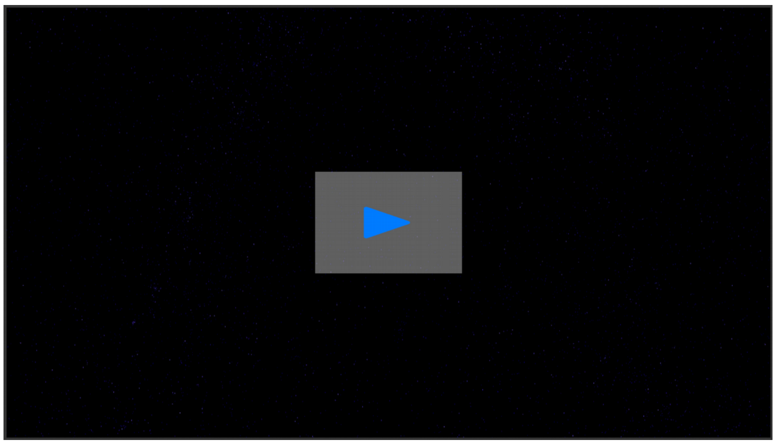
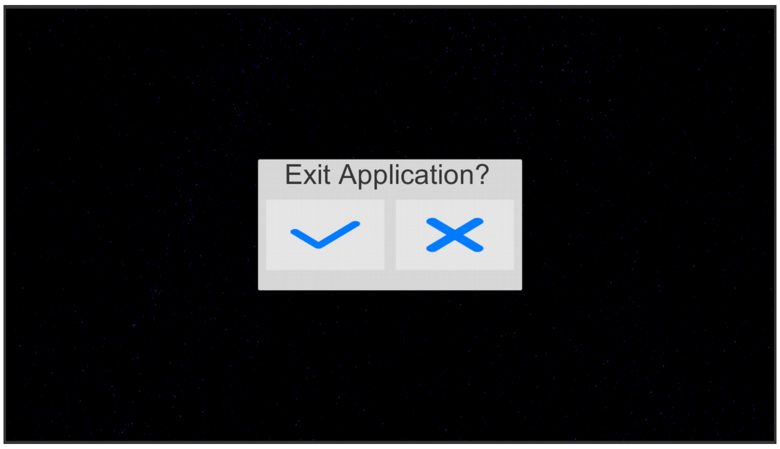
HOW TO USE

We have 2 scenes in the game: Start and Game.

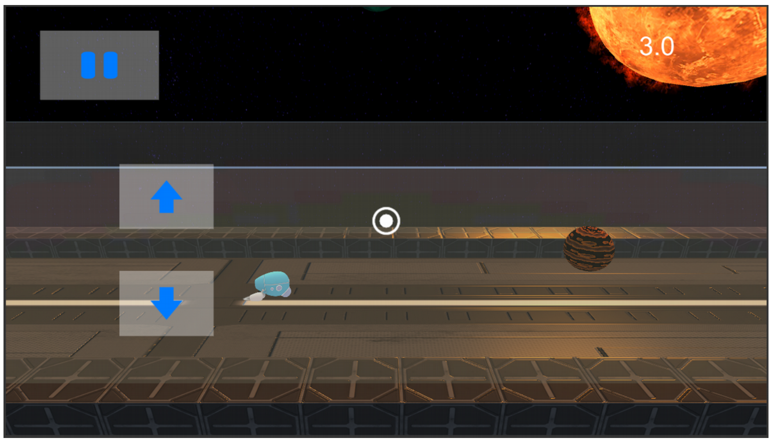
The Start scene have just one button to load the game.



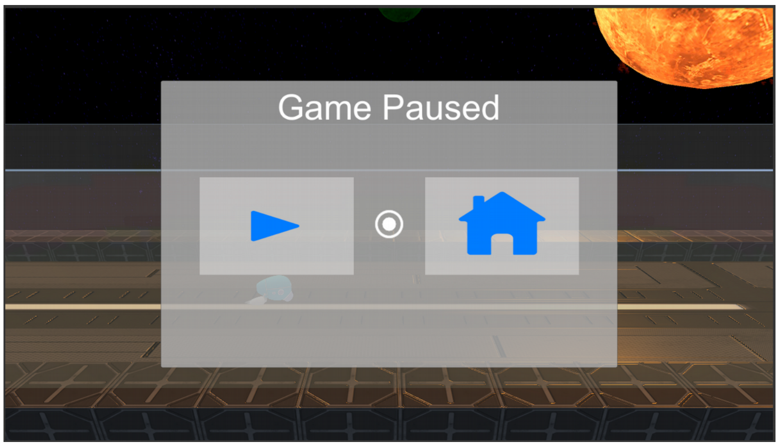
If you press Escape on the PC, or the back button on android, a pop up will show for you to confirm the application exit.



When you click play, the Game screen will show. This is where all the action happens. It's a simple infinite runner, you have to avoid the balls of lava that comes in your direction. The counter on the top right shows you how much you have traveled(a basic score system).
There are two buttons you can use to move the ship, and a Pause button. You can move the camera to view the scenery with you finger.



When you press the Pause button the game pauses and the pause menu appears. From here you can continue playing or go back to the start screen.



When, unfortunately, your ship is destroyed, we are presented the Game Over menu. It shows your score for the current run and your record. You can then try one more time to beat your current record or go back to the Start screen.

Test Requirements:

Tier 0:

Req1: I have implemented a basic initial menu and the application flow. For the game base mechanics i have used an infinite runner.

Tier 1:

Req1: The game is in a 3d world combined with some 2d sprites. The ship may look like a 3d model but it's actually a sprite with some(bad) animations. The UI is in World Space, not overlay, and, as we'll see later, the reticle it's positioned in the 3d world as well. (I researched the unity sites some guidelines for implementing UI in VR, that's why they're in World Space).

Req2: Most of the moving objects have a Rigidbody and are moved using physics. We have collision between the ship and the lava balls, and the latter with an invisible wall on the end of the road to determine when it should "respawn"(object pool). Particles are used when the ship is destroyed and for the animation of sun flares. The lava balls are being constantly spawned from the beginning of the road, I have utilized an object pool to avoid instantiating and destroying a lot of objects.

Req3: There are normal maps in a lot of objects. You can find transparent shaders on the reticle and the windows on the side of road. The sun utilizes a self illuminated shader.

Tier2:

Req1: The planets orbiting the sun utilizes a very basic surface shader with two diffuse textures and a normal map. I exposed the color property so I could change it from scripts, this way each planet begin with a different color everytime the game is played. It's the Planet.shader.

Req2: The camera is moved by touching the screen and dragging. There's no camera zoom for now.

Tier3:

Req1: The reticle is positioned with the other objects in the 3d world and scaled accordingly as recommendation of unity. A modified built-in shader(from unity) guarantees that the reticle is always draw in front of other objects. It's Reticle.shader found in the 'Shaders' folder.

Req2: Being honest I don't think I got the vertex animation correctly. The lava ball has two textures, one for the lava and one for the delimitations. Every time the game starts the lava is painted with a different color. For the animation I utilized a basic sin wave to make the ball wobble.

Assets from the Asset Store:
- 139 Vector Icons 1.1
- Cartoony Spaceship Sprites 1.1
- Textures from Lava Flowing Shader 1.2
- Sci-Fi Texture Pack 1 v2.0
- Vast Outer Space 1.0

Shaders reference: Unity 5.x Shaders and Effects Cookbook. Alan Zucconi, Kenneth Lammers.