Infinite Space V1.0

By Ricardo Concha - Nemoris Games

"Thank you for getting this pack. Let's check how it can help you to improve your game and make things easier"

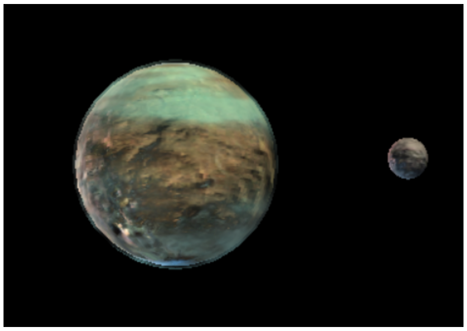
Infinite Space is an art pack to create space scenes with the 3D assets or add our components to randomize the scene and/or add a 4-direction parallax effect.

Art Assets

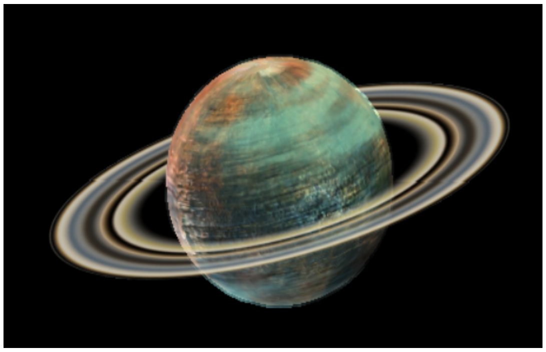
The art assets were created to be used in any kind of game even using most components, the assets will work in any environment other than a parallax kind of game.

This pack allow the user compose complex assets for his project. The prefabs included in this pack are:

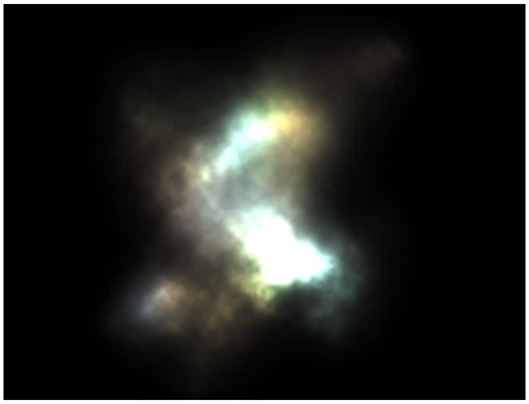
1. **3D Planet**: This asset is the most used and important in the pack. It represent a generic planet with all the scripts needed to be randomize and work in a parallax environment.



2. **3D Big Planet**: This asset is very similar to the planet, but with a 3D model more detailed and a ring surrounding it.



3. : **3D Nebula**: This asset is prepared to see a nebula in a far distance.



4. **Star Group**: A group of stars to be seen from a very far distance.

5. **Star**: A star with a blinking effect perfect for giving a sutil animation to the background.

6. **Space Manager**: This prefab controls the direction of the parallax effect.

Scripts, Behavious and Components

As said, the tipping point in this assets are the behaviours controlled by the components. This scripts are created to be used in any other way or object that the user may need.

1. **SpaceManager.cs**
   This script is used to store important information. In this case, only store and define the **parallax direction**.

2. **ParallaxMovement.cs**
   This is a very important script, it manages the parallax effect in each object using the scroll direction given by the selection in the SpaceManager object(check prefabs).

   The parallax effect uses a random speed given by **minSpeed** and **maxSpeed**. A 0f value will make the object to move along with the camera, a value less than 0f will make the object to move ahead of the camera

   To determine if the object is completely out of screen the script add a **limitOffScreen** parameter that is the viewport's value. For example, if the user want that the object's limit is a half size of the screen, then the limitOffScreen value should be 0.5f

   The **behaviourOnExit** determinate what the object should do when exit the screen (plus the limit). Destroy will destroy the object and Regenerate take the object, repositionate it according to the scroll direction and if it or his children has any random type component, it gets randomize again.

3. **Asteroid.cs**
   This script control the meteor prefabs in the assets. It position it according to the scroll direction, set the direction pointing to the center of the screen

and put the object to 'sleep' until the **spawnTime** value is reached.

4. **RandomColor.cs**
   This script chooses randomly a color from a list of colors settled by the user and applies it to the **<Material>** in the same object.

   It also has a **invisibleProbability** value that, if the random number from 0f to 100f is less than the invisibleProbability, it make the object invisible.

5. **RandomTexture.cs**
   This script chooses randomly a texture from a list of sprites settled by the user and applies it to the **<Material>** in the same object. The script also allows that another random texture can be used in the Albedo Detail map.

6. **RandomNormalcs**
   This script chooses randomly a normal texture from a list of sprites settled by the user and applies it to the **<Material>** in the same object.

7. **RandomRotation.cs**
   Make the object rotate at the speed given by **rotationSpeedMax**. If the **rotate** bool is true, the script will choose a speed between 0f and rotationSpeedMax.

8. **RandomSize.cs**
   This script apply the **multiplierMax** value to the current object's localScale, multipliying it. For example, if the object has a localScale of (1f, 2f, 3f), the new size, asuming that the random multiplier generated is 4f, it will be (4f, 8f, 12f). The multiplier used is a random number generated between 1f and multiplierMax.

Final Note

Thanks for using this asset. I hope you find it useful. I'll try my best to provide support regarding any issue.

I aim to provide support within my best ability. For any question, contact me directly at rconcha@nemorisgames.com

I definitely hope to expand and add more features. So any idea and feedback are welcome.

Please rate this asset and comment it so more people can get awared and improve thir games.

Best regards,
Ricardo Concha – Nemoris Games