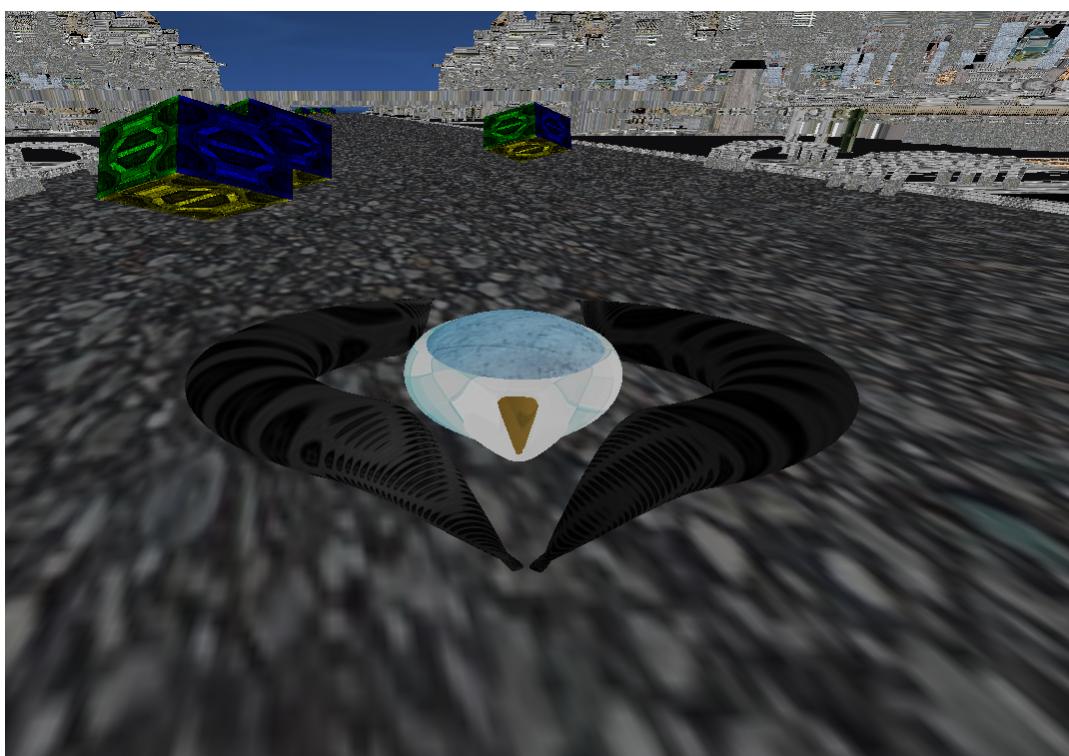
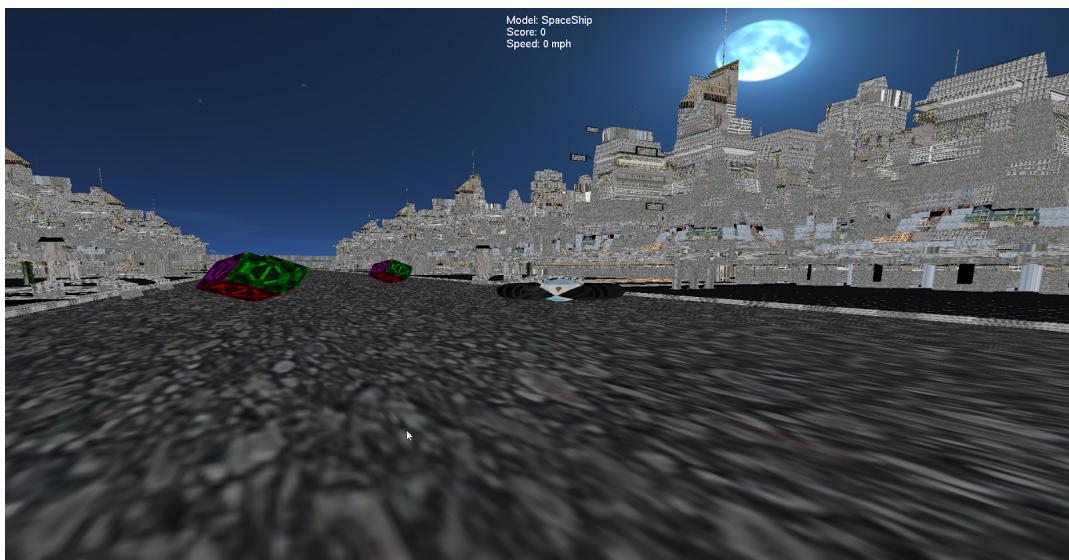
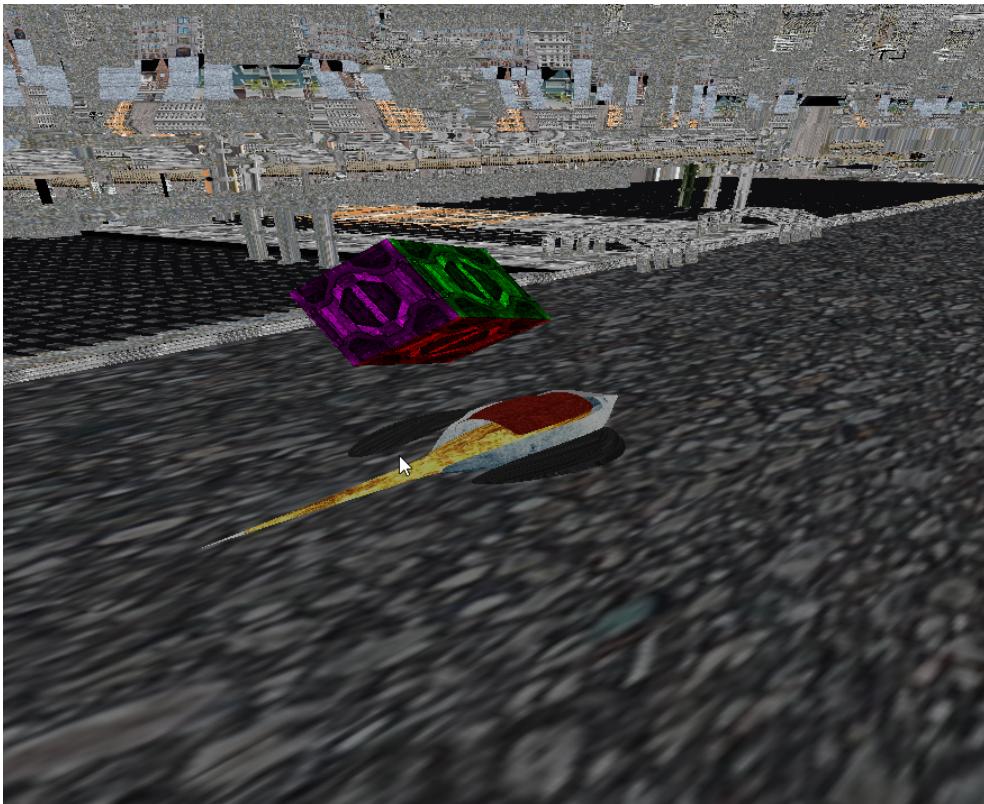
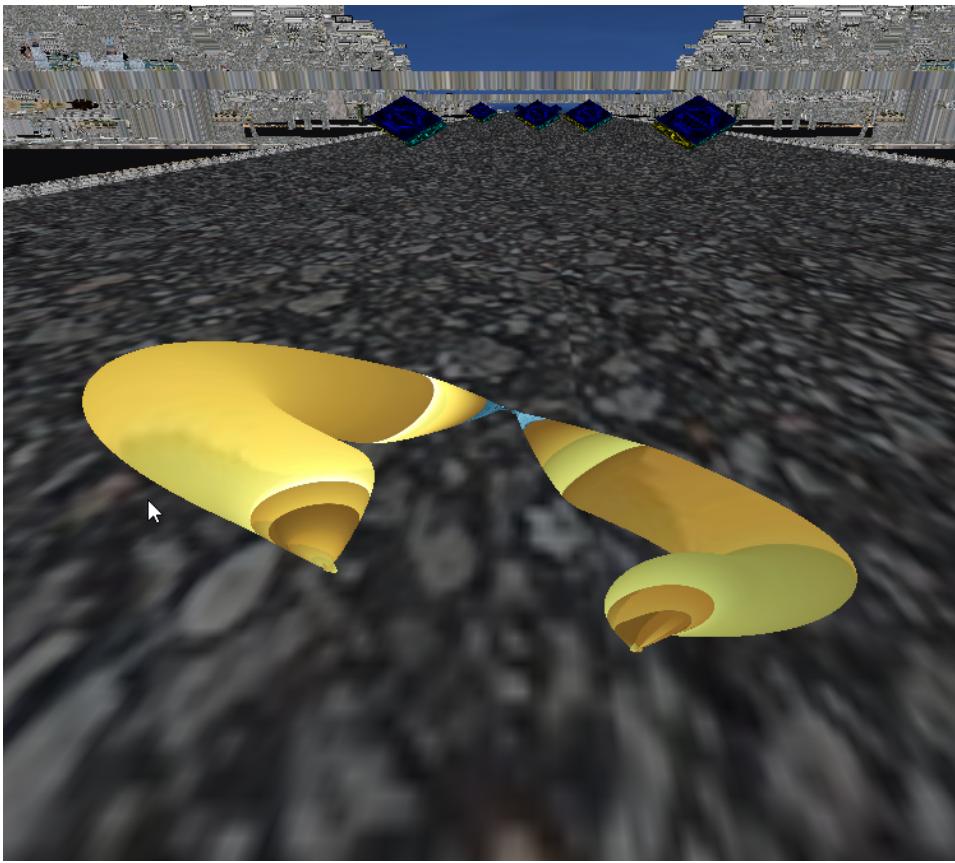


1. The name of your game, and your names

- a. Game Name: Space Race
- b. Arash Parnia, Max Kondrashov, Javier Garrido

2. At least one image (.jpg) showing a typical scene from your game being played





3. How to compile and run your game

- a. To compile this game, you have to open the project in either IntelliJ or Eclipse.

Both project files are provided. We talked to you about it not being able to compile in command line because it throws Package not Found errors, even though our packages are all correct. The IDE's somehow set up the files to all work correctly, so you can compile and run it through Eclipse/IntelliJ.

4. Any special device requirements, such as a particular video card or input device(s)

- a. At least 3 gigs of RAM
- b. Minimum specs: Dual core processor
- c. Recommended: Quad core processor

5. How to play your game, including what things happen and how the scoring works

- a. The game consists of racing down a city and collecting boxes for points.
- b. Every box you collect gives you 10 points and gives you a small speed boost.
- c. You want to avoid NPC or other human players that are also on the track as they will try to bump you towards the edges of the track and slow you.
- d. At the end of the race, your total points will be displayed and the player with the most points wins.

6. What player controls are available (what keyboard keys do, etc.)

- a. Keyboard and the Num Pad with Numlock disabled:
 - i. W,S,D,A – move forward, backward, right, and left respectively
 - ii. Spacebar - Jump/Fly if pressed repeatedly
 - iii. Up – pitch camera up
 - iv. Down – pitch camera down
 - v. Left – look left AND rotate player left
 - vi. Right – look right AND rotate player right
 - vii. HOME – look left without rotating player
 - viii. PGUP – look right without rotating player
 - ix. END – Zoom out
 - x. PGDN – Zoom in
 - xi. F2 - Turn Physics on/off
- b. Gamepad:
 - i. X-axis: move player left and right in the same manner as the A and S keys.

- ii. Y-axis: move player forward and backward in the same manner as the W and D keys.
- iii. A Button: Jump/Fly if pressed repeatedly
- iv. Right Bumper - Look and rotate player right
- v. Left Bumper - Look and rotate player left

7. How it uses scripting, and how an end user would use the scripting

- a. The game uses scripting to specify the amount of boxes that are shown throughout the track. An end user would modify this number to have the game generate more or less boxes depending on what they want. The input is a simple number.

8. An explanation of how the game satisfies each of the requirements listed above

- a. Skybox and Terrain: Our terrain was originally our maze, but since we changed the game, we substituted terrain to be a ground floor and the city mesh in which the race takes place. The skybox surrounds the city with a night sky design
- b. Networked Multi-Player: The initial GUI for the game allows multiple players by having one player be the host and the other/s join his hosted game.
- c. Scripting: Our game uses scripting to set the amount of boxes that will be spawned all throughout the race track.
- d. External Models: We have several different external models - Spaceship, space-pod, spaceship animated, car.
- e. Skinning: Each one of our models uses a custom made skin through UV-unwrapping
- f. Events: We use events to handle collisions with boxes as well as the finish line.
- g. 3D Sound: Our game has several sounds including one for picking up a box and general racing sounds
- h. HUD: Each player has his/her own HUD at the top of the screen showing their current score and speed
- i. Hierarchical SceneGraph: Our car is a hierarchical scene graph made up of several parts
- j. Animation: The animated spaceship has a skeleton and keyframes to allow it to animate it spinning
- k. NPCs: We have several NPCs made up of our external models all of which follow swarm game tactics to try and push human players off of the race track and beat them to the end.

- i. Physics: Most of our game is run by physics, including acceleration and velocity, jumping, collision between avatars and the race track, and collisions with the boxes.
- m. Game first-person (1P) or third-person (3P) view: Player avatars are visible in remote clients
- n. FSEM: Our game has FSEM included and works, but it is turned off by default since it caused some issues on some of our machines.
- o. No hard coded IP: We do not have IP addresses hard coded, instead we provide a GUI for player to enter the IP of the player hosting the game they want to connect to.
- p. Single Player mode and Multiplayer mode: Our game supports both single player and multiplayer mode and it is easy to select which you want in the initial GUI.
- q. Resource relative paths: All resources used are accessed using relative paths and are not hard coded.

9. Any of the requirements that you weren't able to get working

- a. Our game satisfies all requirements

10. Anything technique you used in your game that goes beyond the requirements

- a. Our NPC's follow Swarm Intelligence for their AI

11. The contributions of each team member, including who designed which model(s)

- a. Arash
 - i. Artificial Intelligence for NPCs, Terrain, Hierarchical SceneGraph, 3D Sound, Game logic, Physics, Skybox, City mesh, Cubes
 - ii. Model designed: Mushroom, Chess piece, Car
- b. Max Kondrashov
 - i. Networking, NPC, Physics, HUD, FSEM, Game logic, Camera Controller, Swing Menus, Scripting, Cubes
 - ii. Model designed: Space pod
- c. Javier Garrido
 - i. Scripting, Designed Space Ship Model and Animated Space Ship, Model Texturing/Skinning, Model Rigging and Animation, Documentation, Gamepad controls.
 - ii. Model designed: Spaceship

12. Which RVR-5029 lab machines (at least two – it's networked!) on which your program was tested and is known to work correctly on.

- a. It worked on Pac-Man and Halo before, but as I (Max) talked to you about it not being able to compile from command line because of “Package Errors”, but it worked on those computers before with networking. Also, I found a work around for it not working. You can run the game from Eclipse as well as IntelliJ and we’ve included both of those project files.