CASSANDRA CHAPUT | CCHAPUT

MICHAEL BUZIL | MBUZIL

CONNECT 4 APP

DESIGN DOCUMENTATION

CSCI-C323

4/28/2023

DESIGN DOCUMENTATION A:

GENERAL DESCRIPTION

The app will open with a prompt to have player 1 enter their name and select a color. The same occurs for play 2. Then the connect 4 board will appear and have empty slots. The goal of our app is to create a simple connect 4 app which stores the data of previous games. Our main functionality is to create a connect 4 board for users to use to play the game. The colors chosen during the player info screens will be used to fill the board as a player choses the columns. The results will be stored using core data and will be put into a table view to show the wins and losses of games played. Table view will have the game # as section header, and each cell will have the player's name and winner or loser associated with them

OPENING SCREEN

- user 1 label
- name label & text field
- color label & button for various color options
- next user button to save data to model & reset screen to get user 2 info
    - either different view or change labels and button text?
- repeat process as user 1 did for user 2
- start game button **GAME MODE**
- connect four grid to fit screen
- use the colors for each chip for each user
- check after each player's move if there is a verticle, horizontal, or diagonal line of 4 chips
- check if board is completely filled
- if not continue to next users turn
- touch the column you want to place the chip down
- counter to keep track of how many open slots **GAME ENDED SCREEN**
- table view will have the game # as section header, and each cell will have the player's name and winner or loser associated with them
- save the history of games played and who won by persistent storage

WIREFRAME DIAGRAM / USER INTERFACE SKETCHES

MobileWireFrame.pdf (https://github.iu.edu/C323-Spring2023/Azure/blob/a999ab461a46afc671381543b556b4ad7c901b15/finalproject/wireframe%20copy.jpeg)

UML USE CASE DIAGRAM

Assignment03_UseCaseDiagram.pdf (https://github.iu.edu/C323-Spring2023/Azure/blob/a999ab461a46afc671381543b556b4ad7c901b15/finalproject/Assignment03_UseCaseDiagram.pdf)

DESIGN DOCUMENTATION B:

DETAILED UML CLASS DIAGRAM

UMLClassDiagram.pdf (https://github.iu.edu/C323-Spring2023/Azure/blob/a999ab461a46afc671381543b556b4ad7c901b15/finalproject/UMLClassDiagram.pdf)

DESIGN DOCUMENTATION C:

UML STATE DIAGRAM

Assignment03_UMLStateDiagram.pdf (https://github.iu.edu/C323-Spring2023/Azure/blob/a999ab461a46afc671381543b556b4ad7c901b15/finalproject/Assignment03_UMLStateDiagram.pdf)

REQUIREMENTS: USER INTERFACE:

- input: text, buttons, touch, other GUI elements
- output: text, line graphics, other GUI elements
- at least 3 (three) separate views, organized within at least one **container view controller**, e.g. tab view, split view, etc.
- and at least one **table view controller**

STORAGE:

- Persistent settings/preferences, with the user interface for settings to be either implemented within your app, or by using the iOS Settings app; all relevant preferences need to be preserved between different app runs/launches, or if the app gets force-quit by the user.

FRAMEWORKS COVERED IN CLASS (2):

- core data (persistent storage)
- core graphics

FRAMEWORKS NOT COVERED IN CLASS (1):

- core haptics