Term Project Proposal (TP3)

Name: Zirui Wang
AndrewID: ziuiwan
Section: B

Project Description:

The player starts with a car on a platform that contains 3-5 tracks, and the player needs to pass the blocks that are on the tracks to get score and use skills. Blocks can have various types, and there are also barriers blocking and attacking the car to go forward. Players need to get as many blocks as possible in order to increase their scores and finish the game with shortest time they can. The final score will combine these two subscores. Each player will have a username and a password (can be local). They can use scores to upgrade their vehicles.

Competitive Analysis

This is a music game related to beat detection, specifically beats that exist in different bands of frequencies. The basic model for this project is a mobile game called "Beat Racer." However, in my design, all game music will be in the genre of Future House since it has a very strong mid-range frequency melody line (easy to detect beats) and a relatively static drum patterns (can be used to do some regular jobs in the game). At the same time, different melody lines can represent different game levels depending on how aggressive they are. The general idea that player needs to move the vehicle to get the blocks on the track is the same for both games whereas in my game, I also want to add skill features so that the game will have more playability depending on player's combinations. The real game is a 2.5D video game; however, due to time and arts limitation, the term project will be a 2D video game. In a lot of similar games, players are unable to choose number of tracks and quantize the vehicle's HP and SP, and in my term project, I decide to implement things I mentioned above

Structural Plan

IMPORTED MODULES:

import pyaudio as pa
import wave
import numpy as np
import time
import ~~threading as th~~ multiprocessing
~~#import matplotlib.pyplot as plt~~
import aubio
import math
import string
import random
~~from tkinter import *~~
import pygame
import pyignition
Import pickle
Import sys
from pydub import AudioSegment as aseg

CLASSES:

1. Player
2. Block + SpecialBlock
3. Vehicle
4. Barrier

~~5. Skill # not sure if I should consider it as a class~~
DELETED as time is not enough for me to implement everything, but basic skills are done!
6. Track # contains block, vehicle and barrier

ANIMATION:

~~Tkinter with timerDelay == 10 (to increase animation fluency)~~ **Replaced with pygame due to higher efficiency**

Notes: timeline at the last page

Algorithmic Plan

1. **Music Preload for 2 parts**
    1. Used spectral difference to detect the beats (Jonhatan Foote and Shingo Uchihashi. The beat spectrum: a new approach to rhythm analysis. In IEEE International Conference on Multimedia and Expo (ICME 2001), pages 881884, Tokyo, Japan, August 2001.)
    2. When game starts, music is playing, but all time points have already loaded
2. **Saving/Login Feature**
    1. When player saves his process, all data will be written into another file
    2. When player wants to continue his process, **he just needs to type the saved file into the run function and everything will be back!** #the main file will check out current files to see if a #particular username and its corresponding password exist
    3. ~~Ranking list is stored in a separate file as well~~ **Every player will have his own ranking record!**

Version Control Plan

Each time I modify the file I will save a current version, eg.0.0.1, into my local storage.

At the same time, I will also upload files to Github to keep the right version

Extra Ideas beyond the Project:

1. ~~Change the background color according to low-frequency beats (raw sound data with high pass)~~
2. ~~Put the blocks on tracks according to mid-frequency beats (raw sound data with band pass)~~
3. ~~Adjacent blocks should be either put on the same track or adjacent track~~
4. ~~Barriers are randomly put on the platform,~~ #and they can cover several tracks
5. ~~*Once the player passes the block, it will make sound in which the pitch is identical to that in the original music~~
6. Players can upgrade their cars in terms of HP, SP, Attack, Special Skills, and Jump Ability Done!
7. Each User has its own data set about his vehicle Done!
8. ~~There are more than one skills that player can choose from the skills pool~~
9. Player has to pass special blocks to earn more SP in order to use special skills Done!

Timeline Plan

By Monday 11:59 PM

1. ~~Vehicle can go on the track (actually the moving objects are blocks!) to get blocks (generally not hard, but lots of codes here!)~~
2. ~~No saving/login features, but it can record score for a single game~~
3. ~~Barriers can block and make damage to the vehicle~~
4. ~~Vehicles are able to attack and jump through barriers~~
5. ~~All barriers and blocks are generated randomly without music~~ **And now it is with music!**

By Tuesday 11:59 PM

1. ~~Optimize current algorithm~~
2. ~~Beautify the scene and vehicles by using images~~
3. ~~Debug potential problems~~
4. ~~Finish the rest of Monday plan, if not sufficiently implemented~~
5. ~~Make a TP2 demo! (toughest!)~~

By Wednesday 11:59 PM

1. ~~Add music pitch detection feature (toughest!)~~
2. ~~Preload music to get a list of time points where the block should touch the vehicle if on the same track~~
3. Preload music to get a list of time points where background color should be changed **SKIP**
4. ~~Use tkinter's timer to keep track of current time~~
5. ~~Assign players with accumulated coins and scores to upgrade their vehicles, but not permanent memory~~
6. ~~Add instructions and buttons that can activate different GUIs, or they can be just triggered by keyPresse~~
7. **Changed the platform to pygame!**

By Thursday 11:59 PM

1. ~~Add login/save features and a permanent ranking list (toughest!)~~
2. Expand skills pool **SKIP**
3. ~~Add a memory feature - once it preloads a music, it won't load it again when music is played next time~~ ~~Unnecessary - skip~~
4. ~~Debug everything~~
5. Add a computer player or multi-player features time if time is available* **SKIP**
6. ~~Add a user custom music feature if time is available~~
7. ~~Add a monitoring feature if time is available, that is, when vehicle hits the block, the computer will make a sound that has the same pitch with the original melody line.~~ ~~Added particle effects instead!~~

By Friday 4:30 PM

1. ~~Make a Video Demo~~
2. ~~Finish Design and Documentation~~
3. Check for style problems
4. ~~Beautify more and more if possible (in order to show off at live demo :-))~~
5. Return 42!

>>42