Phase 2 Walkthrough

LogIn & Register System:

- System is connected with online database and users can log in with their account on different devices.
- Checked the checkbox to remember user email and password .
- When user wants to register a new account, the information will be sent to Firebase to check Authentication.

Game Launch Centre:

- The centre is created by by RecyclerView. User can enter corresponding game by clicking the picture. This kind of design is very convenient for developers to implement a new game.

Scoreboard:

- A personal scoreboard can be accessed by clicking the floating button on the Game Launch Centre interface. It shows the highest marks of three games of current user.
- Leaderboard class is shared among three games. The scores are sorted from high to low and current user's ranking together with score are shown at the bottom of the interface.

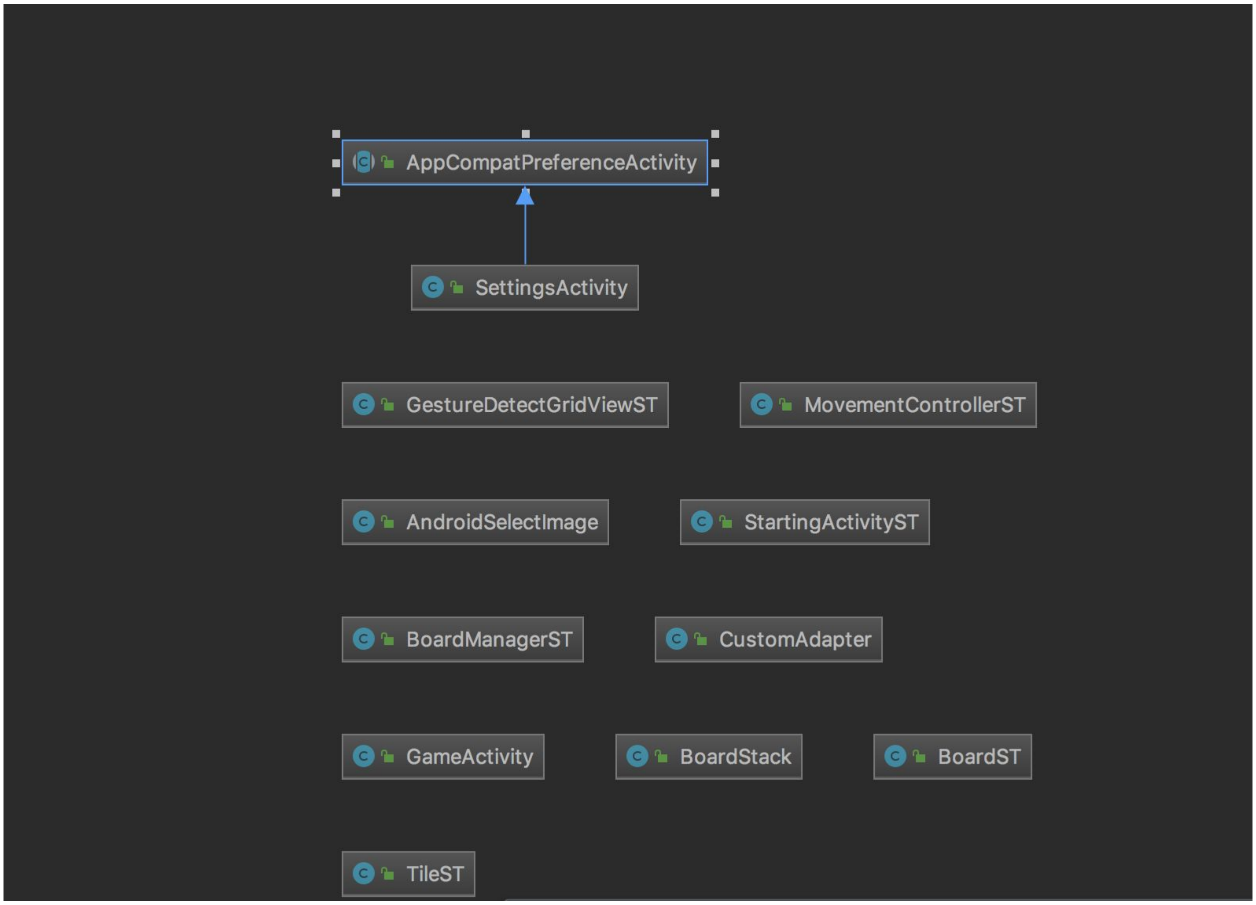
Matching Tiles:

- Main Activity
    - Handles main page
- View
    - Held inside activity screen
    - Controls game view
    - Handles utilities, saving, loading etc.
    - Handles contact with outside classes, such as updating the scoreboard
- Board
    - Handles all data relevant to a specific game
    - Handles specific game logic
    - Contains and manipulates individual tiles
- Tile
    - Controls tile image in the view
    - Holds relevant data specific to the tile

Split into 3 levels. The top level is made up of the main activity which handles the layout and functionality of the main page itself, including the view, the activity is not game-specific. The view handles the presentation of the game as well as all interactions between specific game logic and non-game specific utilities, such as saving and updating the scoreboard. The game logic is left to the Board which in turn is made up of tiles which are used to draw the actual board state.

Sliding Tiles:

- Starting Activity
    - Handles functionality and display of game’s initial menu screen
- Game Activity



    - Activity which controls the game screen
    - Handles utilities, saving, loading, etc.
    - Controls building, updating, and displaying the game board in the view
- GridView
    - Is the view displayed on the game screen
    - Controls touch events on the game screen
    - Handles contact with outside classes, such as updating the scoreboard
- Movement Controller
    - Handles logic to do with movement of the Tiles on the Board
- Board Manager
    - Handles all data relevant to a specific game
    - Handles specific game logic
    - Holds the game Board which is made up of Tile’s

        - Handles undo functionality
        - Handles board solvability
    - Board
        - A data structure built to hold a grid of tiles and manipulate them
    - BoardStack
        - A stack built to hold the state of the board for undoing moves
    - Tile
        - A data structure meant to represent the individual units a board is made up of
    - Preferences
        - Preferences are used to handle the settings, size, undo, tile image

The board and tile data structure is the basis of the game. This board is manipulated by the board manager in relation to the game logic. The data held in the board manager is used to display the board in the game activity screen using the grid view. Where the movement controller handles user touch events on the game screen. The initial menu screen allows entry into the game and changing the game settings and manual use of game utilities such as saving and loading. The undo function of solving tiles is managed by the BoardStack, which has a size and holds previous/current states of the board. Board solvability is checked by running the list of tiles into a solvability checker, which checks the position and id of the tiles against a formula, and returns a solvable board if the current configuration is incorrect.

2048

Game behaviors

- Game autosaves after every move
- Last game state will always be loaded when the game is entered, until a new game is created.
- User has maximum of 3 undos.
- When a new game is created, former undos will be cleared.
- Different user has different save file and matching score loaded
- A game score will be uploaded when a game is loss or won.

Game structure

- Gridlayout(GameView2048) as parent view and Framelayouts(Tile2048) as child views.
- Each child view uses a Tile class with different color and textView settings.
- A tileStack contains game states for undo and a ScoreStack contains matching scores.
- Each user uses their email address as local save file name to create different saves
- The new game, undo, scoreboard button are set in the GameActivity2048

Things we implemented within this project:

- Observer design pattern in sliding tiles
- Inheritance: in sliding tiles, SettingsActivity extends AppCombatActivity

What we decided not to use:

- Mention refactoring matching tiles from original setup using Tiles class, and why we switched to another structure