Statement of Work

INFT 3000 - Capstone

Adam Hemeon - W0434257 | **Brady Getson** - W0430313 | **Craysyn Trottier** - W0226088 | **Jayden Morehouse** - W0427457 | **Kieran London** - W0429281

Table of Contents

Project Information

Project Summary

Project SCRAPBook is an Android game built with Android Studio that takes the gameplay of a classic Rock, Paper, Scissors game which incorporates users' photos and color analysis to win. It will be available to download for free on the Google Play Store.

Project Process

Will use weekly and twice weekly SCRUM meetings to split up deliverables to work on. Discord will be the main project management software, using GitHub branch workflow to version control and merge team member's individual code.

Communication Plan:

Project ScrapBook	Adam Hemeon - W0434257 | Brady Getson - W0430313 | Craysyn Trottier - W0226088 | Jayden Morehouse - W0427457 | Kieran London - W0429281				
Communication	**Method**	**Frequency**	**Goal**	**Owner**	**Audience**
SCRUM Meeting	Online Meeting	Twice Weekly	Reviews current/finished tasks and assign current/new tasks.	Team Members	Project Team
Capstone Deliverables	Discord, GitHub	Upon Deliverable Completion	Submission of completed deliverables (preferably by due date).	Project Manager	Project Team
Issue Logging	Discord "Issues" Channel	On Occurrence	Logging of problems encountered during development.	Team Members	Project Team
Project review	Online Meeting	At milestones	Present project deliverables, gather feedback, and discuss next steps	Project Manager	Project Team
Post-mortem meeting	Online Meeting	At end of project	Assess what worked and what did not work and discuss actionable takeaways	Project Manager	Project Team

Project Budget

The budget for this project is projected to be $25CAD, which is the cost to create a developer account to publish applications to the Google Play Store. Developer time will be free as team members will be completing this project as a college assignment.

Project Deliverables

Project Management (1)

- Education & Training (1.1)
    - Git Branching Workflow & GitHub (1.1.1)
    - SCRUM as Agile methodology (1.1.2)
    - Using the camera / photo library / stock photos as part of an Android project (1.1.3)
    - Color theory and how RGBA works with Rock Paper Scissors Logic (1.1.4)
    - Deploying to Google Play Store (1.1.5)

- Documentation (1.2)
    - Develop GitHub Push/Pull/Branch Code of Conduct (1.2.1)
    - Define Coding Style Guide / Conventions (1.2.2)
    - Define Communication Plan (1.2.3)
    - Project Management Deliverables (1.2.4)

Frontend (2)

- User Interface 1.0 (2.1)
    - Wireframe (2.1.1)
    - Least viable product to code against (2.1.2)
    - Intents / Bundles navigation (2.1.3)
    - Testing (2.1.4)

- User Interface 2.0 (2.2)
    - Color Mockup (2.2.1)
    - Implement the color mockup (2.2.2)
    - Prettify and color fixing (2.2.3)
    - Create XML Animations (2.2.4)
        - Sounds Effects (2.2.4.0) [out of scope]
        - Battle / Round Winning Animation (2.2.4.1)
        - Game Winning Animation (2.2.4.2)
        - Menu Animations (2.2.4.3)
    - QoL changes (2.2.5)
    - Testing (2.2.6)

Backend (3)

- Access Camera & Photo Library (3.1)
    - Selecting up to 9 photos at once (3.1.1)
    - Storing and passing photos the user selected (3.1.2)
    - Testing (3.1.3)

- Storing Stock Photos (3.2)
    - Find stock photos online of varying colors (3.2.1)
    - Use of resources folder (3.2.2)
    - Keeping photos out of memory until needed (3.2.3)
    - Testing (3.2.4)

- RGB Composition Algorithm (3.3)
    - Determine the colour of given photo by the RGB value (3.3.1)
    - Assign Color value to a photo (3.3.2)
    - Testing (3.3.3)

- Rock-Paper-Scissors Game Code (3.4)
    - Compare the user’s and computer’s photos (3.4.1)

    - Implement Rock-Paper-Scissors rules to determine the winning photo (3.4.2)
    - Monitor the score of the game (3.4.3)
    - Testing (3.4.4)

- Computer Player (3.5)
    - Randomly use stock photos from a preselected group (3.5.1)
    - Add strategic rules to react to the player (3.5.2)
    - Balancing (3.5.3)
    - Testing (3.5.4)

- Connecting with Local Network Client-Host (stretch) (3.6)
    - Host a game (3.6.1)
    - Generate a room code (3.6.2)
    - Have client Connect to that game via WIFI/Local Network (3.6.3)
    - Error Handling (Client/Host Disconnects) (3.6.4)
    - Testing (3.6.5)

Deployment (4)
https://developer.android.com/distribute/best-practices/launch/launch-checklist

- Google Play Launch Checklist (4.1)
    - Create Dev Account (4.1.1)
    - Language / Localization Settings (4.1.2)
    - Licensing (4.1.3)
    - Play Store Listing (4.1.4)
    - Choose Distribution Options (4.1.5)
    - Etc. (4.1.6)
- Deployment of the App to the Google Play Store (4.2)
    - Publishing (4.2.1)
    - Testing (4.2.2)
- Clean Up GitHub Repository (4.3)
    - Update README (4.3.1)
    - Delete unused branches (4.3.2)
    - Set Up For Pull Requests (4.3.3)

Project Governance

SCRUM, instructor, self governed team members. Team members are the only stakeholders.

Project SCRAPBook Coding Style Guide and Conventions

Naming Convention

- File names and Class names will be written in UpperCamelCase.
- Resource file names will be written in lower_snake_case.
- All fields and methods within a Class will be written in lowerCamelCase.
- Static final fields (constants) will be written in ALL_CAPS_WITH_UNDERSCORES.
- Treat acronyms as words with respect to camel casing (e.g., XmlHttpRequest).

Java Guidelines

- Braces for a block of code go on the same line as the code before them.
- Always use braces for a block of code, even if the code within fits on one line.
- All Class members will be written in the following order from top to bottom:
    1. Constants
    2. Fields
    3. Constructors
    4. Override methods and callbacks
    5. Public methods
    6. Private methods
    7. Inner classes or interfaces
- For methods that take a Context, it must be the first parameter.
- Similarly, a callback such as a UserCallback must be the last parameter in a method.
- Don’t catch generic exceptions in a try block, and always fully handle each exception case.

XML Guidelines

- Use self closing tags for XML elements without contents.
- E.g., <TextView android:id=”example” />
- Common attributes on an element should be ordered like so:
    1. View ID
    2. Style
    3. Layout width and layout height
    4. Other layout attributes, sorted alphabetically
    5. Remaining attributes, sorted alphabetically

Project SCRAPBook style guide inspired by ribot Android project guidelines:

https://github.com/ribot/android-guidelines/blob/master/project_and_code_guidelines.md

Phase Breakdown

Milestone 0: Project Planning And Researching

Phase Description

This phase is focused on getting the project prepared to be coded, through project management, documentation, and researching possible code solutions. Team member training and education will be the first deliverable before any coding is started.

Deliverables and Assumptions

- Education And Training (1.1)
    - Define Communications Plan. (1.2.3)
    - Researching SCRUM as Agile Methodology. (1.1.2)
    - Define Coding Style Guide / Conventions. (1.2.2)
    - Develop GitHub Code Of Conduct. (1.2.1)
- Documentation (1.2)
    - Researching Git Branching Workflow & Github. (1.1.1)
    - Researching Using The Camera / Photo Library / Stock Photos As Part Of An Android Project. (1.1.3)
    - Researching Color Theory And How RGBA Works / Rock Paper Scissors Game Logic. (1.1.4)
    - Researching Deploying To Google Play Store. (1.1.5)

Milestones +Schedule

- All deliverables of this phase to be completed before February 17th 2021.

Budget +Payment

No finances are required for this phase.

Milestone 1: Least Working Product

Phase Description

This phase is focused on getting a functional user interface working as well as developing most of the backend logic to facilitate the application for the first working presentation on March 3rd. Capstone - Milestone 1 deliverable (10%).

Deliverables and Assumptions

- User Interface 1.0 (2.1)
    - Create a wireframe as a rough guideline for UI development. (2.1.1)

    - Create a *functioning* UI. Appearance is not important at this step, the main focus is on hooking up all the GUI controls (buttons, commands etc…) to the code. (2.1.2)
    - Have the project be able to go from one screen to the next and pass data through said screens. (2.1.3)
- Accessing Camera & Photo Library (3.1)
    - Selecting up to 9 photos at once (3.1.1)
    - Storing and passing photos the user selected (3.1.2)
- Storing stock photos (3.2)
    - Find stock photos online of varying colors. (3.2.1)
    - Be able to store said photos in the resources folder of the application. (3.2.2)
    - Keeping photos out of memory until needed. (3.2.3)
- RGB composition algorithm (3.3)
    - Determine the colour of a given photo by the RGB value. (3.3.1)
- Error handling & testing

Milestones +Schedule

- All deliverables of this phase must be completed before March 1st 2021.

Budget +Payment

No finances are required for this phase.

Milestone 2: Completed Product

Phase Description

This phase is focused on finishing, polishing and deploying the application. Having a more pleasant user interface with animations and possibly sound effects. All game logic works and a player can play against a computer player, as well as deploying to the Google Play Store by March 31. Stretch goal of being able to play against other players as part of a client-host connection model with error handling. Project demo as part of Capstone - Milestone 2 (10%).

Deliverables and Assumptions

- User Interface 2.0 (2.2)
    - Color Mockup (2.2.1)
    - Implement the color mockup (2.2.2)
    - Prettify and color fixing (2.2.3)
    - Create XML Animations (2.2.4)
        - Round Winning Animation (2.2.4.1)
        - Game Winning Animation (2.2.4.2)
        - Menu Animations (2.2.4.3)
    - QoL changes (2.2.5)

- Rock-paper-scissors game logic (3.4)
    - Compare the ‘colour’ of the user's picture to the ‘colour’ of the computer's chosen picture. (3.4.1)
    - Implement Rock-Paper-Scissors rules to determine the winning photo (Red beats green, green beats blue etc..). (3.4.2)
    - Keep track of the game score. (3.4.3)
- Computer Player (3.5)
    - Randomly use stock photos from a preselected group (3.5.1)
    - Add Strategic Rules to react to the player (3.5.2)
    - Computer Balancing (3.5.3)
- Connecting With Local Network Client-Host (3.6)
    - Host A Game (3.6.1)
    - Generate Room Code (3.6.2)
    - Have Client Connect To Game Via WiFi / Local Network (3.6.3)
    - Error Handling (Client / Host Disconnects) (3.6.4)
    - Network Testing (3.6.5)
- Google Play Deployment Checklist (4.1)
    - Create Google Dev Account (4.1.1)
    - Language/Localization Settings (4.1.2)
    - Licensing (4.1.3)
    - Play Store Listing (4.1.4)
    - Choose Distribution Options (4.1.5)
    - Finalize Launch Checklist (4.1.6)
- Deployment the App to the Google Play Store (4.2)
    - Publish Application (4.2.1)
    - Testing Deployment (4.2.2)
- Clean Up GitHub Repository (4.3)
    - Update README (4.3.1)
    - Delete unused branches (4.3.2)
    - Set Up For Pull Requests (4.3.3)
- Error handling & testing

Milestones +Schedule

- All deliverables of this phase must be completed before March 29th 2021.
- Project may overflow into the first two weeks of April

Budget +Payment

This phase will require a budget of $25CAD to create a developer account on the Google Play Store. This will require each member of the team to contribute $5CAD.