**PACMAN PROGRAM DOCUMENTATION**

PacMan.java

PacFrame.java

PacComp.java

Scoreboard.java

Ghost.java

Player.java

Food.java

Map.java

DistanceGhost.java

AmbushGhost.java

HorizontalGhost.java

VerticalGhost.java

1. **PacMan.java:** This is the tester class of our project, which only contains an object of the PacFrame class and the main method.
2. **PacFrame.java:** This is the JFrame object, in which we display the game. We initialize a PacComp class object and put it to the center of this frame. Then we made the basic settings of a frame, such as title, relative position, visibility, etc.

The keyListener of this class is used for getting input from the user for moving the Pacman. If the game is continuing, the space button can be used for pausing and resuming the game. If the player exceeds 12150 points, the game will be won, and if there are 0 lives left, the game will be lost. The actionListener() is used to check the status of the game, and repainting the frame in every 20 milliseconds.

1. **PacComp.java:** This is the class that we used to display the game on the JFrame. We initialize every component that we created so far in this class in order to combine them into a game.

First of all, we fill every empty place on the map with a food. Then we set 4 of these foods as power ups, and 4 of them to be empty places for placing the ghosts. We added all foods in one two dimensional array, and all the ghosts into one array.

The paintComponent (Graphics g) method is basically for drawing the objects of Scoreboard, Map, Player, and Ghost classes. We placed the ghosts according to the position of the Pacman. We also have three Booleans: stopped, win, gameOver. These are repeatedly checked by the paintComponent(Graphics g) method in order to display specific information on the screen.

The reDraw() method is a mutator method that we used to change some Boolean values of the Pacman in order to change the direction and change the values on the scoreboard. It also changes the mode of ghosts from chasing to run away if border of the Pacman intersects with the border of a power fruit. It repaints the game after makes all the changes.

1. **Scoreboard.java:** This is the class that we used as the scoreboard in our game, at the top of the screen. It has two attitudes: Integer lives, initially 3, and Integer score, initially 0.

The class has 3 methods. The first method is draw(Graphics g) method, which basically draws the rectangles as the borders of the scoreboard, and displays the lives and points of the player. The second method is died(), which reduces the lives variable by 1, and the last method is getPoint(), which increments the variable score by 1.

1. **Food.java:** This is the class that we used to display the foods that Pacman will eat. They have 4 attributes: Integer xpos, Integer ypos, Rectangle2D bor, Boolean eaten.

When an object of the food class is initialized, it gets an x-position and y-position form the constructer. Then it creates a rectangle for its border. Finally, it’s Boolean eaten is false. It has one mutator method, which is draw(Graphics g), and it basically draws an oval inside the borders of the food.

It has three accessor methods, one for returning the x-position, one for returning the y-position, and one for returning the borders of the food.

1. **Map.java:** This class generates the map. It uses two dimensional shape array to hold the walls. There are three types of walls: horizontal, vertical, corner. It fills array with corresponding walls and then erases some to create aimed paths in constructor. It has a findRelativeLoc(int i, int j) method. i and j means the indexes of the array and find the relative x, y coordiante on the canvas. It also has intersects(Shape s) method to determine if a object intersects with the walls. It also has draw(Graphics g) method to draw shapes.
2. **Player.java:** This class has variables and methods of the pacman, the player character. It has move() method to change variables. It takes user input via getInput() to change directions of pacman, according user input, booleans that represent directions are changed. The draw() function of this class is called in PacComp ( component class ) to draw pacman.
3. **Ghost.java:** This is the main abstract class that the other ghost classes extend. The ghosts are all declared and initialized from within the constructor of PacComp class. The working principle for the movement of the ghosts depend on 3 different modes: Scatter, Chase & Escape. All ghosts have their own timer instances to keep track of how many seconds they were in a specific mode and change it if that mode’s allowed time value is exceeded. The ghosts always decide which direction to go to at the centers of each square. In scatter mode they choose the order of the directions to test randomly. In chase mode, they try to choose the order based on pacman’s location to catch him with their unique algorithms. In escape mode (which they enter only if pacman eats the special fruits) their first action is to reverse their direction to get away if they were chasing pacman. Then they again decide randomly.
4. **VerticalGhost.java:** This ghost compares pacman’s y to his own and first determines if it should try to go north or south. Then he does the same for x values. It always tries the vertical directions first unless the y values are equal.
5. **HorizontalGhost.java:** The same as vertical ghost, although this one favors the horizontal directions by first comparing the x values.
6. **DistanceGhost.java**: This ghost, during chase mode, checks to see if pacman is within the ghost’s danger zone (approximately less than 3-4 times the length of a square apart). If so, he checks all his direction options and compares the distances between pacman’s current location to itself if it made that move. It takes the direction with the least distance.
7. **AmbushGhost.java:** This ghost aims to catch pacman at the exit of the tunnel pacman takes or end of the road pacman cruises. How he tries to does that is that in essence it works similarly to vertical & horizontal ghosts. However, it takes the square that is 4 blocks in front of pacman according to what direction it goes to.