PhysX Tutorial 01

Denison Linus
dlmtavares@inf.ufrgs.br
http://www.inf.ufrgs.br/~dlmtavares

Game Physics (Real-Time Physics Based Animation)

1. **Project Configuration**
   - Install PhysX System Software
   - Additional Include Directories
   - Additional Library Directories
   - Additional Dependencies
   - DLLs

2. **Physics Manager**
   - Physics Simulation, Collision Detection and Collision Response
   - CPhysXManager

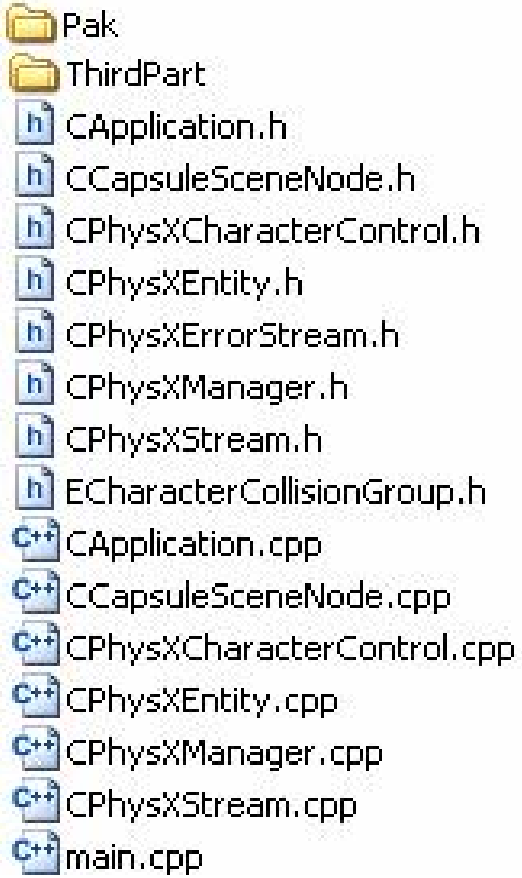
3. **Rigid-Body**
   - Dynamic Actors (implicit solid shape – box, sphere, capsule)
   - Static Actors (explicit solid shape – triangle mesh)
   - CPhysXEntity

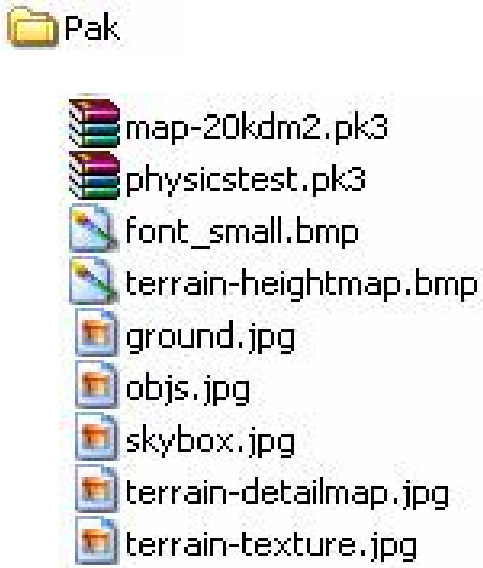
4. **Character Controller**
   - Character control
   - Character interactions
   - Jumps
   - Auto-Stepping
   - Walkable Parts
   - CPhysXCharacterControl

5. **Constraints (Joints, Breakable joints)**
   - Simple Joints
   - Simple Ragdoll

1 - Project Configuration

- **Install PhysX System Software:**
  - http://www.inf.ufrgs.br/~dlmtavares/graduate/INF01019/PhysX_SystemSoftware.zip
- **Source Code:**
  - http://www.inf.ufrgs.br/~dlmtavares/graduate/INF01019/Tutorial_PhysX.zip





1 - Project Configuration

- **Additional Include Directories:**
  - ThirdPart\irrlicht-1.3\source\Irrlicht
  - ThirdPart\AGEIA\v2.7.2\SDKs\Physics\include
  - ThirdPart\AGEIA\v2.7.2\SDKs\PhysXLoader\include
  - ThirdPart\AGEIA\v2.7.2\SDKs\Foundation\include
  - ThirdPart\AGEIA\v2.7.2\SDKs\NxCharacter\include
  - ThirdPart\AGEIA\v2.7.2\SDKs\Cooking\include
  - ThirdPart\irrlicht-1.3\include
- **Additional Library Directories**
  - ThirdPart\Ageia\v2.7.2\SDKs\lib\win32
  - ThirdPart\irrlicht-1.3\lib\Win32-visualstudio
- **Additional Dependencies**
  - Irrlicht.lib NxCooking.lib PhysXLoader.lib NxCharacter.lib
- **DLLs**
  - ThirdPart\irrlicht-1.3\bin\Win32-VisualStudio\Irrlicht.dll
  - ThirdPart\AGEIA\v2.7.2\Bin\win32\NxCharacter.dll,NxCooking.dll,PhysXLoader.dll
- **Docs**
  - ThirdPart\irrlicht-1.3\doc
  - ThirdPart\AGEIA\v2.7.2\SDKs\Docs
- **Character Set**
  - Not Set
- **Warning Level**
  - Level 1 (/W1)

2 - Physics Manager

- **CPhysXManager**

```
bool initialize();
void finalize();
void update();
CPhysXEntity* createRigidBox(...);
CPhysXEntity* createRigidSphere(...);
CPhysXEntity* createRigidCapsule(...);
CPhysXEntity* createRigidMesh(...);
CPhysXEntity* createRigidTerrain(...);
bool createBodySphericalJoint(...);
bool createRevoluteJoint(...);
CPhysXCharacterControl* createCharacterControl(...);
...
```

2 - Physics Manager

- **CPhysXEntity**

```
void update();
core::vector3df applyForce(…);
void setDynamic(…);
bool isDynamic();
scene::ISceneNode* getSceneNode() const;
NxActor* getActor() const;
void setCharacterCollisionGroup(…);
E_CHARACTER_COLLISION_GROUP getCharacterCollisionGroup();
…
```

2 - Physics Manager

- **CPhysXCharacterControl**

```
void update(...);
core::vector3df getCharacterPosition();
bool resetCharacterPosition();
scene::ISceneNode* getSceneNode() const;
NxController* getController() const;
void jump(...);
void move(...);
void setCharacterSpeed(...);
f32 getCharacterSpeed();
void setAutoStepping(...);
f32 getAutoStepping();
...
```

3 - Rigid-Body

CApplication.h

```
#include "CPhysXManager.h"

private:
    CPhysXManager* PhysicsManager;

CApplication(...) : ... PhysicsManager(0)
{
    PhysicsManager = new CPhysXManager(FileSystem,Logger);

~CApplication()
{
    PhysicsManager->drop();

void CApplication::run()
{
      while(Device->run() && Driver)
      if(Device->isWindowActive())
      {
          PhysicsManager->update();
```

3 - Rigid-Body

- **Add Static Rigid Box Surface:**

```
ISceneNode* ground = Smgr->addCubeSceneNode(1);
ground->setScale(vector3df(2000, 1, 2000));
ground->setPosition(vector3df(0, -10, 0));
ground->setRotation(vector3df(0, 0, 0));
ground->setMaterialTexture(0, Driver->getTexture("Pak/ground.jpg") );
ground->setMaterialFlag(video::EMF_LIGHTING, false);

CPhysXEntity* Entity = PhysicsManager->createRigidBox
                        (
                            CPhysXManager::generateUniqueName(),
                            ground,
                            vector3df(2000, 1, 2000),
                            false
                        );
```

3 - Rigid-Body

- **Add Dynamic Rigid Capsule Surface:**

```
scene::CCapsuleSceneNode* capsuleNode = new scene::CCapsuleSceneNode
(
    20,  3,
    Smgr->getRootSceneNode(),
    Smgr
);
capsuleNode->setScale(vector3df(1, 1, 1));
capsuleNode->setPosition(vector3df(20, 80, 0));
capsuleNode->setRotation(vector3df(0, 0, 15));
capsuleNode->setMaterialTexture(0, Driver->getTexture("Pak/objs.jpg"));
capsuleNode->setMaterialFlag(video::EMF_LIGHTING, false);
CPhysXEntity* Entity = PhysicsManager->createRigidCapsule
                        (
                            CPhysXManager::generateUniqueName(),
                            capsuleNode,
                            20, 3,
                            true
                        );
```

3 - Rigid-Body

- **Simple Scene**
  - **Edit loadSceneSimple()**
    - Add 1 ICameraSceneNode
    - Add 1 RigidBox (Ground)
    - Add 1 RigidBox
    - Add 1 RigidSphere
    - Add 1 RigidCapsule
  - **Edit shootBall()**
    - Add RigidSphere at Camera Position
    - Calcule ForceDirection and ForceStrength
    - Apply Force at RigidSphere
  - **Edit shootBox() and shootCapsule()**
    - Like shootBall
  - **Edit OnEvent(SEvent event)**
    - Edit event receiver to add shooting control

PhysX by ageia IRRLICHT LIGHTNING FAST REALTIME 3D ENGINE

3 - Rigid-Body