Snake

Team - HHC

Lixu He - masonhe@bu.edu

Yucheng He - hyc1011@bu.edu

Li Cao - licao@bu.edu

Motivation and Project Idea

- Our team plans to create a relaxed and enjoyable Snake game.The project aims to develop a snake game using Qt graphics library and Linux framebuffer on a BeagleBone, providing an interactive and enjoyable gaming experience for users who are looking for a classic snake game on a Linux-based embedded system with an LCD touch screen.
- Snake Eating is a classic puzzle game that is suitable for children to play and is beneficial for their physical and mental health.
- Bulletpoint List:

  Develop a snake game using C++ and Qt graphics library on BeagleBone.

  Utilize the Linux framebuffer for graphics rendering on the LCD touch screen.

  Implement touch screen input for controlling the snake's movement.

  Create game logic for scoring, collision detection, and game over conditions.

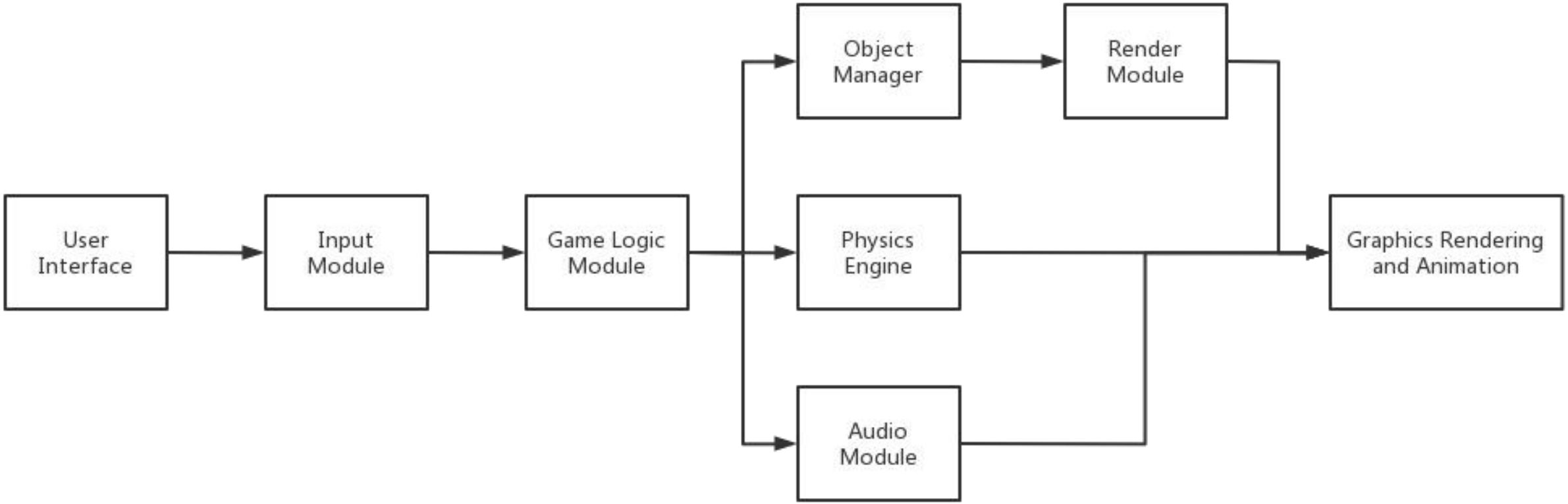
  Add game features such as different levels, power-ups, and high score tracking.

Project Description and Equipment

- **Summary:**
- In the game, players control a long straight line (commonly known as a snake or worm), which constantly moves forward. Players can only control the direction of the snake's head (up, down, left, or right) and collect objects (also known as "beans") along the way while avoiding touching themselves or other obstacles. Every time the snake eats a piece of food, its body grows longer. Eating more food gradually increases the speed of movement for the snake and makes the game more challenging.

- **Equipment needs beyond the provided kits**: N/A

System Architecture



- **User Interface:** Handles user input and output through screen.
- **Input Module:** Receives input from the user interface module and converts it into a format that the game logic module can use.
- **Game Logic Module:** Implements the rules and mechanics of the game, manages the game state, and coordinates the other modules.
- **Object Manager:** Manages the game objects and their properties, including their position, size, and state.
- **Physics Engine:** Simulates the physical behavior of the game objects, including collision detection, gravity, and motion.
- **Render Module:** Renders the game graphics and displays them on the screen.
- **Audio Module:** Plays sounds and music during the game.
- **Graphics Rendering and Animation:** Implements the low-level graphics rendering and animation functions.

Milestones, timeline, and breakdown of tasks

- Milestone 1: Basic game window setup and game board display

  Tasked to: Graphics developer

  Timeline: 2 days
- Milestone 2: Snake movement and collision detection

  Tasked to: Gameplay developer

  Timeline: 3 days
- Milestone 3: Adding food and eating functionality

  Tasked to: Gameplay developer

  Timeline: 3 days

Milestones, timeline, and breakdown of tasks

- Milestone 4: Game over and score display

  Tasked to: Graphics developer

  Timeline: 1 day
- Optional milestone: Adding sound effects and/or high scores

  Tasked to: Graphics or gameplay developer (depending on the feature)

  Timeline: 2-3 days (depending on the feature)
- Overall timeline: 10-12 days (not including weekends or other breaks)