Worm off the String: Technical Risk & Assessment

Delivery Platforms

Windows, Mac, Linux and Modern Consoles (Difficulty: 3/5)

Worm off the String will be released on all three major operating systems (Windows, Mac, Linux) as well as modern consoles due to Unity's ease of creating builds for these platforms. Due to the focus on couch co-op for this game, consoles are a crucial platform due to their ease of local co-op play. However, the high number of platforms could make online multiplayer difficult to implement, as some networking solutions may make developing for certain platforms more difficult, namely the Switch.

Development Environment

Unity

Worm off the String will be developed using the Unity engine (Version 2021.1.19f1). This is because of our two programmers being comfortable with working in this engine, as well as being able to easily create builds for other platforms without massive overhauls in code for the sake of compatibility.

Git

Our choice for version control will be Git, and the main reason for this choice is that our repository service, Redmine, supports Git. While the programmers have experience in other software such as Subversion, all members of the team have experience using Git.

Gameplay Systems

Split Screen Multiplayer (Difficulty: 3/5)

Worm off the String will be primarily developed with split screen multiplayer in mind, with up to four players maximum. This feature may prove to be difficult as neither of our programmers has experience working on split screen based games.

Online Multiplayer (Difficulty: 5/5)

Players will be able to play with their friends in custom lobbies. The online networking solution will require our programmers to ensure physics are updated properly for all clients to ensure that the game feels similar to offline play under good network conditions. Only one of our programmers has made an online game before, so this feature will be difficult to accomplish.

Destructible Objects (Difficulty: 3/5)

With the main objective of the game being destruction, there will be many objects that are destructible, and each object can be destroyed in its own way. The programmers will have to work with the artists to create objects that break in believable ways.

Costume Powerups (Difficulty: 3/5)

At the start of each match, players will have a selection of costumes to choose to give players a special ability to augment their strategy for destruction. One of these could be a grappling hook to traverse the level easier, bombs to destroy a large area, a speed boost, a combo points boost, etc. Ideally, we'd want a large selection of costume powerups so not every match plays out the same.

Worm on a string-esque Movement (Difficulty: 2/5)

Each worm will move as if they are a worm on a string being pulled by an invisible hand. Worms will also have the ability to pull themselves upwards. This will feel like an invisible hand is lifting their string up, causing the worm to fly. This will use a resource, which can be recharged by touching the ground. The main challenge will be fine tuning so that it feels fun to move around while also convincing the player that they are controlling a worm on a string.

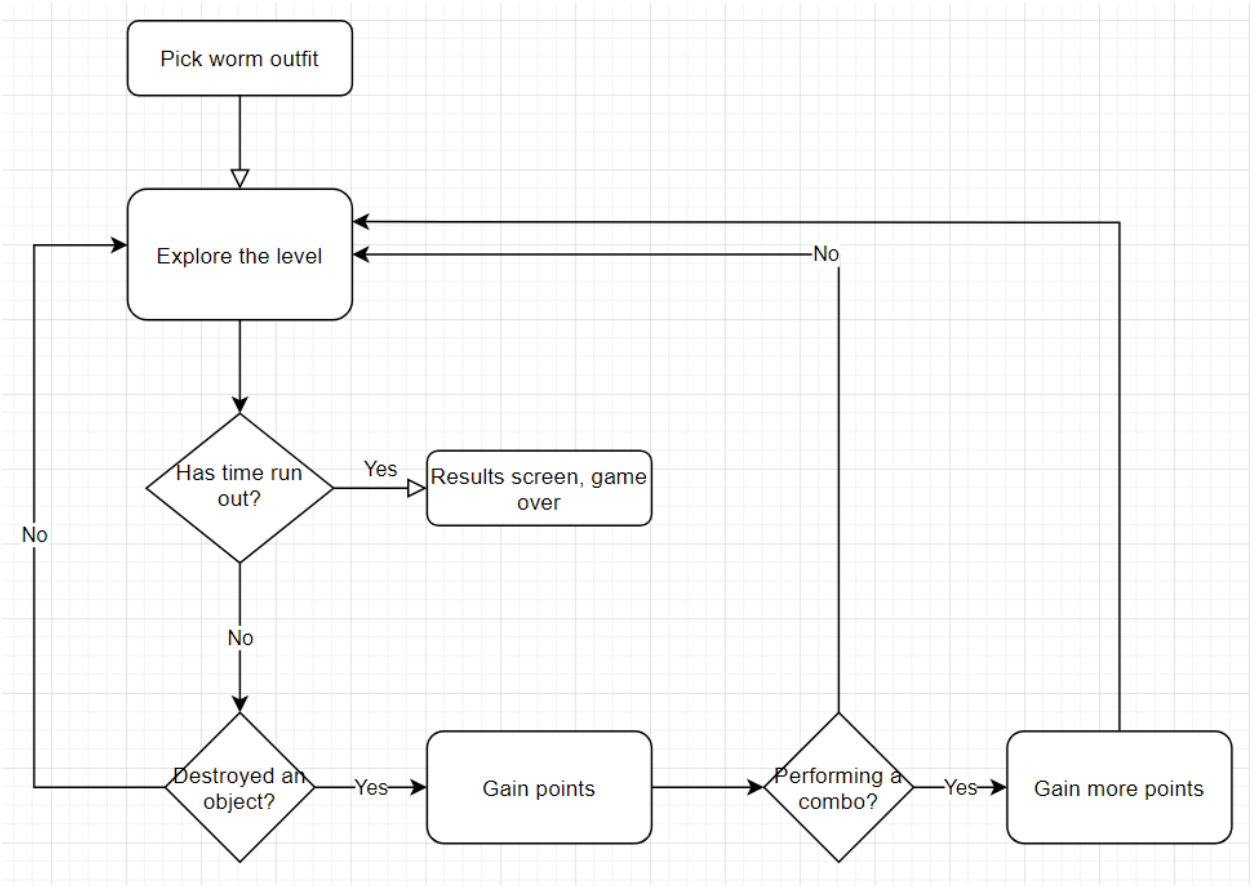
String Grabbing (Difficulty: 2/5)

Worms will be able to use their strings to grab terrain in order to fling themselves across the level, or grab objects to have better control of their destructive capabilities. The main difficulty with this task will be fine tuning our physics systems to ensure it feels good to do, rather than the implementation itself, similar to the movement system.

Scoring System (Difficulty: 1/5)

When an object is destroyed by a player, the player will receive points based on how much speed they had when the object in question got destroyed. In addition to that, there will be a combo system based on a timer where if a player destroys multiple objects in quick succession, they get additional points. This system is fairly simple to implement.

Logical Design Flow



Pipelines

Art Pipeline

Our artists will have a special folder in the repository that they will use to upload art assets for the prototype through Git.

For art pieces that serve as destructible objects, the programmers will likely have to spend time implementing them manually

Sizing

n/a

Naming Conventions

n/a

Design Pipeline

Due to both of our designers having familiarity with Unity, design changes can be easily made by the designer at any time by editing values through the inspector. Nearly any variables that change up gameplay are available to edit without the need of editing the code itself. The designers will meet with the programmers often to ensure they understand their direction fully, and the programmers will maintain a google sheet including all the game's variables, what they do and how to change them for the designers to have an easier time changing things.