Project Proposal: Rekop Poker

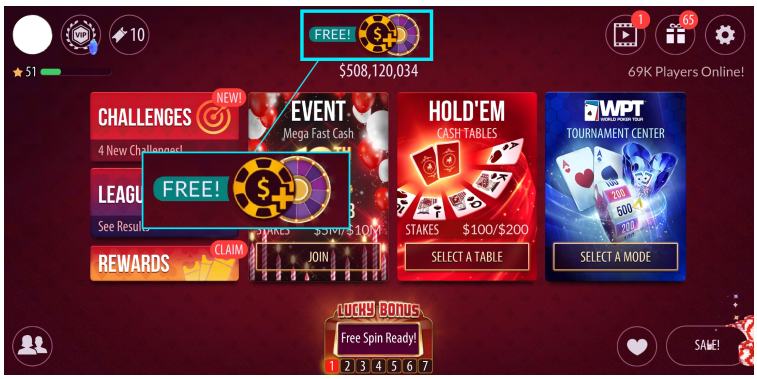
James Scully (4304469)
psyjs20@nottingham.ac.uk
G400 Computer Science

Project Supervisor: Milena Radenkovic

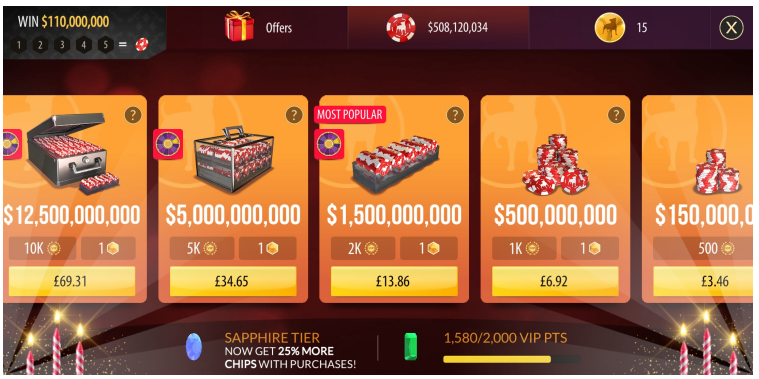
Background and Motivation

Poker is a game loved by many around the world twice over and as such has found its way into mobile applications. Many of these do not give control to the players, and are much like in real life, focused on money, rather than a casual and relaxed game. Zynga Poker for example, has been criticized in the past for "throwing too many unwanted features" in the way of the playing the game, despite seeing an increase in revenue. [1]

A quick search of the Google Play Store reveals 250 poker applications, however most of these include some form of monetization whereby the player is likely to purchase chips in order to continue playing, when they have lost all of their chips. Many people prefer mobile games to be simple, easy to use and not plagued with offers for in-game products. They also typically use misleading labels or actions in order to promote these purchases, as seen in Fig 1.



(a) Label insinuating 'FREE' chips



(b) A multitude of purchases

Figure 1: A common deception used in popular poker applications

As mentioned previously, many of these do not give much control to the user, such as customizing match settings, or the ability to host their own server/lobby to play on. In existing apps, most only have presets and are at the hands of the developers. In regards to user experience, it is best to offer control to the user, as they can tailor the game to their likings. This effect can be seen in a variety of areas not limited to software; think having clothes fit to ones preferences, or the ability to make changes to a home. [2]

Aims and Objectives

The aim of this project is to create a mobile, multi-player poker game as well as a server for people to host their lobbies.
With this in mind, we have some key objectives to consider:

- Players are able to view multiple statistics:
  - Previous hands dealt
  - Total win / losses
  - Accumulated wins
- Develop a server application which:
  - Handles disconnections / interruptions gracefully
  - Is open-source and can be used by the public for private servers
  - Is cross-platform, to ensure it is portable.
- Hands that are given out should not be repeated or predictable i.e. near-complete randomness (not time-based seeds)

Work plan

Research and Plan Documents

Firstly, research will need to be undertaken on areas to innovate upon and frameworks to be used within the app itself, for example if a game engine is going to be used - if so, what API level does it support? How will we host an official server for players to connect to? Next, we'll have to produce requirements; how we're going to meet our key objectives, and ideal features. This will be explained further with the vision and scope document which provides a general vision of a sufficient system and gives an idea of how many features need to be implemented (scope) to prevent feature creep.

Methodology and On-going processes

This project is going to be developed alongside other studies and as such will be obstructed by other pieces of coursework at some points. Because of this, the Agile methodology will be used as this allows for more flexible deadlines and progress. Other more "rigid" methodologies such as Waterfall are not suitable as game development is a ***very*** iterative process, not linear. [3]

It is also imperative that a task management system is to be used such as GitHub issues; this will allow for the project to be easily picked up from a certain point and continued, as there will be a few phases where focus on the project will need to be diverted briefly.

A journal will be kept on the repository for documenting choices made, problems encountered to aid in developing the interim report and dissertation.

Testing and Evaluation

In similar fashion to the journal, development will include regular unit tests on the back-end and can be integrated into Continuous Integration (CI), as this will generate a pass or fail on the unit tests with each commit. However, this mustn't be compared to a Test-Driven Development (TDD) approach, whereby we would write strictly write tests before actual programming, as we do not have too much time to write tests and develop a fully-functional program; using a TDD approach would not help with meeting deadlines.[4]

For the front-end, slightly less frequent testing will be done in the form of black/white box testing as automated unit tests are not suitable for user interfaces.
Evaluation towards the end of development will incorporate feedback from users, findings from quality reviews (both code and user interface).

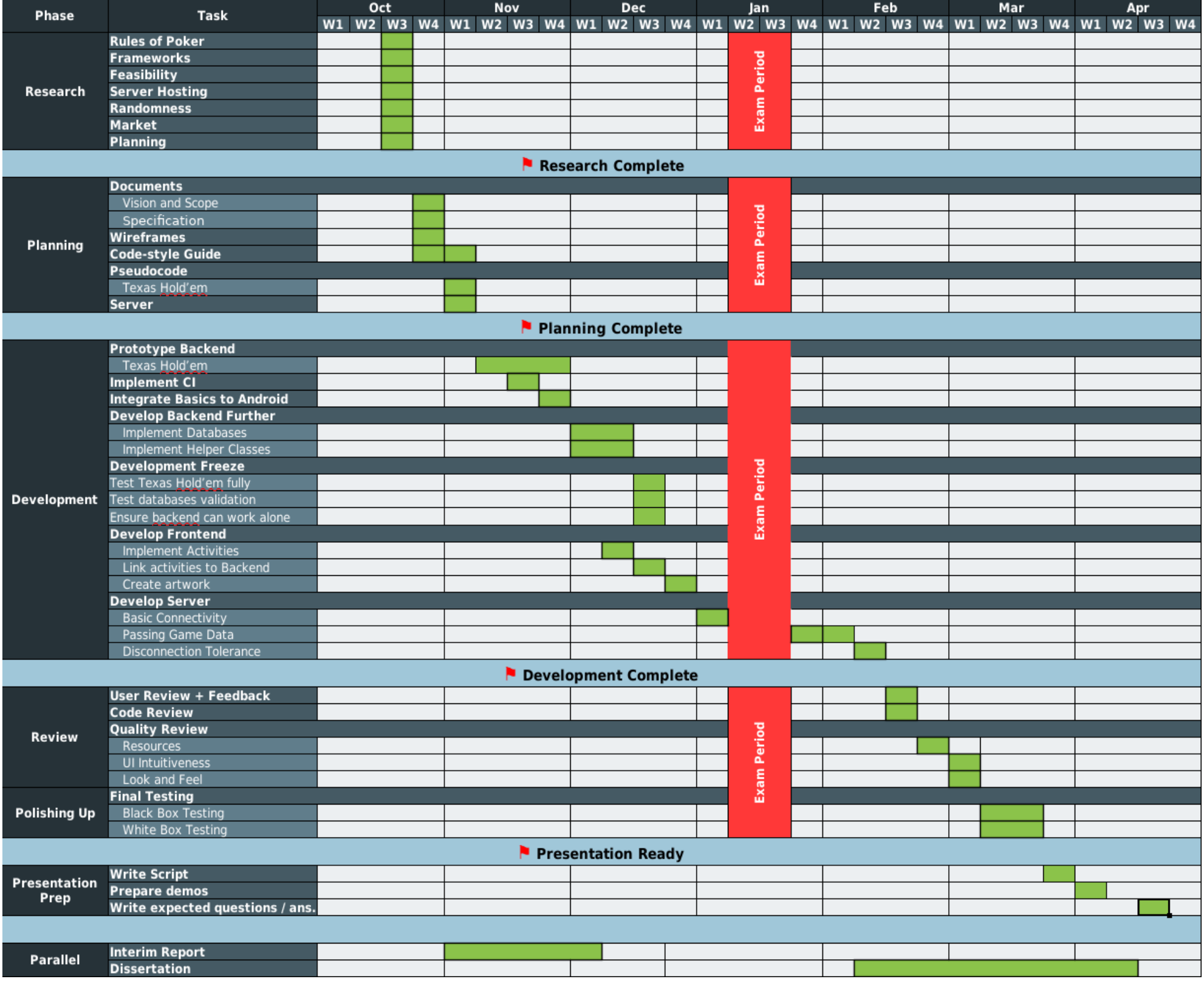
Bibliography

[1] Jeff Grubb, *Zynga stock price jumps 7% as new chief executive replaces Pincus*, 1 Mar 2016
Retrieved from `https://venturebeat.com/2016/03/01/zynga-stock-price-jumps-8-as-new-chief-executive-replaces-pincus/`

[2] NNGroup; Amy Schade, *Customization vs. Personalization in the User Experience*, 10 Jul 2016
Why Customize?
Retrieved from `https://www.nngroup.com/articles/customization-personalization/`

[3] Henna-Riikka Ruonala, *Agile Game Development: A Systematic Literature Review*, 22 Nov 2016, 5.3 Use of Agile Methods; Page 22
`https://pdfs.semanticscholar.org/4bb7/66bee969dd51a5d48ad880523ed475e7c773.pdf`

[4] Boby George & Laurie Williams, *A structured experiment of test-driven development*, 15 Apr 2004,
Retrieved from `https://www.sciencedirect.com/science/article/pii/S0950584903002040#aep-section-id36`


Phase
Task
Oct
Nov
Dec
Jan
Feb
Mar
Apr
W1 W2 W3 W4 W1 W2 W3 W4 W1 W2 W3 W4 W1 W2 W3 W4 W1 W2 W3 W4 W1 W2 W3 W4 W1 W2 W3 W4
Research
Rules of Poker
Frameworks
Feasibility
Server Hosting
Randomness
Market
Planning
Exam Period
Research Complete
Planning
Documents
Vision and Scope
Specification
Wireframes
Code-style Guide
Pseudocode
Texas Hold'em
Server
Exam Period
Planning Complete
Development
Prototype Backend
Texas Hold'em
Implement CI
Integrate Basics to Android
Develop Backend Further
Implement Databases
Implement Helper Classes
Development Freeze
Test Texas Hold'em fully
Test databases validation
Ensure backend can work alone
Develop Frontend
Implement Activities
Link activities to Backend
Create artwork
Develop Server
Basic Connectivity
Passing Game Data
Disconnection Tolerance
Exam Period
Development Complete
Review
User Review + Feedback
Code Review
Quality Review
Resources
UI Intuitiveness
Look and Feel
Polishing Up
Final Testing
Black Box Testing
White Box Testing
Exam Period
Presentation Ready
Presentation Prep
Write Script
Prepare demos
Write expected questions / ans.
Parallel
Interim Report
Dissertation