Team 11

Dylan Herrig
James Glass
Zach Bruennig
Kate Ramge
Ryan Kass

Flingshot

Project Synopsis

An infinite mobile puzzle game with procedurally generated levels using the touch screen to fling the character.

Project Description

Our plan is to create a fun and addicting mobile game with infinite replayability. To accomplish this, the game will be endless - the game only ends when the player messes up. All obstacles will be randomized, so the same pattern will not appear over and over consecutively. That is, every time the user plays the game, there will be a new challenge or arrangement to face. To do this, we will make predesigned blocks that fit together. Each block will have a difficulty associated with it, so as the player gets further the game chooses more difficult blocks/obstacles.

For our team, the technology we will use verges toward the territory of unfamiliar, while still having some concepts which feel natural to us. All team members will have things to learn and master. We have an opportunity to be very creative with our obstacles, and this freedom will allow us to make our app stand out from similar products. As a result our design philosophy revolves around generating the base game, then pack as many cool features as we can into our stages. Ultimately, we want to have a successful, (and potentially profitable) app published on the Apple and Android app stores, as well as a maintenance plan after the conclusion of this project.

Project Milestones

1.) Initial Game components
2.) Visually appealing Game
3.) Boss level components

1	**Initial Game Build**	[Name]	Fri 1/25/19	Mon 2/18/19	25
1.1	Game Boundaries & Obsticles	Dylan	Fri 1/25/19	Fri 2/15/19	22
1.2	Menu	Kate	Fri 1/25/19	Fri 2/15/19	22
1.3	Game movement	Ryan	Fri 1/25/19	Fri 2/15/19	22
1.4	Collision handling	James	Fri 1/25/19	Fri 2/15/19	22
1.5	Obsticle Algorithm	Zach	Fri 1/25/19	Fri 2/15/19	22
1.6	Testing	ALL	Fri 2/01/19	Mon 2/18/19	18

2	**Polished Game Build**		Mon 2/18/19	Thu 3/07/19	18
2.1	Additional obsticles [Algorithm]	Zach	Mon 2/18/19	Mon 3/04/19	15
2.2	Power Ups	Ryan	Mon 2/18/19	Mon 3/04/19	15
2.3	Rising Waters	Dylan	Mon 2/18/19	Mon 3/04/19	15
2.4	Character Graphics/Animations	James	Mon 2/18/19	Mon 3/04/19	15
2.5	Game movement	Kate	Mon 2/18/19	Mon 3/04/19	15
2.6	Testing	ALL	Fri 3/01/19	Thu 3/07/19	7

3	**Boss Game Build**		Mon 3/18/19	Sun 3/31/19	14
3.1	User account (high score/playtime)	Kate	Mon 3/18/19	Wed 3/27/19	10
3.2	Boss Characters	James	Mon 3/18/19	Wed 3/27/19	10
3.3	Boss Areas	Ryan	Mon 3/18/19	Wed 3/27/19	10
3.4	Ads	Dylan	Mon 3/18/19	Wed 3/27/19	10
3.5	Boss Algorithm	Zach	Mon 3/18/19	Wed 3/27/19	10
3.6	Testing	ALL	Mon 3/25/19	Sun 3/31/19	7

4	**Deployment**		Fri 2/01/19	Wed 4/17/19	76
4.1	Testing		Fri 2/01/19	Sun 3/31/19	59
4.2	Submit to Google Play Store		Mon 4/01/19	Wed 4/10/19	10
4.3	Submit to Apple App Store		Mon 4/01/19	Wed 4/10/19	10
4.4	Buffer		Wed 4/10/19	Tue 4/16/19	7

https://docs.google.com/spreadsheets/d/1qAmQyX9Q6cTNOIzZdY5mJiVfLuXhnoH3X8ZYdg443hg/edit?usp=sharing

Project Budget

Resources:

- Software
    - Photoshop, Illustrator, and Blender: Already Acquired
- Hardware
    - Computers & Mobile devices: Already Acquired

Vendors:

- Apple: App Store

Special Training:

- Unity Tutorials will aid in the creation of this app

Estimated cost: $200

- Considering that we are producing this application without a salary. Cost for creating a mobile game like this is rather low.

When Required:

- Apple app store license isn't needed until the project is finished

Work Plan

Dylan - Movement, asset creation, general UI elements, enemy interactions, step in with other elements as needed

James - Design/Implement the character and UI. General coding. Aiming to be a Full Stack developer. :(

Zach - Backend work, algorithms, database management, and general testing. :0

Kate - Menu design, implementation, and functionality. >:(

Ryan - Take Zach's algorithm for map generation and program tile placement in Unity. X(

Final Project Design

Our application will provide a fun and challenging endless runner game that offers replayability, friendly competition, and a challenge to the user. Our app will be built using Unity and its underlying language, C#.

We are choosing Unity because the features that it offers are advantageous to our goal, as well as being the most popular and professional tool available for our project, providing a variety of benefits specific to us. Unity offers an environment that can be easily exported to both iOS and Android Development. It offers a basic physics engine that can be adapted and built upon. Unity additionally offers an environment that can easily integrate a the visual level design with the C# logic coding.

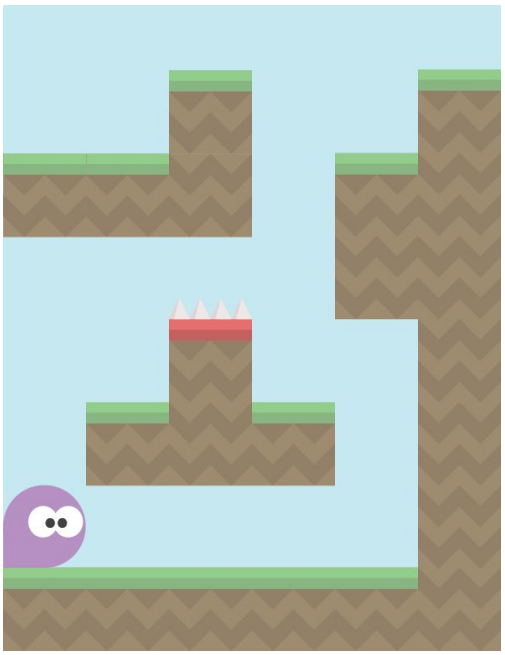
Since our character is a slime, we'll being using animation to simulate soft body physics. This will enable our character to behave a lot like jello, meaning it'll bounce and wiggle when colliding with other objects. At the moment, this won't have an impact on the core gameplay of our game; we plan on using it more as an aesthetic choice. From a development standpoint, this will be additional work with animations, illustrations, and little more. As for movement, we initially wanted to have realistic physics to move our character around. However, upon further discussion and brainstorming, we decided a more fast paced and enjoyable movement option would be to limit movement to the four cardinal directions. Instead of getting hung up on how to move the character around, we wanted the player to focus more on avoiding obstacles and solving puzzles in a timely manner. In addition to this four direction model, we decided our character would stick to walls on collision. We believe that adding this small aspect alone will create another layer of complexity to traversing different sections of the game. Another aspect of our game is that we're focusing on ways we can procedurally generate different patterns of obstacles for the player to traverse. In doing so, we hope this increases replayability and retention among our player base. Our concept will be tailor-made for a grid-based system, which Unity supports extremely well through the use of tilemaps.

Upon launching the application on an iOS or Android device, the player will be greeted with the title screen of the game. This title screen (on the right) has yet to be designed in full but it already has the layout in the picture. The word "FlingShot" will be replaced with an official logo and there will be a background animation of our slime character. This menu has options such as a play button, statistics section, settings, rules, and an about button. The about screen will have information about the game, updates, development, and all relevant information about us, the development team. It will be static in nature and should just output information. Upon selecting rules,

we plan on having a small scripted segment that will guide the player through the various core aspects of the game. We hope to have everything from basic movement, obstacles, and power ups discussed within this section. Upon selecting the play button, the main game will begin and the player will start traversing upwards through the procedurally generated patterns and landscapes.

The gameplay is designed with a few key principles in mind to ensure an engaging experience. The first is always having something to do, interact with, or think through. There should never be a dull moment; no waiting or boredom. This suits our goal of a fast-paced action game. Our second principle is that there should always be some looming danger, something to watch out for and avoid. In this game we have a rising screen, in the form of lava, which must be outran. This requires again, a fast pace, as well as a clear objective of what to do - don't fall in the lava. Finally, one of the most important things to consider is that it should never feel unfair to the player. There should never be a situation where the player loses and they could have done nothing to avoid it. There should always be escape routes to some extent.

Thus, our game is an infinite climber (basic layout seen to the right). There is an ever-looming threat below the player which they must never come in contact with else the lose and must start over. They must climb up, avoiding obstacles and enemies and interacting with the environment to succeed. As the game progresses, and the player manages to survive for longer, the lava will rise faster and faster (up to some manageable upper limit which is still possible). The procedurally generated environment will become more complex over time and trickier to navigate, and overall the climb will slowly become harder over time, yet always possible to work through. While climbing, the player may encounter collectables such as power ups, face special bosses which must be defeated and break up the action, or other unique abilities. There may also be a currency to collect which may be used for special upgrades, or perhaps a second chance should the player lose. These will be calculated into your score at the end of each round, and perhaps a global or regional leaderboard will display top scores daily and all-time.



We also plan on adding a capability to pause the game during runtime. In doing so, the player can select from a few options like exiting from the game, altering game settings, and resuming the current game. During this time, all elements of the game, from the lava to the player, will be frozen. Should there ever be some sort of multiplayer mode, the pause feature will not be enabled. Other features such as a replay option have been considered, but ultimately these may end up being stretch goals if anything.

With the way we have designed our project structure, we have a relatively small hurdle to conquer in producing a working version of the game. Once our initial base project is complete

and we have our initial movement and gameplay, the rest of the time will be open to extend our features as much as we can in whichever directions we decide fit best. This is when our creativity takes over and we can expand it to satisfy our full expectations. Since we can make it very incremental in this manner it allows us to set very clear and concrete objectives in very short terms. This does wonders for workflow, versioning, and parallel design of features, which are essential tools at our disposal for this project.

In general, we want the application's UI to be fluid and reactive. All buttons will have multiple states with respective animations. As an example of this, tapping the button will translate to the button being in a pressed state, therefore, the button itself should appear like an analog button and look like it's actually being pressed down. We also plan on implementing transitions between the different menus and screens within the game. Nothing UI wise should be instantaneous, meaning everything the user interacts with should feel tangible and reactive to some degree.

Advertisements are huge within the mobile development industry. Many applications' main source , and sometimes only, of income stems from the use of embedded advertisements. While we find that most of these applications' advertisement implementations to be rather intrusive, we aim to provide them in such a way that they won't interfere with gameplay and negative affect the player's experience.

Design Constraints

For the most part our app will have very few restrictions. The primary limitation is that we are bound by the software that we are using. We can only do what unity and C# allow us to do, if it is more than the language can handle, then obviously we cannot pursue that idea. Another constraint for us would be the distributor, for us that takes the form of the app stores. Our app will have to abide by both the restrictions of the Apple App Store and the Android Play Store. For example our app icon (seen right) could be denied by either of the app stores.

Ethical Property

The primary ethical issue that we believe our project will encounter is the idea of pay to win. Many games that have come to the app store and gaming as a whole in the last few years have implemented a "pay to win" model. This is where the user can pay money to gain an

advantage over other users that have not paid. In competitive games this enrages the player base and drives interest away from the game. We will strive to deliver a game that is free of this model, and instead will make profit from small non intrusive ads in either the top or bottom banner. These ads will not hinder game play in anyway.

Intellectual Property

An issue that our project may face in terms of intellectual property is the number of games with similar mechanics. On the Apple App Store, there are hundreds of endless runner type games. This means that we need to make mechanics that are unique to FlingShot to differentiate ourselves from the other apps. A few mechanics that will help differentiate us are custom made asset design (including character and platform design), and unique obstacles (including boss battles, portals, and power ups). For example, one app that FlingShot is similar to is Tomb of the Mask. The moving mechanic they use is very similar to what we plan on using for FlingShot. This includes swiping to move in a direction and the character stopping when it reaches a wall. To make our app significantly different from theirs, we are implementing boss battles and taking away the wall of ads that they have among other things. However it is difficult to copyright the mechanic of sliding puzzles, which have been around for much longer than computers have, so I foresee no legal or IP issues.

Change Log

- We have changed how our character will move inside the game. As a result the description of gravity and how the platforms changed to reflect this.
- Updated the definition of how difficulty will work in order to give a more accurate description.
- Completely overhauled the milestones section to reflect more updated and measurable objectives.
- We also updated the Work Plan to better reflect what we will all be doing. It will align better with the gantt chart.
- Unity Pro subscription removed, we will use our own source control system via Github.
- Synopsys updated to include procedural generated aspect.
- Removed unnecessary goals from Milestones.
- Removed and updated relevant IP issues.
- Updated the Final project design section to give more details for each topic