

WALKTHROUGH

CSC207 2018

GROUP_0613

UNIT TEST COVERAGE

- FOR ALL GAMES, THE UNIT TESTS COVERS THE BOARD, THE BOARD MANAGER, AND THE LOGICAL FUNCTIONS IN THE ACTIVITY.
- TWO TESTS ARE PREPARED FOR EACH GAME PACKAGE: ONE SPECIFICALLY TESTS THE BOARD-MANAGER (AND TESTS THE BOARD THROUGH TESTING ITS MANAGER), THE OTHER SPECIFICALLY TESTS THE LOGICAL FUNCTION IN THE ACTIVITY, WITH AN EMPHASIS ON THE CONNECTION BETWEEN THE BOARD-MANAGER (THE CONTROLLER) AND THE ACTIVITY (THE VIEW)

IMPORTANT CLASSES

- ALMOST EVERY CLASS THAT WAS NOT SUBSTITUTED BY A SUPER CLASS ARE IMPORTANT, FOR THE FOLLOWING REASONS:
 - SUPER CLASSES ARE ONLY INTENDED FOR SERVING THEIR CHILDREN.
 - IF YOU CHECK THE GIT LOG, WE SPENT THE MAJORITY OF OUR TIME WRITING THOSE CLASSES IN THE GAME PACKAGES.
 - IF YOU CHECK THE GIT LOG, ONLY 2 MEMBERS OF OUR GROUP WORKED ON THE SUPER-CLASSES, AND THEIR JOB WAS BASICALLY COPYING AND PASTING!
 - WE HAD MULTIPLE DISCUSSIONS WITH PROF. PAUL GRIES REGARDING THE DESIGN, SIGNIFICANTLY LESS THAN 1% OF THE DISCUSSIONS WERE ON SUPER CLASSES.
 - IT IS THOSE CLASSES IN THE GAME PACKAGES THAT MAKES THE GAME GREAT!

DESIGN PATTERNS

- MODEL-VIEW-CONTROLLER PATTERN:
 - THE BOARDS ARE MODELS
 - THE BOARDS' MANAGERS ARE CONTROLLERS
 - THE GAME ACTIVITIES ARE VIEWS
 - THEY COLLABORATES AND MAKES OUR GAME DESIGN CLEAR AND EASY TO UNDERSTAND.
- OBSERVER PATTERN:
 - GAME ACTIVITIES ARE THE OBSERVERS
 - EITHER THE BOARD OR ITS MANAGER ARE BEING OBSERVED DEPENDING ON THE TYPE OF GAME.
 - IMPLEMENTATION OF THIS DESIGN PATTERN ASSURES THAT THE USERS COULD SEE THE MOST UPDATED BOARD.
- ITERATOR PATTERN
 - ITERATOR PATTERN IS USED TO LOOP THROUGH THE CELLS/TILES STORED INSIDE THE BOARD

SCOREBOARD DESIGN

- THERE ARE TWO TYPES OF SCOREBOARD:
 - ONE IS GENERATED BASED ON SCORE DATA OF A SPECIFIC USER,
 - THE OTHER IS GENERATED BASED ON THE TYPE OF THE GAME

SCOREBOARD BY USER

- SCOREBOARD BY USER CONTAINS ONLY DATA OF THE CURRENT USER FOR ALL GAMES HE/SHE HAS PLAYED
- SCORE DATA ARE STORED IN A HASHMAP IN THE USER OBJECT
- *gameName* → *Score*

For example: Scoreboard for User `admin`

Game	Username	Highest Score
Sudoku	admin	90
Picture Match	admin	100

SCOREBOARD BY GAME

- SCOREBOARD BY GAME CONTAINS ONLY DATA OF THE SPECIFIC GAME

For example: Scoreboard for game Sliding Tiles

Rank	Game	User	Score
1	Sliding Tiles	admin	100
2	Sliding Tiles	Jason	90

SCOREBOARD BY GAME

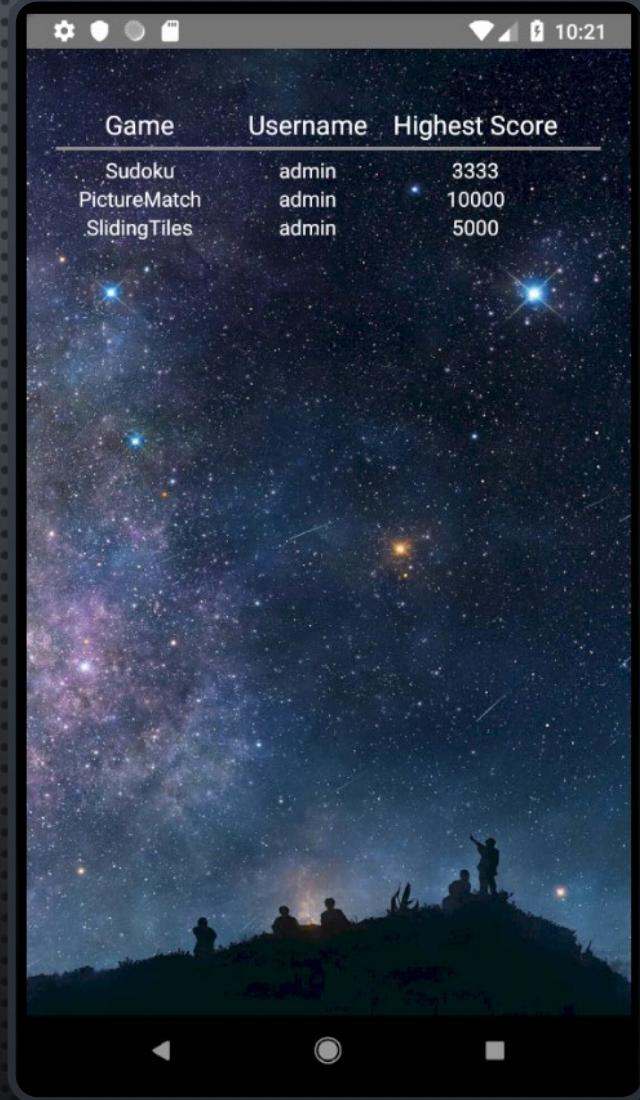
- DATA ARE STORED IN SQL DATABASE
- FOR SCOREBOARD BY GAME, WE ACCESS SQL DATABASE TO GET A LIST OF DATA (RANK, GAME, USER, SCORE) WITH A SQL QUERY (DATA SORTED BY SCORE IN DESCENDING ORDER)

SQL Database Table: dataTable

username	game	score	file
admin	Sudoku	100	admin_sudoku_data.ser
Jason	SlidingTiles	90	Jason_SlidingTiles_data.ser

SCOREBOARD DISPLAY

- SCOREBOARD IS DISPLAYED USING **SCOREBOARDACTIVITY** ON LAYOUT **ACTIVITY_SCORE_BOARD**.
- TITLE IS FIRST CREATED, TO REGULATE THE NUMBER OF COLUMNS THE WIDTH OF EACH COLUMN.
- DATA ARE TAKEN FROM USER OBJECT OR DATABASE DEPENDS ON THE TYPE OF SCOREBOARD AS A **List<List<String>>**.
- EACH INNER LIST REPRESENTS A USER, GAME AND HIS HIGHEST SCORE IN THIS GAME.
- EACH INNER LIST IS PUT INTO A **TABLEROW**, AND THE **TABLEROW** IS ADDED TO **TABLELAYOUT**



SCOREBOARD DISPLAY

