Chat App Design Presentation

Team: Bullet(Team J)

Member:
Siyu Xie(team Lead)
Huijia Zhu (doc lead)
Pengyu Zhang (dev)
Su Zhang (dev)
Suyue Zhang (dev)

Agenda

- Design Decision
- Design Pattern
- GUI Introduction

Technology Stack

Frontend	JavaScript
Backend	Java Spark
Communication	HTTP Protocol

Design patterns we use

1. Strategy Design Pattern
2. Singleton Design Pattern

Use case

1. Movement
User type keyboard up/down/left/right to change Pac-Man's movement direction;
Pac-Man moves straightly toward the current direction.
Movement updates.

2. Pac-Man Collision with ghost
Get the status of Ghost (normal, flash or "eyes" status)
Pac-Man will lose life/get scores or do nothing according to the status.
If Ghost is in normal status, Pac-Man will lose one life. Game will end if Pac-Man loses all lives.

User Case

3. Ghost Movement

Ghost have its movement strategy, every update will call its strategy update function

If the ghost has a GoHomeStrategy, the ghost will call searchHome function to determine the next move

4. Init

Build the matrix with 0 representing wall and 1 representing passway.

Use Case

5. Update

For all Pac-Man and Ghosts in the lists, update its position according to Pac-Man movement and Ghost Movement

6. Control

User type with keyboard up/down/left/right

Pac-man will call function setDirection to set its direction to user input

Interface

GhostStrategy

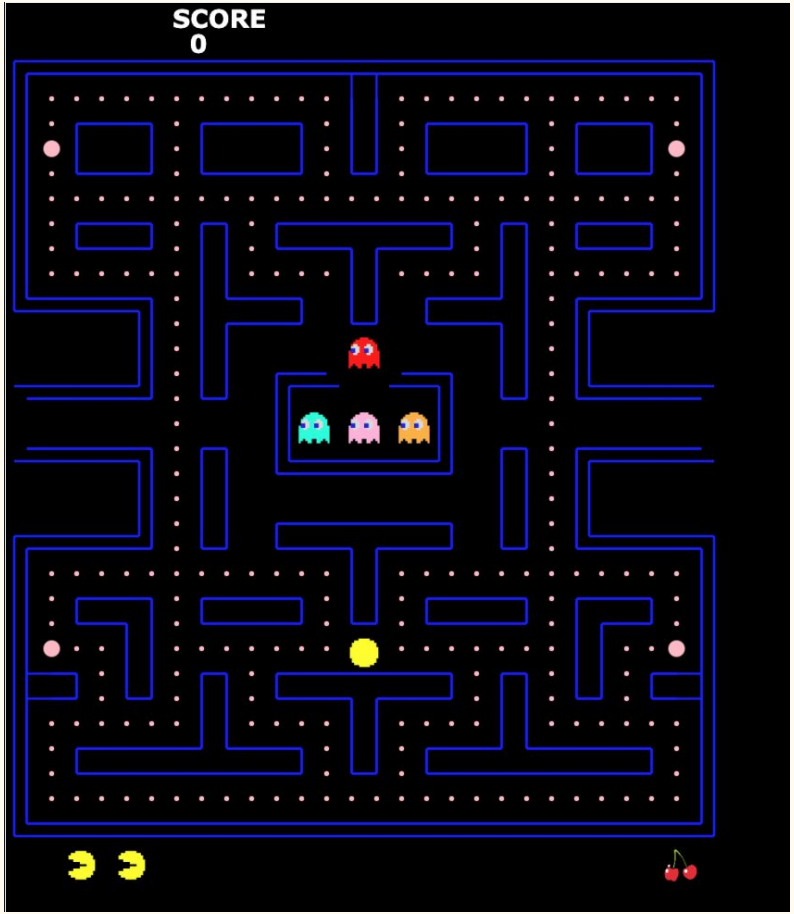
Method	Description
String getName()	Get the strategy name
Void update(Ghost ghost)	Update the ghost using the behavior defined by the strategy

Response

Int life	Current pacman life
Int difficulty	Current game difficulty
Int score	Current score
Pacman pacman	Current pacman information
Ghost[] ghosts	Current list of ghosts
Bonus[] bonuses	Current bonuses

Front end GUI

1. Controller button
2. map



Thank You for Your Listening!

Q&A Session