Draw and Play Video Game
Milestone 2

Kyle Rosenau - Matthew Alonso - Ying Li - Vihn Nguyen

Section 1: Background

At this time this project is in the first stage. The idea for a video game that can be drawn by hand and then played on a phone was proposed to us by Scott Fairbanks, our project partner, and we have started to plan out how this will be accomplished. The team met and talked about what we want the final product to be. The user will be able to draw on a sheet of paper, put an 'x' where they want the ball to be, put an 'o' where they want the hole to be, draw lines for the walls, and take a picture of it to create a simple maze game using the phones tilting mechanics to roll the ball into the hole. The software that we have available to us and that we plan to use is xcode which allows us to write code using Swift to create an IOS application. To have access to this we have gotten accounts through macincloud which lets us use an IOS OS on a Windows computer which we all have. The more specific things we know of and plan on implementing is image manipulation to create walls from lines on a piece of paper, using the accelerometers built into IOS phones to control the ball, and use a physics engine to move the ball around on the screen. As of now, the project is in the beginning planning stages but we do have solid leads on where we want to take this project.

Section 2: Vision

This project aims to establish a solid foundation for other developers to improve upon. Our end goal is to provide a working app that implements image processing and physics manipulation as tools for users to convert their imagination into creative level designs. We do not plan to have a product that is complete and ready to ship out but, ideally, these features will pave the way for future developers to add more tools so that the game becomes less limited and more freedom is granted to the user.

2.1. In addition to the vision, please write two central hypotheses:

2.1.1 Growth Hypothesis:

Similar to games such as Gary's Mod, Minecraft, and Super Mario Maker, the Draw and Play video game will appeal to users looking to apply their creativity to create unique puzzles and level designs. Users will gravitate towards the tools provided by the game such as gravity manipulation where the user will decide how gravity affects the ball and the unique ability to draw the level on a physical piece of paper. Also, the game is available as an iOS app providing easy accessibility to users.

2.1.2 Value Hypothesis:

The value of this project stems from its various potential uses. For example, teachers and students may take advantage of the Draw and Play video game as a learning tool showing off their creativity. Alternatively, some users will use the game as a tool to create unique levels that challenge their friends and family. Finally, this project contains future potential as it can serve as a foundation for future developers to improve upon.

2.2 High-level requirements

2.2.1 Functional Requirements:

1. This App is a IOS application.
2. Users should be able to take a picture of a paper level by camera or choose a picture from the system album.
3. This App should be able to generate a playable game according to the pattern on the paper. The player's spawn point will be in the "O" position and The goal of the game will be in the "X" position. Other shapes will be considered obstacles.
4. Users should be able to control the ball by tilting the phone to change the direction of gravity of the game using the phones built in accelerometers.
5. Users should be able to win the game if the ball touches the goal.

2.2.2 Non-functional Requirements:

1. This App should be designed on a mobile platform.
2. This App can call the camera API to acquire image files.
3. This application can process the acquired images. The different patterns in the image can be separated and information about the shape and position of each pattern can be recorded.
4. The application can use the processed pattern information to generate different game components and place them in positions corresponding to the images.
5. This application contains a physics engine to enable interaction between game components.
6. The app can call the phone's accelerometer interface to change the game's gravity properties in real time.

Section 3: Prioritized Project Constraints

Time Constraints

First, we are planning to learn Swift because we are all new to developing a mobile game on the IOS platform. Next, we are going to learn the way to process images on mobile phones. In addition, we are also going to design a gravity engine. After we have these tools ready, we will start designing the game features and UI. Since we are all senior students, our time will be occupied by the lessons of each class. Thus, our time will be limited for developing this project,

however, we expect each member to finish each task APSP. It's not a mandatory requirement, but please don't put it off until it's too late.

Resources Constraints

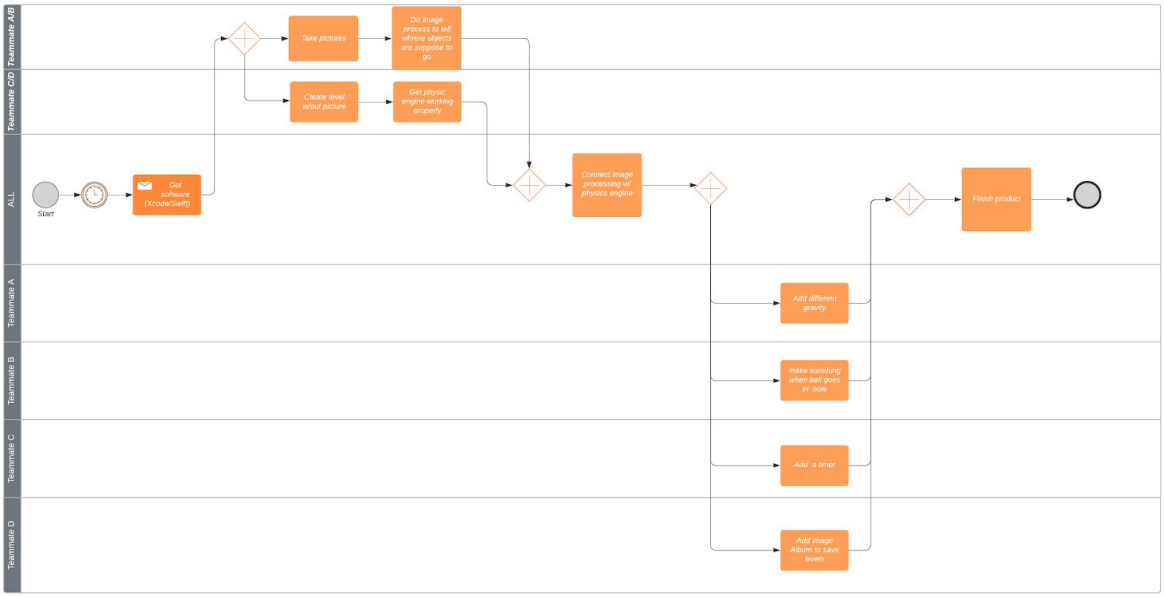
Due to us developing a IOS program, each group member needs a IOS device for testing and debugging, a MAC OS device for coding, building and publishing. In addition, everyone may need to enroll in the Apple development program for distributing our game, which is $99 per year. Everyone in our group had iPhones but no MACs. Due to budget constraints, it is not practical to have a MAC for each person, so we are going to use a cloud virtual machine running MAC OS. Other than that, we shouldn't need much of a budget in development.

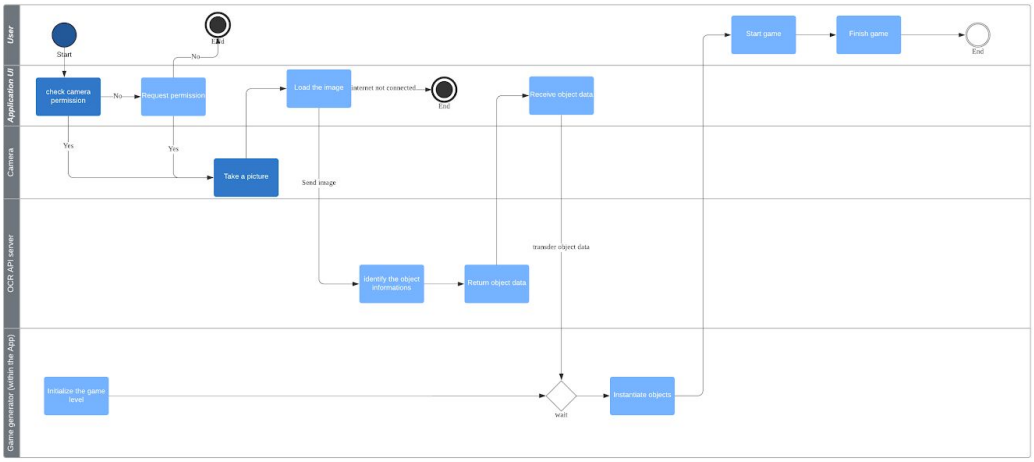
Scope Constraints

Due to the above limitations, we may not have much time to optimize the game's systems. For now, our final goal is to deliver an app that can run on an iPhone that generates playable levels based on pictures. The user must draw simple and loose lines on a clean white sheet of paper. The angle of the camera should be as perpendicular to the paper as possible and the paper should be filled with the contents of the photo. The collider of the game component generated by the application may not exactly match the edge of the original graphic.

Section 4: Scope

4.1 Process Flows:





4.2 User Stories:

As a 12 year old, I want to be able to draw a level and play that level on my phone.

As a 12 year old, I want the game to look good so that I'm not distracted by bad styling.

As a 12 year old, I want to play many times with different drawings so that I get distracted from chores and homework.

As a user, I want to be able to take a picture so that I can capture the level drawing I made.

As a user, I want the drawing to be transformed to a level so that I can play it.

As a user, I want the ball to be movable so that I can move it around the level.

As a user, I want the hole to be made correctly so that the ball will be able to roll into it.

As a customer, I want a playable app to be made so that the game is available to the user to play.

As a customer, I want the developers to keep on track and develop this product within the planned time so that I am getting what I want when I was told I would get it.

As a developer, I want the customer to know that the team has not worked with IOS coding or anything similar to it in the past so that they understand that some steps may take longer than anticipated due to learning as we go.

As a developer, I want the team to have a shared spot where all of our code is stored so that I know exactly where everything is.

As a developer, I want the team to be able to ask each other questions and get responses within a reasonable time so that development can continue and none of us are left behind.

As next year's team, I want the developers to create documentation so that I can see what they were trying to do.

As next year's team, I want there to be a working product that can be played so that we don't need to spend time debugging code that we didn't write before even starting.

As next year's team, I want the developers to stick to the project so that our starting point is in line with what the project's goal is.

Section 5: Iteration Plan and Estimate

Sprint 1:

Get everyone the software needed

Learn Swift

Be able to:

- Take a picture
- Load/ display the image
- Do some basic image processing to make some of the level (user decides what goes into it)

Sprint 2:

Finish work on picture-based level design

- Picture forms the level without user interaction

Work on the physics engine

- using the accelerometers to move the ball
- Make the walls have structure

Sprint 3:

Connect the picture/ image processing to the physics engine to make a working product

- Can take a picture that then creates the walls, ball, and hole

Sprint 4:

Get rid of any bugs/ things not working in the game

- Wall physics not working right
- Ball physics not working right
- Picture not recognizing all data
- Picture adding in extra things
- etc.

Sprint 5:

If everything is going well we could start to add the game functions

- A starting screen
- Something happens when the ball gets to the hole
- Styling
- Different gravity
- etc.