

Poly-Vinyl Pests!



Jeffrey Popek
Team IDX Legends
Poly-Vinyl Pests
Sprint 9

Table of Contents

Delivery Platform	4
Development Environment	4
Engine	4
Version Control	4
Documentation	4
Logic Pro	5
Audacity	5
Photoshop and Clip Studio Paint	5
Maya	5
Substance Painter	5
IDE	5
Discord	6
Systems List	6
Player Movement	6
Score	6
Boss Fights	7
PVP	7
Narrative	7
Pipelines	8
Art Pipeline	8
Environment	8
Rigging/Animation	9
UI	10
VFX	10
Audio Pipeline	11
Design Pipeline	12
Sprint Updates	13
Sprint 0 - Brainstorm	13
Sprint 1 - Bird Game Initial Prototype	13
Sprint 2 - Bug Game Initial Prototype	13
Sprint 3 - Bird Game Prototype Continuation	14
Sprint 4 - Bug Game Prototype Continuation and Step 2 End	14
Sprint 5 - Bungie Industry Review	15
Sprint 6 - Laundry Snail Iteration	15
Sprint 7 - Laundry Snail Phase 2, UI, & PvP	16
Sprint 8 - Bottle Mushroom Boss & LS Phase 2 Iteration	16
Sprint 9 - Industry Review 2	17

Delivery Platform

Our game, Bug Game, will be played on any device that can use two controllers at the same time and run unity games. Our game is mainly aimed at PC players but can be expanded to console players.

Development Environment

Engine

Our game was developed in the Unity Engine because of our team's experience with the engine and also the amount of documentation available online. Unity allows us to easily build our game and let a wide range of players play our game. Unity is also easily accessible for each role in our team. We are using Unity version 2022.3.8f1.

Version Control

Our team will use git for version control because it is supported by BitBucket. Applications such as GitHub Desktop and GitTortose will help simplify the version control process for less experienced git users.

Documentation

Our team uses Google Drive to store and create documentation before finalizing it to be submitted to Confluence. Google Drive is easily accessible to everyone in the team and Champlain gives students a large amount of storage to use for projects. Google Drive also allows each team member to easily access and edit any existing documents for easy collaboration.

Logic Pro

Our team's sound designer uses Logic Pro X to create and export sounds. Logic Pro X is provided by Champlain on the lab computers on campus.

Audacity

Our team's sound designer uses Audacity to edit sounds they created and exported from Logic Pro. Audacity is free and can be used in the labs or on a personal computer.

Photoshop and Clip Studio Paint

Our artists use Photoshop and Clip Studio Paint to create and edit their art. Both of these programs help the artist make creating and implementing art as easy as possible.

Maya

Our artists will be using Maya to model and animate assets. Champlain covers Autodesk licensing and is available on the lab computers on campus.

Substance Painter

Both artists will be using Substance Painter to texture models. The Adobe Suite is covered by Champlain and is available on the lab computers on campus.

IDE

Our Programmers will be using Visual Studio and JetBrains Rider to edit scripts. Visual Studio and Rider are both great options due to their ease of use and integration with Unity.

Discord

Our entire team uses Discord as our primary communication method. Our team has a discord server that the faculty also have access to.

Systems List

Player Movement

Risk: Medium

The players control an ant from a top down perspective using either a controller and the keyboard. Left stick/WASD controls the direction the players move in, right stick/arrow keys control the direction the player is facing, and right bumper/space bar is to attack. For example: move left stick up and right stick right: The player moves upwards while looking to the right. Left trigger/left shift allows the players to perform a dash in whatever direction they are facing, giving them invincibility frames so they can dodge through certain attacks. The risk is medium because implementation will be easy, but getting the player movement to feel good will require iteration and bring the build to testing.

Score

Risk: Low

During a boss fight each player will have a score that can be increased by hitting the boss with your projectiles. The score will be shown as a slider at the bottom of the screen during boss fights and will update each time a player gains score. At the end of a boss fight, the player with the highest score will be crowned the winner, with the results screen showing the winner's icon happy wearing a crown and the losing player's icon crying. The risk for score is low because like the player stats, the score is just numbers that can be updated during runtime.

Boss Fights

Risk: High

The bosses in this game are the main focus along with PVP, so making sure the bosses play well is a priority. Each boss will have a different set of attacks and behaviors, but to decrease implementation time we use the Unity animator as the state machine for the bosses. The animator works more like a behavior tree and can easily be edited to make more complex behaviors. The risk for boss fights is high because it is very time consuming for each discipline to create a single boss fight as well as balancing it.

PVP

Risk: High

PVP is the other focus of our game and is essential to the game loop. PVP will have the players in an arena fighting between the boss fights, with the PVP battles serving to break up the flow of the bosses. PVP will use the same controls as the rest of the game and will only end when one player kills the other. The risk for PVP is high because it is essential to our game and requires a lot of testing to make sure the PVP is satisfying for the players.

Narrative

Risk: Medium

The narrative of this game is not going to be its main focus since we are going for a more arcade system game. The narrative will be told primarily through the environment and through barks spoken by the bosses and players during fights. The risk for narrative is medium because it is closely tied to the art and sound of our game, as the environmental storytelling will not work if those aspects of our game are not strong. Past greenlight, we hope to add short comic panels in between the boss fights to further expand on the narrative.

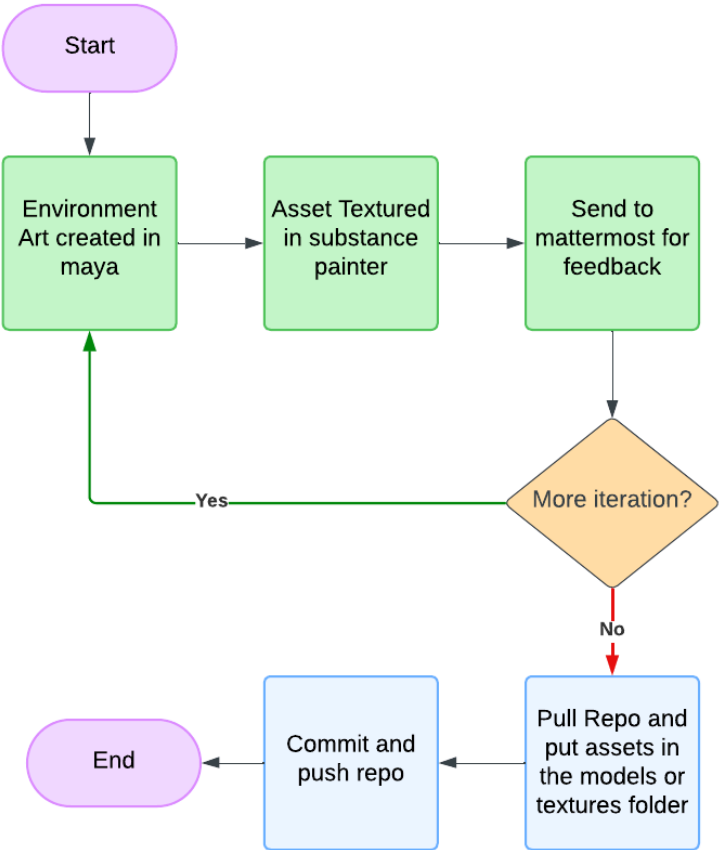
Pipelines

Art Pipeline

Environment

Assets are created in Maya and textured in Substance Painter. Files are exported in fbx and targa formats respectively. After pulling from the repo, put the files in their corresponding folders. Meshes are put into the “Models” folder while textures are stored in the “Textures” folder. Commit and push to repo when finished.

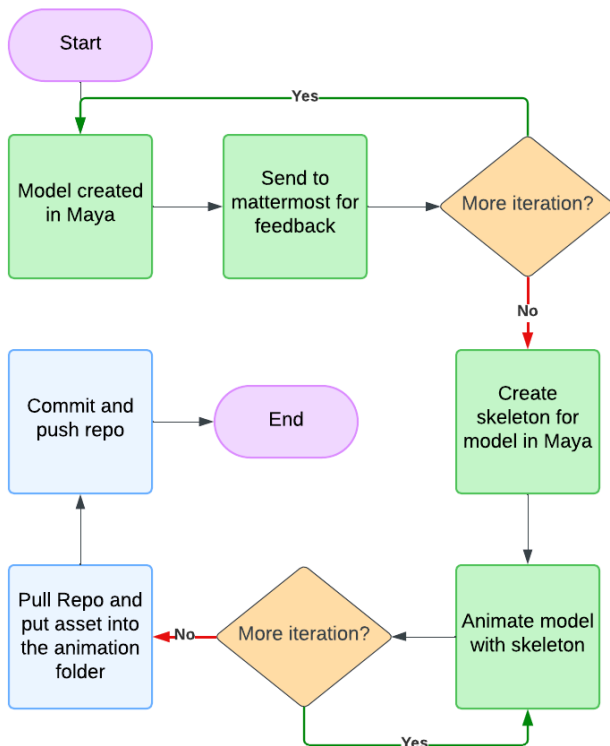
Team IDX Legends Environment Art Pipeline



Rigging/Animation

Blockout models are made in Maya, and reviewed by other Artists/Team/Faculty. The model is then finalized and prepped to rig. The skeleton joints are created and fitted to the model, and parented appropriately to have proper functionality. Once joints are in place and operate smoothly, select all joints and use the orient joints function, delete history and freeze transformations. Once the skeleton is complete, create controls to prepare for animation. Delete history and freeze transformation on these, and parent constrain them to appropriate joints. Before animating, import the model with its skeleton and controls to engine in the Assets > Models folder. Animate the model in Maya, import to engine in the Assets > Animations folder. Work with programmers to link models and animations to player prefabs.

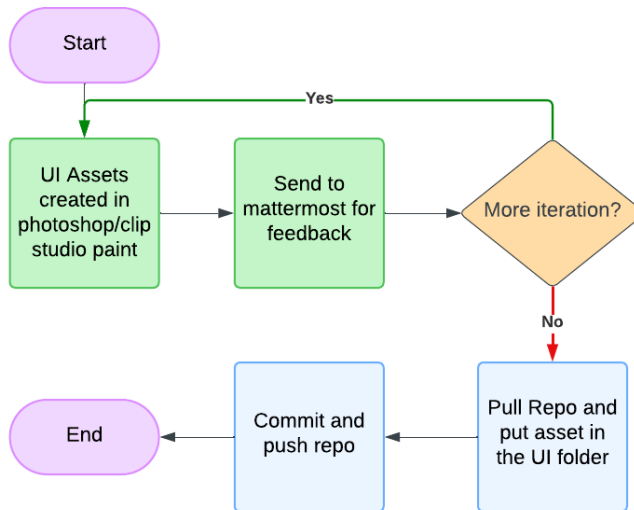
Team IDX Legends Rigging/Animation Pipeline



UI

UI assets are created in Photoshop/Clip Studio Paint and exported in png format. After pulling from the repo, the files are put into the “UI” folder. Commit and push to repo when finished.

Team IDX Legends UI Art Pipeline



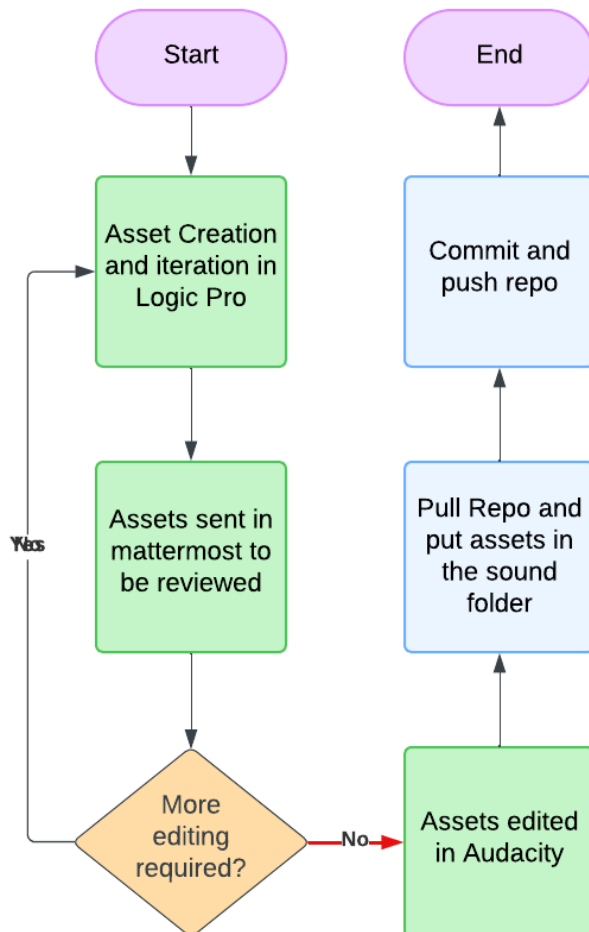
VFX

VFX assets are created in engine with the Unity VFX graph. Work with programmers to link VFX to gameplay features.

Audio Pipeline

Assets are created in Logic Pro, a DAW used for sound synthesis and composition. These files are then sent to the team's Mattermost from the audio labs to be reviewed by the team and given critique. If no further editing is required then these files are taken into Audacity where they are cut up and transformed into individual audio files for each sound/song. After editing pull the repo to make sure it's up-to-date, add audio files into the Assets/Audio folder (SFX for sounds and OST for music for further organization), commit, and push changes.

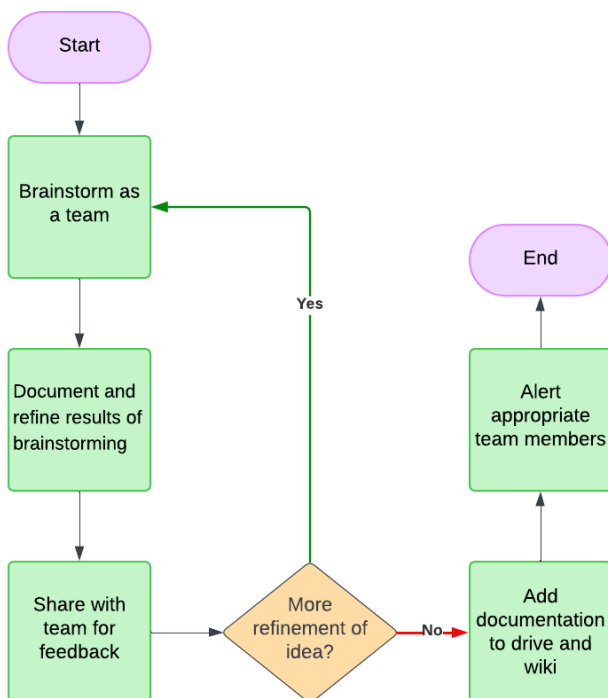
Team IDX Legends Audio Pipeline



Design Pipeline

The Design process begins with the team collaboratively brainstorming the ideas behind a system or boss that we wish to implement in the game. We then proceed to document and refine the results of this brainstorming by working out how feasible the idea would be, and how it fits into the scope of what we are currently working on. Once the documentation process is done, we share what we have with the team at one of our weekly meetings. We ask for feedback and see if what we have could work with the other disciplines, such as if an attack pattern for a boss is too out of scope for the artists or programmers. Once we have gathered the feedback and made the appropriate changes, we add the documentation to the drive and inform the appropriate team members that it is ready, so they can go about finding the best way to implement the ideas.

Team IDX Legends Design Pipeline



Sprint Updates

Sprint 0 - Brainstorm

During this sprint the team brainstormed multiple concepts to bring to class the next week and present. Most of the week was spent meeting and discussing possible concepts to start prototyping in sprint 1. Our team only wanted to work with two different prototypes with multiple overlaps in systems and artstyle. The two concepts we came out of this sprint with was a bird open world flying game and a split perspective game. The main goal of this sprint was to brainstorm 2 game concepts that the entire team was on board to work on for the rest of the semester.

Sprint 1 - Bird Game Initial Prototype

The first sprint was spent prototyping our first concept “Bird Game”, an open world flying exploration game with an emphasis on narrative. The overall theme was that each island would be tied to a specific negative emotion such as fear. The main goal of this sprint was to solidify this concept and create a working prototype to show proof of concept.

- ❖ The programmers began to prototype the systems of the game such as flying, camera work, and exploration mechanics.
- ❖ The designers spent a large portion of the sprint brainstorming the theme of the game as we decided that this concept would be narrative heavy to utilize the narrative designers on our team.
- ❖ The artists began exploring the player model, rigging, and environment art.
- ❖ The sound designer started exploring sound themes such as BOTW.

Sprint 2 - Bug Game Initial Prototype

The second sprint was spent prototyping our second concept “Bug Game”, a couch co-op boss rush that also had the players fight each other with upgrades from the boss. This concept was originally the split perspective game, but after much discussion our team decided to pivot concepts to this one instead. This concept had a heavy emphasis on combat between bosses and players. The overall theme of this game was that the players were ants that would fight against other bugs that have merged with human

trash. Our team wanted to stick to a boss rush and not explore more rogue-like mechanics such as random level generation. The main goal of this sprint was to solidify this concept and create a working prototype to show proof of concept.

- ❖ The programmers began to prototype the systems of the game such as boss ai, two player input, and an item upgrade system.
- ❖ The designers brainstormed and documented the systems and narrative.
- ❖ The artists began exploring the player model, rigging, and environment art.
- ❖ The sound designer started experimenting with different sound themes.

Sprint 3 - Bird Game Prototype Continuation

The third sprint was spent continuing work on Bird Game. In this sprint the team wanted to solidify the theme and art style while continuing work on the systems and mechanics. This sprint has a lot of talk about optimization since this game would be rendering a lot of polygons each frame. Player movement was also a big discussion point. The main goal of this sprint was to work on the player movement feeling good.

- ❖ The programmers continued work on the existing systems, switched the player movement to be momentum based, and started work on boss ai behaviors.
- ❖ The designers started work on level blockouts, player control scheme, and further expanded on the narrative.
- ❖ The artists rigged and animated the player character, experimented with shaders, and model polygon optimization.
- ❖ The sound designer continued work on the established sound themes and created sound effects for the player and the environment.

Sprint 4 - Bug Game Prototype Continuation and Step 2 End

The fourth sprint was spent continuing work on Bug Game. In this sprint the team wanted to solidify systems and mechanics and create a working prototype. This sprint was the end of step 2, so the team met to discuss which concept we wanted to move forward with. We decided to go forward with Bug Game because we could envision the game at all steps (greenlight and end of year) while for bird game we were unsure how it would look by greenlight. The main goal of this sprint was to continue iteration on bug game until we decided which concept to defend for step 2.

- ❖ The programmers continued work on the existing systems: Upgrades, Player movement, Boss AI.

- ❖ The designers continued work on iterating the system's design and created UI designs for the combat and upgrades scenes.
- ❖ The artists rigged and animated the player character, created UI art, and did a boss arena blockout.
- ❖ The sound designer continued experimenting with their theme of power metal to create sound effects and a boss theme.

Sprint 5 - Bungie Industry Review

The fifth sprint was spent preparing for the industry review with Bungie. Since the team only had four days before the build was due, we had to do a lot of work earlier in the sprint to get a working prototype to be tested. The latter of the sprint was spent creating and iterating the presentation that would be shown during the industry review. After feedback we got, we decided to cut upgrades from this semester because we wanted to focus our efforts on making the boss fights and PVP feel as good as possible. In this sprint the team finished a basic game loop with a working boss fight and player PVP. The main goal of this sprint was to create a working build and a presentation to be shown during the industry review.

- ❖ The programmers worked on the boss's phase 1 behaviors, player interactions with the boss and other players, and PVP.
- ❖ The designers created a bullet hell system for the boss to use, boss behavior flow chart, and UI.
- ❖ The artists created a boss model and began texturing and rigging it.
- ❖ The sound designer implemented their sounds into the build and programmed them to play during the game loop.

Sprint 6 - Laundry Snail Iteration

The sixth sprint was spent refining the laundry snail boss fight. This week our team brought our prototype to the GTL. At testing we learned that our boss fight was too easy which was a relief because it would be much easier to make the boss harder compared to making it easier. We also decided that the PVP would be in different stages with each having a different gimmick. The main goal of this sprint was to refine the first phase of the boss fight by implementing better attack patterns, better AI, a refined model, and new combat systems.

- ❖ The programmers worked on improving the boss AI, implementing boss attack patterns created by the designers, and iterating on PVP.
- ❖ The designers continued work on the bullet hell system, brainstormed PVP stages, and implemented UI
- ❖ The artists continued iteration on the boss model, worked on player animations,, and began rigging the boss.
- ❖ The sound designer created a PVP theme based off of the boss theme, created new SFX for attacks, and implemented those SFX into engine.

Sprint 7 - Laundry Snail Phase 2, UI, & PVP

The seventh sprint was spent creating a second phase for the laundry snail boss fight. We also worked on implementing the new pvp arena we previously brainstormed. The main goal of this sprint was to get a phase 2 of the laundry snail working, implement more pvp arenas, and implement UI.

- ❖ The programmers worked on implementing UI, pvp arenas, and phase 2 of the laundry snail.
- ❖ The designers continued work on the bullet hell system, implemented pvp arenas, and implemented UI.
- ❖ The artists continued rigging the snail model and began texturing assets.
- ❖ The sound designer iterated on the pvp theme based off of the boss theme and implemented sfx into engine

Sprint 8 - Bottle Mushroom Boss & LS Phase 2 Iteration

The eighth sprint was spent creating a new boss, the bottle mushrooms, using the code and logic we made for the laundry snail. This boss wasn't too big of a worry for the team because this boss would reuse a lot of code from the laundry snail and overall be easy to implement. Our main concern was the art scope, but we negated that by not having any rigged animations for the mushrooms and instead using unity features to animate the mushrooms. We also spent time iterating on the laundry snail's phase 2 and creating new assets for the environment.

- ❖ The programmers worked implementing the bottle mushrooms and iterating on laundry snail 2nd phase

- ❖ The designers continued work on the bark system, arena blockouts, and ui implementation
- ❖ The artists created animations for the laundry snail and created textures for the laundry snail arena
- ❖ The sound designer iterated on the ost and sfx while also implementing more sounds into engine

Sprint 9 - Industry Review 2

The ninth sprint was spent preparing for our next industry review. Since we had much more in engine compared to last time, we did not have to crunch to get this build completed. This week we mainly focused on polishing the already existing build, implementing art assets, and creating a tutorial blockout. This week was also doc review, but since our team already keeps our documentation up to date it was not a worry.

- ❖ The programmers worked on shaders, post processing effects, UI functionality, boss ai, and bug fixing.
- ❖ The designers worked on a blockout of the tutorial, documentation, ui implementation, and boss ai.
- ❖ The artists continued work on the laundry snail's attack animations, created and implemented props, and created a new splash art screen.
- ❖ The sound designer created a new song for the mushroom boss, implemented sfx, and fixed sound related bugs.