

Regimerunner DOC

Submission CGUE 2017SS by Patrick Hromniak and Robert Fischer

A 3D game with a structured scenegraph and different renderpasses. Currently the game has two separate passes - a renderpass and a depthpass for each camera. Each object is represented by an entity nesting different components, such as models, lights or a watersurface - influenced by the Unity game engine. An entity can also have child objects. Interactions in the game are based on operations which can be added to these objects and processed each iteration. The renderpass is able to differentiate between different objects such as normal meshes, water, skyboxes and also an optional HUD. The same shader is used for all visible models.

Features

First Person view and terrain moving on a pre-generated height map with texture splatting. Objects on the map are placed via an object map which indicates where different objects should be placed. Object collision, terrain roaming and movement is based on PhysX. Moving objects are helicopters which have an attached spotlight. They can focus the player and follow them. Each spotlight on a helicopter casts individual shadows based on the palms on the map. The player has to avoid standing directly in the light beams of a helicopter, additionally the player can hide under a palm for not getting caught by the helicopter. This process is aided by PhysX. The player has to drag wood elements to a pre-defined spot where he can build a raft for leaving the island. View Frustum Culling is implemented using Axis Aligned Bounding Boxes.

Effects

Shadowmaps (1.5 pts)
Spotlights (0.5 pts)
Moving Water (+Fresnel-Shading,Normal Mapping) (0.5 pts)
+ refraction (1 pt)
+ reflection (0.5pt)

TOTAL 4 pts

Effect Implementation

Shadow maps:

The shadow map is seamlessly integrated into the 3D engine. Just create a Spotlight component and all the shadowing is done. The camera renders a shadow map containing the depth information using a Framebuffer Object. When a model is rendered all information regarding light data and shadow data are passed to the shader. The lighting model is a standard ambient, specular and diffuse model, where the shadow casting affects the diffuse and specular color of each rendered pixel. To improve quality of the shadows a simple 3x3 percentage-closer filtering is applied. The shader supports up to 10 shadow maps, which means the engine can have 10 shadow casting Spotlights, currently the game does not use all shadow maps, because of performance concerns.

Water implementation: The water is rendered as a single quad and consists mainly out of three different textures. The reflection and the refraction in addition with a normal map and a dudv map. The reflection and refraction textures are rendered with two different cameras placed and locked onto the player entity. The refraction camera is above the player and the reflection camera is below the player. In order to render correctly, these camera are equipped with clipping planes - otherwise the reflection camera would capture the water or terrain from below. These textures are stored in an FBO and are processed in the render pass where the water model has an own section - all other models are rendered differently. In order to achieve a water surface, both textures are distorted over time by a dudv map. To switch between refraction and reflection a fresnel term is combined with the camera's viewing angle. In a usual position there is not much refraction, but the more the player looks down on the water, refraction is also visible. To achieve specular highlights, the dudv map is also used to distort a normal map to vary the waters normal vectors.

Other special effects

Night effect: the night effect displayed in the game is achieved with slight fog, interpolating a blue value and a skybox.

Blending: The marker for where the woods should be dropped uses simple blending.

Additional libraries

Assimp, Physx, glm, SDL,

Tools for models

The height map is generated by the game based on a simply B/W. The other models are used by various freely available resource websites.

Interaction sequences

To drag wood elements press E. They are then attached to the player - you have to drag them to a spot marked by a green arrow and release them by pressing E again. You need to collect 6 wood planks to finish the game.