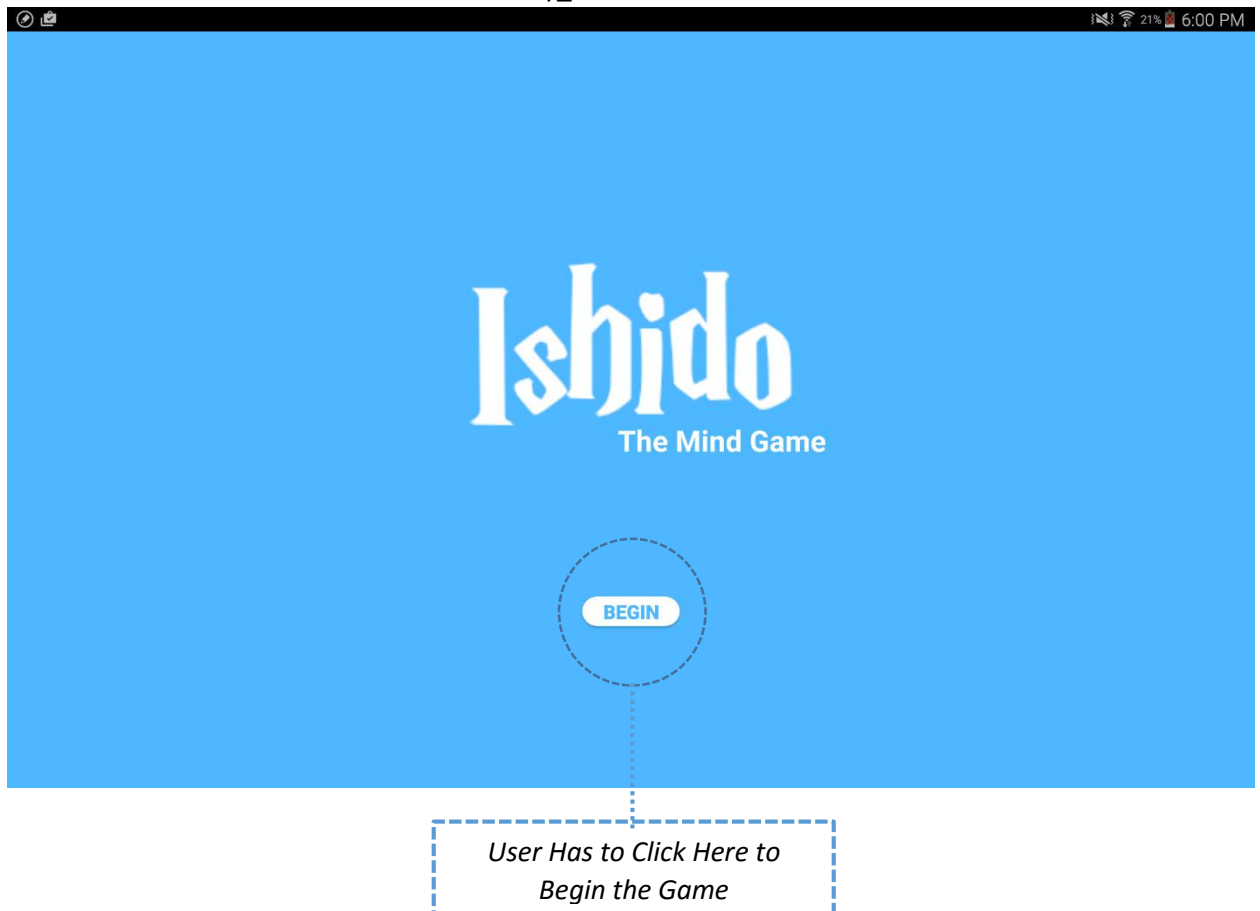# Ishido

ARTIFICIAL INTELLIGENCE, PROJECT 1

Arjun Bastola

RAMAPO COLLLEGE OF NEW JERSEY |
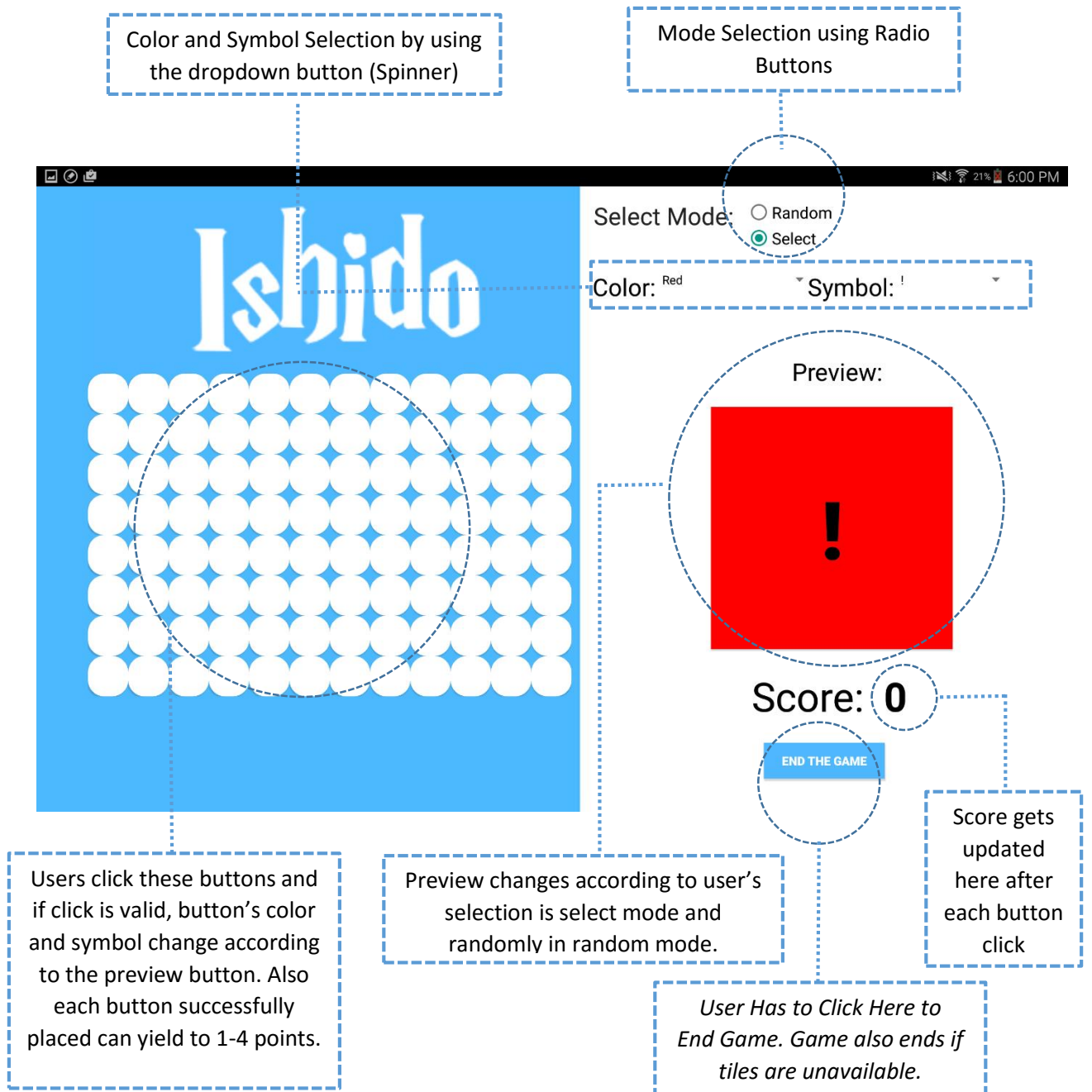COMPUTER SCIENCE DEPARTMENT

Activities:

1. **activity_main:**

   This activity has been used a welcome screen for the app. The activity has the game logo and a start button that sends the user to the activity_board.
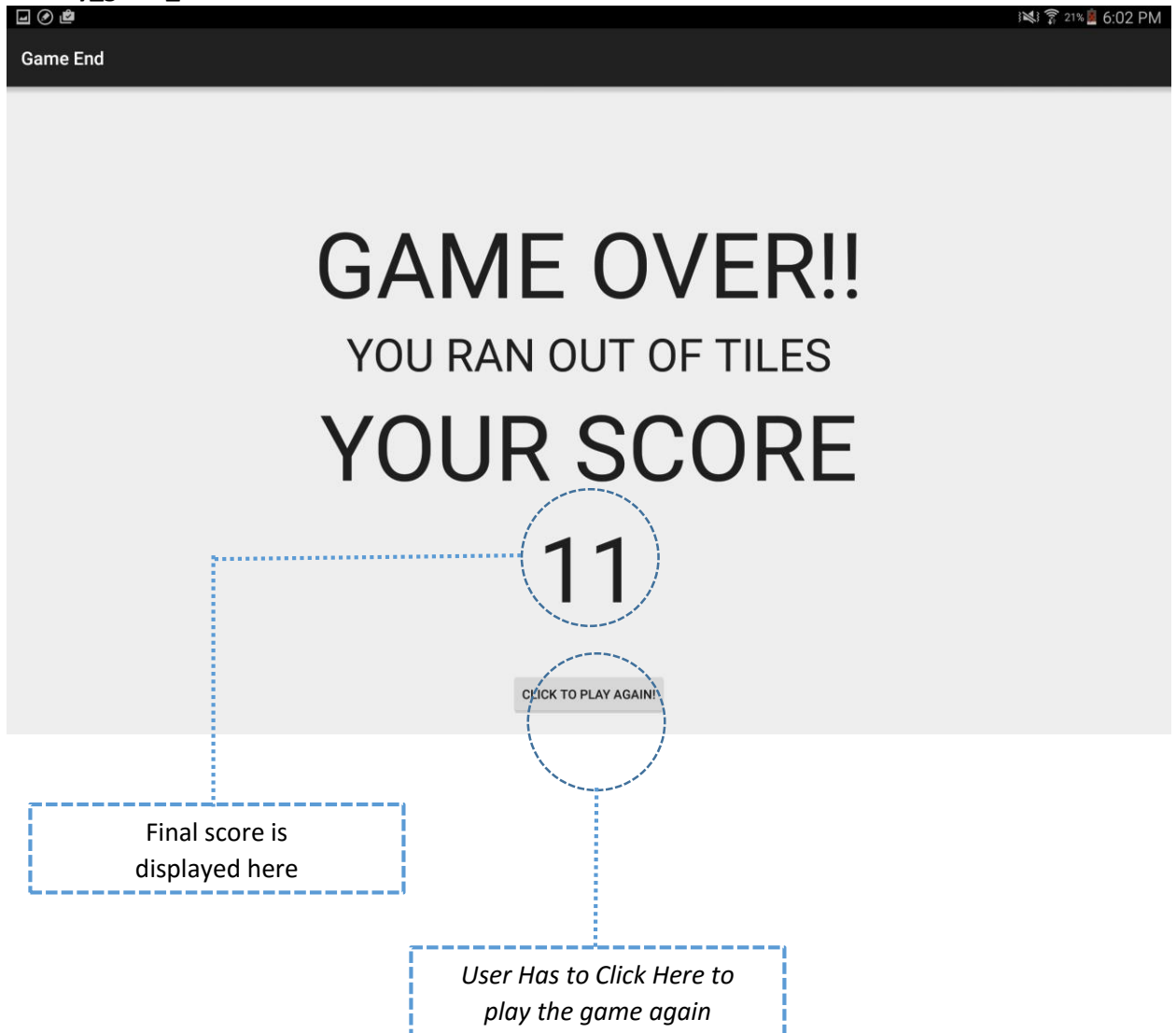
2. activity_board

This activity has the most of the things and is the backbone of the app. Nested layouts have been used in this activity to place contents in right place. The screen has been divided to two vertical parts. First one has the board and second one has elements like radio buttons to select mode, drop down, score labels and generated tiles. Description of each elements can be found below.

Color and Symbol Selection by using the dropdown button (Spinner)

Mode Selection using Radio Buttons

Select Mode: ○ Random
⦿ Select

Color: Red ▾ Symbol: ! ▾

Preview:

!

Score: 0

END THE GAME

Users click these buttons and if click is valid, button's color and symbol change according to the preview button. Also each button successfully placed can yield to 1-4 points.

Preview changes according to user's selection is select mode and randomly in random mode.

Score gets updated here after each button click

User Has to Click Here to End Game. Game also ends if tiles are unavailable.

3. activity_game_end

GAME OVER!!

YOU RAN OUT OF TILES

YOUR SCORE

11

CLICK TO PLAY AGAIN!

Final score is
displayed here

*User Has to Click Here to
play the game again*

Classes/Java Files:

1. Mainactivity
   - Handles onButtonClick Intent and redirects user to board activity
2. Board
   - Changes the preview button according to the user's mode selection.
   - Gets random tile when user selects random mode.
   - Handles all on button clicks of the 8*12 board.
   - Validates if user gesture control selection is valid or not.
   - Updates score accordingly in case of valid selection.
   - Ends the game if conditions are fulfilled.
3. GameEnd
   - Gets final score from board.java and displays it as a TextView
   - Directs the user to homescreen if he/she wants to play again
4. buttonType
   - Holds tiles' information.
   - Stores if or not a tile has been used.
   - Stores how many times a tile has been used.
   - Also has setters and getters to above mentioned variables
5. Tiles
   - Stores information about each tiles' like grid values, whether or not it is filled up.
   - Stores tiles' color and symbol/shape of the tile
   - Also has setters and getters to above mentioned variables

AI Algorithms used:

1. No AI algorithms has been used in this project

Bug Report:

1. Bug with the way the program ends.
   For this project, we had two terminating conditions:
   - When all 73 tiles have been used: This works perfectly
   - When there are no available moves: This doesn't work. User is not informed about whether or not he/she can make additional valid moves.

Features:

1. Added: welcome Screen, toast Messages to inform users about events like button clicks, errors etc.

Log:

Date:	Work Completed
Tuesday, Jan19, 2016	• Completed Setup of Android Environment. • Tested a sample app – Eight Puzzle - by installing the app on the tablet through android studio.
Friday, Jan 22, 2016	• Completed the homescreen (activity_main.xml) • Added button click inner
Saturday, Jan 23, 2016	• Started working out with different kinds of layout. • Tried Grid Layout, didn't work • Tried Relative Layout. But, since I had a lot of elements, changing one element affected all other's position. Had to abandon this idea.
Sunday Jan 23, 2016	• Used Nested Linear Layouts to design the User Interface basic structure. • Completed the basic visual requirements.
Wednesday, Jan 27, 2016	• Added Radio Buttons, Preview Pane and Spinner Drop Down List • Tried getting input from the spinners in order to use them in the preview button. • Tried currentBtn.setTextColor(getResources().getColor(android.R.color.black)); • This didn't work as I had hard time getting the integer value of colors.
Saturday, Jan 30, 2016	• Fixed the color program, by using color."ColorType" • (Color.CYAN) worked. I was able to get int value directly. • Added onClick Button activity to 8*12 Layout • Each button/tiles is an object of Tiles Class
	• Added Scoring System • Couldn't determine if the score was 1, 2 , 3 or 4. • Used a lot of if conditions to check up, down, left and right tiles and was able to get the accurate score.
Wednesday, February 3, 2016	• Added activity_end_game activity. • Added the end game button and also programmed end game algorithm. • Only one condition worked. Program only ended when all tiles were used. • Another condition failed. The program stopped when I ran with the terminating instructions.
Thursday, February 4, 4016	• Tired fixed the error (game termination condition). Didn't work out.
Friday, February 5, 2016	• Finished all the documentations. • Tested the app for final time.