Gold Miner

Team members

- Jingjing Pan
- Yilin Li

Motivation

Gold Miner is a nostalgic and widely recognized game, and possibly many of us played it in our childhood. We plan to develop this game in Java using Swing and focus on applying event-driven logic to handle animations, physics simulations, and score tracking. By developing this game in Java using Swing, we aim to:

- **Strengthen our Java programming skills:** This project will allow us to practice core Java concepts, including object-oriented programming, event-driven programming, and exception handling.
- **Apply object-oriented design principles:** We will design the game using modular and reusable components, ensuring clean and maintainable code.
- **Gain experience in game development:** Building a game involves unique challenges, such as handling animations, physics simulations, and real-time user input, which will broaden our skill set.
- **Create a fun and interactive application:** At the end of the day, we want to build something that is not only technically sound but also enjoyable to play.

This project will serve as a culmination of our learning in CS5004, allowing us to apply theoretical knowledge to a practical, real-world application. We are excited to share that **Prof. Mark has approved our proposal, noting that it is a good and feasible project.**

Project Structure

- **Controller**
  - `GameController`: Manages game flow, state transitions, and player input.
  - `InputController`: Handles keyboard and mouse inputs separately for better code clarity.
- **Model**
  - `GameModel`: Manages game data, scores, and countdown timers.
  - `GoldMiner`: Represents the player character with attributes like `hookLength`, `score`, and `grabSpeed`.
  - `Item` (Abstract Class): Serves as the base class for game objects such as gold and rocks.
  - `Gold` and `Rock`: Concrete classes that extend `Item`.
- **View**
  - `StartPanel`: Displays the start menu.
  - `GamePanel`: Displays the core gameplay area.

  - `EndPanel`: Displays the game over screen.

- **Utils**
  - `Constants`: Defines game constants such as screen width and game speed.
  - `Resource`: Manages images.

- **Main Class**
  - `Main.java`: The entry point for starting the game.

Technical Challenges

While developing Gold Miner, we anticipate several technical challenges, including:

- **Physics Simulation:** Simulating the hook's swinging motion and collision detection with items will require careful implementation of basic physics principles.
- **Animation and Rendering:** Smoothly animating the hook and items while maintaining performance will be a key focus. JavaFX's animation APIs will be leveraged for this purpose.
- **Event Handling:** Managing real-time user input (e.g., launching the hook, pausing the game) while ensuring the game remains responsive.
- **Score and Time Management:** Implementing a scoring system that dynamically updates based on the player's actions and a countdown timer that ends the game when it reaches zero.

Expected Outcomes

By the end of this project, we expect to deliver a fully functional Gold Miner game with the following features:

- A start menu, gameplay screen, and game over screen.
- Smooth animations for the hook and items.
- Real-time score tracking and countdown timer.
- Collision detection and scoring logic.
- A polished user interface with clear instructions and feedback.

Additionally, we aim to produce clean, well-documented code that adheres to object-oriented design principles and can be easily extended in the future.

Timeline

To ensure timely completion of the project, we have outlined the following timeline:

- **Week 1**(Mar 17 - Mar 21): Set up the project structure, initialize the Swing application, and implement the basic UI components (start menu, game panel, end panel).
- **Week 2**(Mar 24 - Mar 28): Implement the core gameplay logic, including hook movement, item generation, and collision detection.
- **Week 3**(Mar 31 - Apr 4): Add scoring, timer, and game-over logic. Polish animations and user feedback.
- **Week 4**(Apr 7 - Apr 11): Conduct testing, fix bugs, and finalize the project. Prepare documentation and a demo for submission.

Division of Tasks

Our team consists of two members, and the tasks are divided as follows:

Jingjing Pan:

- Responsible for the development of the **View** component, including:
  - Implementing `StartPanel`, `GamePanel`, and `EndPanel` to design the game interface.
  - Using Swing to achieve smooth animations (e.g., hook swinging and item movement).
- Responsible for the development of the **Controller** component, particularly `InputController`, to handle keyboard and mouse input.
- Managing the **Utils** component, including `Constants` and `Resource`, to ensure proper loading and management of game resources.

YiLin Li:

- Responsible for the development of the Model component, including:
  - Implementing `GameModel` to manage game data (e.g., score, countdown, item states).
  - Implementing `GoldMiner` and `Item` classes to define the player character and game item attributes.
  - Implementing collision detection and scoring logic.
- Assisting with the development of the Controller component, particularly `GameController`, to manage game flow.

Conclusion

Gold Miner is an exciting project that combines nostalgia with technical challenges. By building this game, we will not only create a fun and interactive application but also deepen our understanding of Java programming, object-oriented design, and game development. We look forward to sharing our progress and final product with the class!