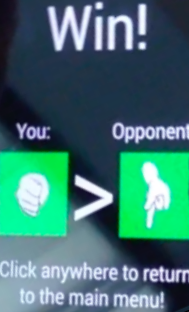
Mobile Applications for Google’s Android, EGN 1935

Final Project: ROCK PAPER SCISSORS XTREME

By: Andrew Shepard, Ryan Hubelbank, and Ladd Brown

**Description:** Our app allows the user to play Rock Paper Scissors on a tablet against his friend and his tablet. Our version of Rock Paper Scissors does not allow cheating (picking slightly after seeing the other player's move). Our app uses NFC chips to communicate between devices.

**How to Play:** The player and opponent each get a tablet and a NFC chip. Press the play button. At the bottom of the next screen the user will see three buttons, each with a picture of a hand in the shape of a rock, paper, and scissors. Click on the button to choose it and then place the NFC chip on the back of the tablet. The opponent will do the same. The user will see a message saying that they have made their selection meaning that their pick has been saved to the NFC chip. At this point, switch NFC chips with the opponent and place the chip at the back of the tablet. The user will now see a new screen stating that they have either won, loss, or tied. The user can then click anywhere to return to the main menu. There the user can play again or exit the app.



**General Future Possibilities:** We would like to give the option to the user to supply their own images instead of the preset images. There would be a “customize” button on the main menu. Once clicking it, the user would be able to take a picture of their hand in the shape of a rock, paper and scissors. The app would save the images and display them instead of the images of hands that are preset in the app.

**NFC Functionality:** The primary driver of the game would be NFC communication between two devices. As such, it was important to develop this functionality first, so that the rest of the app could be built around this primary function. The first approach we took towards getting NFC functionality was a peer-to-peer approach. By tapping two phones together, two users could send information. Unfortunately, we learned that this type of direct transfer was no longer possible; if we wanted peer-to-peer communication, we needed to do it through Android Beam. This method was a bit unreliable and compromised gameplay by allowing for data to only be transferred one way at a time. The solution we decided on took advantage of NFC chips. These chips allowed us to bypass android beam by writing data to them and having users trade chips to play.

**Sending Info To Processing:** We developed a program in processing that could display the results of the game. Unfortunately, when we imported this program into Android Studio, we found that it would be treated as its own set of programs. This meant it couldn’t be communicated with in the same manner as the other java classes. After some research into the interactions between the two, we found that both had access to the Intent object that would be used to start up the processing program. By attaching all of the necessary information directly to the Intent, we could send data from the primary program to the processing-based result screen.

**Future Possibilities for the Code:** More functionality could be developed in communication between devices. One area we could go into is Bluetooth. Theoretically, this would allow the game to be played without the use of NFC chips. Another potential communication development could be in the form of Virtual NFC Tags. Many newer Android devices have the capability to create a virtual NFC tag that acts just like a real one. Many people’s phones could then create their own NFC tags, potentially removing the need to purchase them to play. These may face other problems, however, such as requiring users to pre-allow the specific devices of people they wish to play against with Bluetooth and, in the case of virtual tags, prevent the data from being transferred between both players at the same time, hurting the game by only allowing one of the players to see the end results of the game on their device.

**General Graphics:** The graphics for this application were imported into Photoshop and then modified. The modifications in this app consisted of resizing and changing the color of some parts of the image. Simple pictures found on Google images were used in order to make the game easy to play and understandable by all wielders. These images helped us develop a certain mood that welcomed fellow RPS players. All of the pictures that were implemented into Rock-Paper-Scissors Extreme were labeled for reuse.

**Main Menu Graphics:** When the app is first opened, the home screen pops up. The background image for this screen consists of four separate pictures that were modified and arranged according to the desire of the group. In order to put this image into the app, it had to be turned into a PNG file. Photoshop allows for the image to be saved as a PNG file. This allowed the Java person in the group to apply the image as our background.

**Interior Graphics:** The next screen after the play button is pressed is where the player can choose which option they want. One can choose either rock, paper, or scissors. The little icons for each the choices (rock, paper, or scissors) were made from one original image that I separated into three separate images. I then changed the color of the images to light green to make it look more extreme.

**Colors:** The colors for the interior if this app were chosen by what we thought would be appealing to children. The home screen has a dark and gloomy background. It consists of trees with no leaves on them, and moonlight. The mood is lightened up by the rock, paper, and scissors on top of that dark background. One of those (the rock) has a face on it that is smiling. This takes the scary affect away and shows the player that this is a fun game. The light green was chosen because adolescents like bright colors and it attracts their attention. The Icon of the entire app is an image of three hands with the rock, paper, and scissors hand symbols. The hands are neon green and the background is white.

**Future Possibilities for Graphics:** There are a few things that could be changed. The background of the app icon could be changed to black. It would bring out the green more and would capture one's attention easily.

