TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN

----------o0o----------

BÁO CÁO BÀI TẬP LỚN
MÔN CÔNG NGHỆ JAVA

Đề tài: Lập trình game Tetris

Giáo viên bộ môn: Vũ Huấn
Nhóm: 28
Sinh viên thực hiện: Đinh Thành Nam
Nguyễn Phạm Đức Quý
Lớp học phần: N07

Hà Nội, tháng 5 năm 2023

Mục lục

CHƯƠNG 1. MÔ TẢ DỰ ÁN

1.1. Mô tả yêu cầu dự án

- Lập trình 1 chương trình game Tetris dựa trên ngôn ngữ Java.
- Luật chơi:
  - Di chuyển các khối gạch đang rơi từ từ xuống trong kích thước hình chữ nhật 10 cột x 16 hàng trên màn hình. Chỗ nào có gạch rồi thì không di chuyển được tới vị trí đó.
  - Người chơi xếp những khối hình sao cho khối hình lấp đầy 1 hàng ngang để ghi điểm và hàng ngang ấy sẽ biến mất.
  - Mỗi 2 hàng ngang được lấp đầy sẽ lên 1 cấp. Mỗi khi lên cấp sẽ tăng tốc độ rơi của khối và lượng điểm nhận được.
  - Một khối hình sẽ rơi từ phía trên cùng của màn hình, di chuyển các khối và xoay chúng cho đến khi chúng rơi xuống phía dưới cùng của màn hình, sau đó một khối hình tiếp theo sẽ rơi xuống.
  - Nếu để cho những khối hình cao quá màn hình, trò chơi sẽ kết thúc. Trò chơi kết thúc khi khối gạch không rơi xuống được nữa.
  - 4 phím điều khiển chính:
    + Phím mũi tên lên: xoay khối.
    + Phím mũi tên trái: di chuyển sang trái.
    + Phím mũi tên phải: di chuyển sang phải.
    + Phím mũi tên xuống: tăng tốc độ rơi.

1.2. Phương hướng giải quyết yêu cầu dự án

- Theo yêu cầu của dự án, chúng em bắt đầu xây dựng những lớp, thuộc tính và phương thức đáp ứng từng yêu cầu.
- Các lớp được chia ra làm 2 phần:
  - Handling: gồm các lớp để đọc giữ liệu từ chuột và bàn phím
  - Main: gồm các lớp giao diện (Frame, StartingMenu, EndScreen, HighScore, ImageLoader, Game), các lớp phương thức (Game, Piece, Player, Shapes, Scoring) và lớp Controller để khởi tạo và quản lý game.

CHƯƠNG 2. CÁC LỚP

2.1. KeyListener

- Phương thức:
  - Hàm tạo.
  - keyPressed: đọc nút mũi tên được nhấn và thực hiện hành động tương ứng.
  - keyReleased: xác nhận nút nhấn đã được thả.

2.2. MouseHandler

- Phương thức:
  - Hàm tạo.
  - void mousePressed: xác nhận chuột đang bấm.
  - void mouseReleased: đọc xem chuột nhấn vào nút bấm nào vào thực hiện chức năng của nút bấm đó.

2.3. MouseMotion

- Phương thức:
  - Hàm tạo.
  - void mouseMove: đọc vị trí của chuột.

2.4. Controller

- Thuộc tính:
  - static final int width/heigth: chiều dài và chiều rộng của game.
  - BufferedImage image: lấy hình ảnh.
  - Graphics2D g: lưu trữ những hình vẽ lên game.
  - Enum STATE: tạo các hằng đại diện cho các trạng thái của game.
  - static STATE state: đặt game ở trạng thái mặc định là màn hình menu.
- Phương thức:
  - Hàm tạo.
  - void addNotify: kiểm tra đã có thread hay chưa.
  - void init: khởi động game.
  - void tick: cập nhật game.
  - void render: thực hiện render tùy theo trạng thái của game.
  - void run: chạy game, thực hiện tick, render mỗi 1millis.
  - static void switchClasses: thực hiện đổi giữa các trạng thái game.

2.5. Game

- Thuộc tính:
  - int cellSize: độ to của từng ô trong game.
  - static int gridWidth/gridHeight: chiều dài và chiều rộng của bảng game.
  - int[][] gamePos: tạo mảng 2 chiều lưu vị trí của từng ô trong bảng game.
  - int speed: tốc độ rơi của khối.
  - Rectangle[] buttons: tạo 3 khu vực để vẽ nút bấm.
  - class Clock: cập nhật game khi game đang chạy.
- Phương thức:
  - Hàm tạo
  - void start: khởi động game.
  - void init: khởi tạo các thuộc tính cần thiết cho game.
  - void tick: kiểm tra trạng thái của khối và game.
  - void render: cập nhập khung hình.
  - void moveLeft/moveRight: di chuyển khối sang trái/phải nếu ở rìa thì không di chuyển được.
  - void removeCurrentPiece: xóa khối hình đang điều khiển.
  - void checkRowCompletion: kiểm tra khi lấp đầy 1 hàng thì xóa hàng và cộng điểm.
  - void checkLost: kiểm tra đã thua hay chưa, nếu thua chuyển qua màn hình kết thúc game.
  - void moveDownRows: chuyển các hàng xuống khi 1 hàng ở dưới đầy.
  - void updateCurrentPiece: cập nhật vị trí khối hình hiện tại.
  - void rotatePiece: xoay khối hình.
  - void clearGame: xóa dữ liệu của game để tạo game mới.
  - void backToMenu: trở về màn hình menu.

2.6. EndScreen

- Thuộc tính:
  - Rectangle menu: tạo khu vực để vẽ nút về màn hình menu.
  - Point mousePoint: kiểm tra vị trí chuột chỉ vào.
- Phương thức:
  - void checkScore: kiểm tra điểm đạt được, nếu cao hơn điểm lưu trữ lưu điểm đạt được, xóa điểm thấp nhất trong file lưu trữ.
  - void render: vẽ màn hình kết thúc game.

2.7. Frame

- Phương thức:
  - Khởi tạo 1 khung viền game mới.

2.8. HighScore

- Thuộc tính:
  - Scanner scanner: đọc file.
  - Rectangle menu: tạo khu vực để vẽ nút về màn hình menu.
  - Point mousePoint: kiểm tra vị trí chuột chỉ vào.
- Phương thức:
  - void render: vẽ màn hình điểm cao.
  - void goBack: quay trở về menu.

2.9. ImageLoader

- Thuộc tính:
  - BufferedImage img: lưu trữ hình ảnh đọc từ file.
  - static String squarePath: lưu ảnh các khối.
  - static String tetrisPath: lưu ảnh logo Tetris.
- Phương thức:
  - Hàm tạo.
  - BufferedImage getImage: trả về ảnh lấy được.
  - BufferedImage getSubImage: trả về 1 khu vực trong ảnh lấy được.

2.10. Piece

- Thuộc tính:
  - int x/y: tọa độ của khối.
  - int [][] shape: loại hình dạng của khối.
- Phương thức:
  - Hàm tạo.
  - Hàm get, set của x, y.
  - int getWidth/getHeight: chiều dài và chiều rộng của khối hình.
  - void updateY: cập nhật vị trí khối hình rơi.
  - int [][] getShape: lấy hình dạng của khối.
  - void rotateSelf: kiểm tra có thể xoay hình hay không, nếu được xoay hình 90^{o}.

2.11. Piece

- Thuộc tính:
  - int score: lưu điểm người chơi.
  - int level: lưu cấp người chơi.
  - int linesCleared: lưu số lượng hàng đã lấp đầy.
- Phương thức:
  - Hàm tạo.
  - Hàm get của score, level.
  - void clearedLine: tăng cấp người chơi với mỗi 2 hàng lấp đầy.
  - addScore: cộng điểm người chơi mỗi khi lấp đầy hàng.

2.12. Scoring

- Thuộc tính:
  - static String filePath: lưu đường dẫn đến folder Tetris.
  - static String HIGHSCOREPATH: lưu đường dẫn đến file Highscores.txt trong folder Tetris.
- Phương thức:
  - int[] readFile: đọc file điểm cao.
  - static void createFile: tạo file Highscores.txt.
  - static void deleteFile: xóa file Highscores.txt.
  - static void createDir: tạo đường dẫn đến folder.
  - static void writeFile: viết điểm cao vào trong file Highscores.txt.
  - static String getDefaultDirectory: lấy đường dẫn đến folder Tetris.

2.13. Shapes

- Thuộc tính:
  - static int[][] Block: các hình dạng của khối.
- Phương thức:
  - static int[][] randomBlock: tạo hình khối có hình dạng ngẫu nhiên.
  - static int getColor: lấy màu của khối.

2.14. Shapes

- Thuộc tính:
  - int spawnCount: đếm lượng khối tạo ra.
  - Rectangle[] buttons: tạo 2 khu vực để vẽ nút bấm
  - int speed: tốc độ rơi của khối.
- Phương thức:
  - Hàm tạo.
  - void tick: tạo các hình khối rơi ở màn hình nền.
  - void render: vẽ màn hình nền.
  - void centerText: căn giữa chữ.
  - void highScores: mở bảng điểm cao.
  - void playGame: chuyển qua màn hình game.
  - class Clock: tạo các hình khối khi đang ở màn hình menu.\

PHÂN CÔNG CÔNG VIỆC

Họ và tên	Công việc thực hiện
Đinh Thành Nam	Controller, EndScreen, Frame, Game, ImageLoader, Piece, Shapes, StartingMenu.
Nguyễn Phạm Đức Quý	KeyHandler, MouseHandler, MouseMotion, Player, HighScores, Scoring.

TÀI LIỆU THAM KHẢO

- Tham khảo thuật toán: https://github.com/janbodnar/Java-Tetris-Game