

GROUP 0601 PHASE 2

GAMECENTRE

INTRODUCTION

OUR RESPONSIBILITIES

- ▶ Kailong Huang:
 - ▶ Implement Cold War and all its features + Unit testing
- ▶ Kelvin Fan:
 - ▶ Implement Cold War and all its features + Unit testing
- ▶ Kevin Lie:
 - ▶ Implement Pong and all its features + Unit testing
- ▶ Stephen Utama:
 - ▶ Make Sliding Tiles solvable + Unit testing
 - ▶ Refactoring Sliding Tiles to fit our design and conform to other design patterns
- ▶ Yuwa Yokohama:
 - ▶ Implement Pong and all its features + Unit testing

FEATURES OF PHASE 2

- ▶ Pong Game
- ▶ Cold War Game
- ▶ Solvable Sliding Tiles Game
- ▶ Unit Tests

WALKTHROUGH

DESIGN FOCUS

- ▶ MVC
- ▶ Observer Pattern
- ▶ Object Oriented design
- ▶ SOLID
- ▶ Avoided inappropriate static method/instance variables through thoughtful design

IMPORTANT CLASSES

- ▶ Pong/ColdWar/SlidingTiles GameInfo
- ▶ Pong/ColdWar/SlidingTiles Activity
- ▶ Pong/ColdWar/SlidingTiles Controller
 - ▶ Various utility packages
- ▶ FileSaverModel
- ▶ Scoreboard

HOW THEY WORK TOGETHER

- ▶ All games mainly comprised of GameInfo, Activity, and Controller classes
- ▶ GameInfo classes serves as the "Model" in MVC
- ▶ Activity classes serve as the "View" in MVC
- ▶ Controller classes serve as the "Controller" in MVC
 - ▶ Utility classes are classes designed solely for assisting the Controller classes. Great for unit testing!

SCOREBOARD DESIGN & FILESAVERMODEL

- ▶ Scoreboards for a given game organized in GameScoreboards. Each game has its own GameScoreboards, saved in separate files.
- ▶ FileSaverModel designed to keep code clean and reduce headaches.
- ▶ Case Example: Allowed for easy implementation of scoreboard saving.

UNIT TEST COVERAGE

COVERAGE SANS EXCLUDED CLASSES

- ▶ Pong: 94%
- ▶ Cold War: 90%
- ▶ Sliding Tiles: 81%

EXCLUSIONS

- ▶ All Activity classes
- ▶ GameInfo classes

**CONCLUDING
REMARKS**

WHAT WE HAVE LEARNED AND ENJOYED

- ▶ Designing the app proved to be more difficult in comparison to writing the actual code
- ▶ Building/designing one of the games from scratch was extremely rewarding
- ▶ Although it is easy to find a solution online, it is hard to integrate it into our app using MVC/Design properly
- ▶ Unit tests took more time than expected, but not too frustrating due to good design
- ▶ Saving/Loading was intense
- ▶ Special thanks to StackOverflow :)
- ▶ Great bonding experience

THANK YOU!