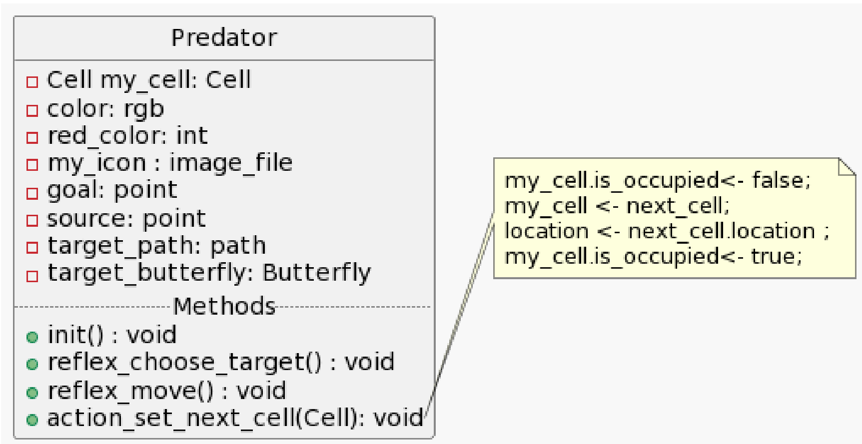

Figure 5: UML of agent predator

We defined the reflex `choose_target` that allows the predators to randomly hunt for the butterflies. Once the predator has selected a butterfly, We set the goal point to the target butterfly cell location and set the butterfly as a target.

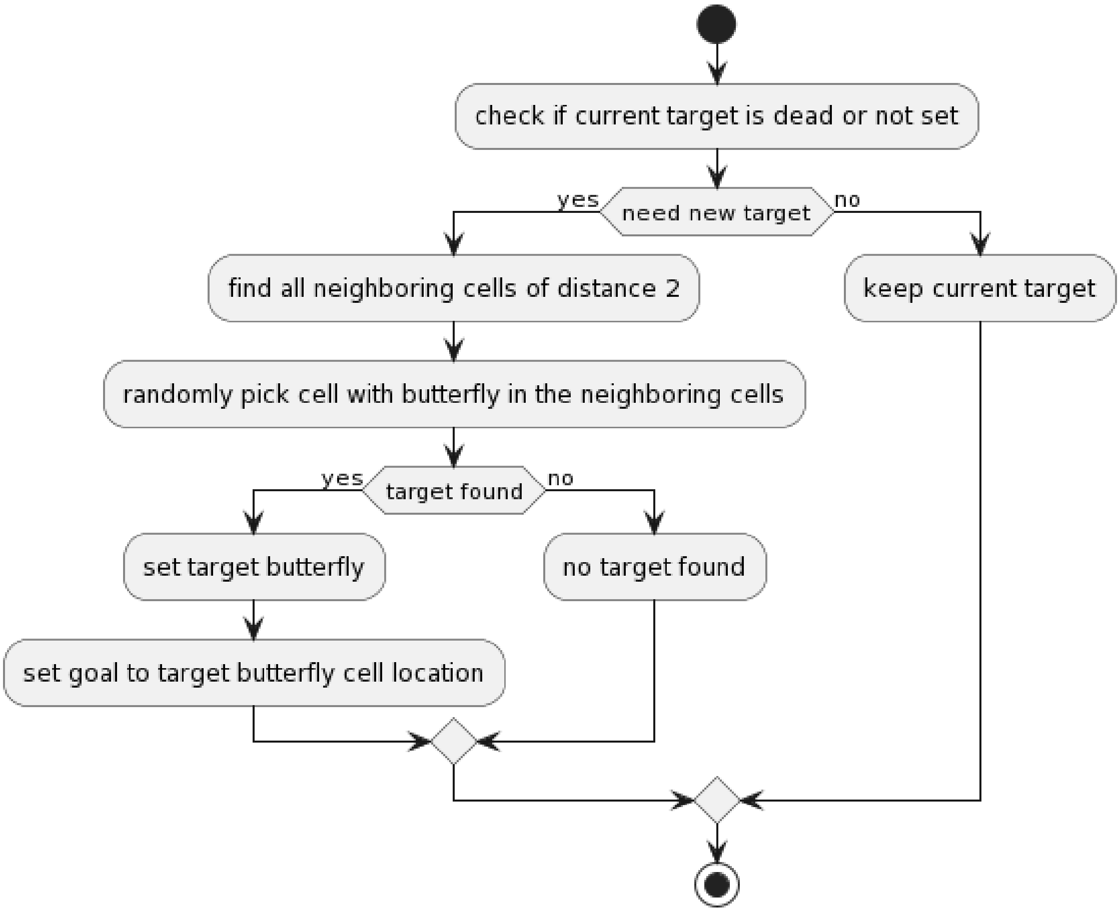

Figure 6: Activity diagram of a predator's target chosen

We define the reflex called `move` that will be activated in each cycle. Initially, We check whether the predator has its own target butterfly. The shortest path is then calculated, and the predator moves a step closer to the predator. It will eat any butterfly on its path to the target butterfly. In case of not having any target butterfly, the predator randomly chooses a cell and moves to it.

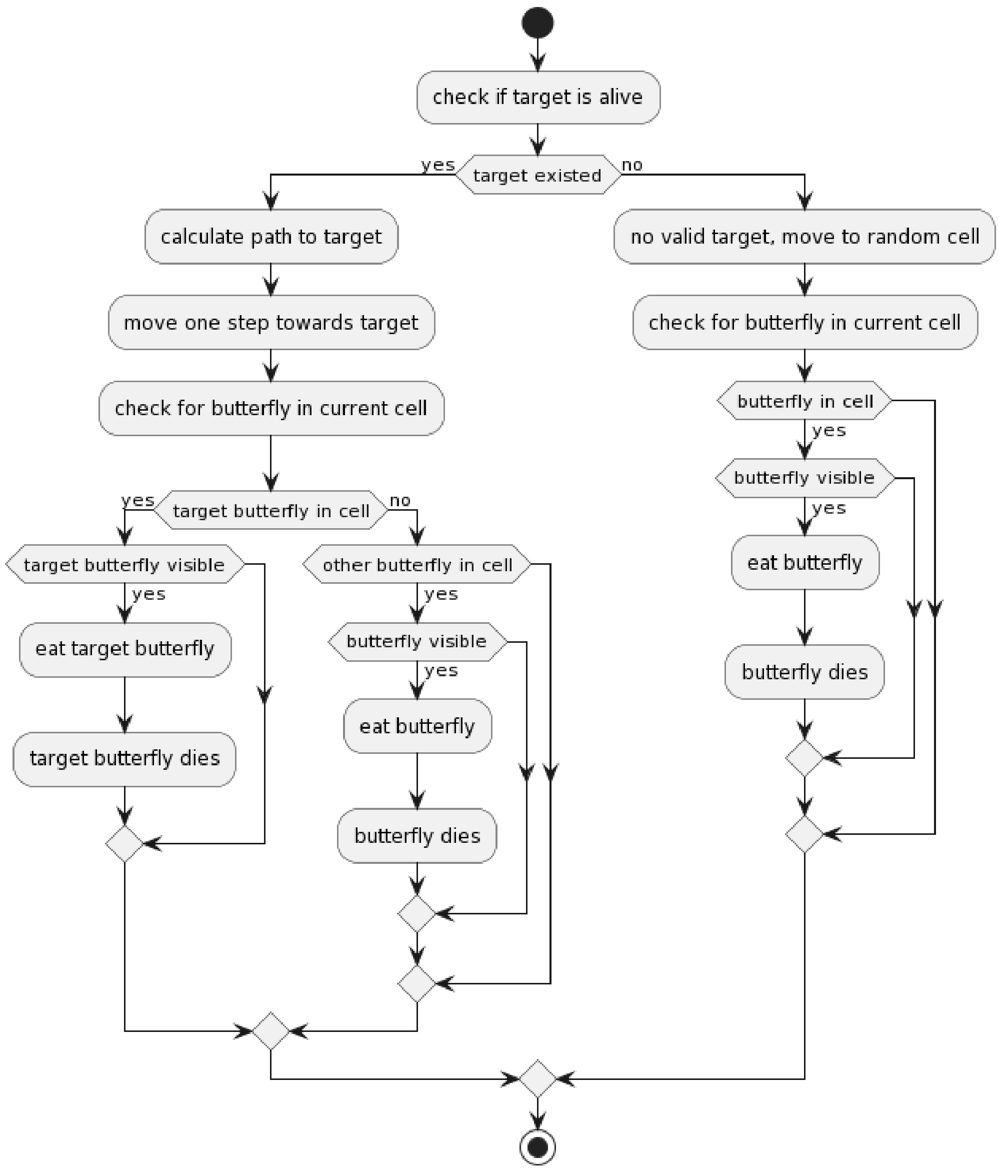

Figure 7: Activity diagram of a predator's movement

We run a batch simulation of 16, and the mean population of each type of butterfly after 200 cycles is shown in Table 1.

Table 1: The difference in the number of butterflies in the base model.

White butterfly	Black butterfly	Gray butterfly
196	197.5	204.6875

Since there is no restriction on how the predator hunts the butterfly, a common scenario will happen where one butterfly type becomes predominant and outnumbers the others. However, with those restrictions on the animal's active time, we achieve a balanced butterfly population.

3 Extension 1

3.1 Overview

In this extension, instead of hunting the butterfly randomly, the predator is chasing the most dominant butterfly in the environment. There are several ways for the predators to hunt the most represented color. We can find the most represented color in the environment and make the predator hunt a random butterfly with that color. Another way is to find the most represented color based on the predator's surroundings. In this project, we will use the latter method for all three extensions. The formulation can be defined as:

- Define the action `get_dominant_color` to get the color of the most dominant butterfly in the surrounding neighbors.
- Update the reflex `choose_target` so that the predators now target the most dominant butterfly in the surrounding neighbors.

We updated the chosen target activity diagram UML to satisfy the requirement. The rest of the code is kept intact.

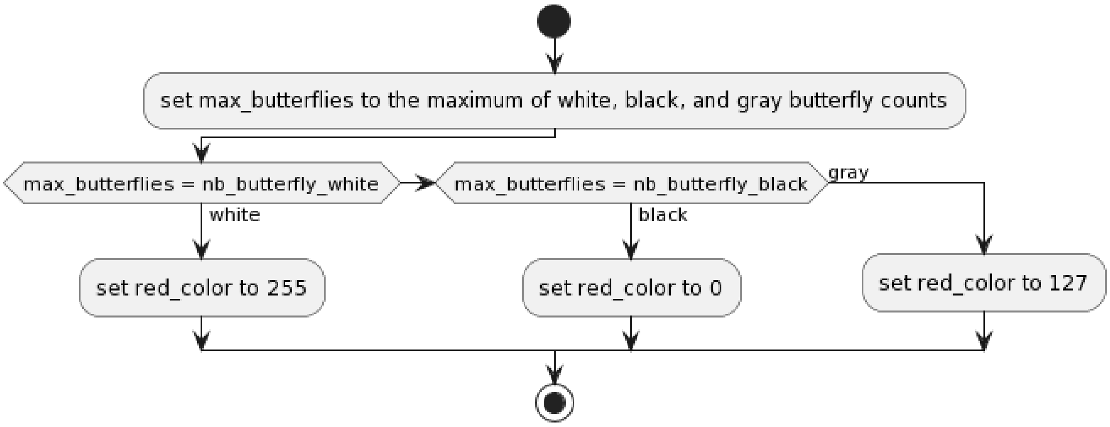

Figure 8: Activity diagram of get most dominant color in extension 1.

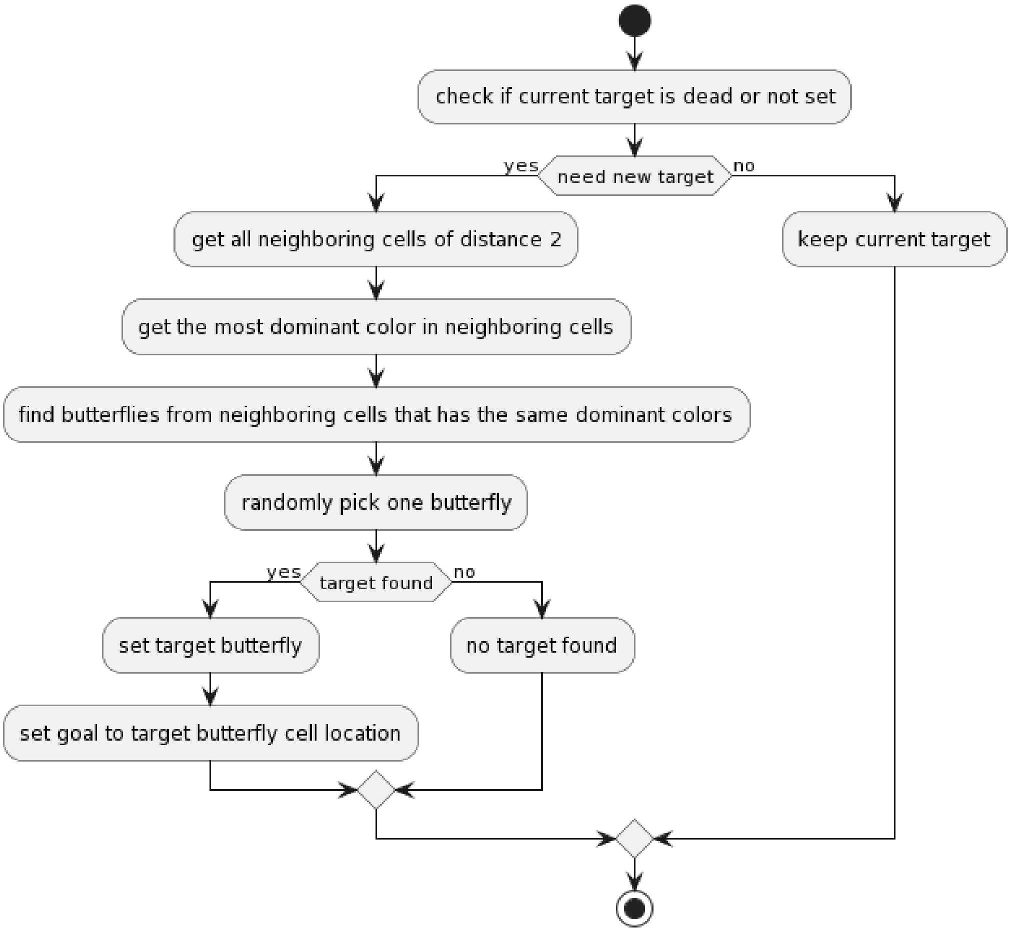

Figure 9: Activity diagram of a predator's target chosen in extension 1.

We run a batch simulation of 16, and the mean population of each type of butterfly after 200 cycles is shown in Table 2.

Table 2: The difference in the number of butterflies in the first extension.

White butterfly	Black butterfly	Gray butterfly
222.125	187.5625	233.5625

Despite the implementation of hunting restrictions for predators in the model, it does not lead to a balanced butterfly population. Instead, there is a significant imbalance, with gray and black butterflies having much larger populations than black.

4 Extension 2

4.1 Overview

In this extension, the model integrates dynamic environmental transitions, where the colors of environmental patches change at varying speeds. This extension explores how different environmental color change rates impact evolution. In this project, we mix the abrupt with gradual transition. When the `current_hour` is from 21:00 pm to 6:00 am, we set the cell color to black. When the `current_hour` is from 6:00 am to 15:00 pm, we set the cell color to white. Otherwise, we define the change in color in 5 different ways, but we only use the first way in this extension; the other five ways will be used in extension 3. The formulation can be defined as:

- Define the reflex `change_color` to change the background color of each cell at each cycle.

The formula for color changes is defined as follows:

- The environmental color transitions horizontally from left to right.

$$\begin{aligned} \text{color\_value} = ((\text{grid\_x} - \text{cycle} \mod \text{cell\_width} + \text{cell\_width})) \mod \text{cell\_width}) \\ \times \text{color\_increment} \end{aligned} \tag{1}$$

- The environmental color transitions horizontally from right to left.

$$\begin{aligned} \text{color\_value} = ((\text{grid\_x} + \text{cycle} \mod \text{cell\_width}) \mod \text{cell\_width}) \\ \times \text{color\_increment} \end{aligned} \tag{2}$$

- The environmental color transitions vertically from top to bottom.

$$\begin{aligned} \text{color\_value} = ((\text{grid\_y} - \text{cycle} \mod \text{cell\_height} + \text{cell\_height}))) \mod \text{cell\_height}) \\ \times \text{color\_increment} \end{aligned} \tag{3}$$

- The environmental color transitions vertically from bottom to top.

$$\text{color\_value} = ((\text{grid\_y} + \text{cycle} \mod \text{cell\_height}) \mod \text{cell\_height}) \times \text{color\_increment} \tag{4}$$

- The color randomly changes with no rules.

$$\text{color\_value} = \text{random\_int}(0, 255) \tag{5}$$

- Mixture of all the aforementioned ways.

The UML for the color chosen is described in Fig. 10.

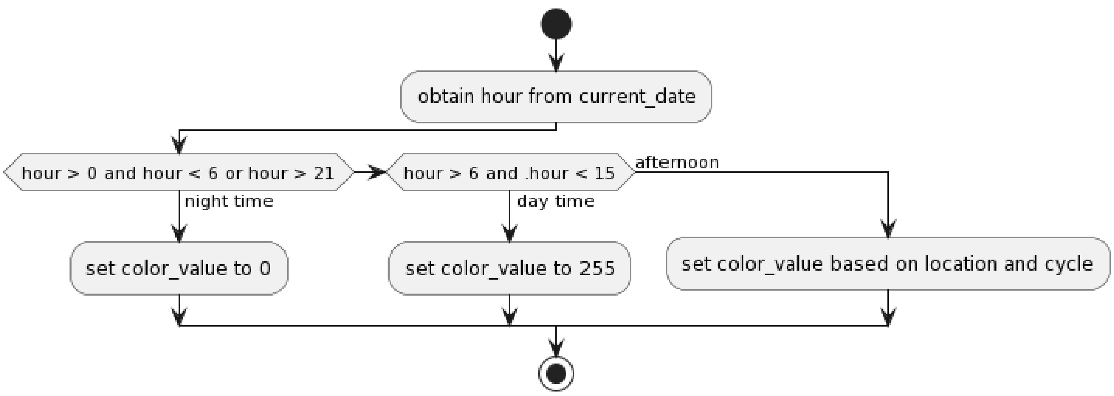

Figure 10: The difference in the number of butterflies in the extension 2.

We run a batch simulation of 16, and the mean population of each type of butterfly after 200 cycles is shown in Table 3.

Table 3: The difference in the number of butterflies in the second extension.

White butterfly	Black butterfly	Gray butterfly
147.0625	149.5625	164.6875

The figure indicates that combining predator hunting restrictions with dynamic environmental color changes results in the butterfly camouflage a smaller number of times. Butterflies are more susceptible to predation, leading to a lower butterfly population than the first extension after 200 cycles. However, this approach does yield a more balanced distribution among the different butterfly populations.

5 Extension 3

5.1 Overview

In this extension, we define a new parameter: `predator_proba_hunting_dominant` represents a mixed type of hunting. The predator will now have a high chance of hunting the most dominant-color butterfly and a low chance of hunting randomly. The UML for this behavior is represented in Fig. 11.

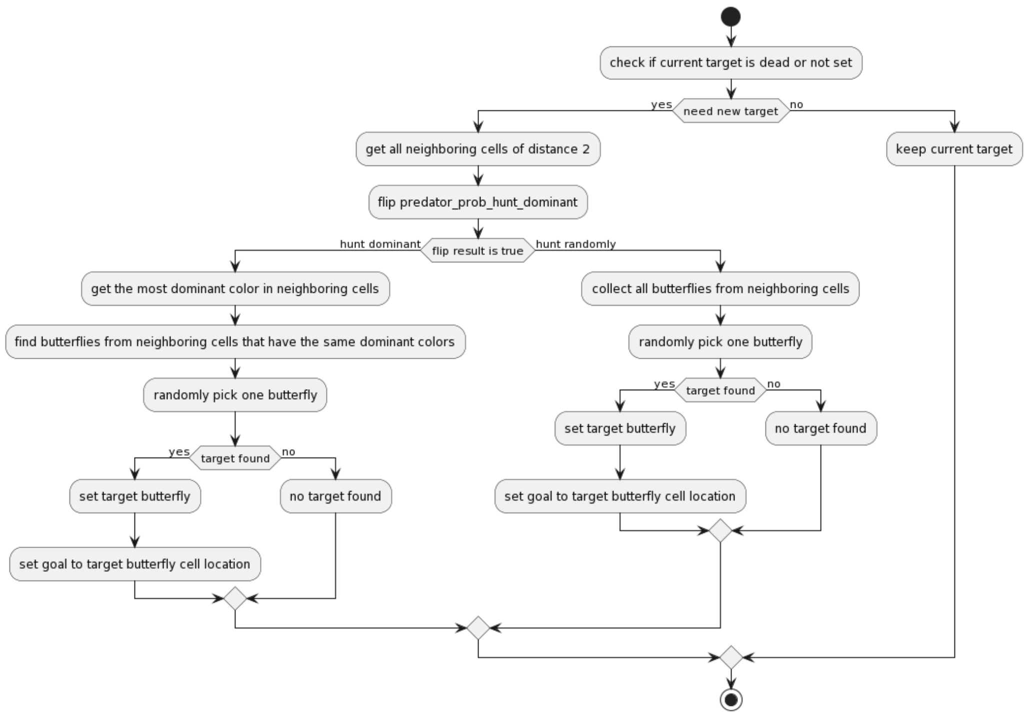

Figure 11: Activity diagram of a predator's target chosen in extension 3.

We use the Tabu Search algorithm to find the best combination of predation pressure and color transition. There are six types of color transitions and nine possible values of predator_proba_hunting_dominant: [0.1, 0.9]. We optimize the difference between the population numbers of each type of butterfly. After running the optimization, the best result we get is the mixture of color transition with predator_proba_hunting_dominant = 0.9. The results of this simulation are presented in Table 4.

Table 4: The difference in the number of butterflies in the third extension.

White butterfly	Black butterfly	Gray butterfly
148.1875	151.625	145.125

Generally, a mixture of color transitions proves to be. By optimizing two parameters, we achieve a well-balanced butterfly population.

6 Conclusion and Future Work

6.1 Conclusion

We have finished all the tasks that have been assigned to the project. We also enhanced the realistic nature of the simulation by adding distinct active times for animals, different types of hunting for the predators, and dynamic background color changes. The combination of predation pressure and a mixture of environmental color transition leads to a balanced butterfly population. Without these factors, one type of butterfly will always outnumber the others.

6.2 Future work

There are several ideas that are subjectively evaluated to be new directions for further trials and experiments. We could incorporate additional ecological factors, such as climate variations. Furthermore, expanding the genetic transmission system could provide deeper insights into the butterfly color evolution. We may use some machine learning algorithms to analyze the relationship between environmental factors and evolutionary outcomes.