How to Use

You can move around the scene using the joystick at the bottom of the screen. The bottom-left button will return you home to where you started. The bottom-right button puts you in presentation mode where all other users will come to you and follow you when you move. You can jump to other worlds by simply clicking on their cards.

Click or scroll down to continue…
Technical Overview

Gallery demonstrates capabilities that are unique and relatively simple to implement in Croquet.

Cards and Behaviors

The Gallery uses several different Croquet cards and behaviors to create this world.

terrain.js

The main environment was copied from Mike Linkovich's Terra demo. I changed out the sky and upgraded the water simulation. This also allows other objects in the world to query the quickly query the height of the terrain at whatever position the object is at. The TerrainActor is used to update the world – ensuring that the grass blows in the wind in the same way on all devices, as well as providing a random gust of wind that shakes things up even more, including increasing the sound volume.

horse.js

This is a great behavior to start with, as it implements a very simple behavior of moving the horse model around the valley in a large circle. The HorseActor class uses a Logo-like control mechanism – forwardBy and turnBy, which is an easy way to describe a circle. The HorsePawn class is listening to the horsePosition message from the HorseActor and updates it's x,y position in the world. It

then uses this to determine what its height should be as it runs on the terrain.

crowd.js

The crowd of bots used instancing with a very simple AI. The bots query the location of the avatars and of the horse that is running around the valley. If they get too close to either of these, they will quickly run away. You can create new bots using the menu accessible from the top right of the world. The color of the bots that a user creates are all the same, but that color is determined randomly.

The CrowdActor class controls the bot behavior, which ensures that they act in the identical way on all machines including their relatively primitive AI. In this case, they determine a new location to move to and then move there – but they are also responsive to user avatars and the horse running through the valley. The user can also send messages to the CrowdActor to increase or decrease the number of bots in the scene.

The CrowdPawn class is used to create and destroy the instanced bots and update their location in the world based upon the CrowdActor update. It uses the same

mechanism as the horse to determine where the terrain is underneath them.

menus.js
The menus behavior simply installs a number of additional menus that can be used to add and remove bots and toggle the sound on and off.

lights.js
The lights behavior is used to generate the cascaded shadow map and load and construct the background sky. I have not added shadows to the ground or grass, but you can see them in the temple and on the tree.

The background sky is also loaded by this behavior into the scene background.

pdfview.js
The PDF viewer, which is likely where you are viewing this document, is used to add documents into a world.

Github
The source for the Mythos world behaviors is available on Github here:

https://github.com/croquet/mythos

Credits

THREE.js Mr Doob on Github
Terra Mike Linkovich (spacejack) on Github
Horse by Mirada for Ro.me
3D Temple by Havolik, modified by Kai Oldman
Above the Clouds texture from HDRMaps.com