Project Description:

Name: Swing and Sketch

Description: My game will consist of a couple of levels which will involve a character that has to travel through the level by swinging and also drawing a path which the character can surf. The combination of these two mechanics will allow the user to control their character so they can move by timing their swings and surfs to constantly move to the right without ever touching the floor.

Competitive Analysis:

My inspiration for my term project was a combination of the games stickman hook and dune surfer. These games both have the separate features that I intend to implement in my game. Stickman hook allows users to swing their character with the aim of avoinging the terrain which is coloured white. Dune surfer has the same aim but the users way of traversing through each level is by drawing what the player surfs on. My game will be unique in the way that it combines both features. Also i intend to design an algorithm which allow me to generate levels and so my game will essentially be endless. Beyond this, I hope to add some unique twists along the way to make the game more unique. For example, I may implement some multiplayer or some vertical parts to the game requiring users to climb vertically.

Structural Plan:

I am going to be using OOP in some capacity for my project. I hope to design a base parent class which will be used as a level base. From this I will expand different children class which will have the core same requirements for a level but will have different features, such as positions of the pivots and also some different functions.

I hope to have multiple files to make everything easier to track. This will involve files for different stages in the game such as menu and the levels.

Algorithmic Plan:

The swinging physics revolves around treating the ball as a pendulum attached to a pivot point. The calculations for this will involve calculating the angle of the ball relative to the point by using the arctan of the vertical and horizontal distance. The angular velocity and angle are updated based on conservation og angular momentum. The angle is then used to calculate the new position of ball. Upon release the tangential velocity is calculated and calculated into its components to let it transition perfectly into falling.

The surfing physics will aim to emulate the behaviour of a ball sliding along a line. First I will detect the balls contact with the line by checking if certain points intersect. If this is true, I will override the balls vertical motion to keep it above the line and preserve the gliding effect.

The gradient of the line is used to determine the slope and balls sliding directions. Adjustments will be made to the surfing mechanics to allow the ball to roll smoothly along the curves and accelerate.

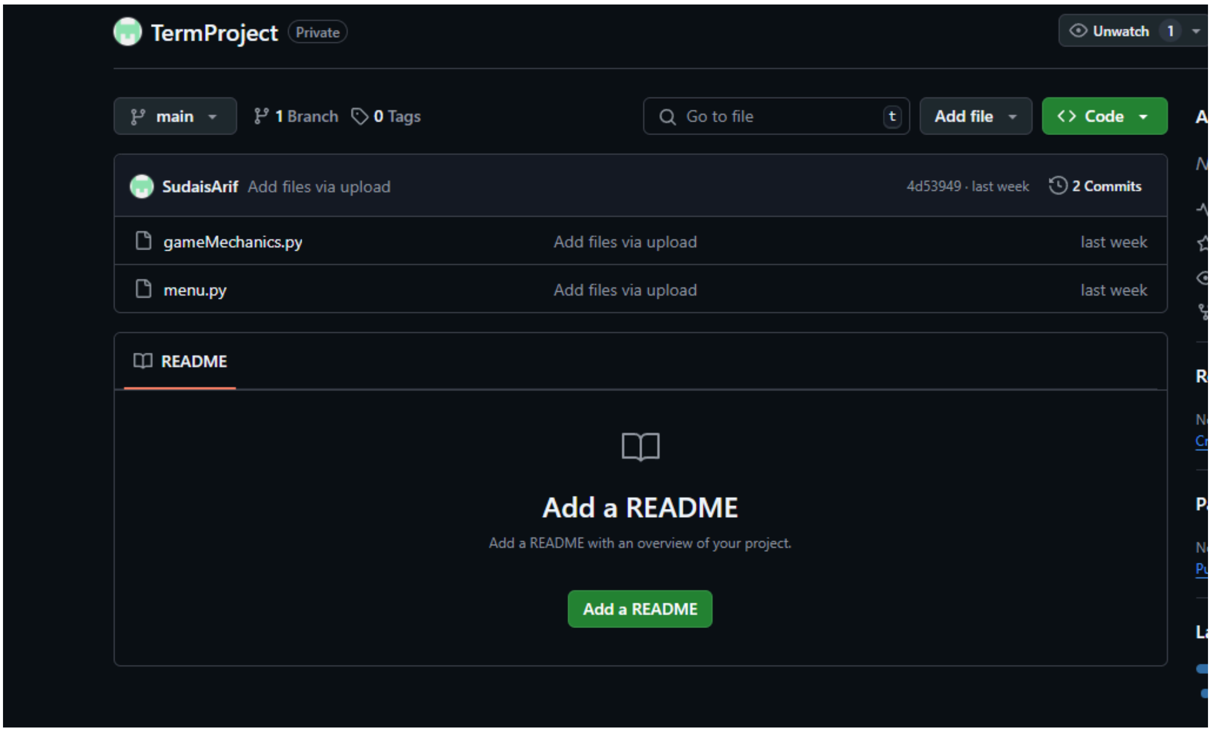
Timeline Plan:

I plan to first create and understand the physics of my game. I hope to do this by the submission of tp1. This will involve me understanding and making the game as realistic as possible. I hope to go about that by using some equations and understand how the speed of a ball is effected by when it is released from a string. I will simply just compute some calculations by hand and then translate that over into my program. For surfing, I will look to see what the best physics will be so that my ball constantly rides the user drawn line and i think this will involve a “quadratic” shape so that the ball loops.

By tp2, I hope to have a large number of levels which have different qualities and levels which are unique. Past MVP, there are some additions which will enhance the experience. This may be changing the sprite where physics are more complicated.

Version Control Plan:

I am implementing a private github repository which I will upload my code to and so will have a backup.



SudaisArif Add files via upload	4d53949 · last week	2 Commits
gameMechanics.py	Add files via upload	last week
menu.py	Add files via upload	last week

Module List:

Up to MVP, I believe will only require the cmu graphics library and also some other standard libraries such as random and math. Past MVP, there are a number of libraries I may implement which could include numpy.

Updated Design Doc:

I have incorporated some new mechanics into my game. Firstly, I have included the addition of flying pigeons which will come across the screen and if the plater collides with them they lose. To combat these pigeons, the user can use some of their line drawing ability to kill the pigeons similair to fruit ninja. I have also added some bullets which are shot from the right screen. These are honing and so follow the player. These can be blocked by a line almost like a shield.

TP3 Update:

For this i have added some physical gestures which can be implemented into the game. For example, it will access the cameras and when the user does a spiderman symbol they will swing. A gun hand gesture will kill the nearest pigeon. I have added an ai which can complete the last level and also a voice controlled level at the end.