COMP47390: CocoaTouch

Individual Assignment 5: Connect4
January 10, 2019

Table of Contents

Objectives

In this assignment, you will build an iOS game to play Connect4 also known as 4-in-a-Row. Connect4 is a two player connection game in which players first pick a colour (red or yellow), and then take turns dropping a coloured disc from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the lowest available space within the column. The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs (`https://en.wikipedia.org/wiki/Connect_Four`).

Your app will let the user play against a deep learning bot, called α-C4, trained to play Connect4, It is packaged into an iOS framework and a startup project demonstrate how to use its API. Your task is to build a responsive, dynamic user interface for the game so that a user can play against α-C4.

The description of this assignment is intentionally vague so that you can express your creativity both in terms of programming skills and UI design. Screenshots are indicative of functionality and your app should not necessarily try to replicate exactly the look and feel.

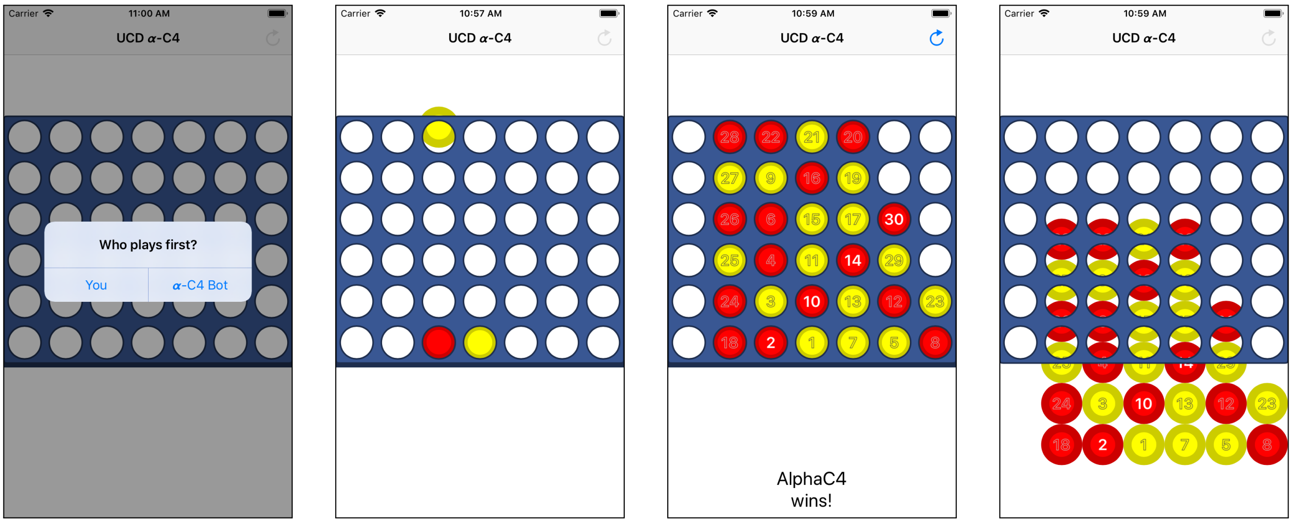
Before coding, you are strongly encouraged to do some architecting and decide what should go into a dynamic behaviour, a controller or a view subclass. There are many acceptable solutions, however devise an architecture plan.

You are required to describe your solution and software architecture in a concise report (pdf format, max 8 pages) illustrating your document with appropriate code segments and screenshots.

Functionality and General Instructions

Keep your game simple and ensure the required functionalities are all complete. Figure 1 depicts some typical screenshots of the Connect4 game. However you are encouraged to propose your own design and not try to reproduce them exactly.

- Create a Connect4 MVC using a Dynamic Animator. *Hint:* you may want to create a `DiscBehavior` subclass of `UIDynamicBehavior` and call `addChildBehavior(...)` to add all associated behaviours to itself.
- When a disc is dropped, some animation must occur before it sets in its final board location (demonstrate how to animate changes to a `UIView`, adding appropriate collision, gravity and elasticity behaviours).
- Use gesture recognisers to let the user interact with your game:
  - a tap gesture to drop discs in columns,



Who plays first?

AlphaC4 wins!

Figure 1: Connect4 App – Screenshots

- swipe gestures to clear the game. animating the drop of all discs through the bottom of the screen and reseting for a new session.

- When the game session ends, your app should display the game's outcome with a message, highlight the winning discs, and show the play sequence by numbering all discs.
- Use custom views with vectorial rendering (use `UIBezierPath`) for the vertical game board and discs. Do **not** use bitmap rendering (severe grade penalty if you use `UIImageView` and `UIImage`).
- Ensure your app is responsive to various interface orientation.
- Add some persistence to your app so that the game states are not lost between application launches
- **Advanced:** use CoreData to save game sessions along with a tab bar controller to present an MVC with game statistics and animated replay of previously completed games.
- **Optional:** integrate your app with GameKit adding leaderboard and other suitable game features in a second tab a suitable MVC to show game statistics and the replay of any past games

Notes

You are not permitted to use SpriteKit or GameKit. Instead, you must use the UIKit physics engine for animations. Discs should participate in collisions with the board and eachother .

Figure 2 provides a possible storyboard architecture for your App.

Submission

For this assignment, testing will be done by running the project on the iOS simulator. We will be looking for the following:

- Project code and report describing your solution:
  - zip archive of your Xcode project called `<Firstname>_<Surname>_<StudentNumber>.zip`
  - pdf report called `readme-part<#>`
- Your project should build without errors or warnings.
- Your project should run without crashing.

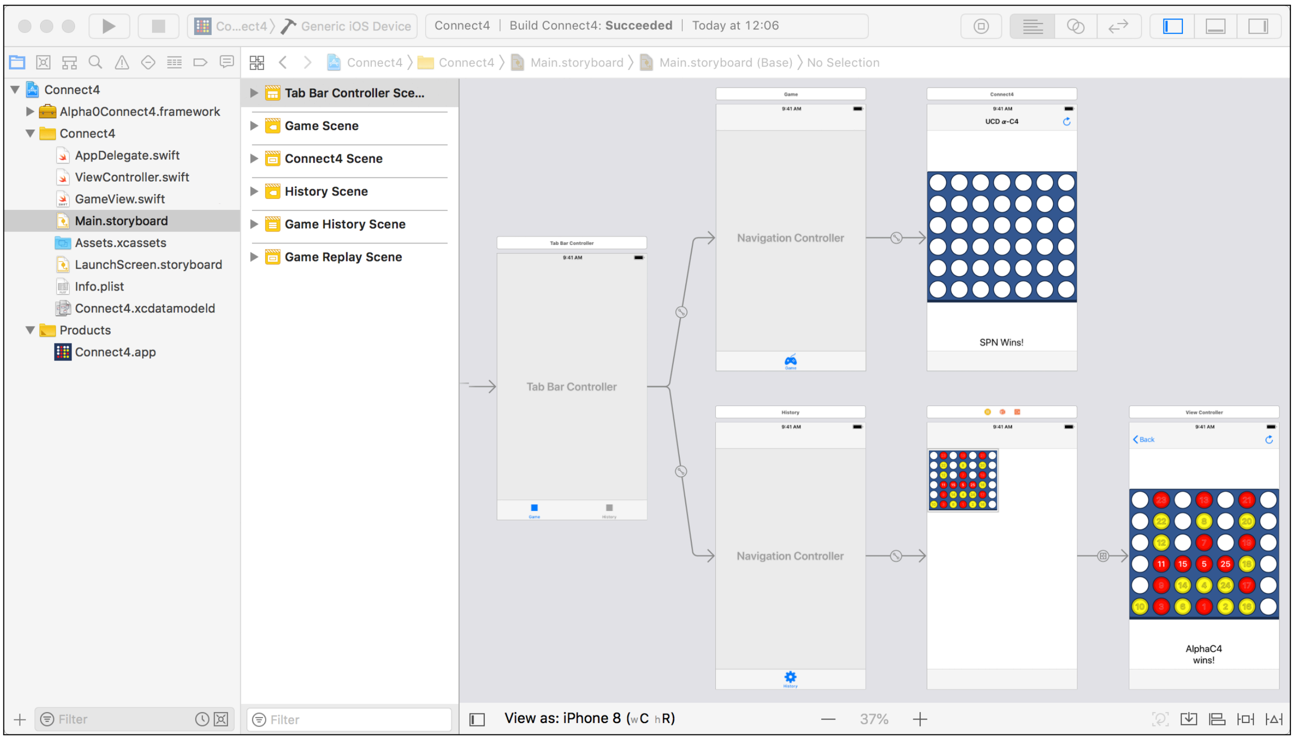


Figure 2: Possible Storyboard for the Connect4 App

- Your project should have a clean user interface, UI elements arranged logically, aligned neatly, etc...
- We will be verifying that all fundamental concepts are understood.
- Ensure your code is easy to read and not visually sloppy (indentation, etc...). Make good use of object-oriented design principles (avoid code duplication, use inheritance appropriately, etc...)
- Your solution should elegant and your code is easy for someone to read (right amount of comments, appropriate variable/method names, good solution structure, self documenting methods, etc...)

— END OF ASSIGNMENT 5 —