

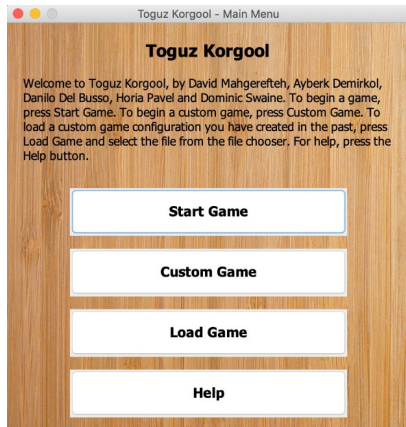
# Software Engineering Group Project

## Agile Project Design Document

By David Mahgerefteh, Dominic Swaine, Ayberk Demirkol, Danilo Del Busso & Horia Pavel

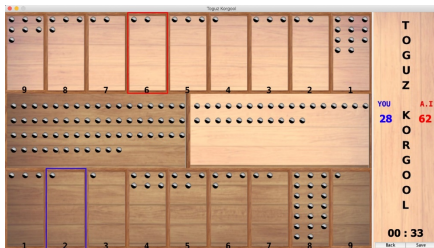
## Interacting with the System

### Main Menu



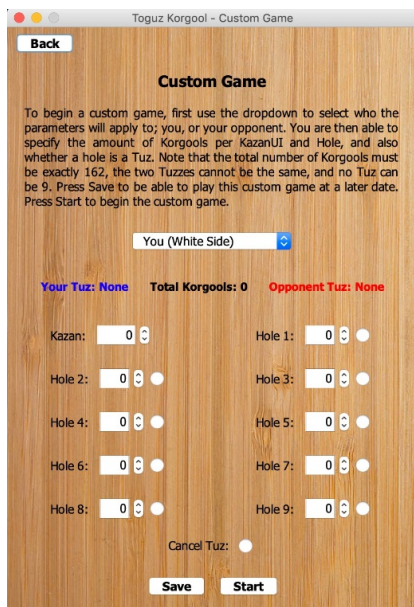
The application starts on the main menu where the user is presented with 4 options. “Start Game” starts a game, “Custom Game” directs the user to a panel where they can specify the parameters for and begin a custom game, “Load Game” enables the user to load a custom game configuration and “Help” displays a popup explaining how the game and system works.

### Start Game



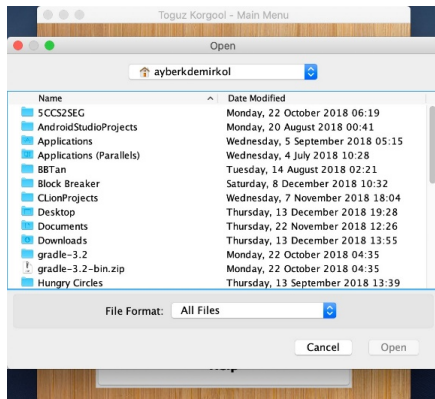
“Start Game” launches a default game. The player’s side is the lighter side and the opponent’s side is the darker one. The blue border highlights the player tuz and the red border highlights the opponent tuz. The scoreboard incorporates a timer which displays how long the current game has been running for. It also displays the score of both the player and opponent. Lastly, the “Back” button navigates the user back to the main menu and the “Save” button allows the user to save the current state of the game so that they can load and resume it at a later date.

### Custom Game



The dropdown enables the user to select who the parameters will apply to. The user is then able to specify the amount of korgools per kazan and hole, and also whether a hole is a tuz. Pressing the “Save” button enables the user to save this custom configuration so that they can replay it at a late date by loading it from the main menu. Pressing “Start” starts a game with this custom game configuration.

# Load Game



The user is able to load a previous game or a custom game configuration by pressing the “Load Game” button and selecting the appropriate file from the file chooser. Once a file is selected, a game is started with that configuration loaded.

# Help



Pressing the “Help” button outputs information displaying how the game and system works.

# Project Structure

- README.md
- build.gradle
- src
  - main
    - java
      - com/dominicswaine/  
seg\_agile\_project
        - Board
        - Logic
        - Resources
  - Test
    - com/dominicswaine/  
seg\_agile\_project
      - Board
      - Package
      - Resources
- .gitignore

Established conventions for Gradle projects were employed during development, with sources and tests placed in discrete directories and subdivided according to their programming language. The front-end user-interface and backend logic code was additionally split into two distinct Java packages, “Logic” and “Board” respectively, with the directory file path for each derived from the reverse internet domain name of an individual within the team.

