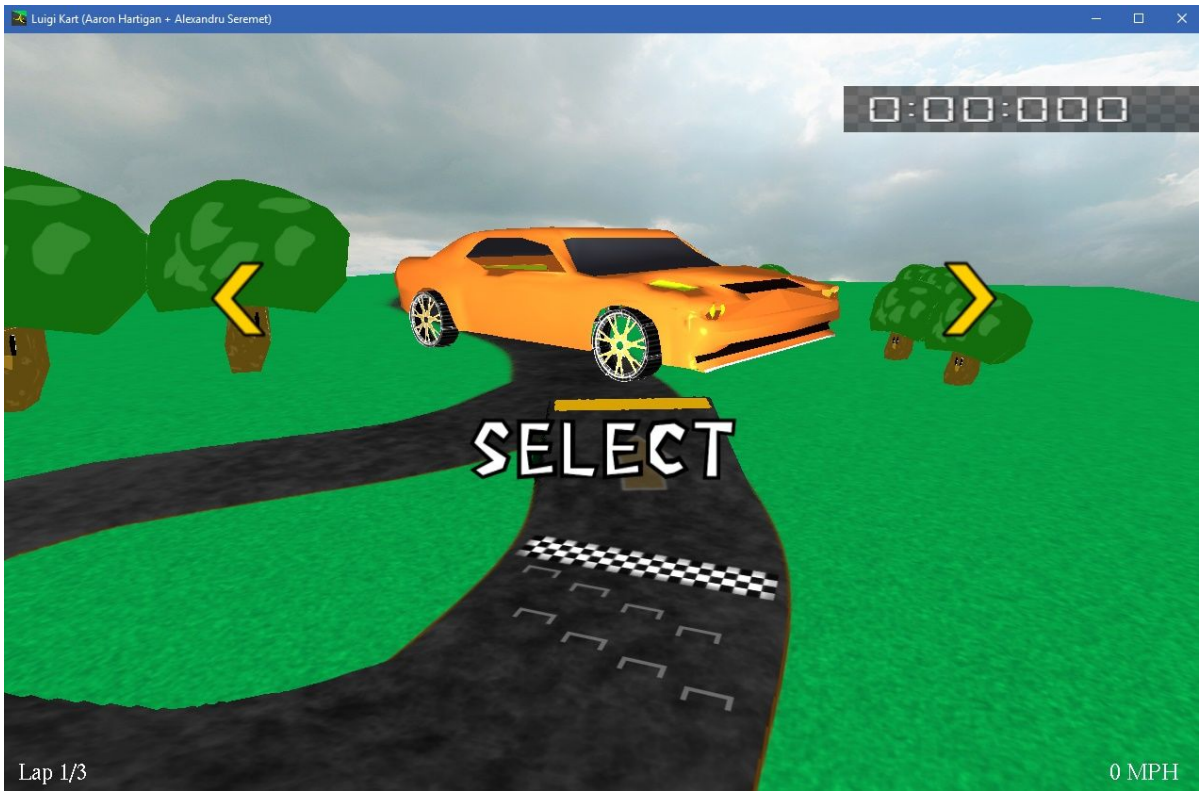
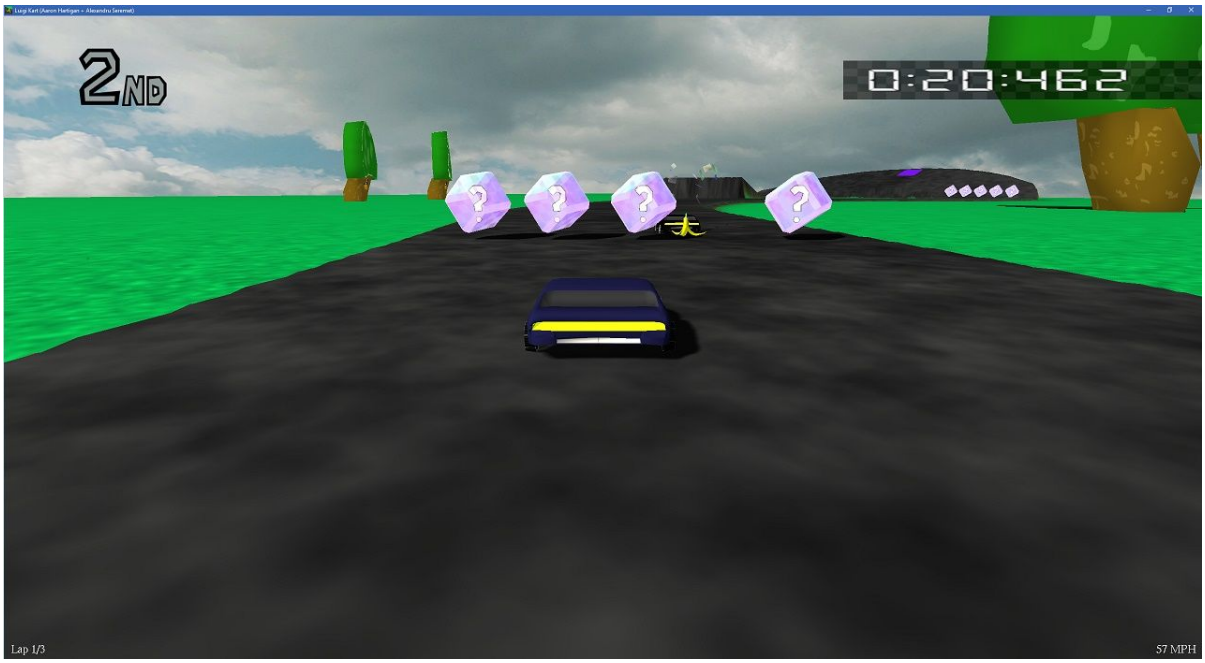
Luigi Kart

1.) **Luigi Kart** made by Aaron Hartigan and Alexandru Seremet
2.) **Screenshots**





3.) Code Compilation

Compiling the code can be done in two ways:

1. Double-click on compile.bat (try compileAlt.bat if compile.bat does not work)
2. Typing compile in the command line (try compileAlt if compile does not work)

4.) Special Device Requirements

The game does not require any particular video card, but the experience won't be the same as when having a good one. User is able to play the game using a keyboard or a game controller (the game was tested with an XBOX controller and a Logitech controller and a keyboard).

5.) How to Play

The game can be played in single-player mode or in networked mode. To run the game in single mode, click run.bat or in CLI use the command run. To play the game in network mode the user must start the server first by pressing runServer.bat or using the CLI command runServer and then launching the game with run.

The game is best played in network mode, but the single mode still works. In single mode the player can drive around the course but will have nothing else to interact with. At the beginning of the game the user can select a color for the car. Colors can be selected by pressing A or D on the keyboard. After selecting car press enter to enter the game (or start on a gamepad). The car will then appear on the track, ready to race. To start the race, press enter on a keyboard (or start on a gamepad) and the server will load all the other players and generate NPCs. The game may be played between up to 8 players. The goal of the game is to complete three loops around the track first. The player's "score" is their ranking in the race. Item boxes exist on the track for players to pick up and use items against each other. Hitting another player with an item such as a banana will cause their car to spinout. Driving off of the road will decrease your car's speed.

6.) Controls

Gui Interaction - Enter (keyboard) or Start Button (gamepad)

Accelerate – W (keyboard) or A (gamepad)

Deccelerate – S (keyboard) or B (gamepad)

Steer Left – A (keyboard) or Left on Left Joystick (gamepad)

Steer Right – D (keyboard) or Right on Left Joystick (gamepad)

Drop Item – R (keyboard) or X or Y (gamepad)

Drifting - Q or E (keyboard) or Left or Right Triggers (gamepad)

7.) Scripting

Scripting is used to control the position, scale and rotation of the trees which serve as decorations.

8.) Game Type

Our game is in the racing genre.

The theme was originally intended to be cartoony, but it ended up being more of a real-world theme.

Motion mostly takes place in a 2D motion on the ground, but the cars can reach the third dimension by driving up inclines.

Activities in the game include driving, picking up items, and throwing items.

9.) Game Requirements

External models: cubes, bananas, trees, cars and wheels. (Wheels are not part of the cart model, and they have own controllers).

Networking: NPCs, items, and the itemboxes are done entirely through the network (they do not exist on any client). In addition, up to eight human players can race together.

Scripting: The scripting is done to control the location, rotation, and size of the trees.

Skybox and Terrain: The skybox is implemented with the RAGE skybox. The racetrack is our terrain and is based on a height map. The physics uses the height of the terrain to handle collisions.

Lights: One light is used at the beginning of the game when user is prompted to choose the car. When the car is selected, that light turns off. The light is turned back on after a race is finished, and the player goes back to the car selection screen. There is another light that is used to light the entire game world.

3D Sound: The game uses Hava Nagila as a background music. 3D sound effect is produced by the cars passing by or flying over when they jump out of ramp.

HUD: Game uses many HUDS. Each corner of the screen during a race displays different information - the timer, the player's position, the lap number, and the player's speed.

Hierarchical SceneGraph: The car is made out from hierarchical scene nodes. The body is one scene node and each tire is another scene node attached to the body.

Animation: Being a racing game we had problems to come up with an animation so we decided to animate the tree by simulating wind.

NPCs: The game can be played by 8 players. If fewer than 8 players join, the unclaimed positions will be filled filled with NPCs controlled by AI.

Physics: We did not use any of the jBullet physics. Our game uses custom physics that we found to be more suitable fo us.

10.) Requirements Not Working

All requirements are working.

11.) Techniques Beyond Requirements

Shadows via shadow mapping, particle explosions, custom shaders for GUI, transparency, and a friction map (similar to a height map).

12.) Contributions

Both team members worked on all of the game functionality. Aaron coded the techniques that were beyond the requirements.

The models were created as follows:

Banana and the item box - Aaron

Car, wheels, trees, and animation - Alexandru

13.) Items Created by Ourselves

Almost every single resource used in this game was created from scratch. The only exceptions are the skybox and the music/sound effects.

The list of textures we created:

0.png, 1.png, 2.png, 3.png, 4.png, 5.png, 6.png, 7.png, 8.png, 9.png, colon.png

banana.png

blue1Tile.png, blue2Tile.png, gray1Tile.png, green1Tile.png, red1Tile.png, yellow1Tile.png

car1.png, car2.png, car3.png, car4.png, car5.png, car6.png, car7.png, car8.png

chevron_left.png, chevron_right.png

first.png, second.png, third.png, fourth.png, fifth.png, sixth.png, seventh.png, eighth.png

tree.png

wheelSpikes.png

timerBackground.png

itemBox.png

questionMarkBody.png, questionMarkDot.png

track1_texture.png

height_map.png

List of skeletons we created:

tree.rks

List of animations we created:

tree.rka

List of meshes we created:

plane.obj

itembox.obj

questionmarkdot.obj, questionmarkbody.obj

banana.obj

tree.obj, tree.rkm

wheelSpikes.obj

car1.obj

14.) Evidence of Permission

For the background music we use a copyrighted melody Hava Nagila by Nick Tacu. We have contacted him to get the permission to use it in our game. Nick speaks Romanian, so Alex wrote him in Romanian to ask for the permission, but he also has translated the emails to English.

Music Permission:

Alexandru Seremet <seremet_alex@yahoo.com>

To:nick_tacu@yahoo.com

May 14 at 10:27 PM

Original: Salut Nicu. Eu mă numesc Alexandru Șeremet. Sunt student la Universitatea de stat din Sacramento (din câte am auzit si tu locuiești în Sacramento). Îmi place Hava Nagila la nai și vreau să te întreb dacă îmi permiți a folosi această melodie pentru un proiect la școala. Vreau cu permisiunea ta sa adaug aceasta melodie ca background la o joaca. Aștept răspuns.

Translated: Hello Nick. I am Alexandru Seremet, student at CSUS. I like the song Hava Nagila on pan flute, and I would like to use your melody in a school project. With your permission I would like to use it as a game background.

- Nick Tacu <nick_tacu@yahoo.com>

To:seremet_alex@yahoo.com

May 15 at 2:57 AM

Original: Salut Alexandru.da o poți folosi,Mi-ar place sa îmi trimiți și mie sa vad când realizezi acel joc.succese...

Translated: Hello Alexandru. Yes, you can use it. I would love you to send me the game to see the outcome. Success...

Nick Tacu

https://www.youtube.com/user/9542570209

www.nicktacu.com

https://www.facebook.com~~/nicktacutenorepanflute~~

Skybox Permission:

https://opengameart.org/content/cloudy-skyboxes

Which gives the license Attribution 3.0 Unported (CC BY 3.0)

Car Sound Effect Permission:

https://opengameart.org/content/car-engine-loop-96khz-4s

Which gives the license Attribution 3.0 Unported (CC BY 3.0)

15.) Machines Tested On

The game was tested on Centipede and Crush in RVR 5029.