

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: miltil

Exquisite Corpse Drawing Game

Description

The exquisite corpse drawing game was created by French surrealists in the early 20th century, but it is still enjoyed today by many children and adults. In the “IRL” version of this game, a piece of paper is folded into either three or four segments and passed amongst a group of people. Each person, upon receiving the piece of paper, draws part of a creature on one segment, without seeing the contents of the other segments. After all segments have been drawn on, the piece of paper is unfolded and the bizarre creature the players created is revealed.

This game is a fun and creative way to pass the time, but there are several instances when it is not possible to play it. For instance, if you don't have anyone to play with, or if you don't have a piece of paper and a pencil on hand. Other problems associated with this game are accidentally seeing another player's contribution while the paper is being passed and space being taken up by storing the old drawings. These problems are solved by creating an app version of this classic drawing game.

In the app version, you can select if you are with one to four players, and can optionally associate names with these players. The app presents the first segment to the first player as a drawing surface. The app times each player's drawing time, and when it is up, prompts the current player to pass the phone to the next player. After all the turns have taken place, the app reveals the picture! (In one player mode, the middle segment is provided by the app, and the player draws the top and bottom sections.) There is an option to delete it or save it in the gallery. In the gallery, all previous saved images can be viewed. If the players entered their names before drawing, their names are associated with that image in the gallery.

Intended User

This game can be played by anyone, but is primarily targeted at kids who want to play a collaborative and creative phone game.

Features

List the main features of your app. For example:

- Organizes turns
- Provides a drawing surface
- Saves past images

User Interface Mocks

Screen 1



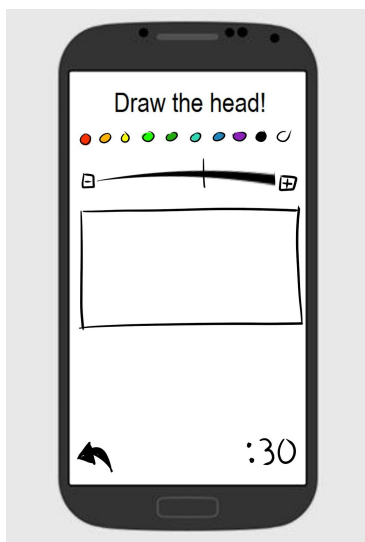
When the app is opened, the screen on the left appears, with options to create a new game, go to settings, or view the gallery. If new game is selected, the popup shown in the middle appears. The user selects the number of players participating, then clicks continue. Next is a screen to enter the names of the participants, which is optional and can be skipped.

Screen 2



Then the game begins. The blank paper is shown below a popup that says which player to pass the phone to to start drawing. If no names are provided, it would say "Player 1" instead of a name. The paper in the background will always be blank so it isn't given away.

Screen 3



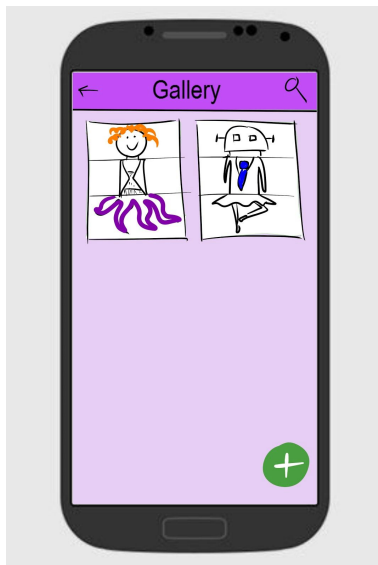
When the player clicks the “ready” button, the drawing screen appears. The player uses their finger to draw on their section. They can pick colors and line width, and undo any mistakes. There is a timer at the bottom (the amount of time given to players can be adjusted in the settings) and when it runs out, the app returns to screen 2.

Screen 4



When all players have finished with their section, the full picture is revealed! Players can choose to discard the image or save it in the gallery.

Screen 5



This is the gallery! The gallery stores images and any player names associated with them. The floating action button starts a new game.

Key Considerations

How will your app handle data persistence?

I'll build a content provider.

Describe any corner cases in the UX.

If the back button is pressed while a game is in progress, a popup appears that asks if the user is sure they are done with the game. If the back button is pressed again, the current game is deleted.

Describe any libraries you'll be using and share your reasoning for including them.

To create the drawing interface, I'll use Android Drawable View (<https://android-arsenal.com/details/1/1510>). This will make creating the drawing surface a lot easier, but I will still have to create the GUI.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Configure libraries
- Set up accounts for Google Analytics and Admob

Task 2: Implement UI for Each Activity and Fragment

- Create any needed images in Photoshop (main title, pencils, etc)
- Build UI for opening screen
- Build UI for drawing interface
- Build UI for reveal screen
- Build UI for gallery
- Build UI for settings

Task 3: Get drawing interface to work

- Create layout
- Integrate layout and library
- Choose colors to offer as options

Task 4: Piece together a picture

- Set up a task that takes the three sections drawn by the player and puts them on one image

Task 5: Build content provider

- Create database
- Store sample images
- Test and debug

Task 6: Create searchable gallery

- Implement a way to search through the gallery to find and display pictures with contributions from a searched name
- Use an AsyncTask to implement search

Task 7: Implement analytics and admob

- Implement these services with the app

Task 8: Build a widget

- Widget will display the most recent picture created with the app

Task 9: Finishing touches

- Check that strings are in strings.xml
- Check that RTL layout is enabled
- Check that tablet UIs still work

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"