PACMAN

Documentation

Amar Zagorčić

Tena Vladović-Relja

Date: 03-05-2022

unaccepted Revision: 0.1

**Table of Contents**

Executive overview 3

Audience 3

Application intentions 3

Assumptions made for this project 3

*Class and method overview* 4

Client GUI 5

Networking connections & Protocols 7

# Executive overview

The final project of the Computational Problem Solving in the Information Domain II for 2021/2022 freshmans was to create a PacMan game in Java programming language. The course sections were devided into groups of two people who were going to be working together on this project. Almost everything we learned throught the second semester of our freshman year was useful in creating this project. The work for the project started out with creating the initial designs for our maze, ghosts, PAC-MANs, and any other design that might be implemented into our game. Those desgins were followed by code, and new designs

# Audience

People who also wish to create Java projects similar to this one. This document would be of help in letting them know that creating a game can be a two person job. This document would answer people’s questions; let them know what was the reason behind this project, how did the creation start, what were the steps along the way, what was expected and what was unexpected, and how did we overcome those unexpected situations.

## Application intentions

The function of this application is not complicated, it is a game of PAC-MAN to be played by one or more people. There is no target audience, anyone can play this game.

# Assumptions made for this project

For this game to work, wee need:

The code we wrote for the game,

The maze background we created,

a connection between two or more people to be able to play this game together in multiplayer mode.

# *Class and method overview*

Overview of the classes and functionality.

Game2DSARTER

* initializes icons which will me visible in the game
  + creates the timer, heart counter(“food”), speed, etc.
* main method
  + launches the game
* start
  + sets the stage and the scene
  + calls initialiseScene
* initialiseScene
  + starts the timer
  + “updates” the images
  + Sets images and their sizes
  + Creates the GUI
  + Key event handler
* sendPacRacer
  + movement of the pacman
* connect
  + connects to the server
  + prints out “connected” when successful
* handle
  + button handler
* sendMessage
  + operates the chat
* class ClientThread
  + connects the client to the server
* class PacManMap
  + sets the map as the background
  + changes the map when 5 hearts are collected
* class PacmanRacer
  + sends the position of the Pacman
  + creates collision
* rand
  + puts the heart in a random position
* beriPos
  + sets the location of the RedBull
* multi
  + creates a dummy pacman to be displayed on screen as movement of the second player
* Game file with Game class
  + Creates GUI for the loading screen with 3 buttons
  + Creates a GUI for instructions for the game

# Client GUI

Lvl 1 map

A picture containing clock, several

Description automatically generated

Lvl 2 map

A picture containing wall, clock, black, orange

Description automatically generated

# Networking connections & Protocols

Client connection to server:

1. Run the server

2. run 2 clients (or run the compound in the Run and Debug section, this will run the server and 2 clients by itself)

3. press “ctrl” to enable chat and create a connection

4. the client writes their name and clicks connect

5. press the letter “K” to create a connection

Port number(s): 32002

# Data used

File 1: Game.java – a file which initializes the loading screen with buttons to start, exit, and get instructions

File 2: Game2DSTARTER – holds all code necessary to run the game

For connection, the port number 32002 is used