Greg Budhijanto

15112 Term Project Design Documents TP3

TP3 Update

Since TP2, I've updated the user experience by setting up a main menu page and an interactive tutorial to introduce the player to the game controls. Additionally, I've created different difficulty modes to chose from. I've also made some tricks I previously coded locked for the user to unlock after reaching higher scores. These locked tricks also score more points. Last but no least, I set up a high scores page from the main menu. If the player earns a new high scores, they are able to record it with their initials. The game will recall these scores from a separate txt file each time it is open. See new storyboard below.

Project Proposal

Project Description (TP2)

I changed the name of my term project to 'Shredder 112' and it is still a 2D physics based, side scrolling skateboard game. The skater has 30 seconds to get his/her highest score. Starting out, the skater can only ollie/jump, the terrain will be flat, and the obstacles will be easy cones. However, as he/she reaches certain score, more tricks and terrains (that would yield higher points) would be unlocked.

Competitive Analysis (TP2)

Similar games I've found online include: *Stickman Skater: 360 Epic City* and *Skateboard Hero.* My game will be similar to these game in that they will also be 2D side scrolling game however, I plan to allow the user to have more creativity in trick choice, speed control, and overall better physics. Stickman Skater and Skateboard Hero perform arbitrary tricks and do not simulate a realistic physics when pushing and cruising the terrain.

Structural Plan (TP2)

My structural plan is that there will be a welcome screen that allows the user to pick a game mode (time trail, free skate, instructions). In time trail, you can unlock more complex terrains (as possibly characters too) from earning scores that meet the map level's required score. In free skate you can select what types of obstacles and terrains you would like to skate based on what you've unlocked in time trail. Instructions are how to play.

In the game play, the main three classes are skater, obstacle, terrain. The skater class primarily composes of different tricks and their associated sprites from the skater's sprite sheet. In this class, the skater also has self.state which indicates whether the skater is pushing, cruising, or in air. This will help determine what tricks are allowed based on the skater state and if what he is doing will trigger a collision. The obstacle class is a sprite sheet with the different obstacles that would be called upon from app.started. In app.started, there will also be a list for the obstacles to populate as the enter and exit the screen. Like the obstacle class, the terrain class will be called upon from app.started and will populate a app.terrains list.

In keypressed, different keys will be assign for different tricks/functions. Based on the skater's state, there may be keys that will not work. In timerfired, different calls to the skater, map, or terrain's location will be updated based on the skater's state or the speed he is moving at (which is listed as app.run). Also in timerfired, a collision check will be called to see if the

skater intersects an oncoming obstacle. If so, it will trigger a bail state which would animate his brutal fall and bring his speed to a stop.

Redraw all draws the terrain, obstacles, and skater as expected

Algorithmic Plan (TP2)

For my MVP, I've integrated vector physics on an incline plane into my project. The increase or decrease of the skater's speed is a trigonometric function based on the slope angle. Additionally, for collision, the skater's speed is recorded and multiplied from for the skater's distance of travel. For the smart map generation, new functions are made to record the object the skater falls on. This object is added to a list that new obstacles are randomly generated from. So if the skater fails on more on a specific obstacle, more of that obstacle will show up in generation.

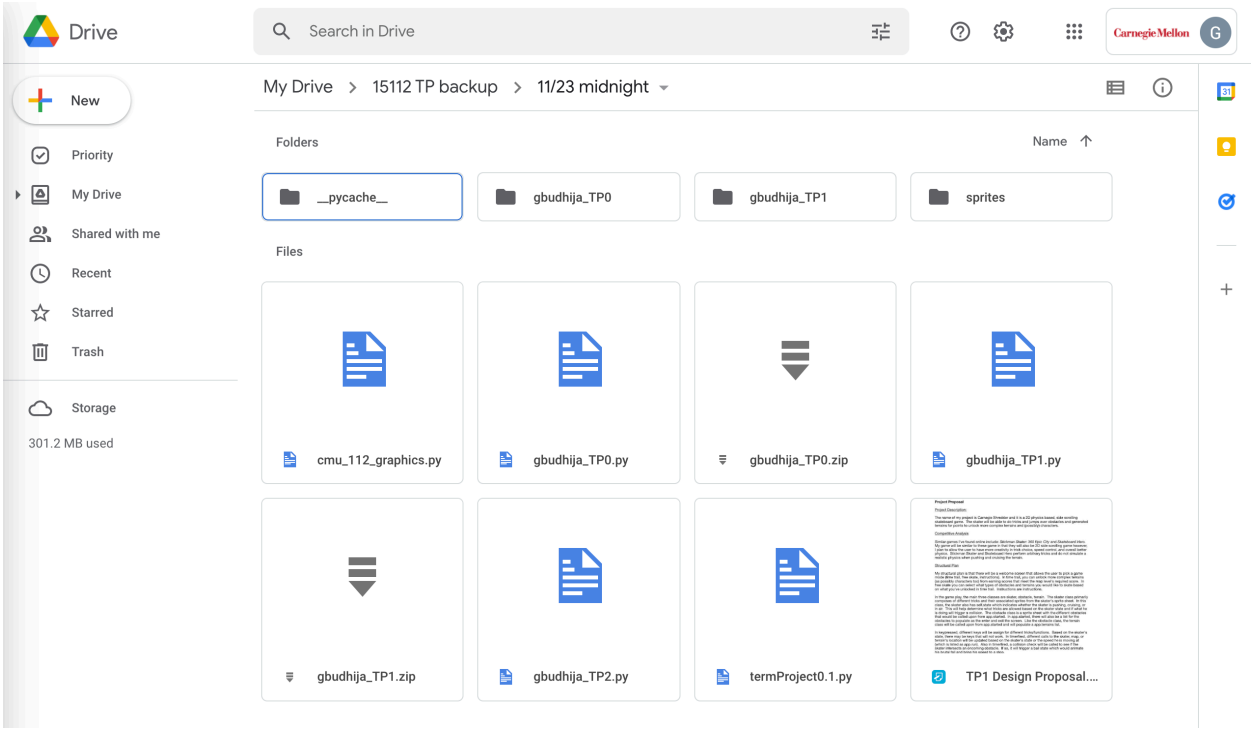
~~The hardest part of my project is incorporating the physics into my speed and collision. To do this, I plan on looking up and using vector based formulas to increase and decrease speed based on the terrain. Likewise for collision, I will consider his velocity when he hits the object to determine his trajectory on falling.~~

Timeline Plan (TP2)

Hopefully, I've achieved MVP at this point (TP2). My goal leading up to final submittal is to add complexity to the game play by adding some different game modes and other terrains/ obstacles.

~~At this point, I have a pushing and cruising skater with 3 tricks/jumps. I also have different 2 obstacles coming and a randomly generating terrain. My goal for TP2 is to better incorporate the physics features and smart map generation that will keep track of failed tricks. Then after tp2 (Tuesday, November 23) I plan to work on adding more tricks and obstacles and updating the overall user experience of the game.~~

Version Control Plan (TP2)



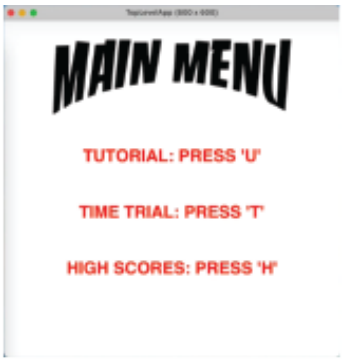
I am backing up my files on google Drive

Module List

None

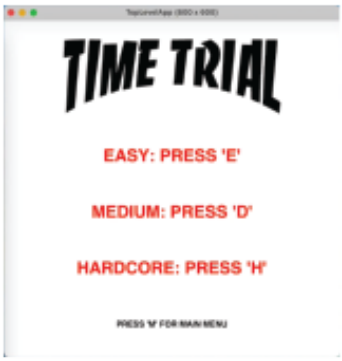
Final StoryBoard

WELCOME SCREEN



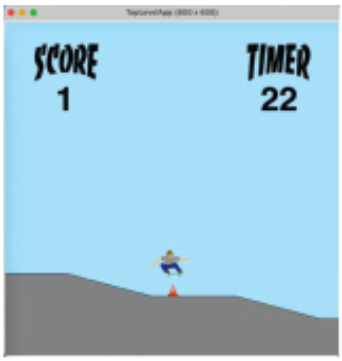
MAIN MENU

TUTORIAL

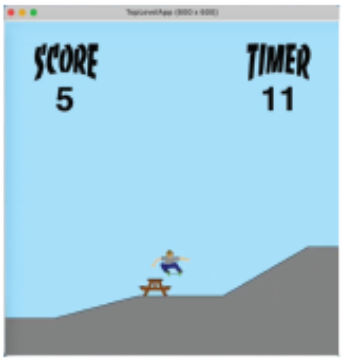


TIME TRIAL MENU SCREEN

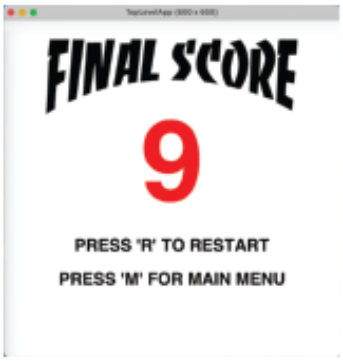
EASY GAME MODE



MEDIUM GAME MODE



HARD GAME MODE



FINAL SCORE SCREEN

UNLOCKED TRICK

NEW HIGH SCORE

HIGH SCORES