

Virtual Pet Game

version 1.0, 2024-12-07

produced by Yan-Bo Chen

Overview

In today's fast-paced world, many people desire to have pets but struggle with the time commitment required for pet care. This virtual pet game offers a lightweight way to experience the joy of pet ownership.

Developed in Java using MVC architecture, the game simulates real pet care through real-time state monitoring and interaction mechanisms.

Players need to attend to various pet needs (hunger, cleanliness, tiredness, entertainment) and take appropriate care measures, fostering both entertainment and a sense of responsibility. Have fun !!!

Features

1. State Monitoring System

- Real-time health monitoring (0-100)
- Four basic need indicators (0-10): Hunger, Cleanliness, Tiredness, Boredom
- Automatic state degradation system

2. Interactive Functions

- Feed: Reduces hunger
- Clean: Improves cleanliness
- Play: Reduces boredom
- Rest: Reduces tiredness

3. Special Mechanics

- Sleep system (automatic/manual wake-up)
- Automatic health recovery (condition: no critical states)

- Multi-state priority management
- Pet revival functionality

Running the Application

1. Ensure Java Runtime Environment (JRE) 11 or higher is installed
2. Download VirtualPet.jar file from the PetModelRedo/ res/ directory
3. Open terminal and navigate to the jar file directory
4. Execute command:

```
java -jar VirtualPet.jar
```

- No additional parameters needed
- GUI will launch automatically

Usage Instructions

1. Basic Operations

- Click buttons to perform corresponding actions
- Monitor status bar for current pet needs
- Check message area for real-time feedback

2. State Management

- Normal state: All indicators below 10
- Critical state: Any indicator reaches 10
- State Priority: Based on weight system (Dirty > Tired > Hungry > Bored)
- Death condition: Health drops to 0

3. Special Features

- Sleep mode:
 - Entry condition:
 - Available in tired state score value > 0 and
 - there are no critical states of hunger, cleanliness, or boredom
 - Auto wake-up: After 1 minute
 - Manual wake-up: Click "Wake Up" button
 - No actions allowed during sleep mode except wake-up

- When pet is in sleep mode, health would recover through time and other states score would not change
- Revival function:
 - Trigger condition: After pet death
 - Operation: Click "New Pet" button

4. Game End

- Game stop when pet health reaches 0, you can start a new game by clicking "New Pet" button or
- Click the X button in the top-left corner of the Virtual Pet window to close the application

Implementation References

Key implementation details can be found in the following files:

1. State Management:

- Pet.java:94-112:updateState()
- PetState.java:1-25:enum PetState
- PetStateTest.java:1-224

2. Action Handling:

- PetController.java:37-58:handleAction()
- Pet.java:94-102:performAction()

3. Time Management:

- TimeManager.java:45:startScheduler()
- TimeManager.java:53:restart()
- TimeManager.java:105:notifyStateChange(String)

4. Display Logic:

- PetView.java:59:initializeUserInterface()
- PetView.java:160:updatePetIcon(String)
- PetView.java:188:updateState(String)
- PetView.java:197:appendMessage(String)

- PetView.java:280:updateStateScore(PetState, int)

5. Controller Logic:

- PetController.java:50-79:handleAction()
- PetController.java:130:updateView()

Assumptions

1. Time Settings

- Base time unit: 1 second

2. State Mechanics

- State update frequency:
 - Hunger: +3 points every 5 seconds
 - Cleanliness: +5 points every 15 seconds
 - Tiredness: +4 points every 15 seconds
 - Boredom: +2 points every 10 seconds
- Updates start when game begins and pass each state's interval
- States update at fixed intervals
- Multiple states can change simultaneously
 - State scores are capped at 10 (critical state), and if there are more than one critical states,
 - The pet would display the state with the highest default weight
 - And the GUI view message will demonstrate the state with the highest default weight every time the state changes
 - And the message also encourage the player to take care of the pet
- Health decreases when any state reaches 10
- the pet can not go to sleep when there are critical states of hunger, cleanliness, or boredom
- the pet can not perform any actions when it is in sleep mode except wake up
- Happy and Sleeping is two special state, so they are not put in enum.
 - Happy state: only show up 2 seconds when the pet is fed, cleaned, or played, not is rest
 - Sleeping state: only show up when the pet is in sleep mode

- Sleep state:
 - can only be entered when tired state score value > 0 and no critical states
 - auto wake-up after 1 minute
 - manual wake-up available
 - all the other state scores would not change during sleep mode, but the health would keep recover through time

3. Health Mechanics

- Initial value: 100
- Health capped at 0 to 100, when health reaches 0, the pet would die (dead state)
- Critical state decrease rate: -2 per second
 - no matter how many critical states, it would only decrease 2 per second
- Recovery rate: +5 per second (no critical states)

Limitations

1. Functional Limitations

- States are managed by a priority system based on default setting weights
- Actions can be performed as long as the state score is not 0
- No actions allowed during sleep state except wake up
- Sleep mode is only available when the pet has tired state score value > 0 and no other critical states
- New pet creation can only be implemented and the only allowable action when the pet is dead
- When the Virtual Pet window is closed, the game stops and the application terminates, the game cannot be resumed
- And we don't have a pause function, so the game can not be paused

2. Technical Limitations

- Single pet support only
- Sometimes the time interval between the state changes is not accurate because of the method of time management
- No save other functionality except above-mentioned
- Very basic pet care simulation UI
- The score value for each state increases in a segmented manner rather than changing continuously

- No custom pet appearance options
- No custom state weight settings
- No custom state update intervals
- No custom health update rates
- No custom sleep duration

Citations

1. Image Resources:

- Black shiba inu icons created by Chanut-is-Industries - Flaticon
- Source: [Black shiba inu icons created by Chanut-is-Industries - Flaticon](#)

2. Development:

This project is an original design, inspired by Professor Lino and Teaching Assistant David. All functionalities and mechanisms are independently developed without external references except for the pet images mentioned above.

JAR File Instructions

1. Building the JAR

- The project is developed with JDK 11
- Compile and build with Java 11 compatibility:
 - Set source compatibility to Java 11
 - Set target compatibility to Java 11
- Include all required resources in the JAR:
 - All class files
 - Images directory and its contents
 - Any configuration files
- Set the main class in the manifest file:

```
Main-Class: main.Main
```

2. Running the JAR

- Minimum requirement: Java Runtime Environment (JRE) 11

- Recommended: Latest JRE 11.x version
- Command to run: `java -jar VirtualPet.jar`
- The GUI will launch automatically
- No command-line arguments needed
- Click the X button in the top-left corner of the Virtual Pet window to close the application

3. Development Environment

- JDK 11 is used for development
- All dependencies are included in the JAR
- No external runtime dependencies required