Design Document for Final Project – Roaring Engine OL

Young Fan, SNO:10389048

July 11, 2012

1 Introduction

This is yet another racing car game, however it has two significant features. First, it's an online game implentmented in Python. Second, players are able to *drift their car* while racing, which depends a lot on the application of physical principles.

Here's the definition of "drift" on Wikipedia.

`http://en.wikipedia.org/wiki/Drifting_(motorsport)`

And a short video about drifting.

`http://www.youtube.com/watch?v=E877e3xZOmw`

2 Game Logic

2.1 Rule

The track

The tracks in this game is similar to the ones in reality or in other games. The players should drive along the road, and there will be bars or somethings else to keep them from running outside the road. There will be a given number of laps for one game, the first one who finish all the laps wins the game.

Play mode

There're two play mode available, the practice mode and the LAN play mode. In the practice mode, there's only one player(car) on the track. This is a good place for the player to practice his/her drifting technique. In the LAN play mode, several players compete with each other on the same track.

2.2 Entitis and their Relationship

Threr are 5 layers in this game, namely, (from bottom to top)

- background
- ground
- main
- status
- message

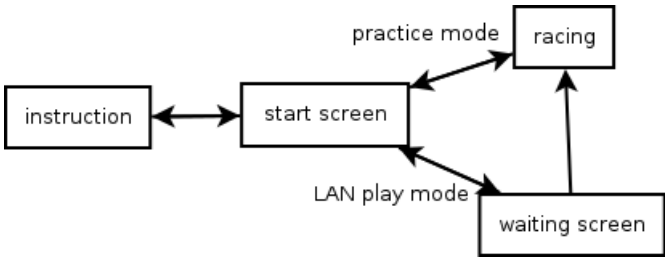
The track is placed on the background layer, Cars are on the main layer. They

- Collide with other cars on the same layer.
- Collide with the bars of the track.
- Leave the drifting trace on the ground layer.

Collision will cause speed loss. Moreover, collision between cars will affect their driving directions.

The status layer contains some game status, and the message layer is used to display pop out message.

2.3 State Diagram



The start screen is displayed first, asking the player to choose a game mode to play, or to read the instruction. If practice mode is chosen, the game start directly. Otherwise if LAN play is chosen, the player has to wait for other players in the waiting screen until all players get ready.

3 Milestone plan

The whole project is divided into 4 stages.

Stage 1:Physics Details

In this stage I'll do some research about the physics related with driving and car drifting. I'll try the Pymunk[1] library, which simplifies physical analysis.

Stage 2:Networking Details

In this phrase I'll be working on the connecting problem. I should choose a good strategy for connecting, e.g. UDP or TCP connection. Also how data is organized and transferred should be considered, too. Maybe I should synchronize all player before every frame update. Another problem is that how players can find each other so that a match can be organized quickly(maybe through UDP broadcast?).

Stage 3:Building the working game

With the progress gained in stage 1 and 2, I should be able to build the working game. Graphical stuff will be done with GIMP or mtpaint. I'll work on the music later using Hydrogen, Rosegarden, Jack, Ardour, etc.

Stage 4:Improvement

I want to do these two improvements if time permits,

- Refine the graphical and musical design
- Make the game playable through the internet.
- A match can be "recorded", so that the player can play with himself/herself in the practice mode.

4 Data plan

Track

The track is stored in a bitmap-like fashion. Every pixel has its properties, for example, whether it's an obstacle, how about the fraction at this place.

Synchronization

The only thing that will be transferred is the commands(turn left, brake, etc.) by the players. The commands from other player controlled *his/her* car on *your* screen. So action should be synchronized before the update of every frame. And the game should promise that the same series of commands will always produce the same outcome.

[1] https://code.google.com/p/pymunk/

Record

If time permits I'll try to implentment recording the game. Since the same series of commands will always produce the same result, the only thing that need record is the commands.