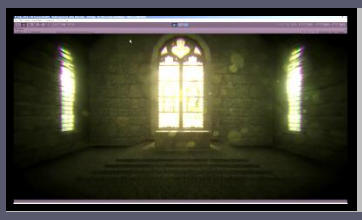
PETER MO
Game Art and Real-Time FX Reel 2018 Breakdown Sheet
(408) 218-4419 * pmo321@gmail.com * www.petermo3d.com



Rapper Run Game Level #1 – *Unity3D, Maya, Photoshop*
I set up shaders, UVs, set up lights, baked lighting, animated camera movement, and added post-processing effects to this published iOS game from Fantasy Hip Hop, Inc. This level had an old, abandoned church, a graveyard, and a secret underground dungeon. This was one of ten levels I worked on with just one other artist.

Rapper Run Game Level #2 – *Unity3D, Maya, Photoshop*
The second level I chose to show was the interior of a nightclub. It gave me a lot of options on the lighting side. I became very good at light baking and setting up either custom or auto-generated lightmap UVs.

Real-Time Kitchen in 1st Person – *Unity3D, Maya, Photoshop*
This was a recent test in Unreal 4.18.2 in which I imported the kitchen scene I once rendered in V-Ray for the CG Society Lighting Challenge. The model was given, but I needed to UV-unwrap everything for both stacked layouts and efficient light-mapping layouts. Lastly, I created my own C# scripts from scratch for character movement, look controls, and a working clock with adjustable time offsets.

Trouble Brewing VR – *Unreal, Maya, Photoshop*
I was invited to participate in the SIGGRAPH 2017 VR Game Jam in which I worked with a small, hand-picked team of 3D and audio students to convert the Trouble Brewing animated short I'd worked on (see my lighting reel) into a VR experience. I supervised my team, worked on optimization for all textures and meshes, edited a custom hand rig the player controls, and edited some Unreal Blueprints.

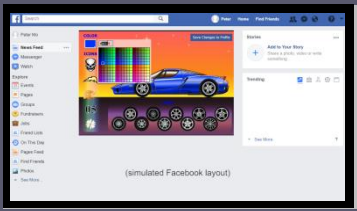
Trouble Brewing VR (cont.)
The final product was put together, complete with music and sound in just 4 days with students who were all Unreal novices. We had to convert all assets into a VR experience that ran at a silky-smooth constant 90 fps on the Oculus Rift. Many guests came to our SIGGRAPH booth and were blown away by what we made!

Unreal Engine Skin Shader – *Unreal, Maya, Photoshop*
I created an Unreal material network and made it into a Material Instance with custom controls for editing all textures, such as intensity and saturation level of Diffuse, tiling amount for the Detail Normal Map, SSS Intensity, and much more. I then took an IBL latlong map from smartibl.com and used it for both the SkyLight cubemap and the texture on the EditorSkySphere. Post-processing effects include Color Grading, Bloom, DOF, and Ambient Cubemap.

Secret Agent - *Maya, Photoshop, Unity3D, XCode*
I worked with one other person, a programmer, to create this side-scrolling 2.5D platform game. I was the only artist, so I created *all* the graphics. The sprites used in the game were either pre-rendered in Maya using Mental Ray, or put together from stock images in Photoshop. I created efficient sprite sheets to help increase game performance, back when the first iPhone had just been out a while.

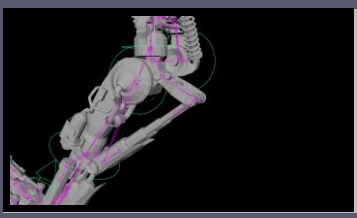
Secret Agent (cont.)
In addition to doing all the art, I also dabbled on the programming side with Objective-C in XCode. I was a novice coder, but eventually learned to help with editing gameplay variable throughout the code to balance the game. Stock and custom models were used for the in-game characters. I created the boss, a guy in a shiny, sci-fi armor, from scratch. For the laser gun, I developed a vaporization effect using a 2D particle container with an image mapped to it.

Beat Tap - *Maya, Mental Ray, Photoshop, Motion Builder*
This rhythm/dancing game was, sadly, never published, but it gave me good experience with game UI/UX design, developing custom rigs which accepted both mo-cap input and user animation. I also got a taste of cleaning mo-cap data and calibrating rigs in Motion Builder. The models were provided, I rigged, UV-unwrapped, textured, lit, rendered, and image edited.

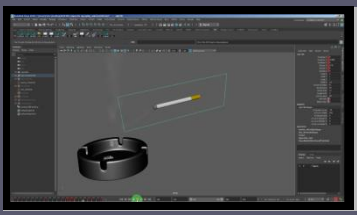


Speed Racing –*Flash, Illustrator, Photoshop, ActionScript 3*
This game was published on Facebook in 2007 and became a Top 5 app under Most Daily Active Users. I created around two dozen cars, several car rims designs, and decals to allow players to customize their rides in Flash and Illustrator. Photoshop was used for some of the shading effects on the cars and I worked in ActionScript 3 to develop the color picker and customization selections. The SWF was composited onto my Facebook page for presentation purposes.

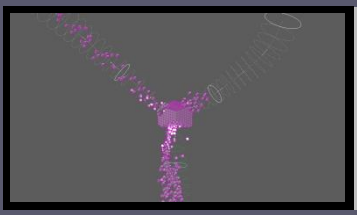
I See Ewe - *Maya, Mental Ray, Photoshop, Flash*
I created all the art content for a publish iOS children's memory game, I See Ewe (the logo was a sheep). The images shown represent stock 3D models that I rendered in a toon-like style. I gained experience in working with vector artwork in Flash.



Mecha Cat Rig – *Maya*
Model by Nico Strobbe. I used helical curves as wire deformers for best spring deformation. Set-Driven Keys and non-flipping Aim Constraints were used for telescoping hydraulics. Set-Driven Keys were also used for claw controls for Spread, Curl, and Side-to-Side. This is an efficient setup using mostly parenting and direct connections, with no binding and few constraints. The FK controls have limits set to establish realistic extents of movement.



Cigarette Rig - *Maya*
Two nParticle emitters are constrained to the tip of the cigarette; one for smoke, one for ash. Custom attributes were added to the controller to affect particle behavior: Rate controls emission rate and Duration controls lifespan for both emitters. Smoke Rise controls the magnitude of an upward Air field. Smoke Drift controls the magnitude of a Turbulence and a Vortex field. Ember Glow controls the Incandescence ramp color on the cigarette tip. The sim in the video was cached.



Particle Effect for Rebus Farm - *Maya*
This nParticle system was set up from a promotional animation made for Rebus Farm. A thousand particles filled a cube and instanced as smaller cubes. Volume Axis fields were added to vacuum away the cubes. Cached and played in reverse to make the big cube appear to be coalescing. The final rendered result can be seen in my lighting reel.