

Advanced Mobile: iOS Development - Proposal
Student number: 15026592

Project Aims

The application, *Diner Deduction*, will be a guessing and reasoning game loosely based on the 1970 board game, *Mastermind*, by Mordecai Meirowitz¹. However, unlike the original game's emphasis on pure logical inference, the app's purpose will be to provide a fun casual gaming experience, enhanced by appealing 2D graphics and the use of humour. If published to the App Store it would be a paid game, with a free version consisting of only a few levels available for trial purposes.

The user will be asked to construct fast food dishes for someone else, with no knowledge of what they want. They will attempt to determine the desired dish through a series of repeated attempts, where they will receive feedback about the number of components they have correctly guessed. The emphasis will be on speed, so there will be a timer to track the number of seconds a user takes to successfully assemble a dish. The timer will also expire after a set amount of seconds, triggering the game fail screen.

As the user progresses through each level, the number of ingredients requested will increase to escalate the difficulty. To keep the game varied, new types of dishes may be requested, with different ingredients. These could be pizzas, ice creams, burritos or salads. Strange ingredients could also be included to introduce humour and surprise, as could flash events such as "Finland Day" or "Christmas", where the music would change and the items would become themed accordingly. These events would be triggered randomly and appear in the game for short durations to add to the hectic pace.

In Hsu and Lin's analysis of purchase intention for paid apps, social value was determined to be a key factor of an app's desirability.² Social networking may be incorporated into app as a means of increasing its social capital, personalising the experience for the user. By enabling Facebook login into *Diner Deduction*, the mystery meal demander could be one of the user's friends, adding an element of amusement. The option to share a high score on the user's Facebook page could also increase the app's reach.

Due to time restraints the game graphics will be 2D, though a future version of this app could involve food construction using AR (Augmented Reality), where the user builds virtual dishes in their own front rooms using 3D graphics. Users could then take snapshots of their creations to share with their friends on social media, further enhancing the social currency of the game.

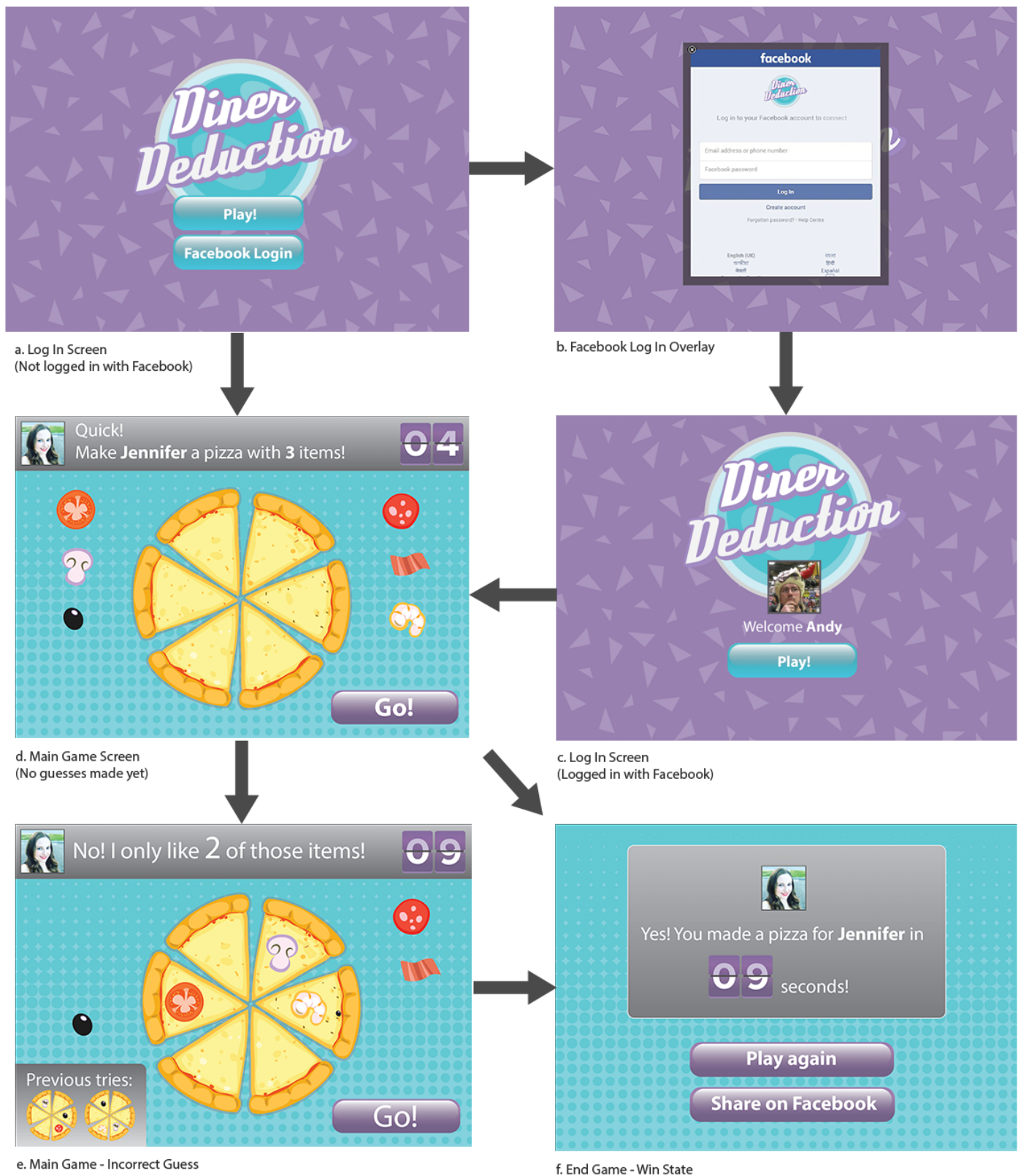
Target Devices

The game will be developed for iPhone and iPad (all variants), and the UI will be fully scalable. Considerations will be taken to ensure the UI is still viable at smaller resolutions, though the game experience will be improved on an iPad. The game will only be playable in landscape mode – the user will be prompted to rotate their device if the orientation is incorrect.

UI Design/ Blueprint

The main storyboard is displayed in Figure I, using the example of pizza. Screens a, b, and c show the flow of Facebook log in.

Figure I – App Storyboard

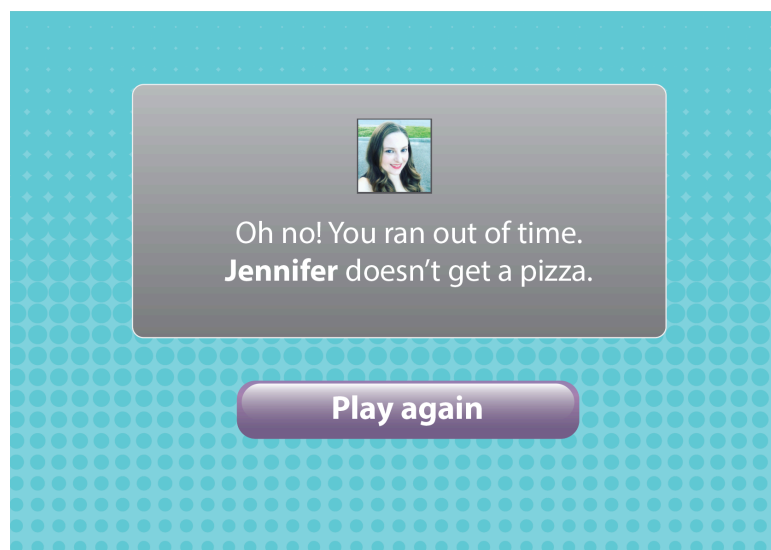


When the user has logged in, their name and profile picture will be displayed on the log in screen (c). The app will be playable without logging in to Facebook, though it means the user will only see the basic log in screen (a), and there will be no friends' pictures and names displayed throughout the game. Any names will be replaced with the anonymous "me".

At the main game screen (d) users drag and drop items with their finger onto the pizza and click the "Go!" button when ready. Items may be removed from the pizza by dragging them off it; they will then snap back to their original locations. Functionality may be added to submit the pizza when the screen is sharply tilted to make the selection even faster, thus adding to the hectic pace of the game. This would require user testing to determine if it feels intuitive. After the 'Go' button is clicked, users are either taken to the win screen (f) or the incorrect guess screen (e) to have another turn. The previous pizza combinations are indicated in the lower right corner of the incorrect guess screen to ensure the game's emphasis is on deduction rather than memory. The win screen also has the option of sharing the result on Facebook. This link would take the user to the Facebook app (if installed) or open a link to Facebook in the user's browser.

When the user has run out of time and no guesses have been made correctly, they will be taken to the fail state screen (Figure II) and given the option of playing again. If they choose to play again, a new game would begin and the user will be taken to the main game screen (d). Further buttons may be added to this storyboard to quit the app completely, perhaps on the main menu (a and c) and win screen (f) as well as on this failed game screen.

Figure II – End Game - Fail State



The buttons and menu items will use custom styles to match the game's identity, rather than using the standard iOS buttons. They will all be consistent, to ensure the aesthetic cohesion outlined in Apple's *Human Interface Guidelines*.³ All game assets will either be custom made for this project, or correct licensing will be obtained to ensure that no intellectual copyright is breached.⁴ As the app features no in-game microtransactions or adult content, and does not store the user's personal information, it may be submitted to the Kids' Category on the App Store.⁵

Design and Development Pattern and Discussions

The app will utilise the MVC framework (Model, View, Controller). This ensures that the game logic is kept separately from the view, and the controller acts as an adaptor to communicate between the view and the model. Objects defined in this way tend to be more reusable and extensible, and their interfaces better designed.⁶ Additionally, many Cocoa technologies are based on MVC principles.

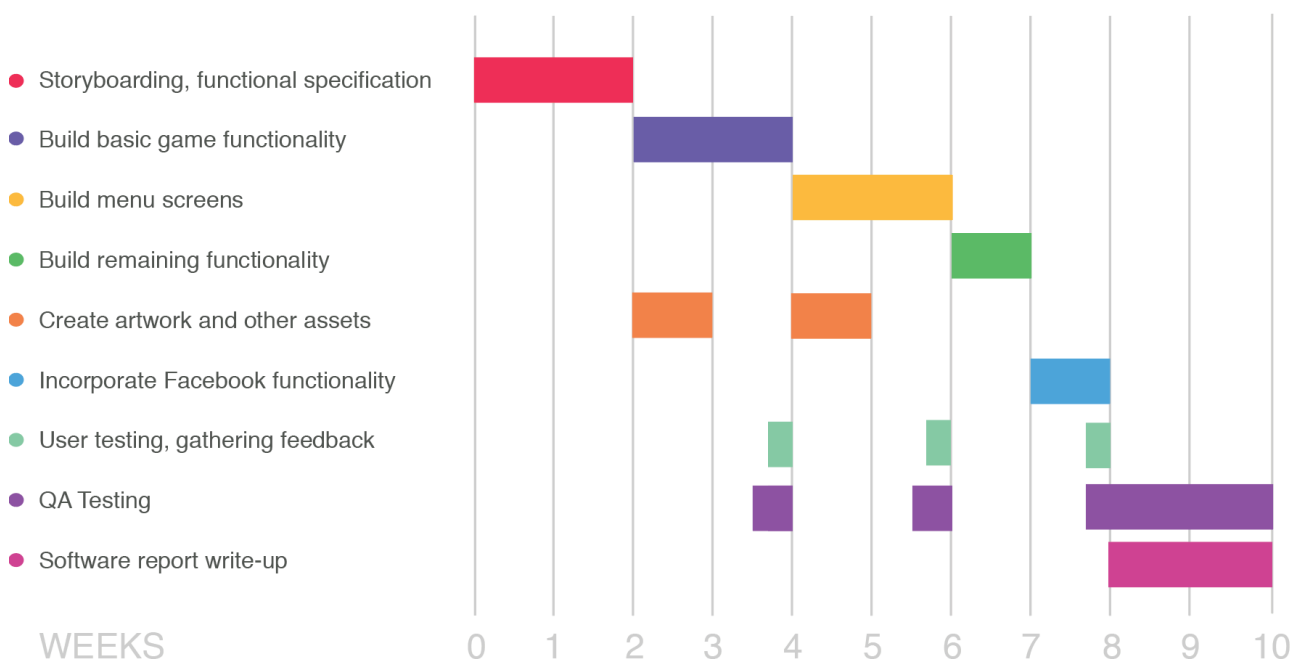
The State pattern will also be implemented to create a finite state machine for the game⁷, ensuring that varying game behaviour is encapsulated based on its internal state. This will enable game objects to change their behaviour at runtime without resorting to a large block of conditional logic. The state machine instance should be a Singleton to ensure that game state is managed in one place only. The Observer pattern might also be used with the state machine to ensure that game entities observe any state changes, and respond accordingly (for example, they destroy themselves when the game is complete).

Timeline for Development

Figure III shows the project schedule, based on the time available. Development for the game will occur as a series of short iterative cycles interspersed with user testing to ensure the app remains engaging for its audience. The idea of “minimum viable product”⁸ will be utilised, so the core game functionality will be developed first. When this minimum slice of work has been completed, it can then be audience reviewed, and any feedback will be incorporated into the next iteration. This will ensure that the game stays on track and remains relevant to its users throughout development. Non-essential functionality such as Facebook integration will occur in a later iteration.

QA testing will occur after each slice of work, and a full regression test will be undertaken at completion. Artwork and other assets will be created as needed in parallel with development.

Figure III – Development timeline



References

- ¹ Board Game Geek. <https://boardgamegeek.com/boardgame/2392/mastermind> Accessed 23rd July 2017.
- ² Hsu, C.-L. and Lin, J.C.-C. (2015) What drives purchase intention for paid mobile apps? – An expectation confirmation model with perceived value. *Electronic Commerce Research and Applications* 14, 46-47.
- ³ Human Interface Guidelines: Design Principles. <https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/> Accessed 30th July 2017.
- ⁴ App Store Review Guidelines: Developer Information. <https://developer.apple.com/app-store/review/guidelines/#developer-information> Accessed 23rd July 2017.
- ⁵ App Store Review Guidelines: Kids' Category <https://developer.apple.com/app-store/review/guidelines/#kids-category> Accessed 30th July 2017.
- ⁶ Iulia, T. Ciocarlie, M. and Ciocarlie H. (2011) Best practices in iPhone programming: Model-view-controller architecture - Carousel component development *EUROCON - International Conference on Computer as a Tool*, 1-4.
- ⁷ Gamma, E. Helm, R. Johnson, R. and Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. p305 – 313.
- ⁸ MVP: A Proven Methodology to Maximize Return on Risk <http://www.syncdev.com/minimum-viable-product/> Accessed 29th July 2017.