AstroFrog

EDG-220-06 | Section 6 | Team 1 | Project 3

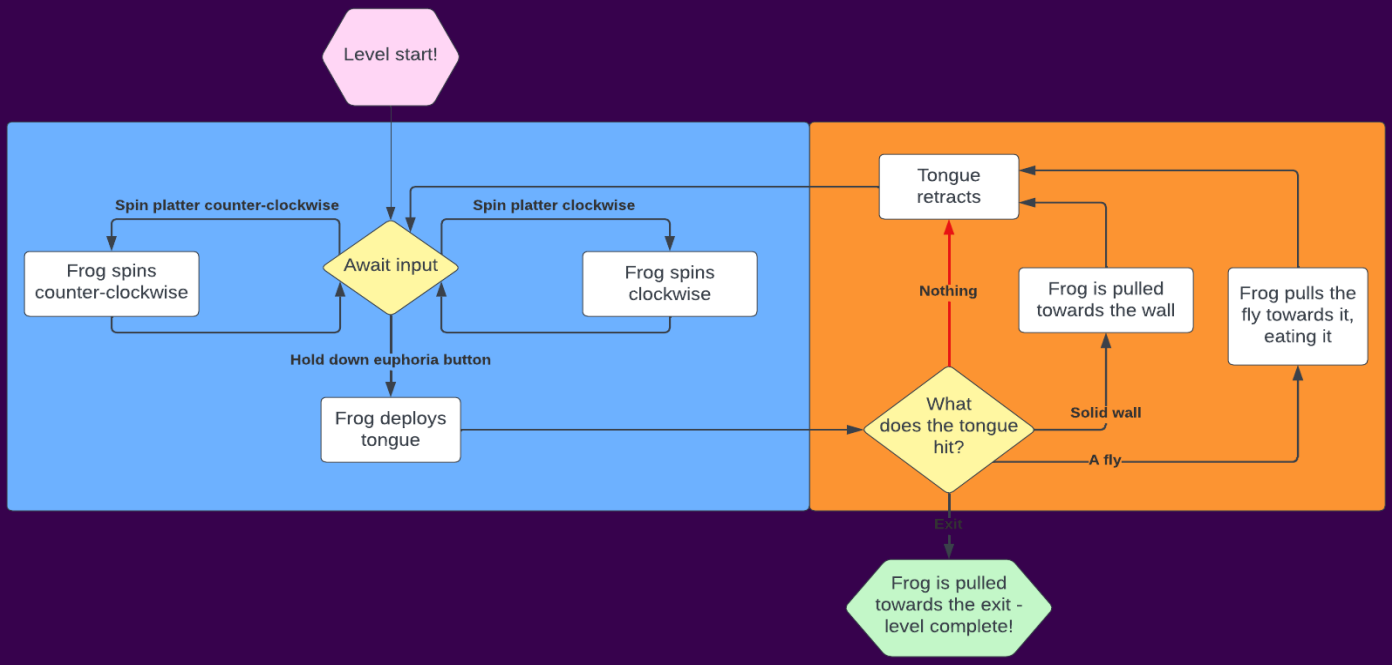
Technical Document

Table of Contents:

Risk Assessment Chart:

1	Safe	Implementation or use of the item is safe to use without creating errors.
2	Low Risk	Implementation or use of the item has a low chance of causing hindrances and problems.
3	Moderate Risk	Implementation or use of the item has a moderate chance of causing hindrances and problems.
4	High Risk	Implementation or use of the item has a high chance of causing hindrances and problems.
5	Near Impossible	Implementation or use of the item has a 99% chance of causing hindrances and problems.

Logical Flow Diagram:



Delivery Platform:

Windows | 1

Windows is an OS for PCs that features stability and an adaptive working space. Since our game is educational, the most common way to play our game would be on school lab computers, which most often run windows. This is safe due to it being one of the most used platforms in the world, making the distribution of our game easy.

Development Tools

Unity | 3

Unity is a game engine for PC that specializes in creating both 2D and 3D games. Unity comes with tools for code implementation and worldspace design tools that will be vital to our game. Risks of using Unity are complexities within the engine that will have to be learned, as well as the engine having small compatibility issues with our repository management software.

GitKraken | 3

GitKraken is a GUI for the popular repository management software GitBash. GitKraken will be the main tool our team will use to push fast and easy updates to the game and documentation for the other members of the team. Risks of using GitKraken are mainly due to the potential of merge errors and lost data if members of the team do not communicate about assigned workspaces.

Adobe Photoshop | 2

Adobe Photoshop is a piece of photo editing software that excels in creating rasterized art. This will mostly be used for sprites in our game, as well as backgrounds and potential UI art. Risks of using Photoshop fall under the time it takes to create art in order for it to be implemented into the game.

Adobe Illustrator | 2

Adobe Illustrator is a design creation software that excels in using vector art to create 2D images. This will mostly be used for the elements of our game that require smooth art with curved outlines, such as the objects or placement zones. Risks of using Illustrator fall under the time it takes to create art in order for it to be implemented into the game.

Game Systems:

Turntable Movement | 1

Whenever the player starts the game, they will be able to use the DJ Hero turntable to spin the player left and right. This will also change the direction in which the player can aim and shoot their tongue. The turntable will also be able to select menu options on UI based scenes.

Button Tongue Trigger | 1

On the DJ Hero turntable, the singular button above many of the sliders and knobs can be used in our game to fire the player's tongue. This button will also be used for our game menus and UI functions.

Fly Counter | 2

In the bottom right corner of the screen, a piece of UI will display the amount of enemy flies remaining in the level. This will not include the boss fly. This counter will not complete the level when finished, but will help completionists finish the levels to 100%.

Level Select | 3

From the main menu, the player will be able to navigate to a level selection menu. From this menu, the player will be able to load any level in the game and play through it.

Start Menu | 2

When the player boots up the game, they will be greeted by a title screen. This title screen will allow the player to start a new game, select a level, exit the game, change game options, or see the credits for the game.

Level Complete Menu | 2

Whenever a player completes a level, they will see a pop-up screen that tells them specific things about how they completed the level, such as the time remaining and the flies they collected. The player will then be prompted to continue to the next level.

Frog Characters | 3

To add a more interesting and fun experience to our game, the player would be able to play as many different characters throughout the game. The player would also be able to switch between these characters at the beginning of each level.

Character Select | 3

At the main menu or the beginning of each level, the player would be able to select from a pool of other playable characters with different stats and abilities. These characters would not change the gameplay dramatically, but add some spice to how each level feels, and could potentially fit someone's playstyle better.

Game Mechanics

Zero-Gravity Physics | 1

The zero gravity physics of AstroFrog are the games main challenge. The player has no way to move in a traditional manner, as the zero gravity prevents the player from moving in any way. Within the zero gravity environment momentum is incredibly important, as even the slightest bit of momentum in any direction will cause the player to float in that direction until they hit an object that stops that momentum.

Grapple Tongue | 3

The grapple tongue is the main method of movement for the player, when the player presses a specific button their tongue will be launched out. The tongue will move forward until it either hits an object or until it reaches its reach limit, then the tongue will retract. Once the tongue hits an object it will check what that object is, if the tongue can interact with the object it will perform a specific action. If the object is a wall the tongue will latch on and drag the player to its current location. If the object is a collectible then the object will be attached to the tongue and the tongue will return to the player. The player has to hold down the grapple tongue button to continue grappling. If the player releases the button at any time while the tongue is out, then the tongue will drop whatever it's connected to and retract.

Eating Flies | 2

The fly eating mechanic is the main way collectibles are acquired by the player, once a player touches a fly they get eaten, being eaten removes the fly from the game world. Once a fly is eaten the flies collected counter goes up by one.

Puffer Flies | 2

The poison fly is a with a special ability, when the player character gets within a certain distance of this fly, they puff up and protrude spikes. When the player eats the fly in this state they lose a chunk of their remaining oxygen.

Moving Flies | 2

Moving flies are a special type of fly that moves back and forth between two walls. The fly will move until it comes into contact with a wall, once that happens it will reverse its direction.

Shield Flies | 3

The shield fly is a fly with a special ability, they have a shield that blocks the players tongue and prevents the fly from being eaten from that side. This shield protects the side of one of the sides of the fly; if the player's tongue comes into contact with the shield the tongue is forced to retract. The player must eat the fly from an angle that isn't protected by a shield.

Oxygen Tanks | 3

Oxygen tanks are another type of collectible found in the game's levels. These tanks act much like flies, as they can be grabbed by the tongue and can be eaten. When the oxygen tanks are eaten a portion of the oxygen in the players oxygen meter is restored.

Boss Flies | 1

The boss fly is the end goal of the game's levels. Once the player eats this fly the level will end and take the player to the end screen. This fly is exactly like a normal fly, but doesn't add to the flies collected counter.

Art Pipeline

Naming Conventions

Enviroment_[Name].png

Sprite_[Name].png

Implementation

Art for the game will be developed in either Photoshop or Illustrator, and exported as a .png file when the asset is finished. The asset will then be pushed to the repository in the file under Art Assets that it pertains to. If it is an environment or tilemap asset, it will be placed in Art Assets/Environments/, and if it is a sprite or singular asset, it will be placed in Art Assets/Sprites/ (NOTE: The first time that an asset is pushed, the folder will need to be set up). Once art is in these folders, the art will be imported into the games' assets folder, and then be sorted in a similar fashion to the repository (Materials/…). When the art is fully imported into the game, the programmers/designers will place the art onto whichever asset it is required by.

Audio Pipeline

Naming Conventions

SFX_[Name].png

Music_[Name].png

Implementation

Sounds for the game will be sourced from and downloaded by the designers/programmers, and imported directly into the game project. It will be sorted into its respective folder: Sound/SFX/ or Sound/Music/. From there, it will be placed into the AudioManager class and object, and that will be hooked up to the respective audio source from there.

Design Pipeline

Naming Conventions

Level_[Name]

Screen_[Name]

Level Design

For every level in the game, a designer will create a new scene and drag in the required prefabs, such as the Game HUD, RoomID, Audio Manager, Scene Switcher, and Button Manager. This list will be updated as more managers are required for scenes. The designer will then create a tile map and draw the level using the tiles provided by our artists. Once the level is drawn, a collider will be added to the walls of the level, and the designer will then place all the necessary enemies and collectible objects around the level. Once the level is fully implemented and has been tested to be functional, the level will be added to the build order, as well as its build order ID published in the SceneIndexes ScriptableObject. Once this is done, the programmer will enter the level and polish around the edges, as well as hook up the level to other scenes, such as the level select and the adjacent levels in the game.

Sprint Updates

Sprint 1

- ★ Establish game concept
- ★ Create documentation and flesh out ideas for game

Sprint 2

- ★ Work on concepts for systems and mechanics
- ★ Establish Unity project to contain prototype
- ★ Implement basic mechanics into project

Sprint 3

- ★ Fix bugs with basic mechanics
- ★ Set up current feature to allow for implementation of new features
- ★ Create flies and a basic level, along with art for the UI

Sprint 4

- ★ Test game for mechanical feel
- ★ Polish and finalize basic mechanics

- ★ Begin to finalize scope and planning on all features

Sprint 5

- ★ Test level 1 in the game build to get responses on new features
- ★ Polish and implement UI art
- ★ Create animations for characters and implement art

Sprint 6

- ★ Implement character selection for levels
- ★ Finalize art and mechanics for UI and levels
- ★ Implement levels 2 and 3 with already created mechanics and prefabs

---