Pacman

Sophie Weikovich, Ranveer Singh, Andreas Frauendorfer

Game design/details

- For this implementation of pacman we have 2 core classes, which are called Character.java and Game.java respectively.
- The Character class holds our design for the chasers(the ghosts) and our player(pacman).
- The Game class which holds the logic of the game implementation including our public static void main.

Character class

- Our character class is abstract which implements comparable and is inherited by the chasers and player.
- This class holds the name of the character, if it is a chaser or player, and its position in the game.
- The functionality implements the next move, which is the way how characters move around the grid respective to their X and Y values
- The compareTo method will calculate the distance between two character, and when it returns it will give a positive number if the first passed in character is higher in the plane, and negative if the second passed in character is in a lower position in the plane.

Character Class Continued

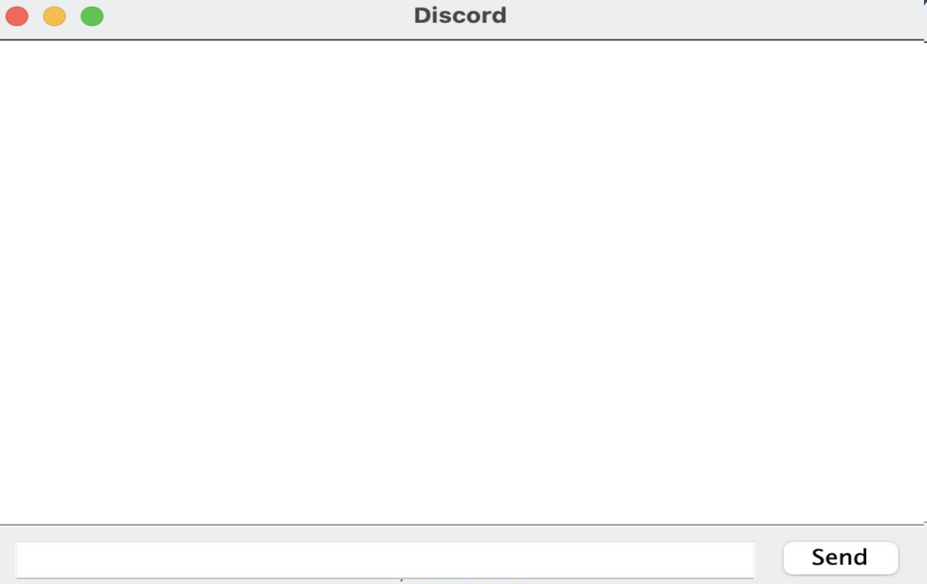
- The chasers were designed by each one of us and respectively do:
  - SophieCharacter: This chaser moves randomly one field in every direction like the King in chess
  - AndreasCharacter: This chaser moves diagonally over the playing field
  - RanveerCharacter: This chaser moves by jump randomly around playing field except that it cannot land in the middle of the board
- Each chaser also has a class attached to it that will print the position of where on the grid that chaser would be.
- The player class is controlled by the user and uses input from the user and moves around the map. The X and Y values are updated and in their position matrix.

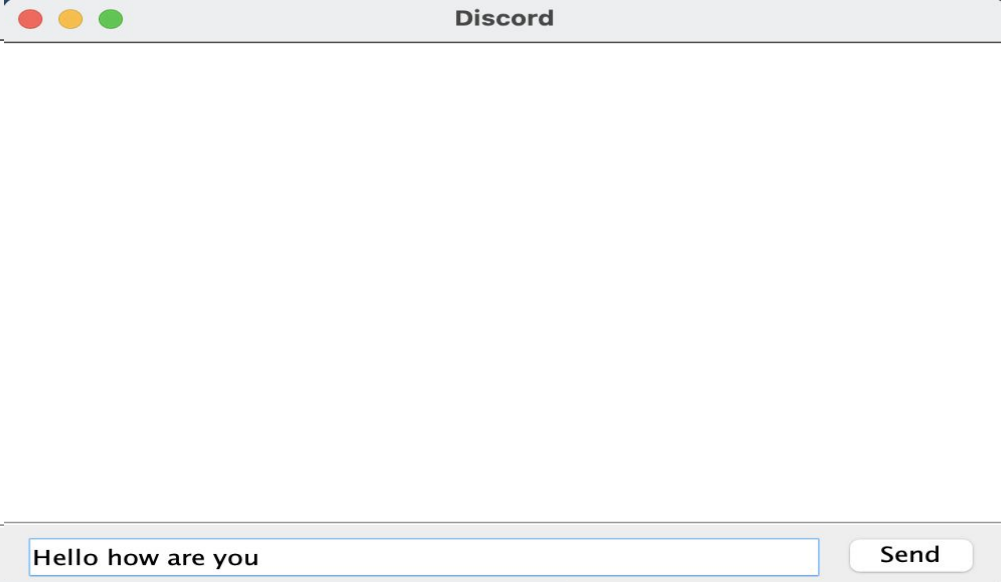
Game Class

- In the game class we have our game play out over a fixed sized grid with 3 chasers and one player.
- Our Public Static Void Main does the following:
  - stores last and best score in a stack read from a file
  - initializes random positions to the players and 3 chaseers
  - The player and chaser each move once each round
  - warns if player gets to close
  - displays the playing field with the Player and 3 Chaser
  - checkPlayerDead: compares the position of the player with the position of each chasers and returns true if they align
  - sumUntilNumber: sums up all numbers until n

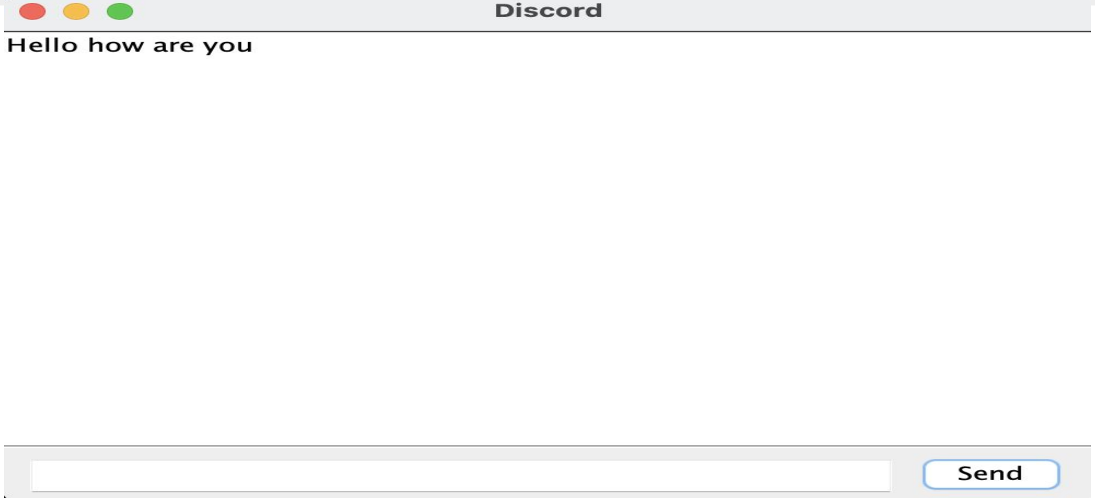
GUI

- Since the swing gui would not update the screen properly when running the pacman grid we decided to show our implementation with a simple ‘discordlike’ chat bot.
- This gui uses both the Java swing and awt libraries to run.
- Once it is ran it pops a window up that pops up, which gives the user a prompt to enter a message. After clicking enter the text is submitted and is displayed in the chat log.





Hello how are you



Hello how are you