

UOIT CSCI 4100U | Mobile Devices | Term Project

Phase 2: Design Document and Prototype

Development Philosophy

Iterative design functions strongly with a small team and for applications like this one which undergo many changes to the original design frequently. For these reasons we are making use of this design philosophy.

Requirements and Constraints

Requirements:

- Network connectivity, likely over a socket connection, to allow for multi-player games.
- Set up archive files to store the state of the game if the user quits in the middle of a game or is interrupted, as well as storing high scores.
- Different game modes, for example time attack, and time trial.
- Audio for the selection of tiles, as well as allowing the user to select their own background music
- Increased Accessibility through the support of multiple languages.

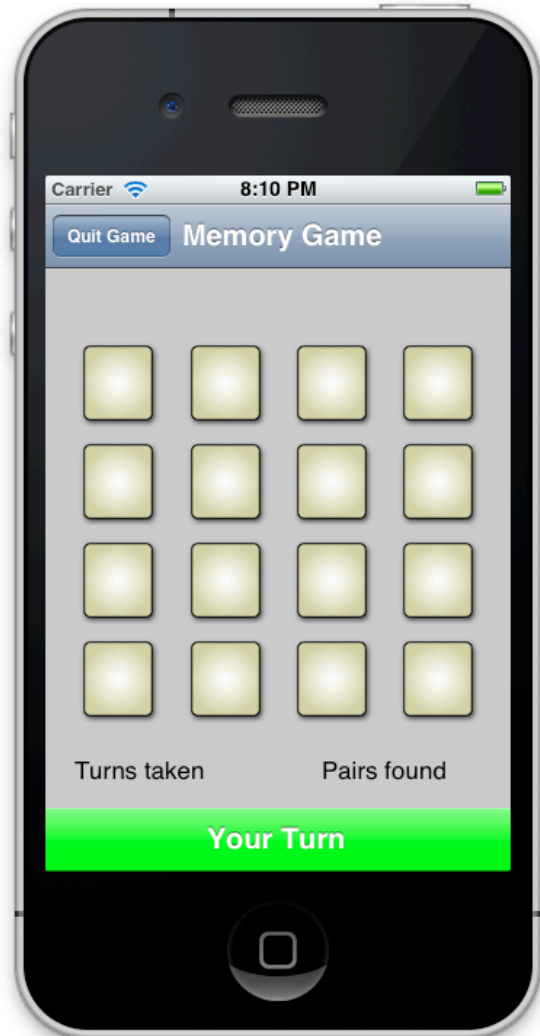
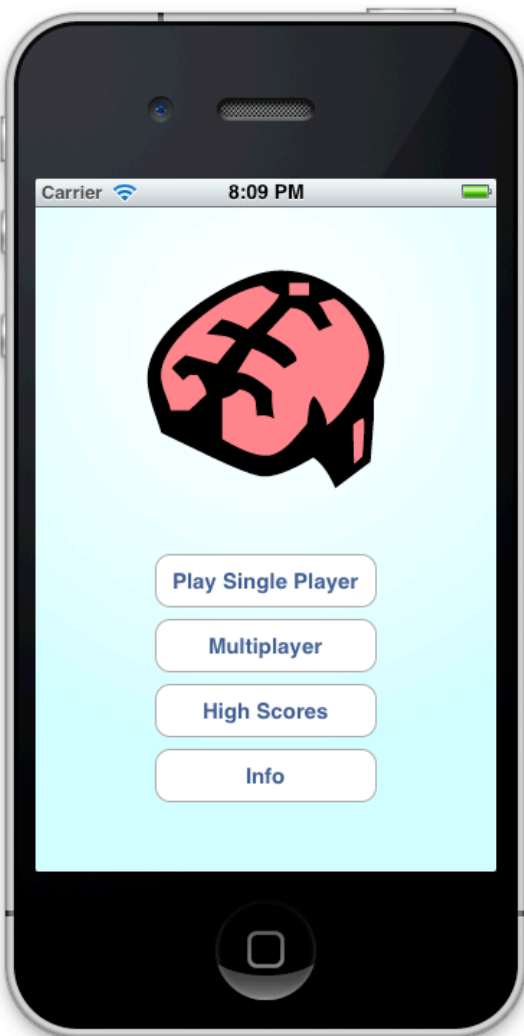
Constraints:

- Running the application in a sandbox, is a constraint placed upon us due to the nature of iOS development. In our particular case this should not be an issue as the game will not require memory allocated to other applications, however depending on how networking is done, this may change.
- Due to the mobile nature of the application, we must be able to resign control to the device very quickly, and when the user resumes have the current game resume play.

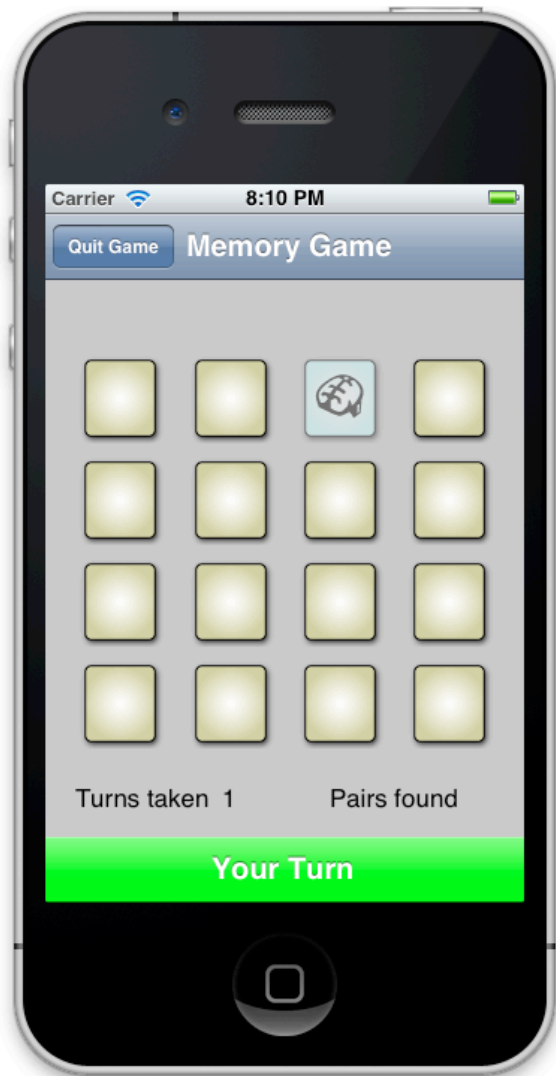
Prototype Description

The prototype only supports a single player game for now. The game will not reset the cards once it is done, or after going back to the main menu(the app has to be reopened).

UOIT CSCI 4100U | Mobile Devices | Term Project
Prototype Screenshots

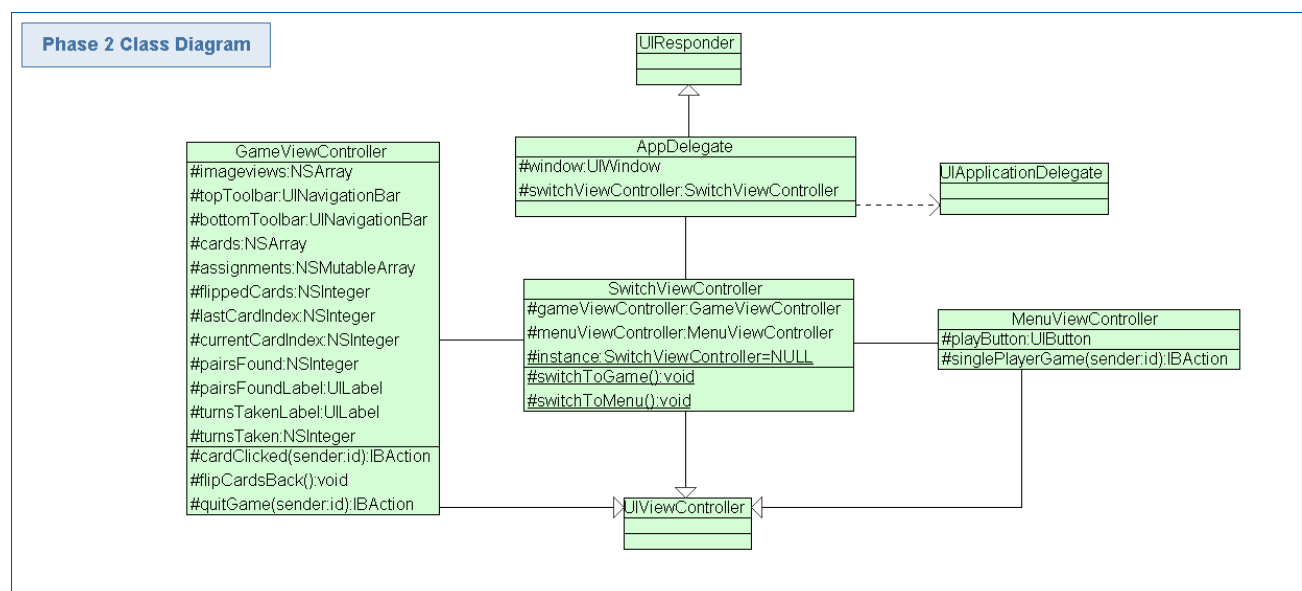


UOIT CSCI 4100U | Mobile Devices | Term Project



UOIT CSCI 4100U | Mobile Devices | Term Project

Class Diagram



AppDelegate	Aside from its known behavior, our delegate instantiates a SwitchViewController and sets it as the root view controller
SwitchViewController	This class contains functions to switch from one view to another
switchToGame(void)	This function will change the current view to the single player game mode view (through a gameViewController)
switchToMenu(void)	This function will change the current view to the main menu view (through a menuViewController)
GameViewController	This class contains all the logic for the single player game, as well as handling the button presses in the

UOIT CSCI 4100U | Mobile Devices | Term Project

	single player.
cardClicked(sender.id):IBAction	Checks when a user touches a back facing card and flips it up. If two cards have been flipped, it calls the flipCardsBack function.
flipCardsBack(void)	Checks whether the two cards that were flipped were the same. If they're the same, it will hide them from the view, if they're not it will flip them face down. It also keeps track of how many turns have been taken and how many pairs have been found.
quitGame(sender.id):IBAction	This function calls the switchToMenu function from SwitchViewController.
MenuViewController	This class is in charge of handling events from the menu view. Currently it only has the single player button implemented (the rest are not wired yet).
singlePlayerGame(sender.id):IBAction	This function calls the switchToGame function from SwitchViewController

This is a class diagram for the prototype submitted along with this document. It is missing the classes which deal with networking aspect of the application since we haven't acquired this knowledge yet but it is still a top priority feature. There are a few views and view controllers missing as well: the multiplayer view and the high scores view. The relation between the classes is as follows: The app delegate will instantiate *one* SwitchViewController which in turn will instantiate the respective view controllers for menu and game views along with any future views we decide to add.

Additionally you have been added as a collaborator to the Github project, so that you can view the prototypes, as well as the current versions of the code.