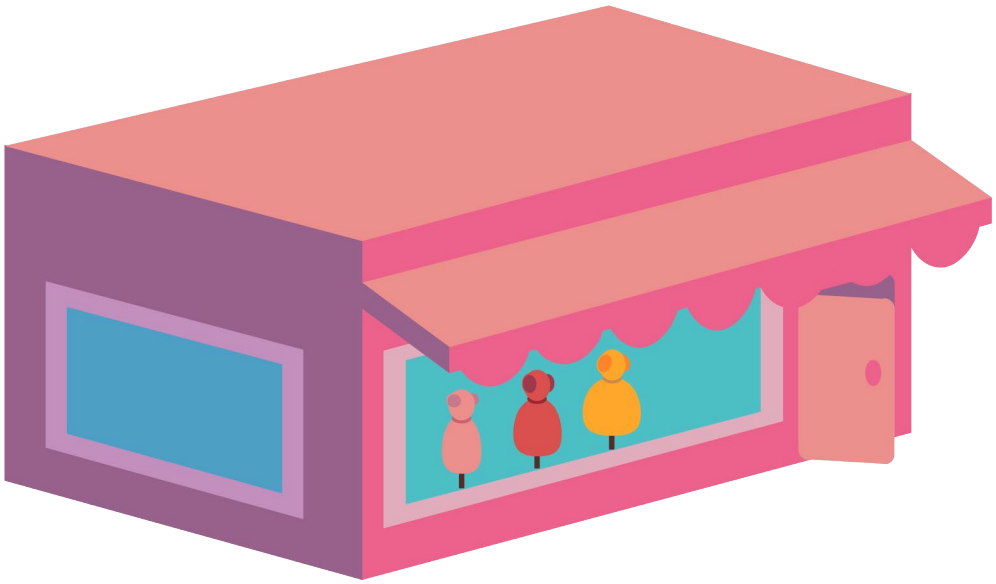
PoliCity



By Jared Mason & Priya Jurasinski

Policity was created by Priya Jurasinski and I as part of a semester long Independent Study focused on mobile game design. My goal behind the independent study was to develop my coding and game design abilities through a new, mobile perspective. To do this, it was decided we would develop an Android app using OpenFrameworks. This software utilizes the C++ coding language and allowed me to refresh my C++ knowledge while learning the new workflow and interface. Initially, we had hoped to include phone specific tools, like the camera, but we ended up focusing on utilizing the touch screen.

The first challenge of the project was convincing OpenFrameworks to build apps for an Android system. While there is an Android version of OpenFrameworks, it appears to be the newest, and least supported, version. It was recommended to run OpenFrameworks inside the Android Studio integrated development environment (IDE). After installing everything, I discovered compatibility issues that arose from the version of Android Studio I was using. Further research told me Android Studio 3.1.3 was the latest supported build, so I was required to reinstall everything onto this older version. I still encountered errors occasionally (like a specific library error that seemed to occur randomly and could only be fixed by starting a new project and copying over all the code) but usually, everything worked as expected.

Next, I created a test app to learn the software, and decide if I wanted to continue with this project. This app was simple, but it helped me learn basic input methods and what OpenFrameworks was capable of. My app allowed the user to draw dots by touching the screen while also moving another, bigger dot with directional buttons.

Since it was now clearly possible to create an Android app, the next step for us was to generate ideas for our game. We decided Policity, or “World Farming” as it was originally called, was our best idea. The original game pitch was to create a world where the player makes environmentally based choices that change the appearance of the world. The feedback we were given was to make the questions more open ended (with no clear right of wrong answer) and create a way of interacting with the game that make use of the unique aspects of the mobile platform.

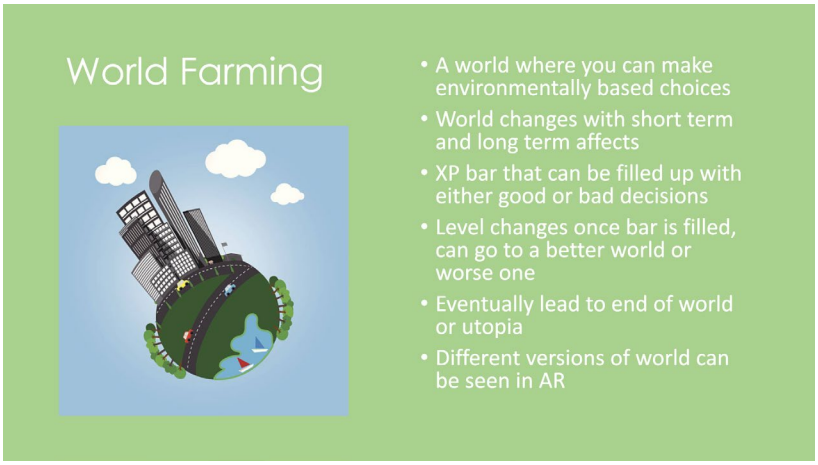


Figure 1: Original Game Pitch

With this feedback we refined our game by enlarging the world and moving the bulk of it off the bottom of the screen. Players would now swipe (our mobile game aspect) to rotate and explore the world.

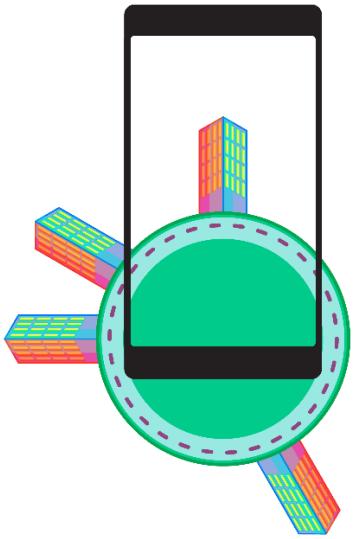


Figure 2: Mockup Game Layout

Now that we had a refined game idea, the next step was to put it into practice. I focused on creating a basic, working game prototype. I quickly created a temporary world and building asset in Adobe Illustrator, then imported and placed them using OpenFrameworks. I used push and pop functions to rotate them both around the temporary world's center when the player swiped the screen. Originally, this felt clunky as the default OpenFrameworks swipe function does not utilize the swipe's velocity. However, I learned that Android does calculate the velocity, OpenFrameworks was just discarding it in favor of simple directions. After sifting through the numerous steps between Android and OpenFrameworks, I managed to modify the swipe function to now contain velocity. This let me implement it so the world would rotate more or less depending on how fast or slow the player swiped the screen.

At this time, Priya created clickable buttons for our questions to use. I brought her code into the prototype so whenever the player tapped on a building the buttons would appear, and the player could no longer interact with the world behind them, until the question was answered.

In order to show which objects would contain these questions, I made a white circle (a glow) appear behind the object when it was close to the middle of the screen.

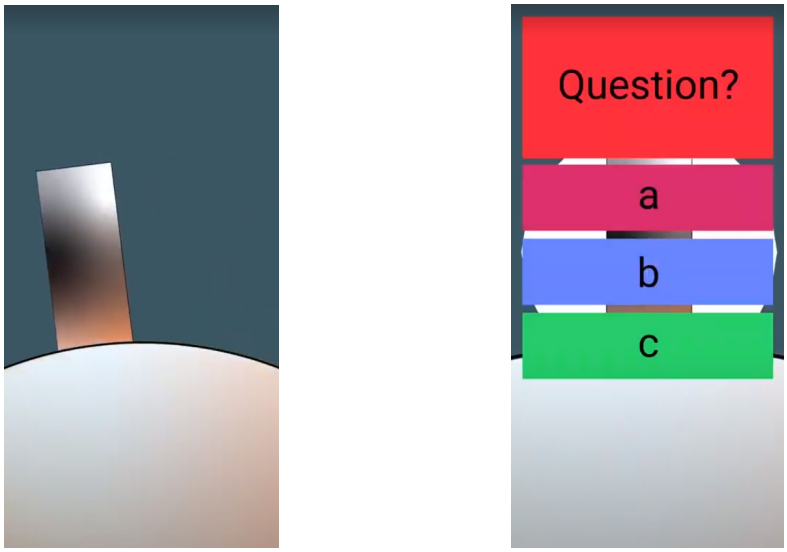


Figure 3: Prototype World and Questions

Now that we had a functional prototype, the next step was to create questions and the assets associated with them. While Priya did most of the legwork here, I did create a few assets, and assisted in brainstorming question ideas. I also took the assets and generated a white outline around them to replace the previous white circle glow. More of my time, however, was spent refining and future proofing our prototype. Once the assets and questions were complete, I

implemented and placed them in the scene. Three questions would start discoverable, and each time you answered one another would take its place till all 15 were answered.

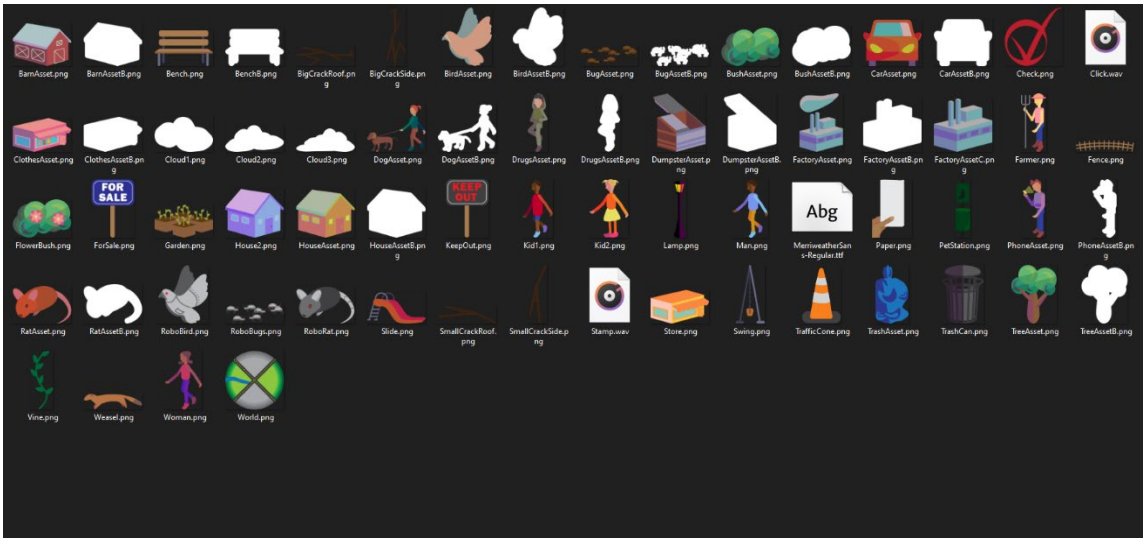


Figure 4: Final Folder of Game Assets

Our next step was to make the questions actually impact the world. To do this, we wanted to create 5 variables that would change with each question and reflect on the world. We ended up settling on happiness, economy, nature, waste, and air pollution. Reflecting these in a way that was noticeable to the player, yet not painfully obvious was another struggle of ours. Air pollution, nature, and waste were all fairly straight forward. For air pollution we decided to make the sky look more or less grey, and thus more or less polluted, as air pollution changed. As nature and waste increased or decreased, we changed the number of natural objects and trash that appeared around the world. For happiness, we initially thought to make faces appear over the characters that reflected their emotions, yet we were not happy with how this looked. We decided on affecting the colors of the people to show happiness, by making them more or

less vivid. Similarly, to showcase economy we aimed to make the buildings look more or less depreciated.

While I am happy with how the variables turned out, I do wish the changes would be more noticeable to the players. To compensate for this, we tried to make some of the questions directly change the world. Like when you decided to “replace the bushes with prettier ones” they are changed to bushes with flowers on them. With this, the player should immediately see the change, and understand why it occurred. With this, there are both obvious and subtle changes.

Figure 5: Example of Direct Change

At this stage in the project, we needed to tie everything together with a theme. We decided to cast the player as a mayor (or a similar government position) who is traversing the town and making decisions based on what they see. To show this, we chose to present the questions as

sheet of paper and have the player “stamp” their answer with their seal of approval. Here, we also reworded the questions to better fit this theme.

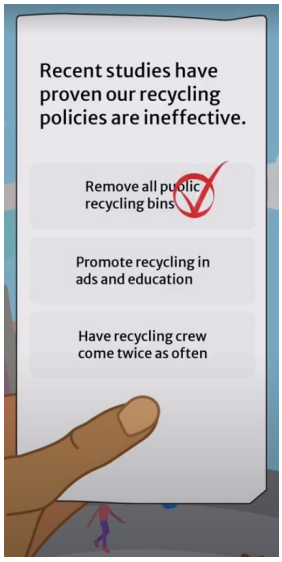


Figure 6: Final Question Visuals

While Priya worked on audio and font, I struggled with modifying the Android specific aspects of the App. By this I mean the loading screen, app icon, installable apk file. For the loading screen, I created a new xml style sheet and told the app to utilize it while loading. The app icon was problematic, till I realized I needed to update the SDK version to 25 to utilize Android’s new adaptive icons. Luckily, this proved an easy fix and simply involved changing a few numbers and clicking yes when prompted to download a missing file. Building the app was also simple enough once I understood how the process. It involved generating a signed apk by creating a

keys.jks file that stored the app's signature and password. With this, the generated apk file could now be distributed and installed on other android devices.

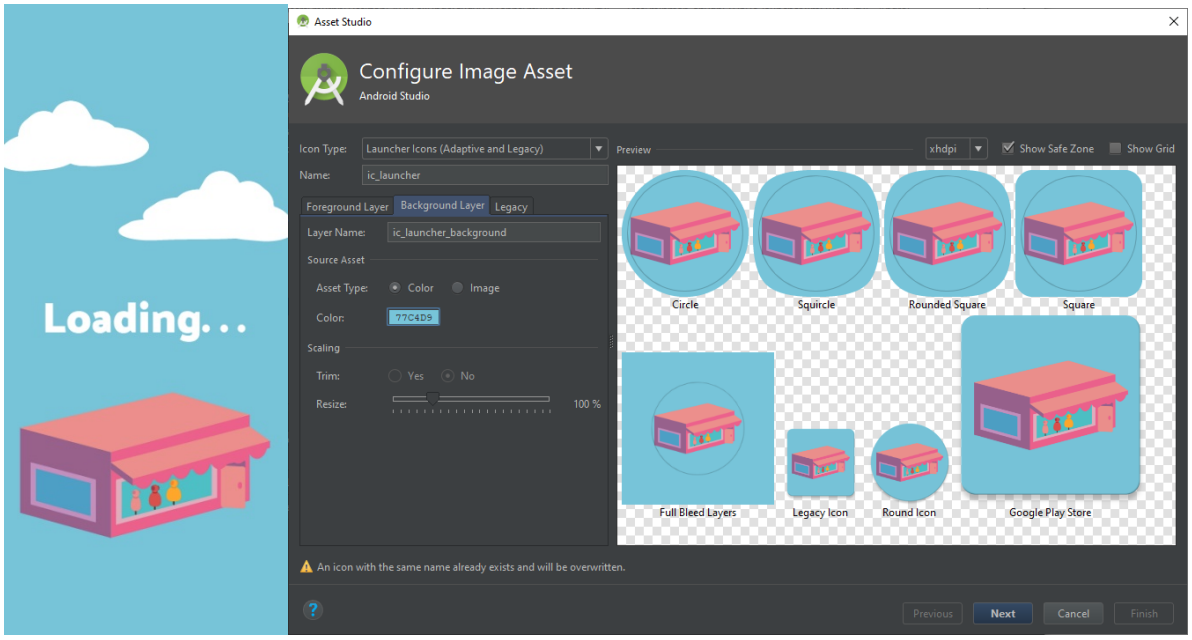


Figure 7: Loading Screen and Launcher Icon Examples

With one week left in the semester, the last thing we wanted to acomplish was implementing an ending screen. This screen is available after the player has answered all of the questions and tells the player what varriables they increased the most, and therefor cared about the most,

during their play through. It also includes a buttons to return to view the world they have creatred, or to restart the game completely and try different answers.

Figure 8: Example End Screens

With this, Policity was complete. I still hope to upload the app onto the play store, but I am satisfied with where the app itself is. While the process was frustrating at times, I am glad I worked through it. While OpenFrameworks may not have been ideal, it allowed me to learn for the new experience. Which, in the end, was my overall goal for the independent study.