KOÇ ÜNİVERSİTESİ
COMPUTER ENGINEERING

COMP132: Advanced Programming

Programming Project Report

Food Chain Through Time: Simulation Design and Development

<Arda Albayrak, 86489>

<Fall 2025>

Part 1- General Demo Information:

1.1 Register / Log-in

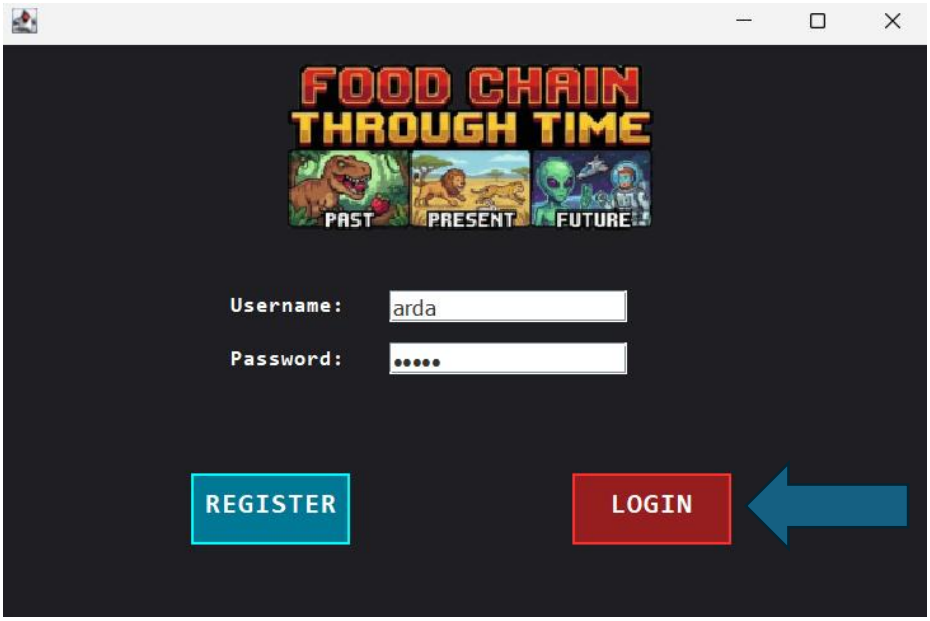
1.1.1 Pre-registered Player

Username: arda

Password: 12345

How to Log-in:

- Start the application.
- Enter previously registered username and password in the respective fields.
- Click the “LOGIN” button from the login screen.



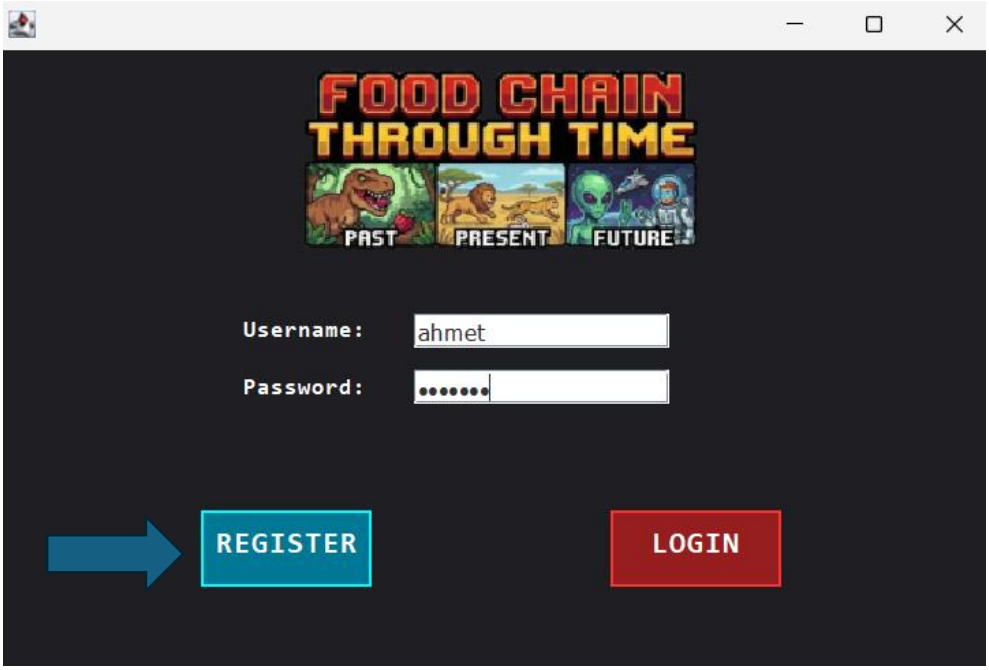
- If the username and password match a previously registered user’s datas which are recorded in the users.txt file, the game settings selection screen is going to open.
- If datas does not match, the following warning message will be displayed:

1.1.2 New Player

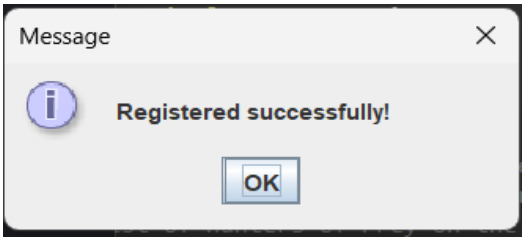
Username: Ahmet
Password: xyz1234

How to Register:

- Start the application.
- Enter the username and password in the respective fields.
- Click the “REGISTER” button in the log-in screen.



- After clicking the register button, player will see a registration message on their screen similar to the following:

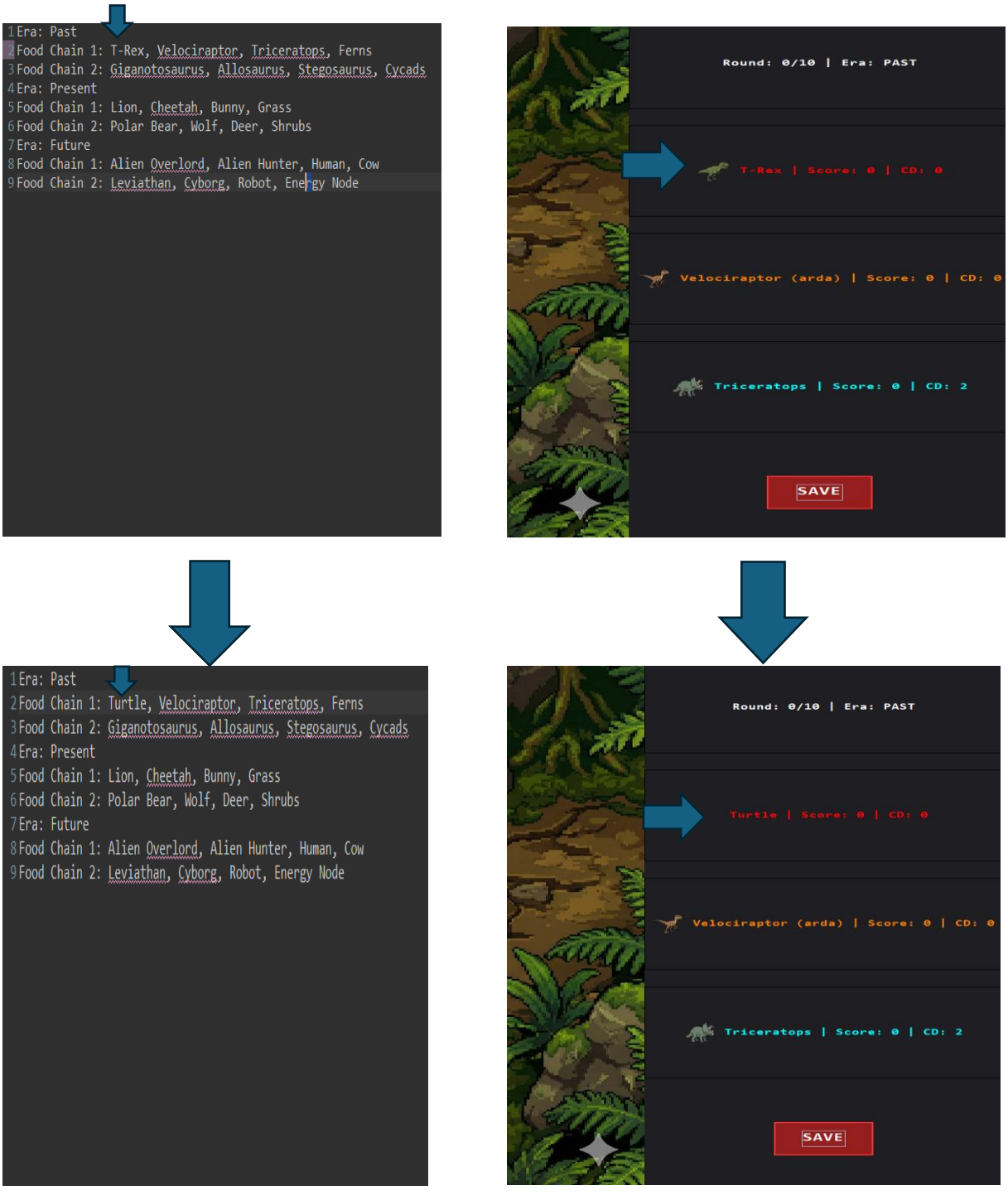


- After clicking the “OK” button on this screen, the player will be registered to the panel and can log in as described in section 1.1.1.

1.2 Configuration via “foodchain.txt”:

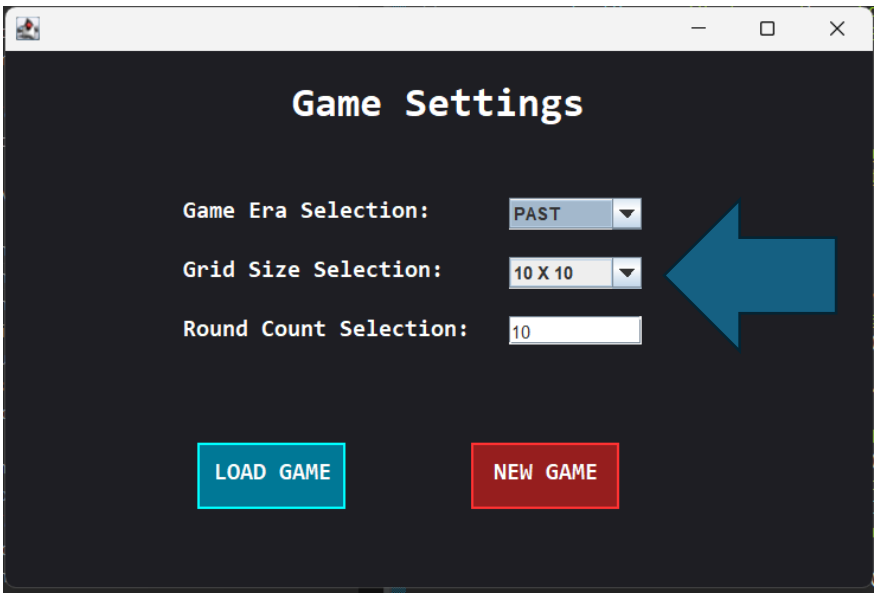
- The game utilizes a data-driven architecture where entity properties are not hardcoded into the source code but are parsed dynamically from an external file named foodchain.txt. This allows for flexibility and easy modification of game parameters without recompiling the code.

- As demonstrated below, modifications made to the text file are immediately reflected in the application. For instance, changing the entity name from 'T-rex' to 'Turtle' in the configuration file results in the game updating the relevant UI components upon the next launch.

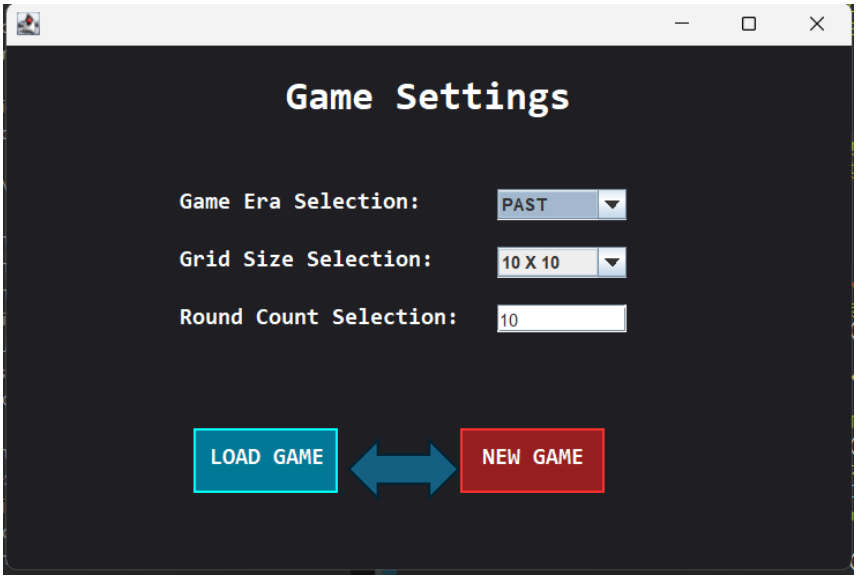


1.3 Game Settings Selection Screen:

- After the players log into the game, they will be able to select the game's era, number of rounds, and grid size on this screen.



- Possible Eras: Past, Present and Future.
- Possible Grid Size Selections: 10x10, 15x15 or 20x20.
- The number of rounds cannot be less than 10.



- The player can also start a new game by pressing the “START GAME” button on this screen.
- Additionally, if the player has saved a game in the past, they can press the “LOAD GAME” button to continue that game directly.

1.4 Gameplay and Mechanics

- In the game's main mechanics, the player controls only the Predator character, The player must both try to escape from the ApexPredator and hunt the Prey object. Also, the player can see all the squares they can move to, as they are transparently colored - according to the movement rules specified and shown in sections 1.4.1 and 1.4.2 below - and can change its position by clicking on these squares with the mouse.

- The Prey and ApexPredator objects are controlled by a computer (AI).

- Prey objects behave in a way that attempts to optimize both the distance between themselves and their closest predators, as well as the distance between themselves and their food source. They try to both eat the food and avoid their predators.

- ApexPredator objects, on the other hand, choose the Prey or Predator object that is closest to them at that moment and try to hunt it.

1.4.1 Common Basic Move of Animals:

- All animals (predators, prey, and apex predators) share the common default ability to walk, allowing them to move to any of the 8 squares around them in any round of any era.

- As it can be seen from below, all the squares the Predator (Player) can go to with this ability are indicated in transparent white.

1.4.2 Era-Dependent Special Skill Moves of Animals:

- Also, all animals possess special skills that vary according to eras and can be used with cooldown periods or other restrictions.

- The special abilities of the Predator Object controlled by the player in different eras can be seen below. (transparent yellow squares)

PAST ERA:

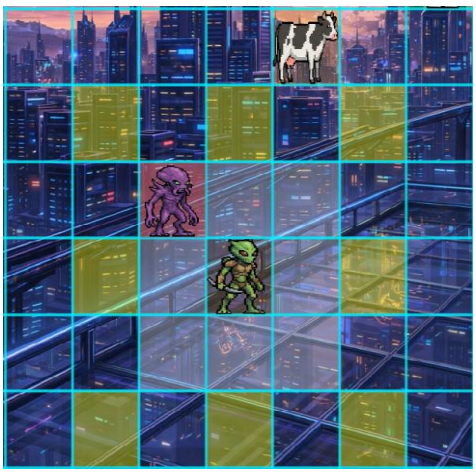
- Horizontally or vertically move 2 cells, diagonals prohibited

- Cooldown: 2 rounds

PRESENT ERA:

- If ApexPredator is adjacent to Predator (1 cell distance), move 1 cell twice in the same round

- No cooldown



FUTURE ERA:

- Move 2 cells one of the 8 directions (diagonals included)

- Cooldown: 2 rounds

- Predators and apex predators also possess special abilities as listed below:

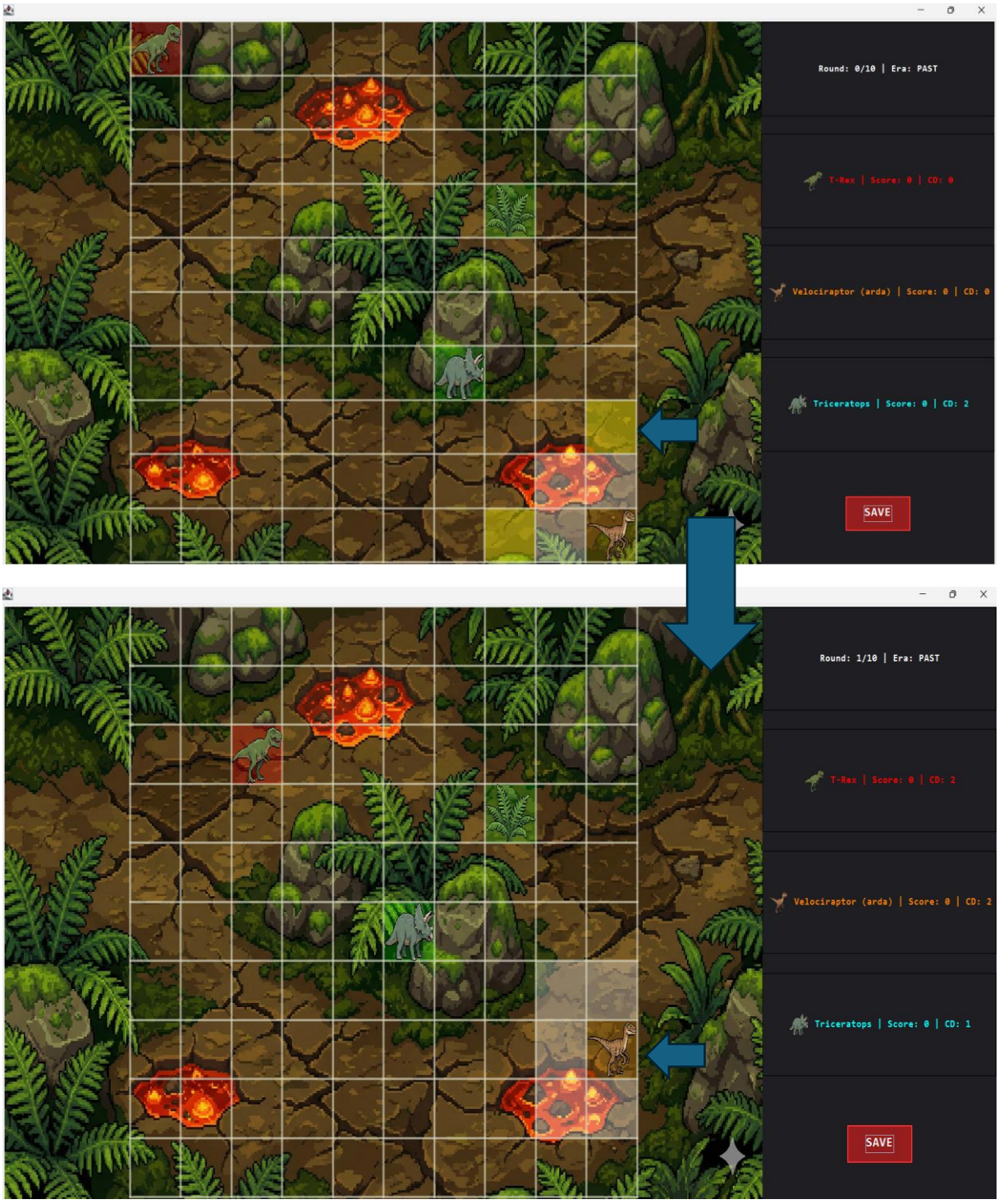
Entity	Control Type	Past Era Ability	Present Era Ability	Future Era Ability
Apex Predator	AI (Automatic)	SPRINT: Move 2 cells in a straight line (diagonals included), CD: 2	SPRINT: Move up to 3 cells in any direction (diagonals included), CD:3	SPRINT: Move to any cell within a 3-cell radius, CD: 3
Prey	AI (Automatic)	HOP: Move 1 cell, then immediately shift 1 extra cell sideways relative to the first move (like a zigzag). CD: 2	HOP: Move 2 cells immediately in any direction (diagonal included) CD: 3	HOP: Move 3 cells but cannot eat food this turn. CD:3 Cannot eat food

- NOTE: Foods do not have any move ability.

1.4.3 Movement of Player:

- The player can move by clicking on the transparently colored squares. (The special ability is present in the top photo but not in the bottom one)

- The information panel on the right is updated (cooldown)



1.4.4 Scoring and Respawning Mechanism:

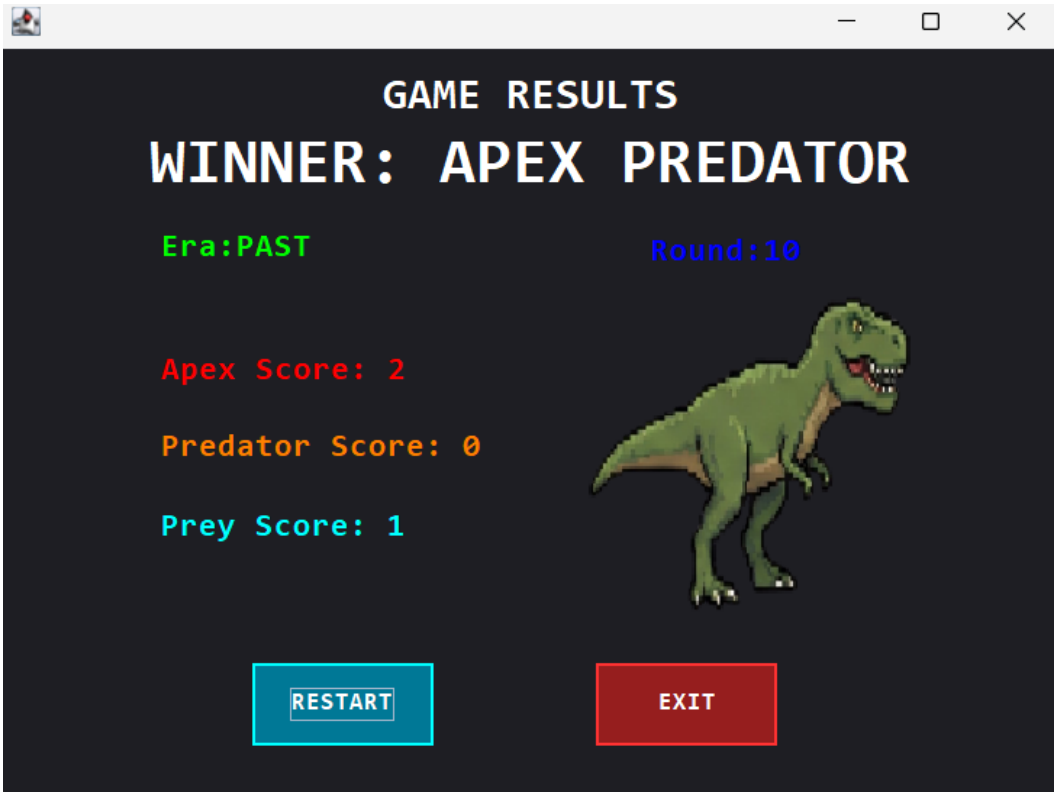
Score Changes and Respawning of the Organism:

- Prey gains +3 points when it attacks and consumes the cell containing food
- If Predator can guess the cell where Prey is located and make a move, it can consume Prey and gain +3 points.
- Apex Predator gains +1 point if it attacks and eliminates a cell containing a Prey or another Predator.
- If Apex Predators or Predator Food attack and consume a cell containing them, the score does not change and the food respawns on the map.

- In the images above, Prey ate the food and earned 3 points, and a new food item respawned on the map.

- All organisms other than the Apex Predator will respawn in a random location in map if they are hunted by their hunter and the game hasn't ended yet.

1.4.5 Game Completion



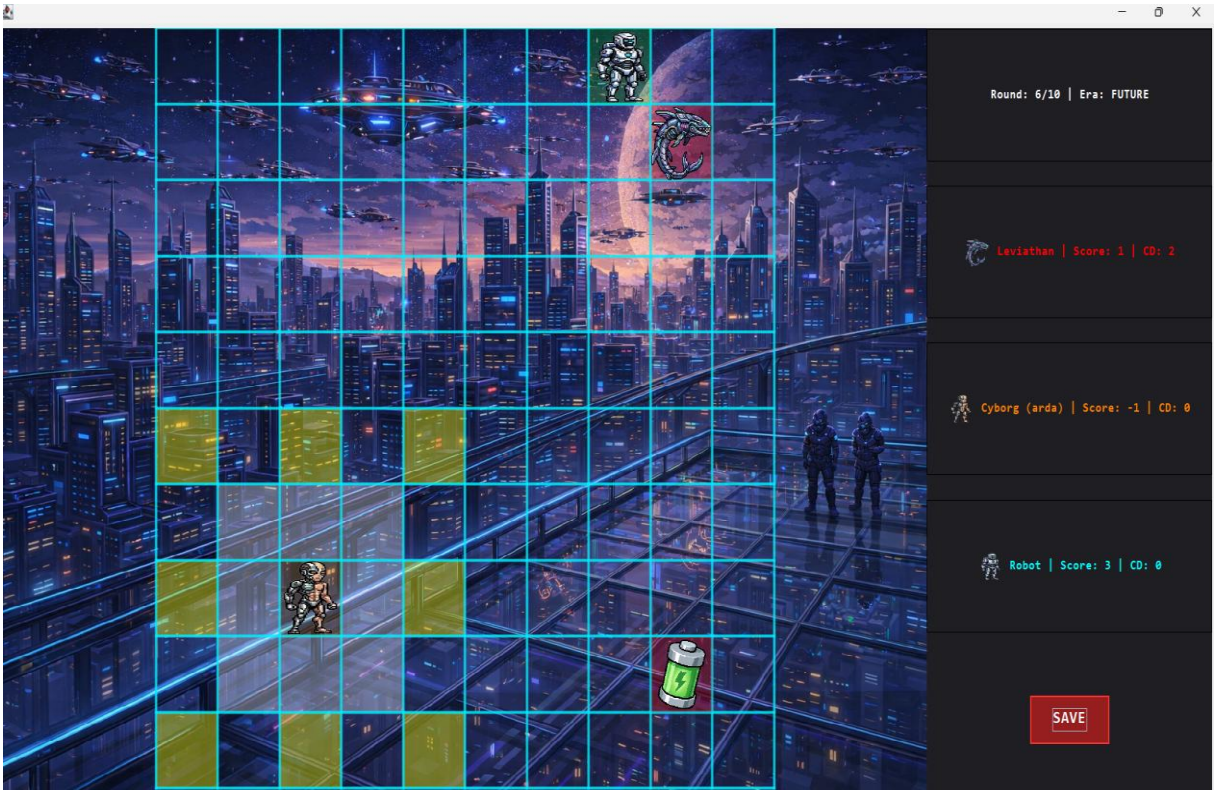
1.4.6 The log.txt File of the Game Above

```
[Oca 08 04:01:49 2026] === NEW GAME STARTED (Era: PAST) ===
[Oca 08 04:01:49 2026] Round 0: AI (PREY): Triceratops moved to (6,6)
[Oca 08 04:02:04 2026] Round 0: AI (PREY): Triceratops moved to (5,5)
[Oca 08 04:02:04 2026] Round 0: ACTION: Hunter used SPECIAL SKILL! (Era: PAST)
[Oca 08 04:02:04 2026] Round 0: PLAYER: Predator moved from (9,9) to (9,7)
[Oca 08 04:02:04 2026] Round 0: AI (APEX): T-Rex moved to (2,2)
[Oca 08 04:02:07 2026] Round 1: AI (PREY): Triceratops moved to (6,4)
[Oca 08 04:02:07 2026] Round 1: Player used basic move.
[Oca 08 04:02:07 2026] Round 1: PLAYER: Predator moved from (9,7) to (8,6)
[Oca 08 04:02:07 2026] Round 1: AI (APEX): T-Rex moved to (3,3)
[Oca 08 04:02:21 2026] Round 2: AI (PREY): Triceratops moved to (7,3)
[Oca 08 04:02:21 2026] Round 2: EVENT: Prey ATE FOOD at (7,3) | New Prey Score: 0
[Oca 08 04:02:21 2026] Round 2: Player used basic move.
[Oca 08 04:02:21 2026] Round 2: PLAYER: Predator moved from (8,6) to (7,5)
[Oca 08 04:02:21 2026] Round 2: AI (APEX): T-Rex moved to (4,3)
[Oca 08 04:11:36 2026] Round 3: AI (PREY): Triceratops moved to (6,1)
[Oca 08 04:11:36 2026] Round 3: Player used basic move.
[Oca 08 04:11:36 2026] Round 3: PLAYER: Predator moved from (7,5) to (8,4)
[Oca 08 04:11:36 2026] Round 3: AI (APEX): T-Rex moved to (6,1)
[Oca 08 04:11:36 2026] Round 3: EVENT: Apex CAUGHT Prey at (6,1)
[Oca 08 04:11:40 2026] Round 4: AI (PREY): Triceratops moved to (5,3)
[Oca 08 04:11:40 2026] Round 4: Player used basic move.
[Oca 08 04:11:40 2026] Round 4: PLAYER: Predator moved from (8,4) to (7,5)
[Oca 08 04:11:40 2026] Round 4: AI (APEX): T-Rex moved to (5,2)
[Oca 08 04:11:42 2026] Round 5: AI (PREY): Triceratops moved to (4,4)
[Oca 08 04:11:42 2026] Round 5: ACTION: Hunter used SPECIAL SKILL! (Era: PAST)
[Oca 08 04:11:42 2026] Round 5: PLAYER: Predator moved from (7,5) to (5,5)
[Oca 08 04:11:42 2026] Round 5: AI (APEX): T-Rex moved to (4,3)
[Oca 08 04:11:46 2026] Round 6: AI (PREY): Triceratops moved to (3,2)
[Oca 08 04:11:46 2026] Round 6: Player used basic move.
[Oca 08 04:11:46 2026] Round 6: PLAYER: Predator moved from (5,5) to (4,6)
[Oca 08 04:11:46 2026] Round 6: AI (APEX): T-Rex moved to (3,2)
[Oca 08 04:11:46 2026] Round 6: EVENT: Apex CAUGHT Prey at (3,2)
[Oca 08 04:11:51 2026] Round 7: AI (PREY): Triceratops moved to (9,9)
[Oca 08 04:11:51 2026] Round 7: Player used basic move.
[Oca 08 04:11:51 2026] Round 7: PLAYER: Predator moved from (4,6) to (5,6)
[Oca 08 04:11:51 2026] Round 7: AI (APEX): T-Rex moved to (5,4)
[Oca 08 04:11:52 2026] Round 8: AI (PREY): Triceratops moved to (8,9)
[Oca 08 04:11:52 2026] Round 8: Player used basic move.
[Oca 08 04:11:52 2026] Round 8: PLAYER: Predator moved from (5,6) to (6,7)
[Oca 08 04:11:52 2026] Round 8: AI (APEX): T-Rex moved to (6,5)
[Oca 08 04:11:57 2026] Round 9: AI (PREY): Triceratops moved to (9,9)
[Oca 08 04:11:57 2026] Round 9: ACTION: Hunter used SPECIAL SKILL! (Era: PAST)
[Oca 08 04:11:57 2026] Round 9: PLAYER: Predator moved from (6,7) to (8,7)
[Oca 08 04:11:57 2026] Round 9: AI (APEX): T-Rex moved to (7,6)
[Oca 08 04:11:57 2026] GAME OVER: APEX PREDATOR WINS! Final Score: 0
```

1.5 Save and Load the Game

1.5.1 Save the Game:

- Press the save button and close the game.
- The arda_save.txt file is created.

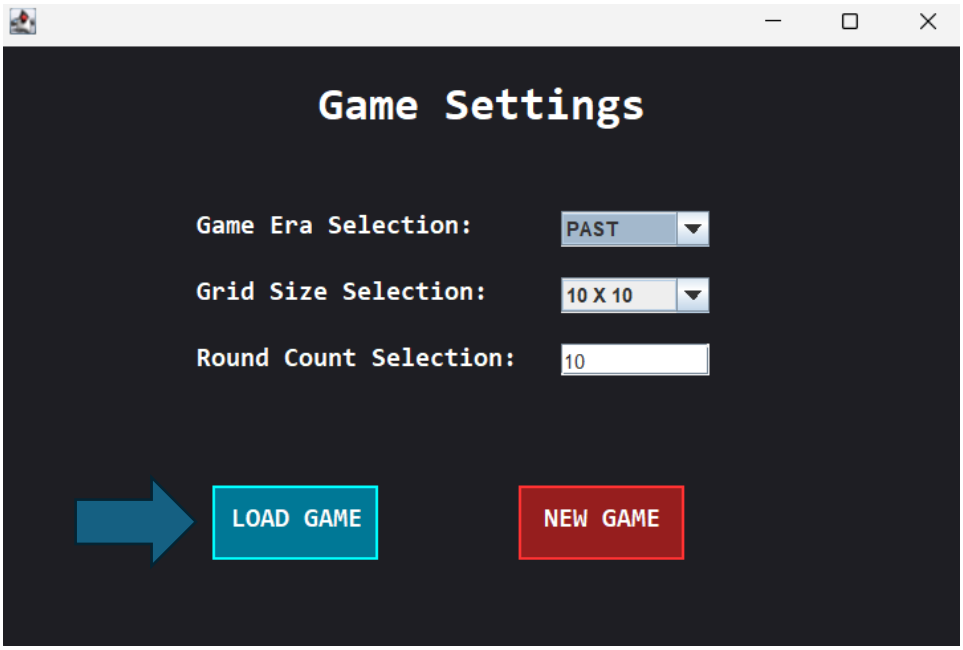


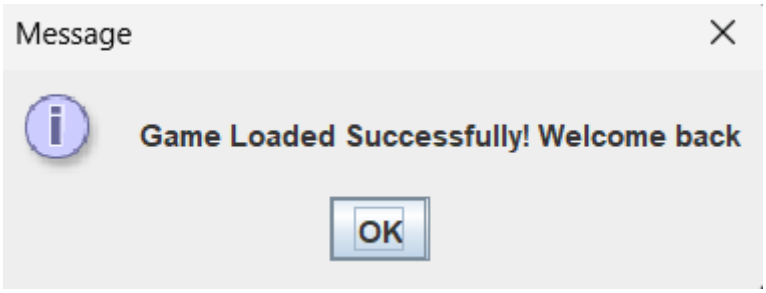
arda_save.txt file:

```
1 FUTURE,6,10,-1,1,3
2 PREDATOR,Cyborg,2,7,0
3 APEX,Leviathan,8,1,2
4 PREY,Robot,7,0,0
5 FOOD,Energy Node,8,8,0
6
```

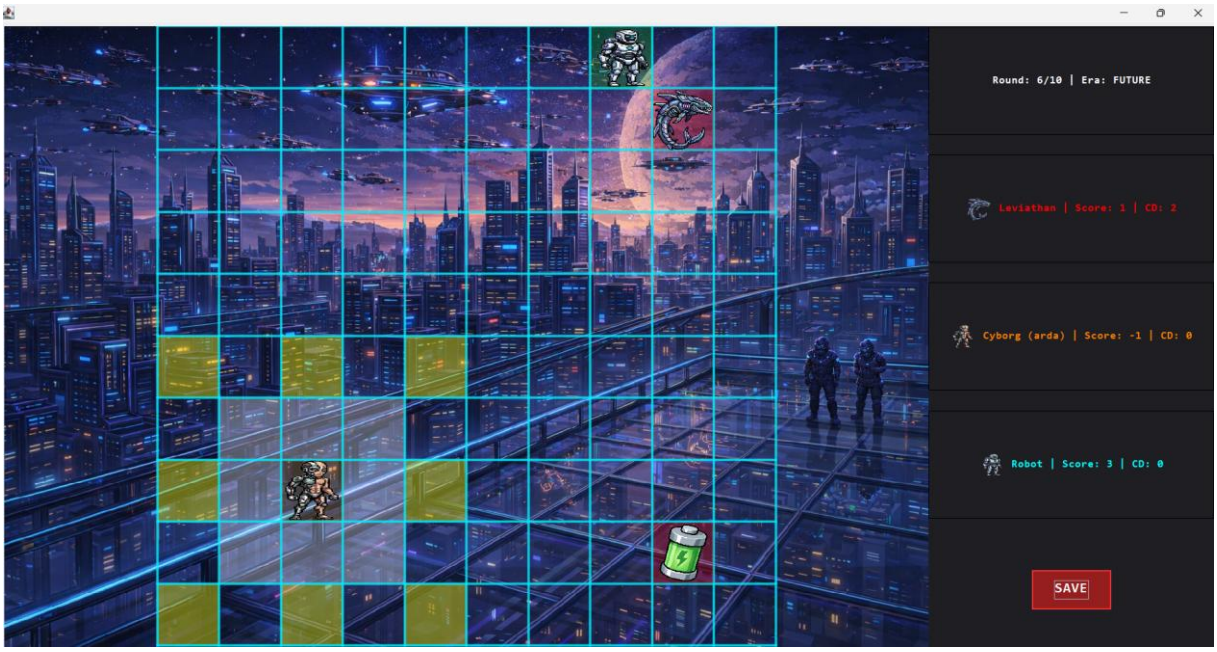
1.5.2 Load the Saved Game Above:

- Start the application again.
- Log-in the game again.
- Click the LOAD button on the game settings screen.
- A notification screen will pop up stating that the saved old game has been loaded.

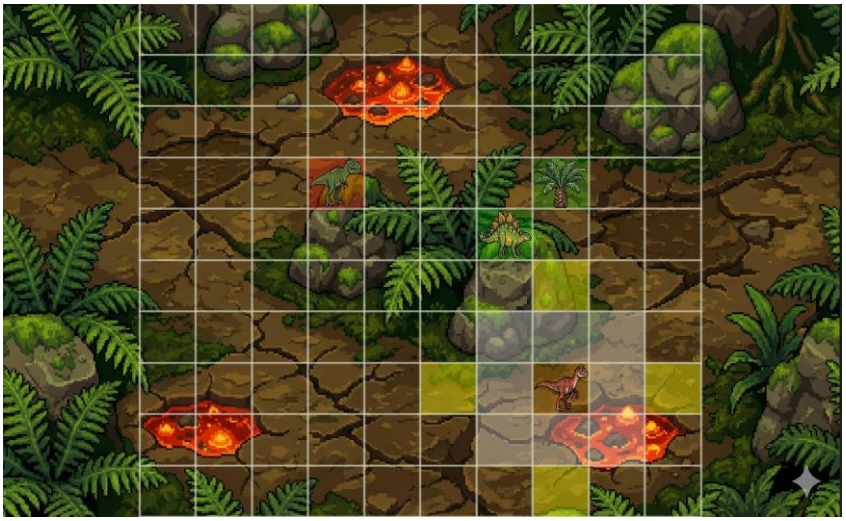


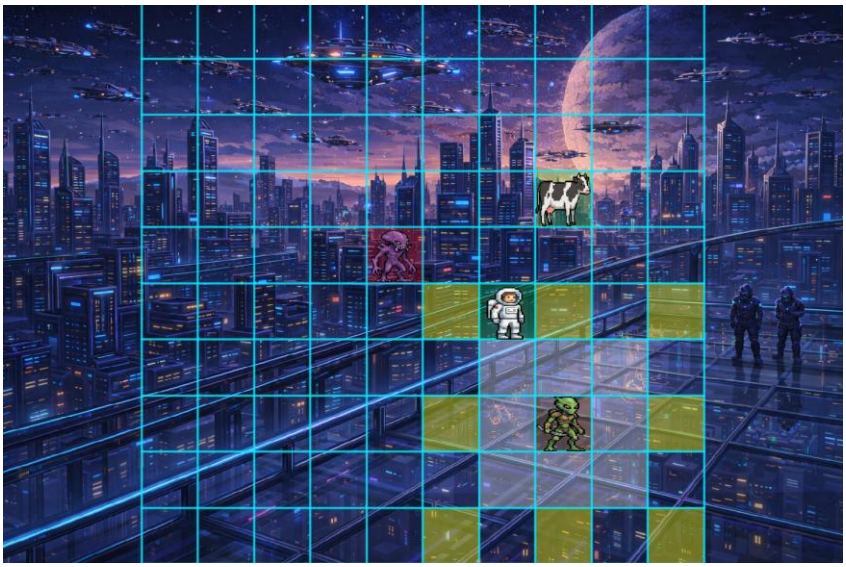


After Loaded:



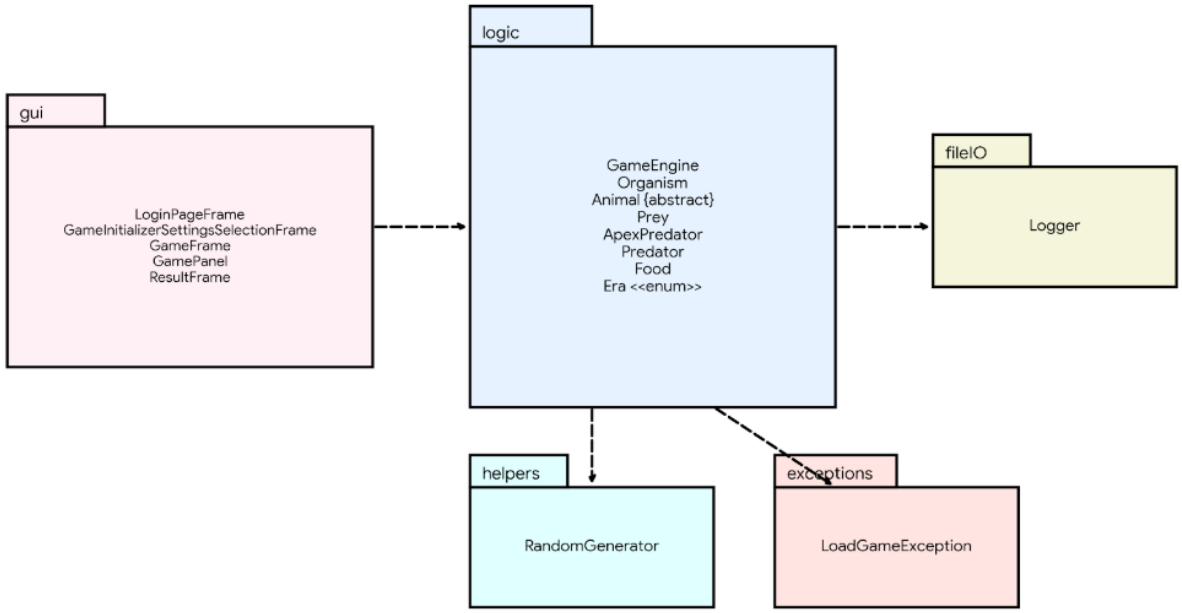
1.6 Different Eras, Icons, Special Skills





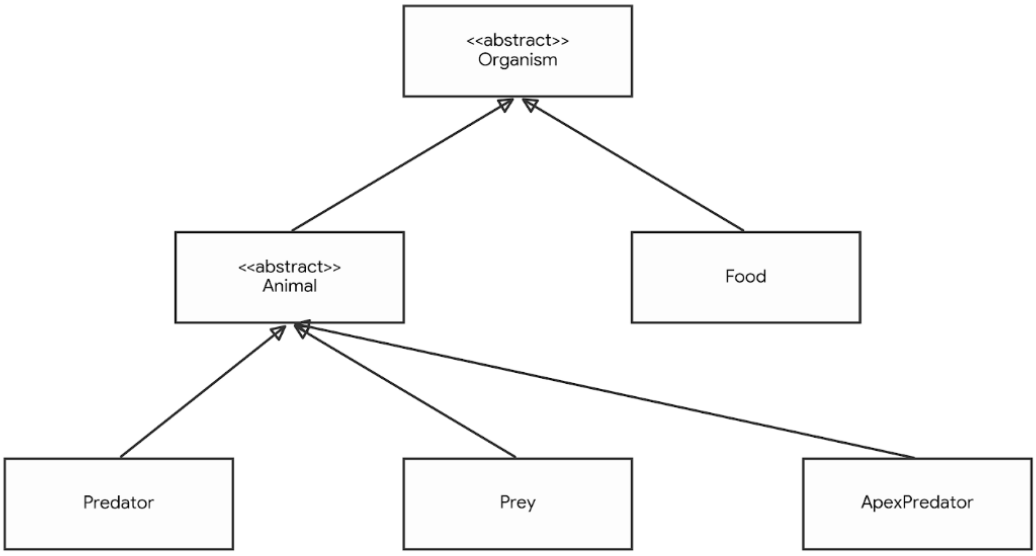
Part2: Project Design Description:

2.1 Project Structure:



- Additionally, the resources folder contains images used in the game.

2.2 Inheritance, Type Hierarchies, and Abstract Classes



- **Level 1 (Root):** Organism is the top-level abstract class, defining the fundamental properties shared by all entities on the grid, specifically their coordinates (x, y).

- **Level 2 (Intermediate):**

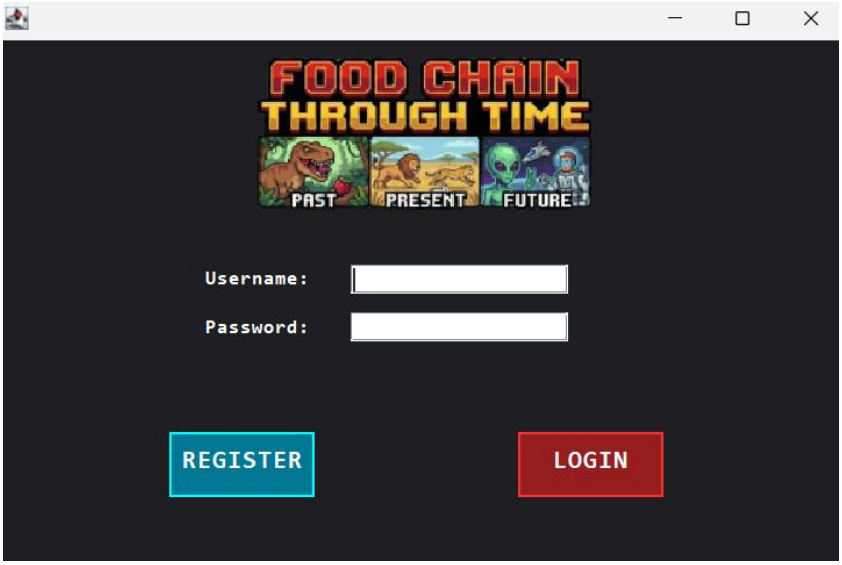
- Animal (abstract) extends Organism and adds properties specific to living and moving creatures, such as name, era, and cooldown mechanics.
- Food extends Organism directly, as it has coordinates but does not share the complex behaviors of animals.

- **Level 3 (Concrete):** Predator (Player), Prey, and ApexPredator are concrete classes extending Animal. They inherit common traits but implement their own specific movement algorithms (move, catchTarget) and unique abilities (specialSkill) through Polymorphism, overriding the base methods.

2.3 GUI Components:

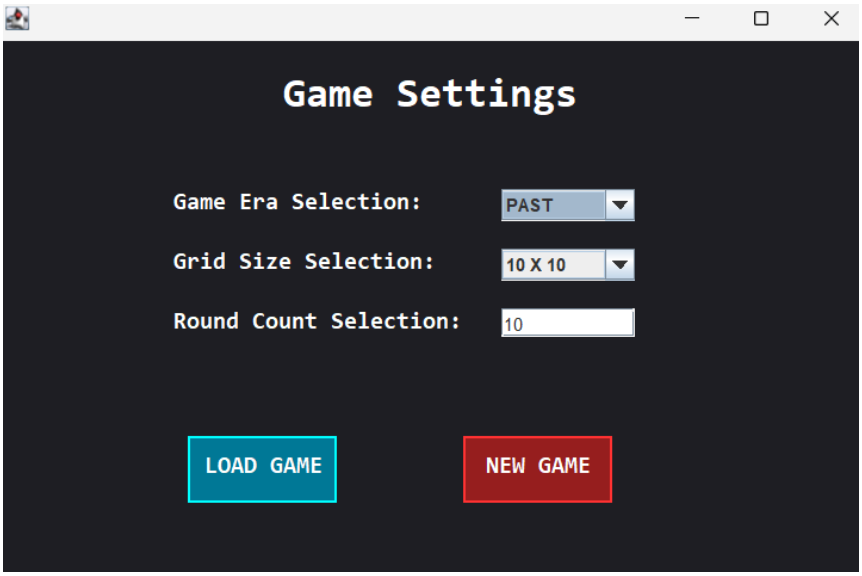
LoginPageFrame:

On this screen, players can register with the system using the "REGISTER" button and log in using the "LOGIN" button.



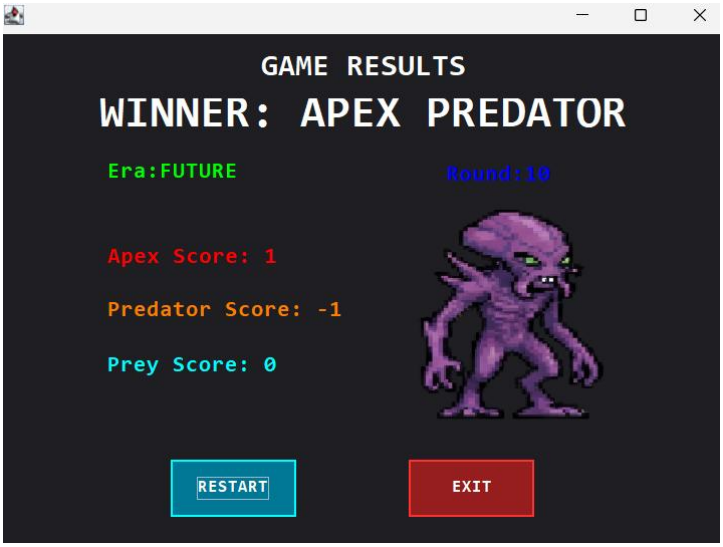
GameInitializerSettingsSelectionFrame:

On this screen, players can adjust pre-game settings, setting the era, gridsize, and number of rounds. After making adjustments, they can use the START GAME button to start a new game, or press the LOAD button to reload a previously saved game.



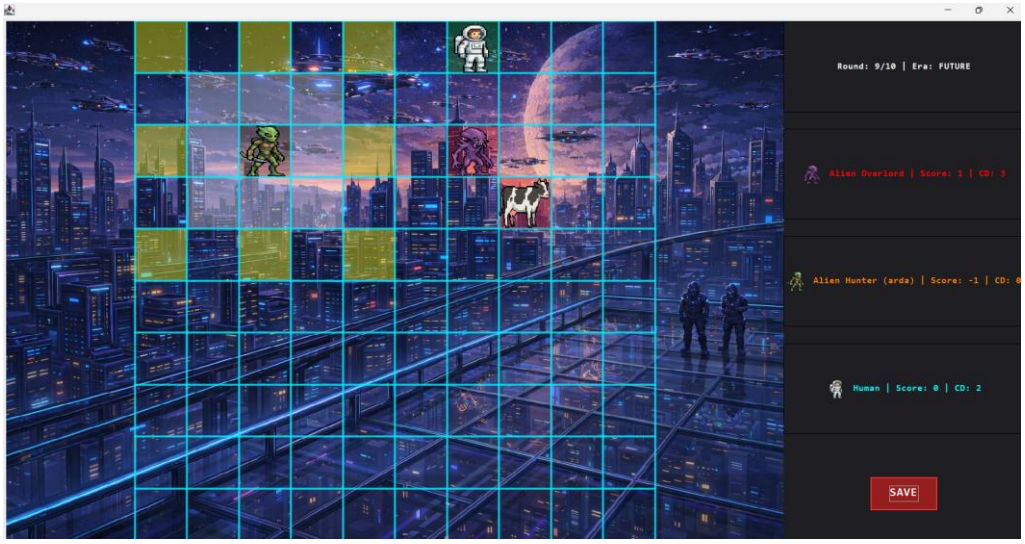
ResultFrame:

This screen opens when the game ends and announces the winner, shows the scores of all animals, and indicates era and round information.



GameFrame:

It is the large screen on which the game is played. It also has an information panel that displays information such as score, round, era, and cooldown.



GamePanel:

It is the class that draws the playing area of the game and detects clicks made on that area. This panel is added to the GameFrame screen and creates the area where the game is played.

2.4 File Processing:

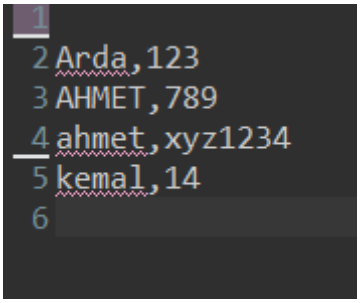
2.4.1 Logger Class:

- The Logger class is a dedicated utility within the fileIO package designed to record every event occurring during the simulation. It ensures the transparency and verifiability of the game logic.

- File Writing Mechanism: The class utilizes BufferedWriter and FileWriter in "append mode." This allows the application to continuously add new logs to the log.txt file without overwriting previous turns' data.

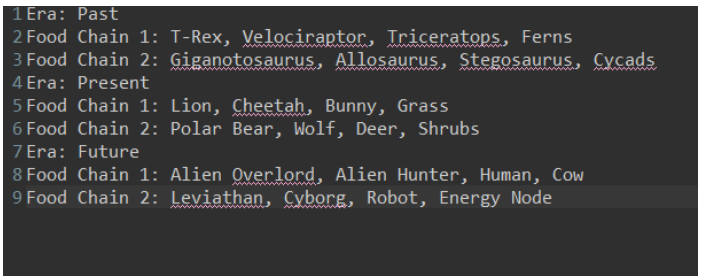
2.4.2 users.txt:

A text file containing the usernames and passwords of players registered in the system.



2.4.3 foodchain.txt:

This text file contains the names of the animals owned by each era, and at the start of the game, a random animal set is selected from this file to begin the game.



2.4.4 username_save.txt:

This is the file where the current game is saved when the player presses the save button and saves the game.

2.5 Implementation Details:

- Animals have the ability to move according to different rules depending on the era. The algorithms associated with these abilities are classified as ApexPredator, Predator (Player), and Prey.

- ApexPredator's algorithm detects the nearest prey or predator at any given moment and approaches it to catch it.

- In Prey's algorithm, while trying to stay away from predators and apex predators, it also tries to optimize the situation by getting closer to the food source

- In these algorithms, AI characters (Prey and ApexPredator) decide which square to go to by evaluating the scores of the surrounding squares and moving towards the square with the highest score according to their own criteria.

- Also, in these algorithms, the distance between animals is calculated using the Manhattan Distance formula. (abs(x1-x2) + abs(y1-y2)).

2.6 LoadGameException:

Custom exception class. When loading a saved game using the loadGame method in the GameEngine class, if the array created by splitting the values in the username_save.txt file with "," is not 5 in length, this exception will be thrown.

References

1. Oracle Java Documentation, "The Java Tutorials: Trail: Creating a GUI With JFC/Swing". [Online]. Available: https://docs.oracle.com/javase/tutorial/uiswing/
2. Deitel, P., & Deitel, H. (2017). "Java How to Program, Early Objects". Pearson Education.