***Instructions:***
*Sign in with dvd12345*
*Password is 12345678*

***Contributions:***

- Daniel Portillo
    - Worked on getting the game implementation working (some of which is not in the submitted version)
        - Wrote the Tile class
        - Wrote the Board class
        - Wrote the GameViewController class (partially)
        - Wrote the GameScene class (partially)
        - Wrote the Array2D class
        - Wrote the Utils class (partially)
        - Wrote the Block class (partially)
    - Worked on the parse implementation for
        - Signup
        - Login
        - Reset password
    - Worked on getting the button icons
        - Play
        - Friends
        - Leaderboards
        - Settings
        - Sound on/off
- Don Negro
    - Storyboard
        - View Controllers/Segues for all buttons
    - Homescreen
        - Buttons and icons for going to other view controllers and the game scene
        - On/off switch and volume slider for background music
    - Background Music implementation
        - File conversion - .mp3 to .caf
    - Parse implementation for user accounts
        - Log in page
        - Sign up page
        - Reset password page
    - Labels and buttons for game screen

- Daniel Dao
  - Worked on the game implementation
    - Wrote the block class
    - Wrote the GameScene class
    - Wrote the GameViewController class
    - Wrote LosingScene, WinningScene
  - Wrote the friend utils class
  - Wrote the users utils class
  - Set up the Parse backend and set up/drew out the relationships (one to many friends, one to many games, etc.)
    - Laid out the table relationships
    - Friends have a one to many with a relationship called Users
      - Friends has two columns one for user, and the friend
    - Games have a one to many relationship with user
    - Challenge has two one to many relationships with users
      - The challenger and the person being challenged
  - Wrote the parse queries
  - Wrote the FriendsListTableViewController
    - Made parse queries to the Parse backend and pulled all the friendships for each person
  - Wrote the ChallengersTableViewController
    - Made parse queries to the Parse backend and pulled all the challenges
  - Wrote the leaderboards table view controller
    - Made parse queries to the Parse backend and pulled all the scores for all the users and ordered by descending
  - Set up alert controllers for friendship cases
    - **All cases will be making a parse query to the backend to check if relationship exists, if username exists, if you're trying to add yourself.**
      - Lots of nested lambdas/callbacks
    - If friendship was already there then alert controller would pop up success
    - If friendship wasn't there then alert controller would pop up failure
    - If username entered doesn't exist then alert controller would pop up failure
    - If friendship with self was made, then alert controller would pop up notifying can't add self
  - Set up alert controllers for challenge cases
    - Makes parse queries to pull the challenges and makes a join based on the challenger and the user to the games table
    - Concatenate the scores and display

    - Set up StoreTableViewController
        - Filled it with dummy items and rows

***Differences:***

- Our main difference is the hosting feature. We decided that hosting was not very desirable since people would not prefer anonymous players. Instead, when you press a player in the friend list, you have an option of challenging them. By challenging them, you compare your highest score against theirs. Your challenges will be displayed in the challenge table view, (look at home page with the two crossing swords icon).