CSE 241 Winter Project

In this Project Firstly I started to make a view board representation. I made a view for cellView that class contains setting color setting cellSize etc. After that I made a BoardView by extending Relative layout each cell is right and bottom of the before for every cell in a row and I accessed to cells by id. Every Cells has own id that is created by a static id generator which starts from 1000. Before updating the board I reset that id generator.

The Main activity is to hold board and options menu that contains save, load, new game reset etc.

In the setting activity I took all game parameters to construct hex object and then I sent the hex object which is seriable as I did hw-7 , to main activity with intent.

I search for algorith for AI part I learnt minimax algorithm. But icouldnt find a library for hex AI so I did it without help. The idea mine was that I made graph classes and converting the board to graph and then with dijkstra algorithm I find shortest distance between borders.

And then in minimax algorithm I took all free cells to a list and shuffle them and I selected and find minimax of every state recursively that was so complicated because of recursive functions. It takes lots of times to calculate the best place for cell of computer So I made its deepness 2 because it doesnt take lots of time to show demo. U can change deephness in minimax algorithm 2 to to many number .But as I said it works but takes so long time. I couldnt optimize it because I havent so much time. I learnt this project in staring of the school. I had 2-3 homeworks were hard like this. Sorry for that sir.

Views

1) BoardView: Takes cells and initializes board view Sets Cells according to screen size and board size.

- init() : initilizes the boardView by taking required arguments
- setColor(Hex) : *sets colors of board cells*
- makeBoard : *makes board row by row*
- private void addCellToRow : *adds a cell to row*

- private void setCellSize() : *calculates the appropriate cellSize*

private int dpToPx(float dp) *: Converts dp to px*

- cellView.onTouchListener : handles board touches

private void showWinner() throws Exception : *shows a winner with a toast message*

public void save: Takes name and creates file

private void setCell(Location location, CellView.CellColor color) : *sets the board cells color*

public void updateBoard() : *updates the board*

2) CellView holds location and contains color has an id .

public Location getLocation(int boardSize *returns the location of the cell*

public void setColor(CellColor cellColor): *sets the color of cell*

private void setCellLengths(int cellSize) : *sets the lengths of the cell according to cellsize*

Main Activity : *Contains BoardView and options menu. Options menu contains all required operations for game like save ,load, new game, compare etc.*

public boolean onOptionsItemSelected() : *sets all the options of the option menu*

public boolean onCreateOptionsMenu(Menu menu) : *inflates options menu and add*

private void loadGame() : *loads the game by showing list of the saved games*

private void compare() : *compares the games by showing all saved games*

private void newGame() : *starts a new game redirects to settings activity*

protected void onCreate(Bundle savedInstanceState) : *oncreate method that initilizes all views and setting up them*

Setting Activity : Takes all parameters for the hex game and initilalizes game and sends hex object to main activity with intent . I setted some listeners for radio buttons and numberpicker

public void colorOnClick(View view) : sets color

public int dpToPx(float dp) : converts dp to px

public void start(View view) : *makes an intent puts the hex game to in intent and starts the main activity (board representation)*

private boolean requestWritePerm() : *requests the write permission from user*

HexEngine : Hex engine is completely same as I did on hw-7

Minimax

DijkstraAlgorithm : Its an algorithm for finding the shortest path between 2 node in a graph

public void execute(Vertex source) : *takes source*

private void findMinimalDistances(Vertex node) : *finds minimum distances*

private int getDistance(Vertex node, Vertex target) : *returns the distance*

private List<Vertex> getNeighbors(Vertex node) : *finds neigbours and returns*

private Vertex getMinimum(Set<Vertex> vertexes) : *returns the minimum vertex from the given vertexes*

public LinkedList<Vertex> getPath(Vertex target) : *returns path and returns null if not exist any path*

Edge : The class to hold edges of a graph that holds
- Id
- Source vertex
- Destination vertexes

Graph : The class for graph structure contains all
- vertexes and
- edges

Vertex : The class for Vertexes that contains
- Id
- Location
- CellState

Minimax

The class for minimax algorithm that calculates scores of users and try to find best location for Computer which contains:
- hex
- left, right, up, down
- deepth : for minimax algorithm can be changed
- steplimit

private int shortestPathOfUsr2() throws Exception : *returns the shoretest path Size for user2*

private int shortestPathOfUsr1() throws Exception : *returns the shoretest path Size for user1*

private int getWeight(LinkedList<Vertex> path,Graph graph : *gets the weight of the path*

public int getScoreOfUser2() throws Exception : *returns current score of the user 2*

private int getWeight(LinkedList<Vertex> path,Graph graph): *gets the weight of the path*

private Graph getGraph(Hex.CellState userSign) throws Exception : *returns the graph for a users cells*

public Location bestLocation(Hex hex) throws Exception : *tries to find best location for the user computer*

private int max(Hex hex) throws Exception : *returns maximum score of the given the given clone of hex game and makes all possible samples and put them into min function returns maximum one*

private int min(Hex hex) throws Exception : *returns minimum score of the given the given clone of hex game and makes all possible samples and put them into max function returns minimum one*

Mehmet Hüseyin YILDIZ
200104004095