The super-mario game

Description

This is a simple game of super mario, where the player has to collect coins and avoid enemies.
The player can move left and right, jump, and grow to big mario. The player dies when hit by an enemy, fall into a hole, or time is up.
The goal of the game is to reach the flag at the end of the level.

Design decisions

Problem Description

Our task is to attempt to highly restore Super Mario and to make the level arrangement and modification more flexible based on it. This requires us to fully analyze the game development details of the original Super Mario and add our own understanding and ideas to it.

Assumption

As a constraint on task completion, we only need to implement some base maps and some item and effect functions. It is assumed that we can successfully implement similar functions.

Constraints

- Time constraint:

  The Super Mario project started on May 15, 2023, and is expected to be submitted by June 4, 2023. There is about three weeks to complete the project development and improvement during this period.

- Business constraint:

  The integrity of the overall game experience needs to be guaranteed, so the game clearance should be prioritized. Special effects and props can be added appropriately according to time and task volume. For example, only two Mario forms are developed, and more attention is paid to the interaction and display details between the screen, characters, and players.

Engine Descion

1. Litiengine
   - Advantages

   Depends on a pure Java library, which keeps the number of external dependencies to an absolute minimum, ensuring that the library is clean and lightweight, making it easier to understand and use.

   - Disadvantages

   The engine documentation is lacking, which is not conducive to designers' learning and understanding, and may encounter difficulties in game design in the future.

   The API usage experience is poor, and some functions in the Super Mario game may be cumbersome and complex when calling its API, which will increase the project design time.

2. Libgdx
   - Advantages

   Strong compatibility and high efficiency can save a lot of time for the project.

   Clear architecture and quite a lot of development tools make it easy for us to understand and get started.Libgdx's encapsulation of the physics engine is also great.

   - Disadvantages

   For some developers, the learning curve is steep.
   Support for iOS is based on RoboVM, and RoboVM itself is still in the development stage, making debugging difficult.

Decision

In the long run, alternative 2 is the better choice. We have team members with strong learning abilities, and we currently only plan to complete the game design on Windows. For other operating systems, we have not yet considered them, so iOS is not very necessary for us.

How to run the game

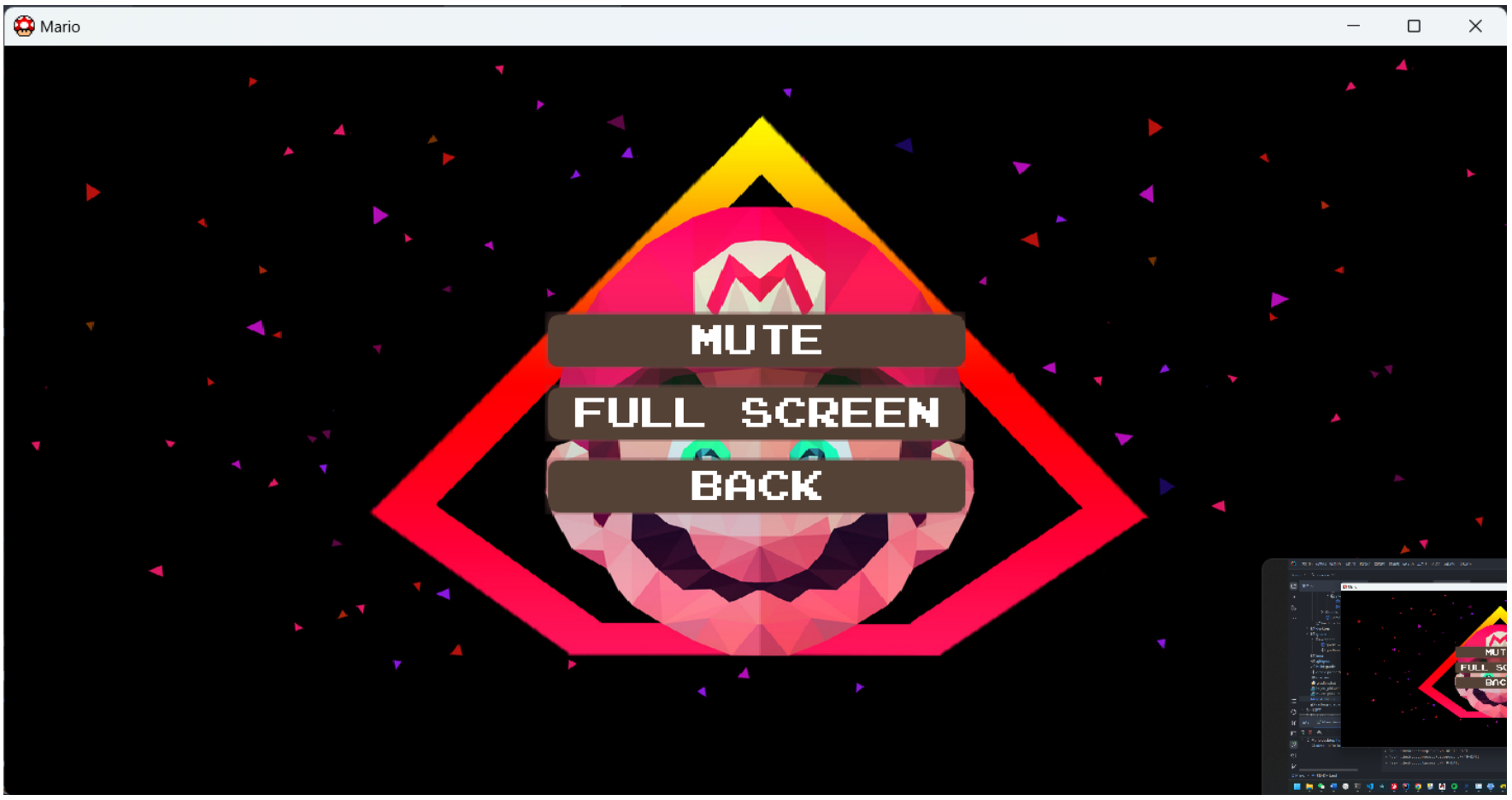
Requirements

- Java 17 or higher
- Gradle 7.5.1 or higher

```
# when get into the project folder
cd Mario

# compile the project
# in windows
./gradlew.bat desktop:dist

# in linux and mac os
./gradlew desktop:dist

# run the game
java -jar .\desktop\build\libs\desktop-1.0.jar
```

features

- [x] Audio
  - [x] Background music
  - [x] Sound effects
- [x] A settings menu with the following options:
  - [x] Change the audio mode
  - [x] Change the screen mode (full screen or not)



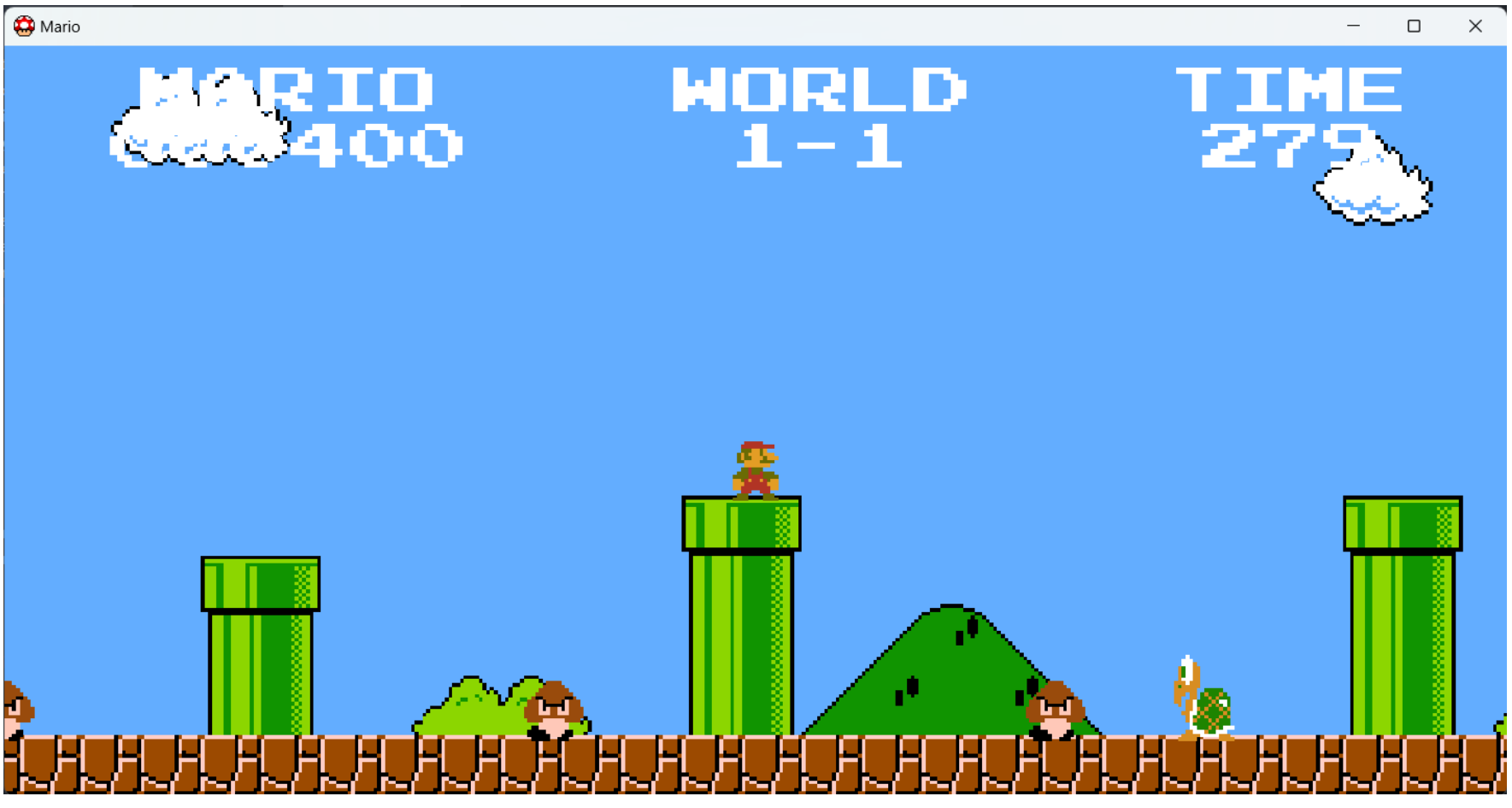
- [x] An instructions menu with the following options:
  - [x] How to play the game
  - [x] Use mouse to navigate

- [x] A level selection menu with the following options:
- [x] Select the level to play
- [x] Different map for each level

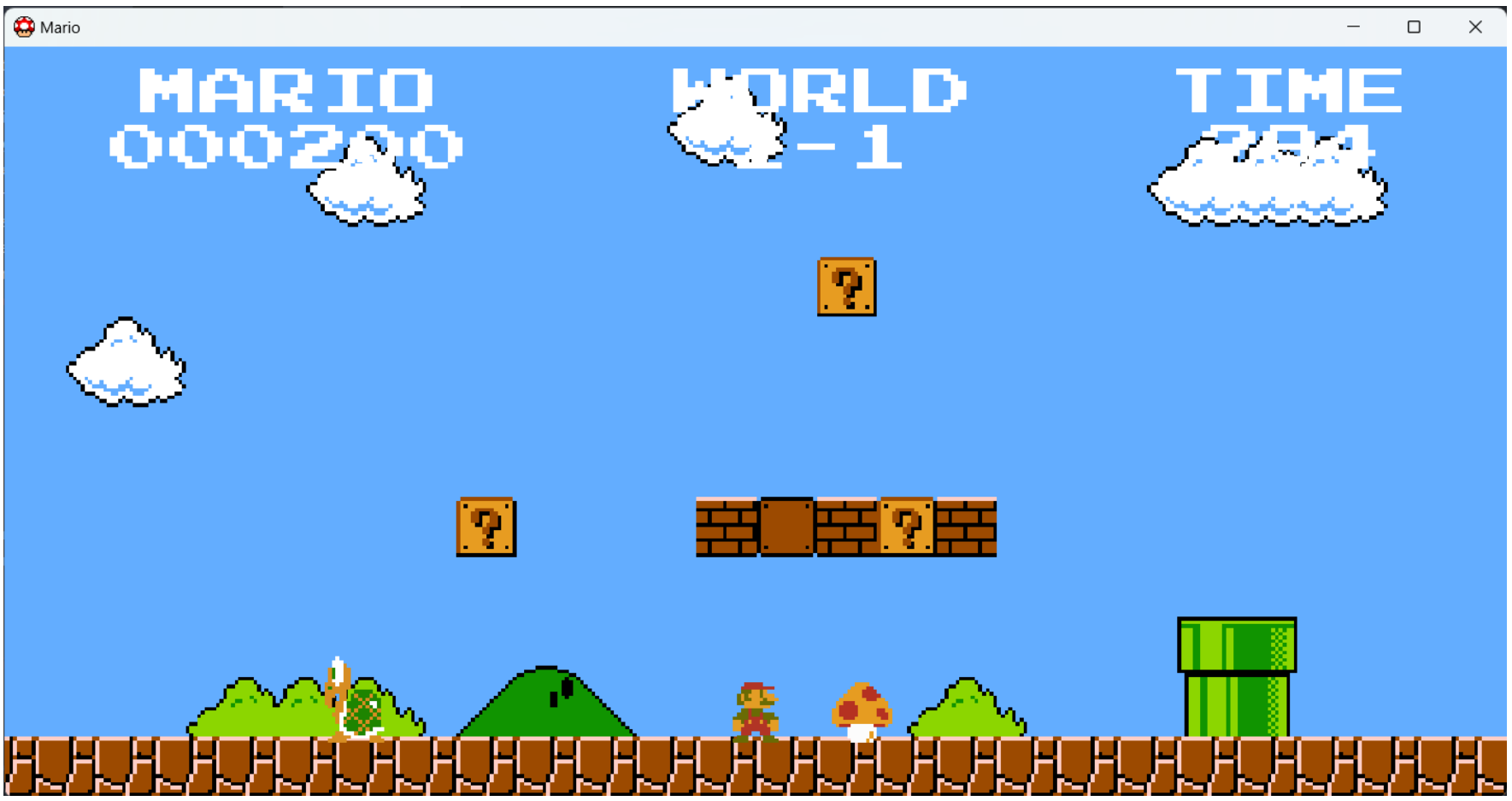
```
![level 1](./imgs/level1.png)
```

- [x] A player (mario) with the following features:
  - [x] Move left and right
  - [x] Jump
  - [x] Grow to big mario
  - [x] Die when hit by an enemy
  - [x] Die when fall into a hole
  - [x] Die when time is up
  - [x] Die when run ou

- [x] Enemies with the following features:
  - [x] Move left and right
  - [x] Die when hit by mario
  - [x] Die when mario jumps on it
  - [x] The turtle can hide in its shell



- [x] Brick (Coin) with the following features:
  - [x] Disappear when mario touches it
  - [x] Increase the score when mario touches it
  - [x] Mushroom comes out when mario touches some of it



- [x] Flag with the following features:
  - [x] End the level when mario touches it
  - [x] Enter the castle when mario touches it
- [x] A board with the following features:
  - [x] Show the current score
    - [x] Increase the score when mario touches a coin
    - [x] Increase the score when mario kills an enemy
  - [x] Show the current time
    - [x] Decrease the time every second
    - [x] End the game when time is up
  - [x] Show the current level

Team members

- BINSHUO ZU (21012854):
  - Implement the object include enemy, item and player.

    - Implement the score and time board.
    - Collect and make enemy and item sprite sheet.
- LIMIN ZHOU (21012853):
    - Design game introductionScreen and MessageScreen.
    - Design the style and effect of the keys.
    - Fix bugs in the game development process.
- ZIYE ZHANG (21012874):
    - Complete the selection level screen and Settings screen.
    - Improve and fix the functions of multiple screen.
    - Fixed bugs with characters jumping and moving in the game.
- ZHAOHUI LIANG (21012755):
    - Complete the presentation of the team project.
    - Select the appropriate architecture and engine to import.
    - Complete some human-computer interaction operations, such as customs clearance, death, etc.

Video Link

- Presentation
- Game Demo