Project 1 | Technical Risk Assessment

Vedant Chaudhari & Tyler Chapman

Delivery Platform

Penguin Shotgun Baseball is being developed for the MacOS and Windows platforms. Console platforms will not be targeted as of now and our team will build a mobile (Android and iOS) version in the future. By developing for these platforms we will be able to rapidly build our prototype and have access to a large install base.

Development Environment

Penguin Shotgun Baseball will be developed using the Unity Engine due to it's well documented and easy to use 2D development environment. Furthermore, Unity's polished and easy to access tools such as Prefabs and UI Systems will make the technical process more accessible to both artists and designers. Additionally, we are using git scm as our version control software of choice.

Game Mechanics & Systems

<u>Power</u>

The player taps and a bar appears using Unity built in slider UI. The slider repeatedly charges up and down between 0 - 100 which will be stored in an integer. The higher the slider the larger the force modifier used to fire the penguin.

<u>Swing</u>

The player taps again to pitch the penguin and tap one more time to begin the swing animation. The baseball idle and swing animation are implemented using Unity's Mecanim. Using *AnimationState.normalizedTime* to determine at what point during the animation the player is at when it makes contact with the penguin. Based on the value, a predetermined firing angle is selected.

<u>Flying & Shooting</u>

Every shotgun the player purchases has a fixed amount of shells stored in an integer. Whenever the player fires this value is decremented by 1. The player uses the shotgun to boost the penguin forward and up whenever a hit successfully lands. Collision checks will be performed between the reticle and the penguin when fired to determine if a hit has landed. Debuffs and buffs can be collected by the player which can boost the penguin further or stop it. These will be implemented using modular prefabs that are then spawned in procedurally in the world generation code. Certain buffs and debuffs are tied to different heights of the map. When the penguin's velocity is 0 for more than 2 seconds, the camera returns to the player and the player is awarded money equal to the distance they traveled.

Buying

Money earned by the player can be spent at the shop built using Unity's UI tools to get better and more powerful shotguns as well as increased ammo. To open the shop the player must hold down the space button for 1.5 seconds and then release. To navigate the shop the player uses tap to scroll down, double tap to scroll up, and hold to select. Shotguns and shells will be implemented using base generic classes with certain stats or attributes that can be inherited and expanded upon with specific shotgun classes.

Winning

Feature cut due to time constraints

Art Pipeline

This section outlines how art assets make their way to the current development build

Sketch

Assets will be conceptualized by artists on sketching paper. This process will

Photoshop

2D assets greenlighted in the sketching stage by both the artists and designers will be digitized using Adobe Photoshop.

Discord

Digitized assets will be sent to the art channel in discord for review by both designers and programmers

Repository

Project files will be uploaded to a git repository by the programmer

Unity

Assets committed to the repository will automatically be added to the Unity project folder for use. In Unity, the pixel to unit ratio will be consistent between all objects. Art will be implemented into the game by the programmers.

Unity Mecanim

For animations, sprite frames will be added to the repository and will automatically be added to the Unity project folder for use. These assets will be implemented using Unity's built in mecanim animator. Via Unity animator tab, parameters and triggers will be used to trigger animations.

Review

Once an asset has been implemented, an artist and designer will review them in order to certify they have been implemented correctly and are working as intended. If not, the artist and designer will work with a programmer to bring the asset to a satisfactory state.

Design Pipeline

This section outlines the design process for Penguin Shotgun Baseball

Modifier variables such as slider speed, penguin starting force, penguin speed, etc. will be made available in the Unity inspector for designers to modify and tweak the gameplay.

For organizational purposes, scripts have been split up into folders by functionality. For example the world generation code is separated from the spacebar control code in order to decouple our code and keep it organized and readable.

Prefabs of common game objects will be available in the Unity asset folder for the designers to tweak and adjust.

Future Goals

In the future, the team would like to add a win state and flesh out the game's featureset. Furthemore, we would like to add more polish and feedback to the core gameplay loop.

The team is also interested in bringing Penguin Shotgun Baseball to all major mobile platforms as it could do well in the casual mobile gaming market.

Risk Assessment

Game Development is a cutthroat industry that moves at a very fast pace. It is rifled with major failures that lead to studio failures, but also success stories that lead to the meteoric rise of many studios.

We have thoroughly researched our target audience to make sure that we have a consumer base that would be interested in our product and our designing it very closely with focus groups. We strongly believe our game could be a moderate success on mobile platforms as it lends itself well to that form factor.