Pacman Design
Final 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
- API Design
- GUI

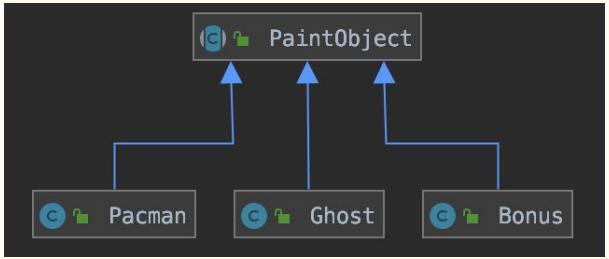
Technology Stack

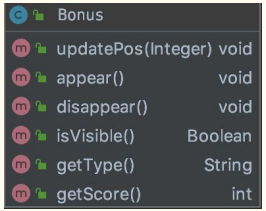
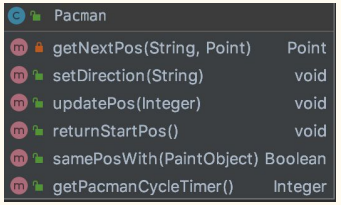
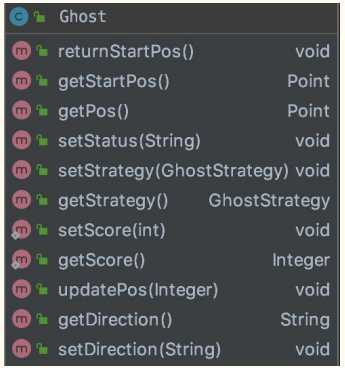
Frontend	JavaScript
Backend	Java Spark
Communication	HTTP Protocol

Design patterns we use

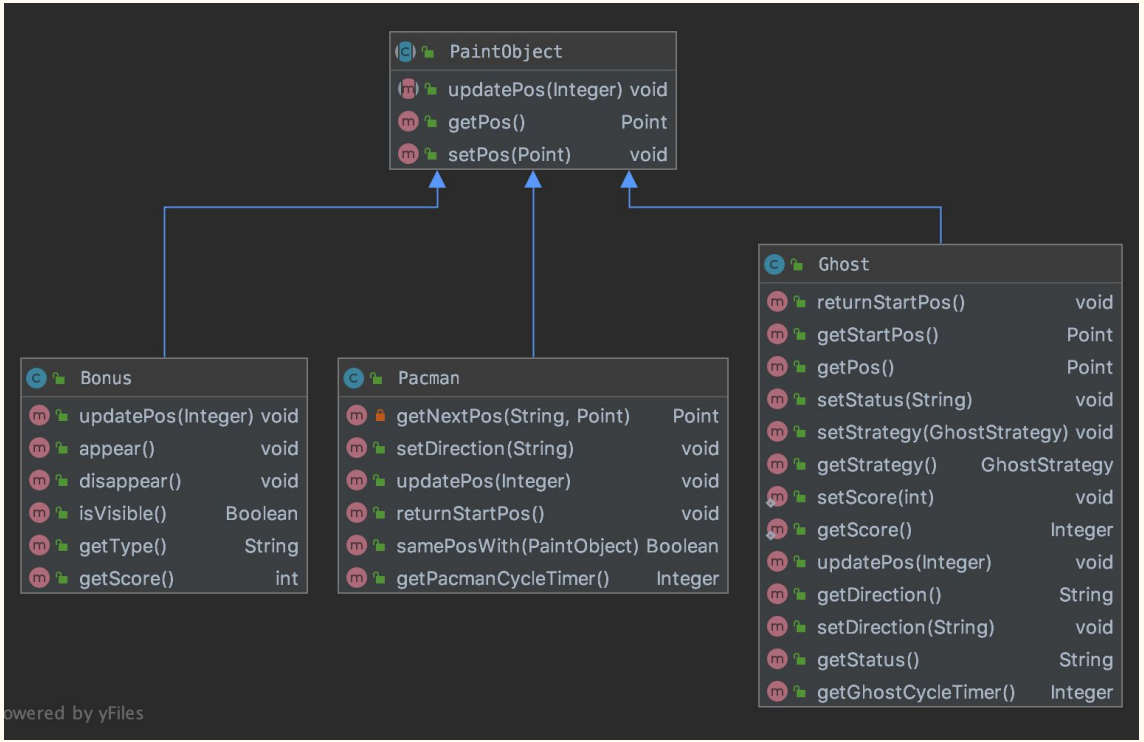
1. Strategy Design Pattern
2. Singleton Design Pattern

UML of PaintObject

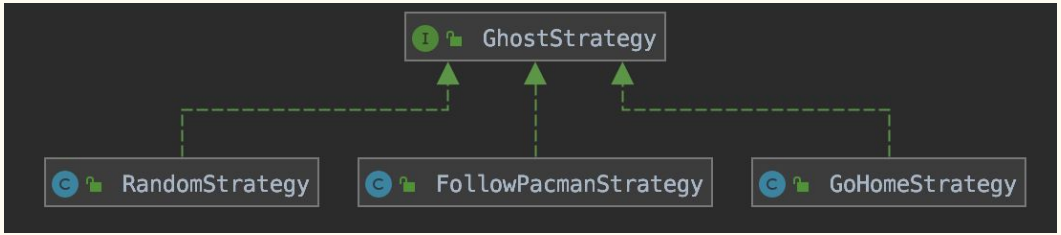


Bonus	Pacman	Ghost
		

UML of PaintObject

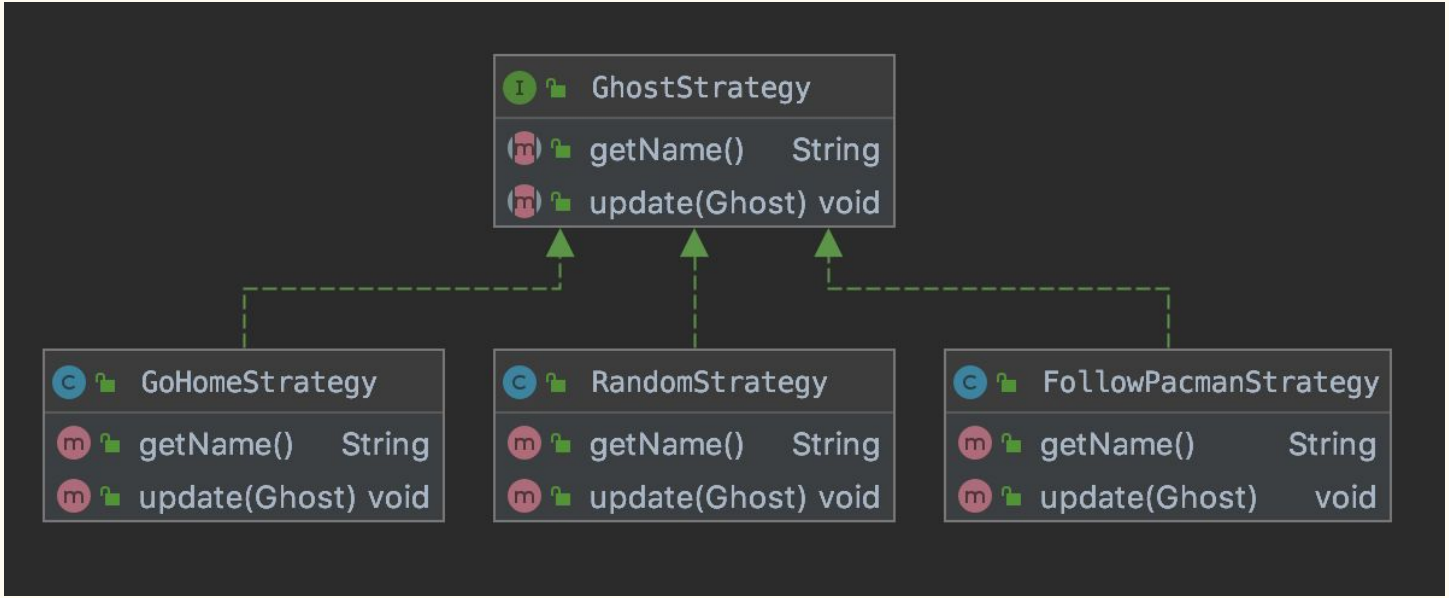


UML of Strategy



RandomStrategy	FollowPacmanStrategy	GoHomeStrategy
GoHomeStrategy getName() String update(Ghost) void	RandomStrategy getName() String update(Ghost) void	RandomStrategy getName() String update(Ghost) void

UML of Strategy



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.

Use case (Pacman movement)

Flow of Events	1. User type keyboard up/down/left/right to change Pac-Man’s movement direction; 2. Pac-Man moves straightly toward the current direction. 3. Movement updates.
Alternative Flows 1	1. If User types a keyboard up/down/left/right, Pac-Man will set up a new direction according to the user's input. 2. Movement updates.
Alternative Flows 2	1. If Pac-Man’s forward board is a wall, then Pac-Man will stay and not move. 2. If Pac-Man’s forward board is a ghost, then Pac-Man will do Collision with Ghost flows. 3. If Pac-Man’s forward board is a bonus (fruits, small dots and big dots), then Pac-Man will get corresponding scores and bonus objects will be set “invisible”. 4. Movement updates.
Entry Conditions	Update times meet pac-man movement cycle

Use Case (Pac-man collision with ghost)

Participating Actors	Pac-Man, Ghost
Flow of Events	1. Get the status of Ghost (normal, flash or “eyes” status) 2. Pac-Man will lose life/get scores or do nothing according to the status.
Alternative Flows 1	1. If Ghost is in normal status, Pac-Man will lose one life. Game will end if Pac-Man loses all lives. 2. All the Ghost and Pac-Man will return to its initial positions.
Alternative Flows 2	1. If Ghost is in flash status, Pac-Man will get a current ghost score. 2. All the Ghost scores will increase. (the first ghost Pac-Man collides with is worth 200, the second is worth 400, the third is worth 800, and the fourth is worth 1600) 3. Ghost will change its status to eyes status, and set up its movement strategy to GoHomeStrategy.
Alternative Flows 3	1. If Ghost is in “eyes” status, Pac-Man and Ghost will do nothing.
Entry Conditions	Ghost and Pac-Man are in same position

Use Case (Ghost movement)

Participating Actors	Ghost
Flow of Events	1. Ghost will have its movement strategy, every update will call its strategy update function. 2. Movement update.
Alternative Flows 1	1. If the Ghost has a RandomStrategy, then the Ghost will randomly move up/down/left or right. 2. If the Ghost's next move will collide with the wall, Ghost will stay and do nothing. 3. Movement update.
Alternative Flows 2	1. If the Ghost has a FollowingPacManStrategy, then the ghost will call SearchPacMan function to determine the next move. 2. Movement update.
Alternative Flows 3	1. If the Ghost has a GoHomeStrategy, then the ghost will call SearchHome function to determine the next move. 2. Movement update.
Entry Conditions	Update times meet Ghost movement cycle

Use Case (Init)

Participating Actors	/
Flow of Events	1. Build a board matrix with “0” of wall and “1” of passaway. 2. Generate Pac-Man, 4 Ghost and dots. Place them in the correct place. 3. Return Game object to front-end (including board matrix, Ghost lists, Dots lists and Pac-Man).
Participating Actors	/
Flow of Events	1. Build a board matrix with “0” of wall and “1” of passaway. 2. Generate Pac-Man, 4 Ghost and dots. Place them in the correct place. 3. Return Game object to front-end (including board matrix, Ghost lists, Dots lists and Pac-Man).
Participating Actors	/
Flow of Events	1. Build a board matrix with “0” of wall and “1” of passaway. 2. Generate Pac-Man, 4 Ghost and dots. Place them in the correct place. 3. Return Game object to front-end (including board matrix, Ghost lists, Dots lists and Pac-Man).

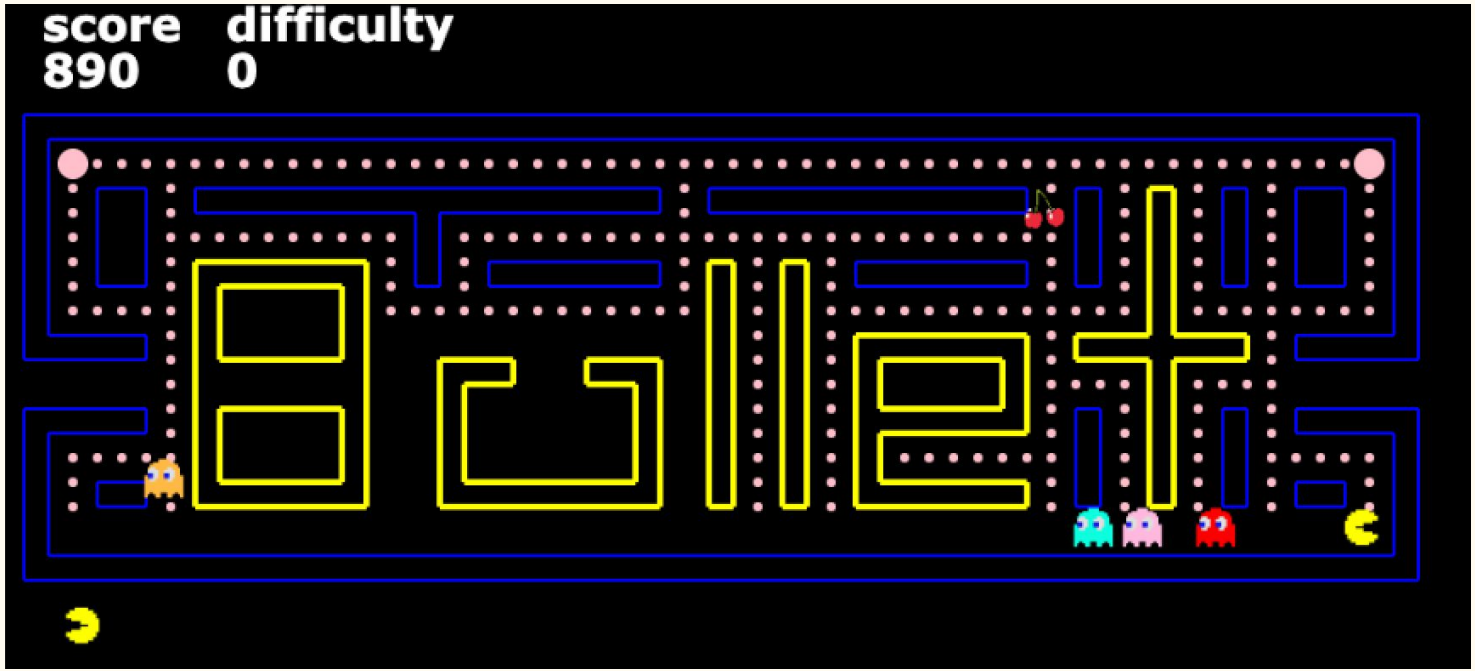
Use Case (Update)

Participating Actors	/
Flow of Events	1. Call game object's function update function. 2. For all Pac-Man and Ghosts in the lists, update its position according to Pac-Man movement and Ghost-Movement. 3. For Pac-Man, update it's status (life or score), all ghosts status, bonus status according to Pac-Man Collision with Ghost workflow. 4. Package these scores/difficulty/pac-man/ghosts/bonus and return response.

Use Case (Control)

Participating Actors	User
Flow of Events	1. User type with keyboard up/down/left/right. 2. Pac-man will call function setDirection to set its direction to the user input.

Design of game map



Thank You for Your Listening!

Q&A Session