

24783 Group Project Component Design

Instructor: Soji Yamakawa

Project Name: PacMan

Group Name: BugOnTheFly

Team Leader: Angel Chu

Team member: Zixuan Zhang, Yanqiao Wang, Yujia Wang, Weilin Zhang

Introduction

To recover most of the functionality of original PacMan game while adding novelty to make it more interesting, we design the following program structure:

I. Core Implementation

In this part, we will realize most of the core functionality of the PacMan game, with the following component constructed:

- A **Board** class recording the map information such as location of obstacles, as well as current location of pacman and ghosts.
- A **Movement** base class with implementations of moving a character from the *Board* class in this game.
- A **Position** class recording current location for a character. It may also help evaluate the distance between two characters.
- A **Ghost** class simulating the general behavior of a ghost character. Since it is a moveable object, it inherits implementations from *Movement* base class.
- A **PacMan** class describing the behavior of a pacman character. Similarly, it also belongs to a subclass of the *Movement* class.
- A **Pellet** class depicting the functionality of a pellet if eaten by a pacman.
- A **Timer** class working as a timer for the game.
- A **GameController** class working as the entrance of the game.
- Planning **ChaseAlgorithm** for four different ghosts: different route planning algorithms to be added to four different ghost characters.

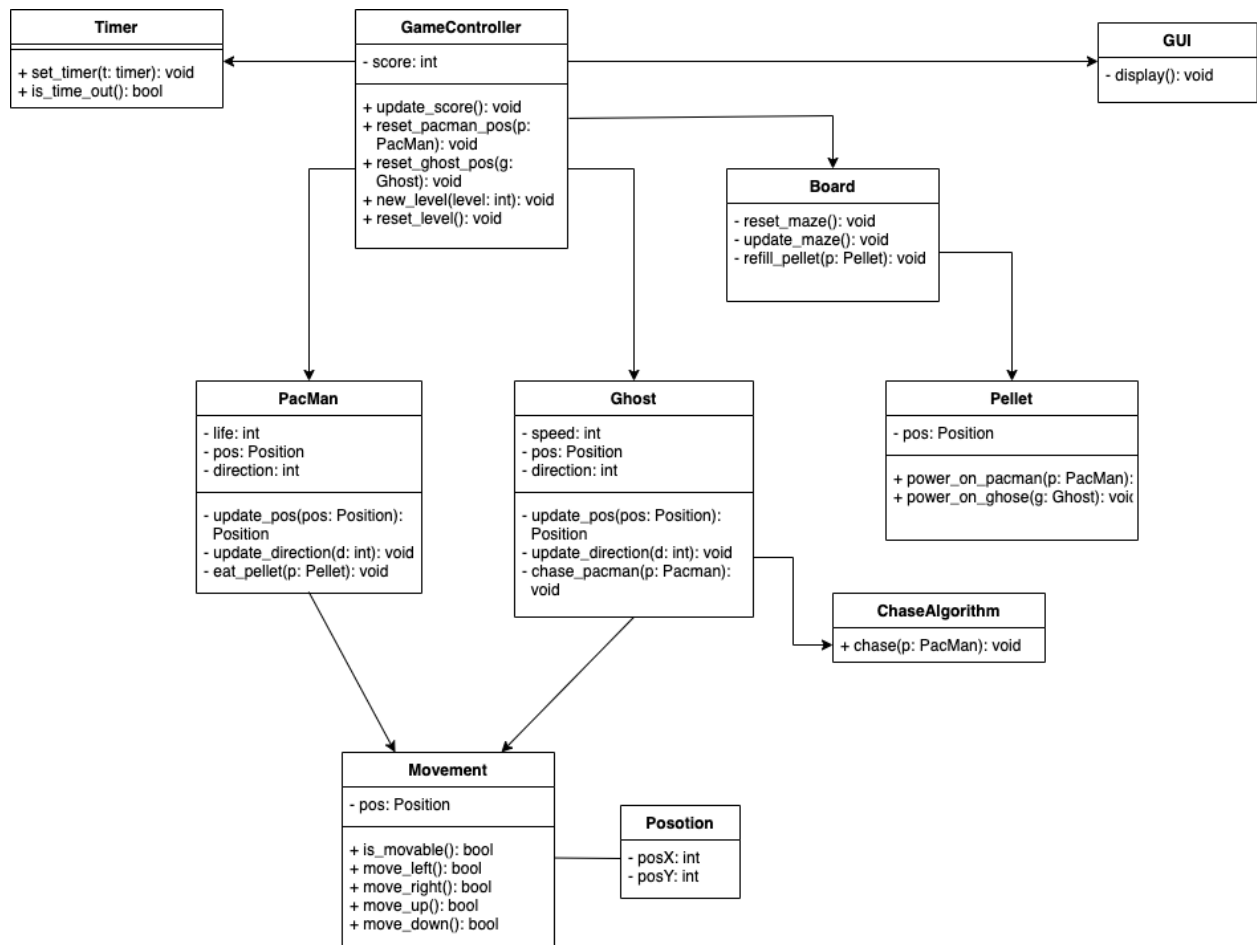
II. GUI Implementation

For the GUI programming, we decide to utilize OpenGL as our graphic library assisting with our GUI rendering. This part involves in the following components:

- A welcome page as the start of the game
- A main GUI 1 displaying level 1 maze design, movement of pacman/ghosts.
- A main GUI 2 displaying level 2 maze design, movement of characters.

- Pop-up windows informing players of victory or failure.

Components In Detail



In this part we would like to talk more about our functional design in detail:

GUI: Display game interface.

display(): void

Board: Initialize game board with walls and positions of pellets, pacman and ghosts.

reset_map(): void

update_map(): void

refill_pellet(p: Pellet): void

Movement: both pacman and ghost are going to refer to this method.

pos: Position
is_moveable(): bool
move_left(): bool
move_right(): bool
move_up(): bool
Move_down(): bool

Position:

posX: int
posY: int

Timer:

set_time(t:timer): void
is_time_out(): bool

Pellet:

pos: Position
power_on_pacman(p: PacMan): void
power_on_ghose(g: Ghost): void

Game_Controller: it is going to monitor the operations of timer, score board, pellet, map, ghost and pacman. When ghost and pacman's current position_t are the same, pacman loses a life.

score: int
update_score(): void
reset_pacman_pos(p: PacMan): void
reset_ghost_pos(g: Ghost): void
new_level(level: int): void
reset_level(): void

Pacman:

life: int
pos: Position
direction: int
update_pos(pos: Position): Position
update_direction(d: int): void
eat_pellet(p: Pellet): void

Ghosts:

speed: int

pos: Position
direction: int
update_pos(pos: Position): Position
update_direction(d: int): void
chase_pacman(p: Pacman): void
switch_state(): void

Algorithm for 4 Ghosts: unlike pacman, ghosts need to move by themselves.

Blinky (red ghost): get pacman's current location and direction, and then find the shortest path to chase the pacman.

Pinky (pink ghost): get pacman's current location and direction, choose the destination as 4 cells in front of the pacman, considering both direction and location (walk in front of)

Inky (blue ghost): walk randomly in the map

Clyde (orange ghost): get pacman's current location, find the shortest path despite chasing after or walking in front of the pacman. Find the second shortest path, and set the current pacman's location as destination

set_destination(position: Position_t, dir): void
find_shortest_path(position, dir): void
find_sec_shortest_path(position, dir): void
move_blinky_ghost(position, dir): void
move_pinky_ghost(position, dir): void
move_inky_ghost(position, dir): void
move_clyde_ghost(position, dir): void

Unit Tests:

We come up with three tentative categories of unit tests for three important component:

1. Pacman test

Test(P_test, createPacman): we test the pacman constructor class and its expected behavior when no keyboard input affects the movement of pacman

Test(P_test, movePacman): we test the up/down/right/left movement of the pacman and check if there is any abnormal behavior if obstacles are met.

Test(P_test, resetPacman): Assert whether all conditions of pac man goes back to initial state.

2. Ghost test

Test(G_test, collideWithPacMan): test whether pac man loses one life when a ghost touches it.

Test(G_test, abnormalCondition): test whether a ghost will be eaten by a pac man when the pac man eats a frozen pellet.

3. GameController test

Test(C_test, set_pellet): test whether a pellet could be set and then eaten by a pac man. Should be reflected in a score increase and change in ghost condition.

Test(C_test, reset_map): test whether ghosts, pellets and pac man are reset to the initial position during game start.

Test(C_test, update_score): test if the score of game will increase differently if the pac man eats normal pellet, frozen pellet and partner pellet.

Responsibility

Since we separated the program into different components, we assign each of our group members with a unique responsibility, where everyone takes charge of

Team Member	Responsibility
Angel Chu	GUI
Zixuan Zhang	Board Design, Movement Class
Yujia Wang	Position, Timer, Pellet Class
Yanqiao Wang	PacMan, Ghosts Class
Weilin Zhang	Route Planning Algorithm