CARBON

Kynapse in Eve Online

David Press

davidp@ccpgames.com

Who is CCP?

- 600 person company.
- Working on 3 AAA games.
- Eve Online – 370k subscribers, 65k PCU
- Dust 514 – Upcoming FPS integrated with Eve.
- World of Darkness – Upcoming MMO.

What is Carbon?

- Shared technology platform.
- Parts used in all 3 games.
- Kynapse used in Eve Online and World of Darkness

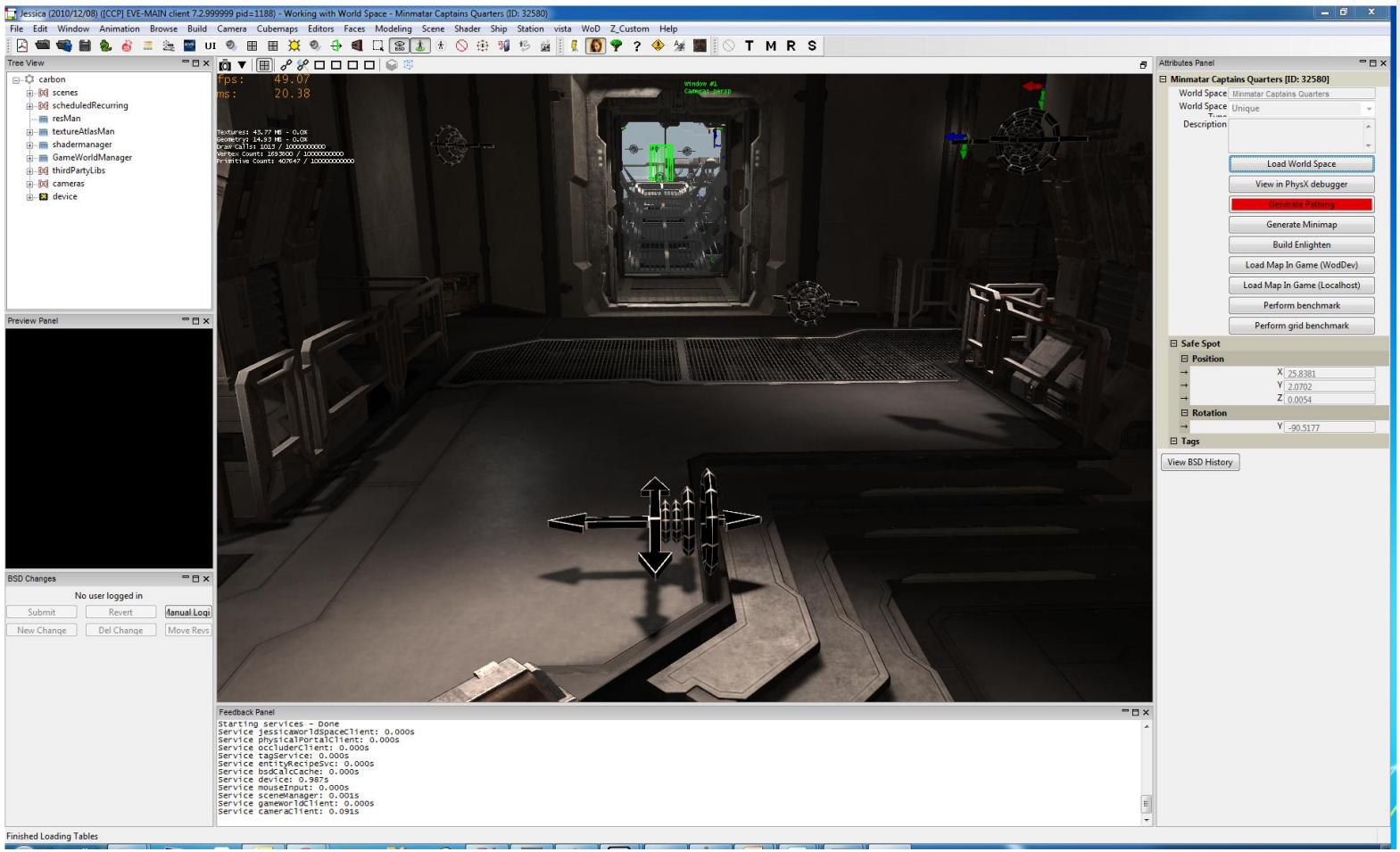
How is Kynapse used in Eve?

- Incarna expansion
- Captain's Quarters
- Designers wanted double-click movement around Captain's Quarters, because that's how Eve in-space controls work.

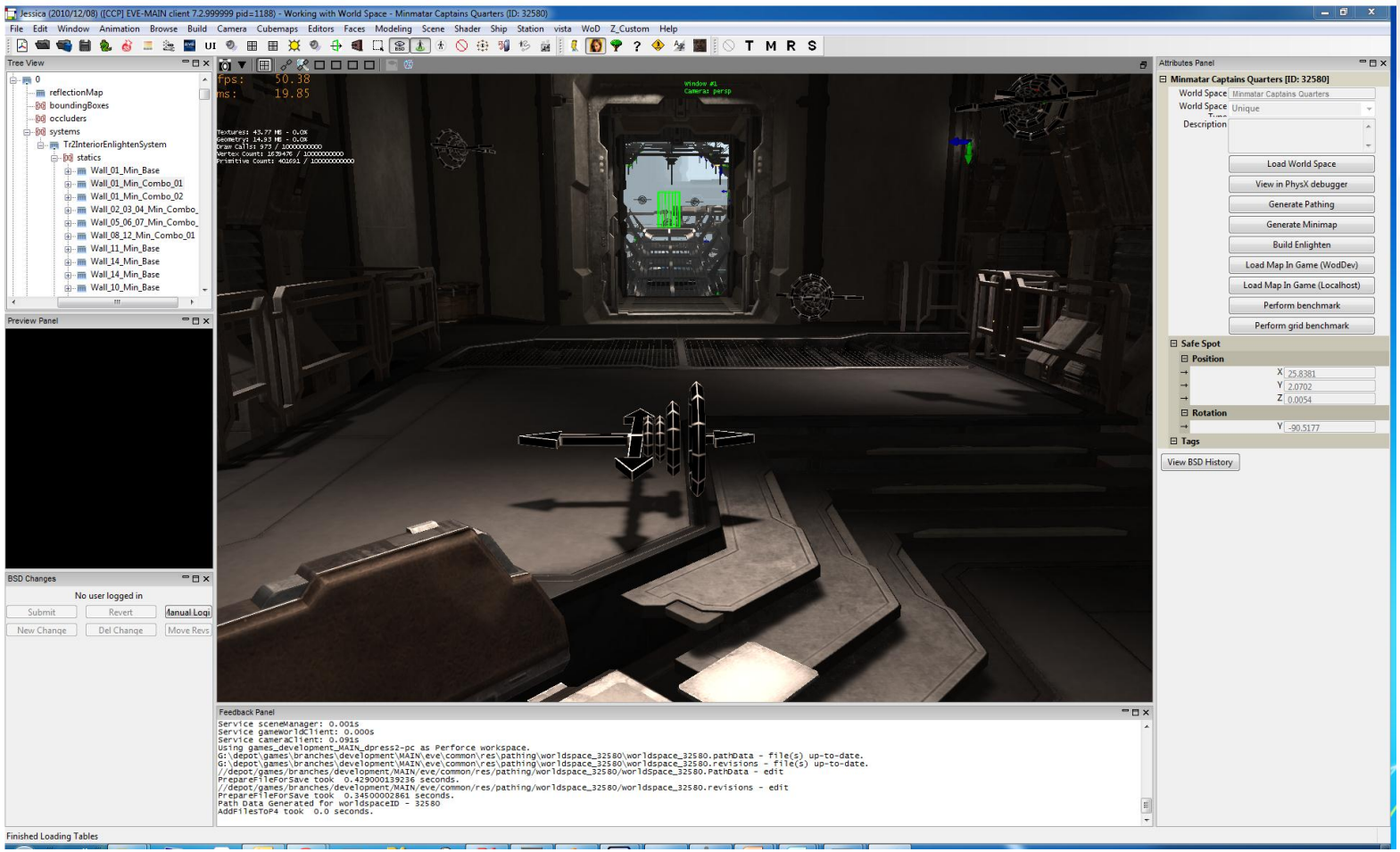
How is Kynapse used in Eve?

Movie

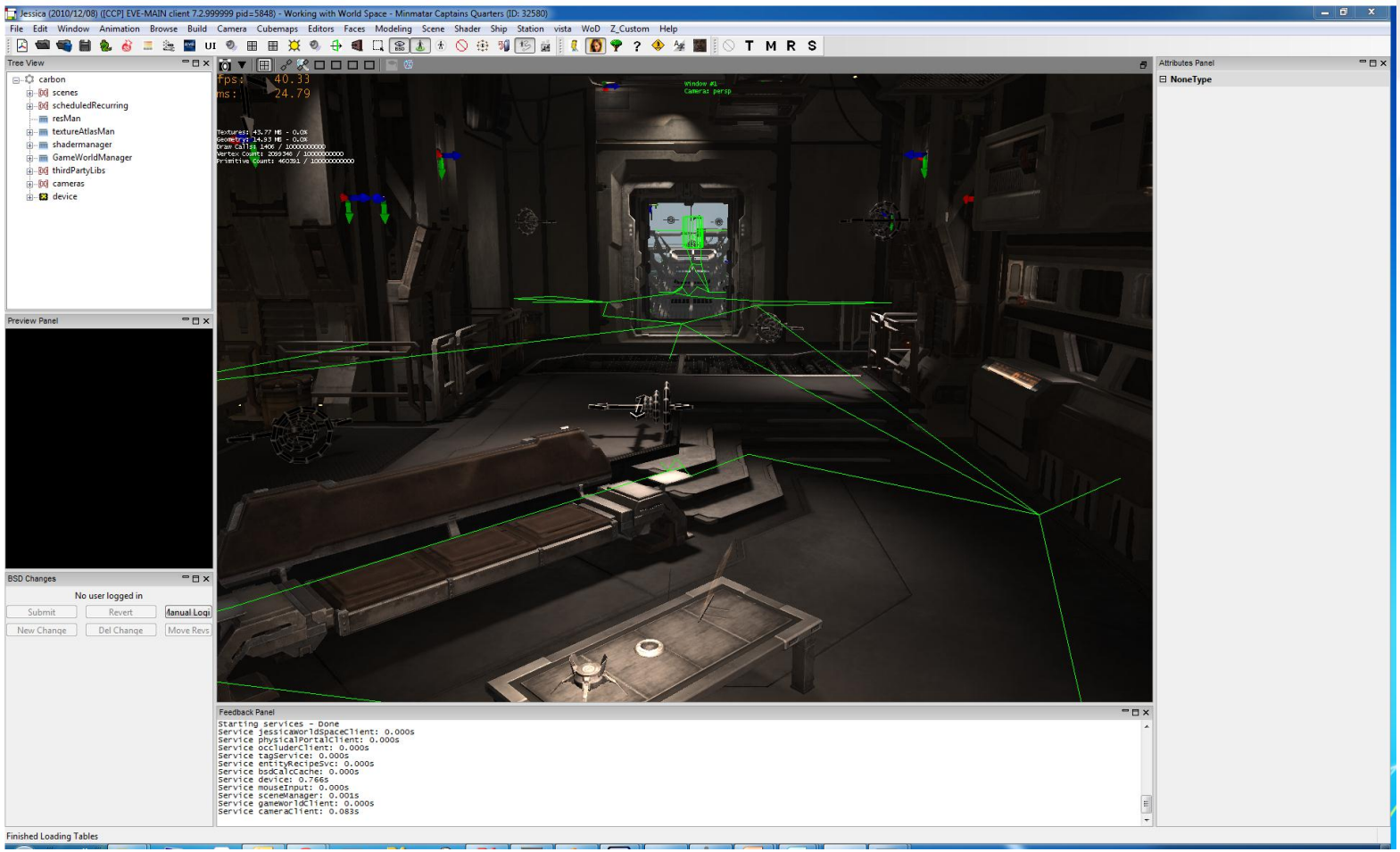
Kynapse for the Level Designer



Kynapse for the Level Designer



Kynapse for the Level Designer



Kynapse for the Level Designer

- That's it
- If your collision works, your pathing works.
- High quality auto-generation of pathing data with Kynapse is, IMO, its "secret sauce".
- Super fast
  - < 1sec for Captain's Quarter.
- Just works.

Kynapse Pathbuilding Integration

- Kynapse 7 comes with a standalone pathbuilding tool.
- We wanted pathbuilding integrated with our level design tool.

Kynapse Pathbuilding Integration

CARBON

```
KyResult KynapsePhysXInputProducer::Produce( Kaim::ClientInputConsumer &inputConsumer )
{
   int test = 0;
   const GWStaticShapeVector &statics = m_gameWorld->GetStaticShapes();
   size_t numStatics = statics.GetSize();
   for( size_t i = 0;i<numStatics;i++ )
   {
      TrackableStdList<NxDebugLine> lines( "KynapseInputProducer line vector" );
      statics[i]->GetPhysXDebugLines( lines );
      GenerateTrianglesForConsumer( inputConsumer, lines );

   }
   return Kaim::Result::Success;
}

void KynapsePhysXInputProducer::GenerateTrianglesForConsumer( Kaim::ClientInputConsumer &inputConsumer, const TrackableStdList<NxDebugLine> &lines )
{
   TrackableStdList<NxDebugLine>::const_iterator lines_it = lines.begin();
   while( lines_it != lines.end() )
   {
      const NxDebugLine *line = &( *lines_it );
      Kaim::Vec3f a( line->p0.x, line->p0.y, line->p0.z );

      lines_it++;
      line = &( *lines_it );
      Kaim::Vec3f b( line->p0.x, line->p0.y, line->p0.z );

      lines_it++;
      line = &( *lines_it );
      Kaim::Vec3f c( line->p0.x, line->p0.y, line->p0.z );

      lines_it++;

      KyResult res = inputConsumer.ConsumeTriangleFromPos( a,b,c, 1 );
      if( res != Kaim::Result::Success )
      {
         CCP_LOGERR_CH(s_kynapseChannel, "KynapsePhysXInputProducer::GenerateTrianglesForConsumer-Something went wrong." );
      }
   }
}
```

Kynapse Pathbuilding Integration

CARBON

```
void PhysXCollisionBridge::RayCast( const RayCastQuery& query, /*[out]*/RayCastResult& result )
{
   if ( m_gameWorld == NULL )
      return;

   Vector3 start = KynapseBase::ConvertKynapseToVector3( query.start );
   Vector3 end = KynapseBase::ConvertKynapseToVector3( query.end );
   Vector3 hitLoc, hitNormal;
   GWObject *closest = m_gameWorld->LineTest( start, end, COLLIDABLE_ALL, hitLoc, hitNormal );
   if( closest )
   {
      result.hasHit = true;
      result.hitNormal = KynapseBase::ConvertVector3ToKynapse( hitNormal );
      result.hitPosition = KynapseBase::ConvertVector3ToKynapse( hitLoc );
   }
}

bool PhysXCollisionBridge::HasRayCastHit( const RayCastQuery& query )
{
   RayCastResult result;
   RayCast( query, result );
   return result.hasHit;
}

void PhysXCollisionBridge::SphereCast( const SphereCastQuery& query, /*[out]*/SphereCastResult& result )
{
   float radius = query.radius;
   Kaim::Vec3f kynCenter = Kaim::GetCoordSystem().KynapseToClient_Pos( query.start );
   Vector3 center;
   center.set( kynCenter.x, kynCenter.y, kynCenter.z );
   bool hit = m_gameWorld->SphereTest( center, radius,COLLIDABLE_ALL );
   if( hit )
   {
      result.hasHit = true;
      RayCast( query, result );
   }
}

bool PhysXCollisionBridge::HasSphereCastHit( const SphereCastQuery& query )
{
   SphereCastResult result;
   SphereCast( query, result );
   return result.hasHit;

}
```

World of Darkness

- Big, open world.
  - 500mx500m portion of dense city takes under 4 minutes to generate pathdata for on a Core 2 Duo, including rooftops.
- Pathing runs on server for NPCs and on client for PCs.
  - Don’t want to burden server with PC pathing.
- Follow behaviors, fleeing behaviors, patrolling behaviors, wander behaviors.

Kynapse 2012

- If you haven't used Kynapse in a while, give 2012 a try.
- It's basically a whole new product.
- Same section of world that takes 4 minutes to build now used to take 2 hours.
- Memory use way down.
- Integration is much easier and less intrusive
- Better support for multithreading and MMOs.

Questions?

CARBON

We’re Hiring!

http://ccpgames.com/jobs