

ETGG3802

Lab1: Physics "picking"

Points: 20

Assigned: 1/5/2010

Due: 1/12/2010¹

Objective:

Review the engine classes from last semester. Also add the ability to "pick" an object by casting a ray into the scene (here, from the camera into the world)

Tasks:

1. Determine your tools.
 - a. In the past, we've used ODE (create a dRay type dGeomID) – problem: only works with physics-enabled objects.
 - b. MOC (Minimal Ogre Collisions) seems to work well
 - i. <http://www.ogre3d.org/forums/viewtopic.php?t=45267>
 - ii. Uses the AABB to do a rough test and then checks triangle-by-triangle for a refined test.
 - iii. Not a huge performance hit.
 - iv. Will work for physics and non-physics objects. We may need to do some tweaking to get it to work with bones (later)
 - c. You're welcome to look for other tools too!
2. Give the user the ability to click in the scene and find the EngineActor (which could also be a class derived from EngineActor, such as EngineActorPhys was hit by the ray).
 - a. To me, the logical place to put this functionality is in EngineLevel – we're searching its actor list.
 - b. The function should return a EngineActor pointer.
 - c. Using an overlay (probably just one field in the debug overlay), show the picked object (if any).
 - d. Note: MOC has the ability to create the ray for you, but here, I want you to create the ray yourself.
 - i. You may want to do this in FrameStarted.
 - ii. hint: There's a method of the camera class that will generate a ray for you.
3. Efficiency:
 - a. Your addition shouldn't drop the fps too much.
 - b. In debug mode, I was running at 32 fps without the hit-check.
 - c. I dropped to 28 fps with it (with an animated ninja, and two physics objects).
 - d. Minimize the changes to other areas of the engine – you'll have some, but think carefully about them...

¹ Remember: this semester I'm trying an experiment: **NO LATE WORK** accepted, unless otherwise specified.