CS425 GAME PROGRAMMING

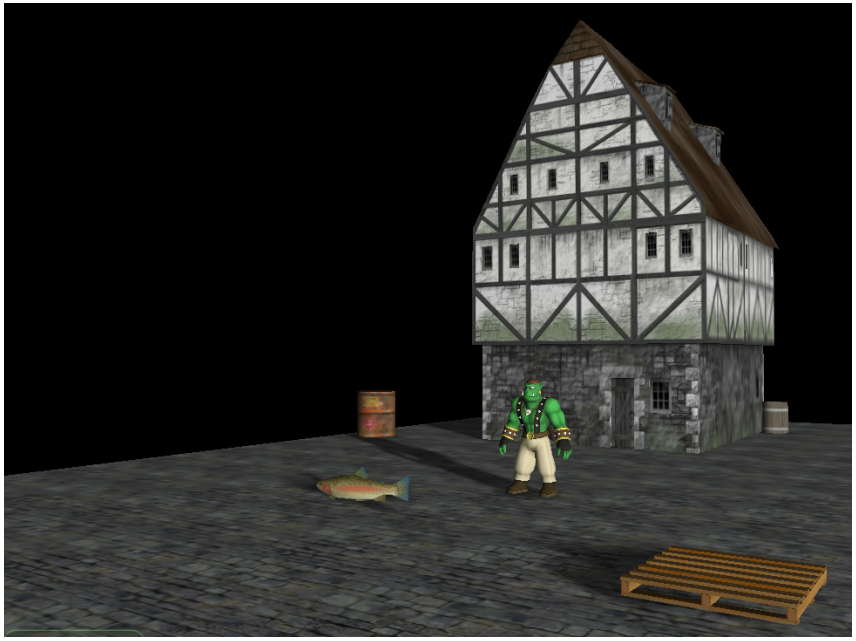
Level Loading and Management

Announcements

- Assign #01: Is already out on github, but details are still coming.
  - Basically, your goal is to create an animation or a character walking and stopping using OGRE and Unity default resources.

Today

- Review OGRE tutorials
- Level loading (Chapters 14 and 15)



Anatomy of a Game World

- World Chunks
- High-level Flow

- Static Background
- Dynamic Foreground

Skyrim

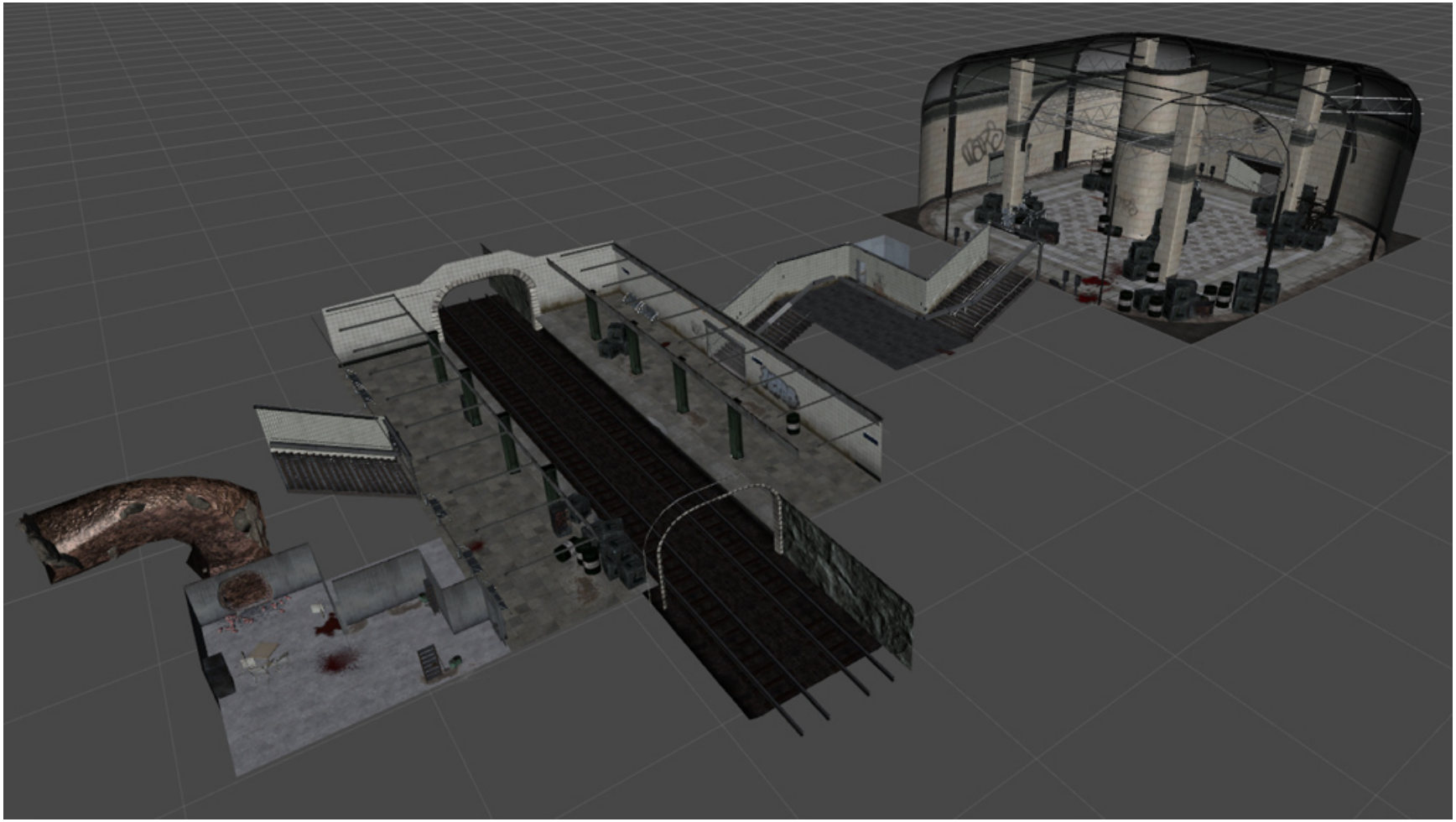
World of Warcraft

Level Loading

- Why bother?
- What are we loading?
- What info do we need about what we're loading?

- Remember what we are going over is generally important. The specifics relative to OGRE are not as important.

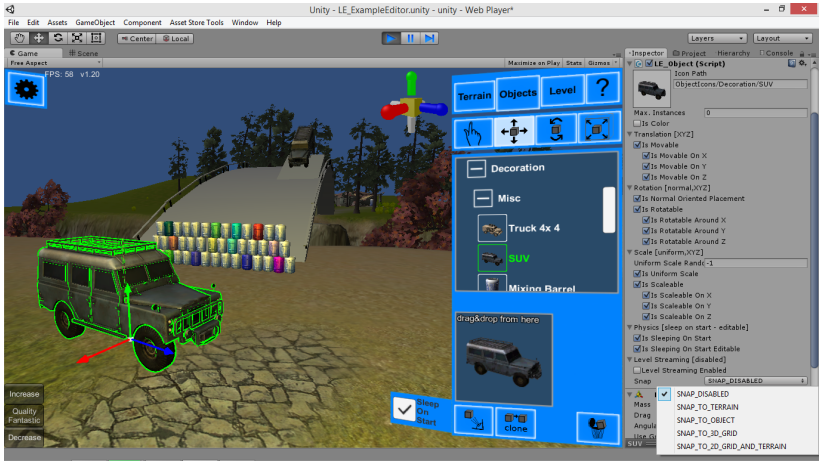
Level in Unity Editor. Good flow?

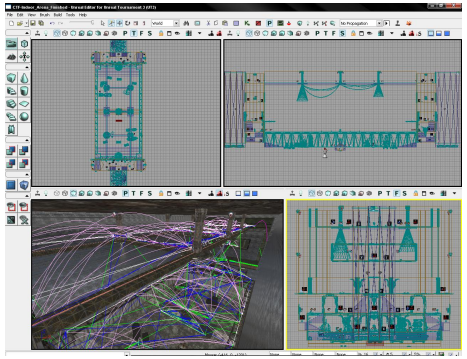


Even Scratch has it

Editor Features

- Chunk creation and management
- Visualization
- Navigation
- Selection
- Layers
- Property grid (attributes)
- Object placement and alignment aids
- Special objects: lights, particle emitters, sounds, regions, splines, nav meshes, custom data, video



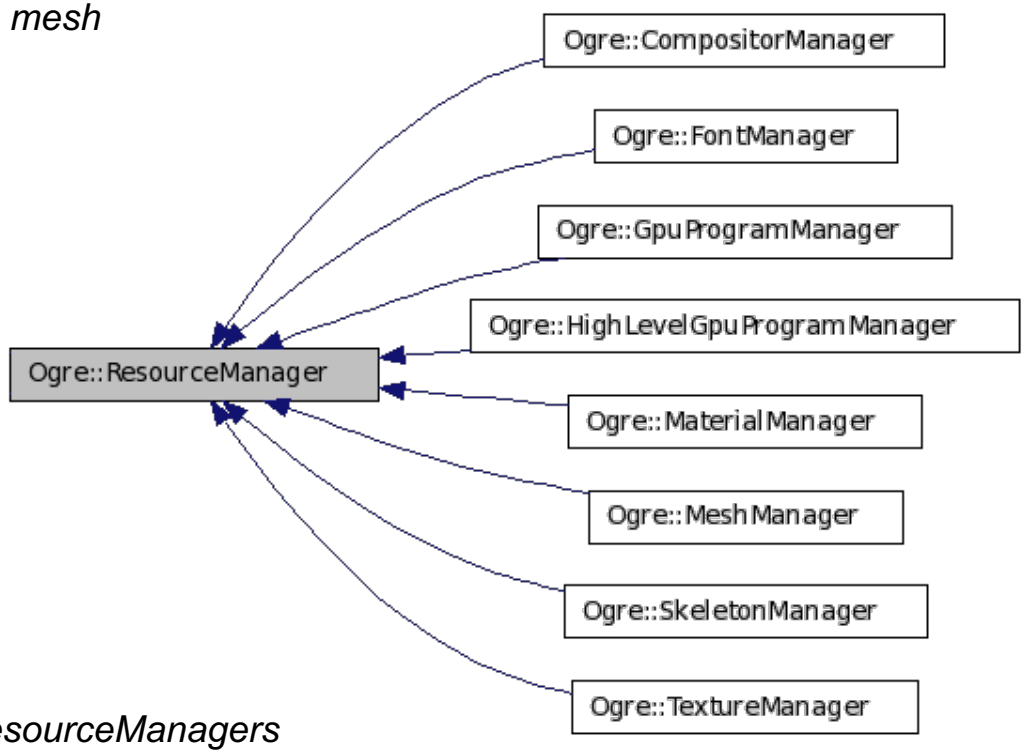


Asset management is no joke

- Loading and Unloading and Reloading
  - From where and to where?
  - Dependencies of assets (some assets require other assets to work together)
- Synchronous vs. Asynchronous Asset Manager
  - Loading everything upfront or when needed?
- Most modern game engines mange assets automatically
  - Unreal (**Asset Manager, Asset Tree**), Unity (**AssetBundle)**, etc.
  - Asset manager usually exists in both Editor and in the game
- How do you manage assets in OGRE?

OGRE Resource Management

- Ogre needs to know where to search for "resource" files
  - Via configuration files (e.g., resources.cfg)
  - OGRE creates its resource manager in Root::Root
- When you say:
  - *Entity *e = mSceneMgr→createEntity("MyEntity", [MeshFilename]);*
  - *OGRE Resource manager finds the mesh*



See details in
http://wiki.ogre3d.org/Resources+and+ResourceManagers

Resources.cfg Example

```
# Resources required by the sample browser and most samples.
[Essential]
Zip=app/native/Media/packs/SdkTrays.zip
Zip=app/native/Media/packs/profiler.zip
FileSystem=app/native/Media/thumbnails

# Common sample resources needed by many of the samples.
# Rarely used resources should be separately loaded by the # samples which require them.
[Popular]
FileSystem=app/native/Media/fonts
FileSystem=app/native/Media/materials/programs
FileSystem=app/native/Media/materials/scripts
FileSystem=app/native/Media/models
FileSystem=app/native/Media/particle
…
Zip=app/native/Media/packs/cubemapsJS.zip
Zip=app/native/Media/packs/fresneldemo.zip
Zip=app/native/Media/packs/ogredance.zip
…
[General]
FileSystem=app/native/Media # Materials for visual tests

[Tests]
FileSystem=app/native/Media/../../Tests/Media
```