Doodle Jump!

Project Video: https://www.youtube.com/watch?v=GaYx0dsX-SI&feature=youtu.be

Project Repo: https://github.com/saloni-saxena/doodle-jump

A randomly generated single/multiplayer game that allows users to continuouslyjump on platforms to progress farther in the game. Users can shoot at monsters to gain points and each other to win. It is based off of the original created by LimaSky.
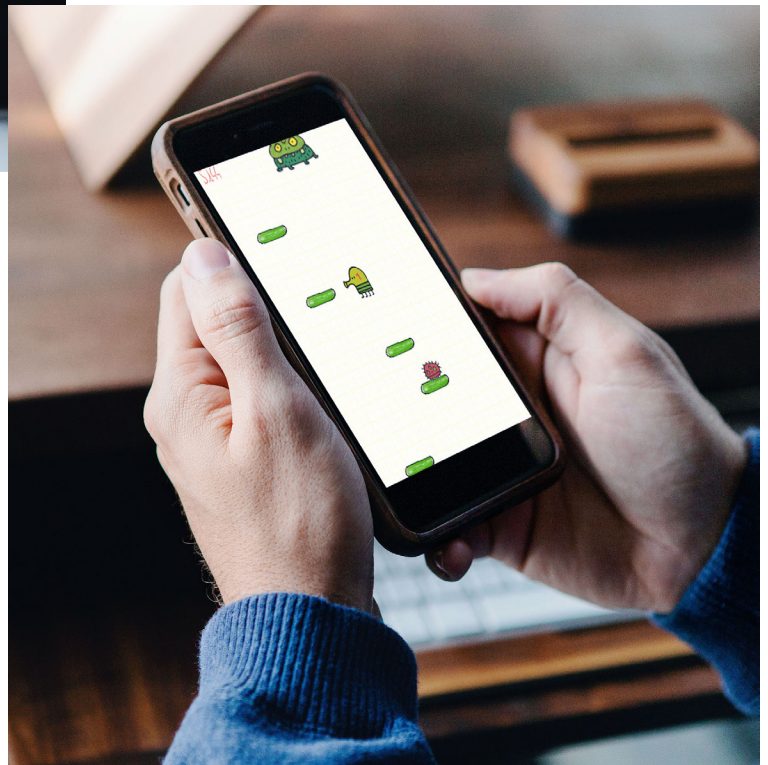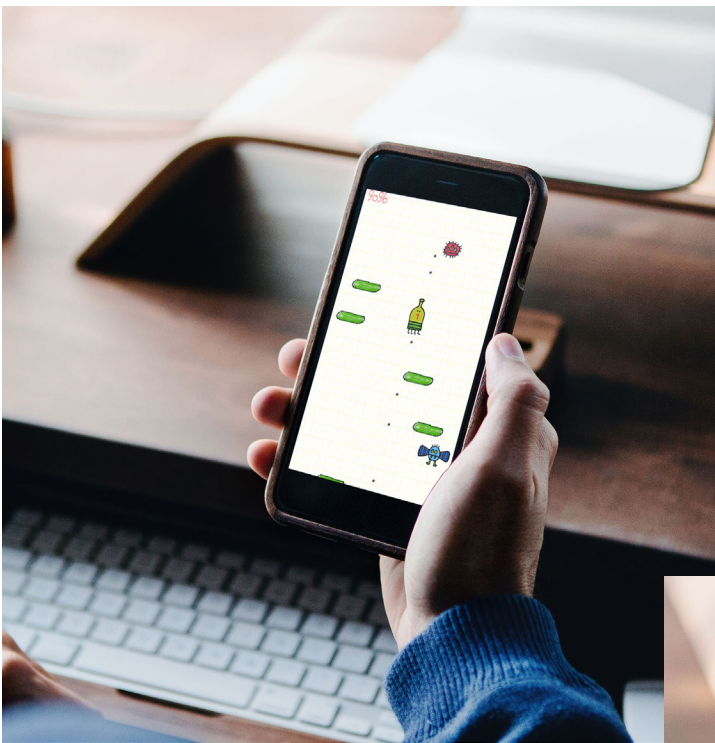
Role
Frontend | Backend | UX

Development
The most important aspects of the game require procedural generation. As a player progresses through the game, the platforms and monsters are created randomly. Therefore, the game could be played forever and a unique version of the game is created each run.

Another key facet is the implementation of physics concepts. As the player constantly jumps on platforms, the player's velocity is calculated each second to mimic the effects of gravity. The same method is used to shoot bullets.The game also heavily uses detection methods. The player must be able to detect platforms and monsters to be able to jump, shoot, and survive. Falling off of a plaform or hitting a monster result in game over.

Additionally, the game includes a multi-player feature, where two people can play each other. Players can choose to work together to shoot monsters and progress through the game, or to shoot each other to diminish each other's energies and win.

Reflections
I really enjoyed developing this game, especially since I played it constantly when I was a child. In terms of visual design, I liked the classic UI/UX of the original game so I made the decision to leave it untouched.

In terms of improvements, there are many features that I would like to add to the game in the future. For one, I would like to add a socket option to the multi-player feature, where two people can play from different computers. Additionally, to make the game more interesting, an AI could be implemented, perhaps allowing the users to play the computer or allowing the monsters to shoot bullets back at the player. It would also be fun to include different types of platforms (ex. moving, breaking, single-jump, etc.) and gifts that the player can collect to advance higher.

CyberCut

Allows users to visualize different haircuts on their face in real-time web-cam feed through augmented reality. Detect's the user's face, calculates the user's face shape, and recommends hairstyles that best accentuate their facial shape.
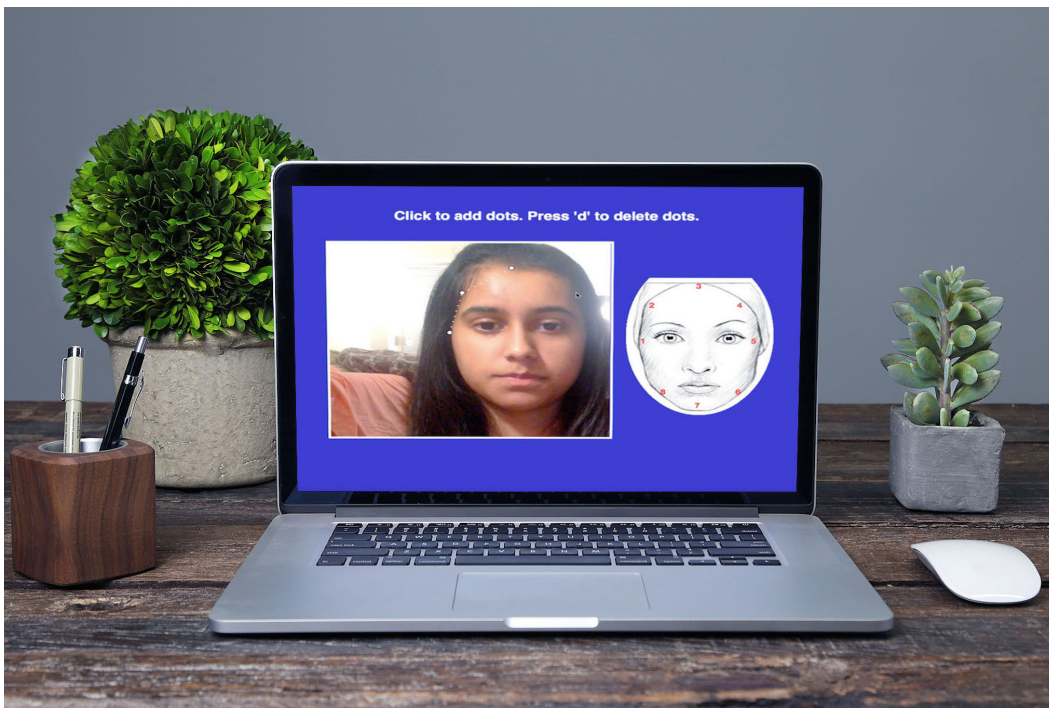
Role
Frontend | Backend | UX

Development
The most important functions in this program require facial detection. OpenCV functions activate the user's webcam and detect the users face. This information is then used to track the user's face as it moves across the screen. From there, the most challenging aspect of the program is implemented. When a user picks a hairstyle, it is projected on the user's head and scales and moves accordingly. This is done by replacing the pixels in the webcam image that pertain to the appropriate area with the image of the hairstyle. Another cool feature of the program is facial-shape calculation. The distances between key points on the user's face is used to calculate their face shape. For example, an equally wide forehead and jaw correlate with a heart-shaped face. From there, the program recommends styles that best complement facial shape.

Reflections
I found this project to be very difficult and time-consuming. Learning OpenCV proved to be quite challenging and I spent a lot of time debugging my code. However, I eventually overcame those problems and I am very happy with the finished product.

This program has a lot of room for improvement. It would be nice to add more interesting features, like photo upload and a color-picker, where a user can use a photo to try on styles and pick any rgb color. I would also like to refine the UI of the program, to make it more engaging and modern.

Walkly

Project Video:https://www.youtube.com/watch?v=G7RHAKcuBEo

Created for hackCMU 2017. Android application that encourages users to exercise by tracking their steps and pledging money to local organizations. Designed and implemented all frontend components, including UI/UX.

Team: Manini Amin, Pallavi Bannai, Arushi Patel, Saloni Saxena