Have you heard of the high elves?

By Ivan Parkhomenko, Dylan Kirkby and Brian Cowan

A lone elf teleported to a sky island via haywire portal – you have survived this long with just your kettle and the apple trees. The two portals here are linked to each other, but you might see equal luck to your prior lack of!

# **Technical features**

### **Particle System Component**

A buffer of transform matrices for particles is created on CPU and filled with appropriate data received from simulation of their moment, which is done based on their lifetime, velocity, position, and scale. Afterwards this data is sent to GPU and instantiated.

- Used for fire particles.

### **Animation Blending**

Previously operating on a queue system, animations have been moved so that each one performs a blend into any animation that is not identical. This is done by a by using smoother step based on the blended animations time to calculate the transform ratio for both animations, which is then applied to the transform of every bone and added together. There are several checks in place to ensure that some animations can not be blended out of early, effectively locking you in. The transition time to exit idle animations is half of any other animation for ease of use.

- Used for animation.

### **Portal**

The scene is rendered into FBO for each camera used by portals. Cameras’ positions and directions at which they are looking at are calculated based on the main camera and portals’ positions. To avoid the issue with sidelines cameras are not placed at the same position as their portals, instead they are placed behind them at the appropriate distance and angle. That creates the distance issue (distance from camera to portal is applied twice), which is fixed by scaling up the texture of portal to the camera resolution.

- Used for two portals.

### **Dynamic Reflection**

Six cameras are placed at the object’s position and looking up, down, back, front, left and right. The scene is rendered into cubemap FBO once for each camera with 90 degrees FOV and then used as texture for this object and blended with its actual texture.

- Used for teapot near the campfire.

### **Threaded Texture Loading**

While default textures are created and loaded on the main thread, everything else is created on a separate thread. Each texture is loaded on the main thread when it’s confirmed that it has been created.

- Used for textures and skybox textures.

### **Console**

Hooked up to a command parser to dispatch to the mediator and each subsystem, the console takes quite a few different commands. Since this is a game jam instead of for the engine stuff, we added the cheat codes: ‘tgm’ (toggle god mode) and ‘hover’.

The console opens as a new window with shared context to the games main window and applies text glyphs as a texture for rendering. The window is translucent so you can still see the main window behind it and pauses all game updates while open.

- Used for console.

# **Technical features (extra)**

In case if something from above is not qualified

### **Instanced Transform Component**

Instanced Transform Component holds a vector of Transform Components, data in which is used to create a buffer of transform matrices. Afterwards this data is sent to GPU and instantiated. Some of Transform Components are paired with physics components (like trees and rocks).

- Used for trees, rocks, grass, flowers.

### **Off Screen Effects (Blur, Overlay, etc.)**

Off Screen Effects are used to modify UV coordinates and pixel colour in fragment shader. All of them can be assigned in .ini file for any second pass object. For this game only 3 of them are used: Full Screen Effect, Blur Effect and Overlay Effect.

- Used for portals.

### **Animation Based Kinematic Character Controller**

The animation system is integrated with the physics system such that a bullet kinematic character controller is ran based on what animation is active on the entity. For in place animations, the animation system dispatches dynamic motion to the physics component. For non in place animations the physics system uses Inverse kinematics based on the current animation to update itself.

- Used for animation.