



Useful Tools for Making Video Games

Part II

An overview of **IRRLICHT**



Sample Games

- Slam Soccer 2006
- H-Craft Championship
- Di Jabolo
- Gekkeiju Online
- Inevitable Online



Features

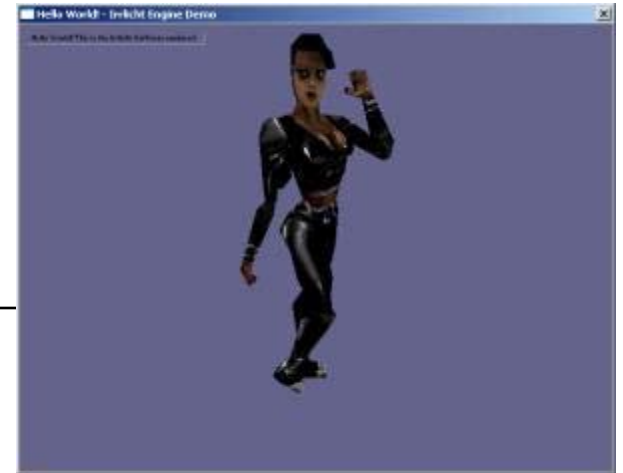
- Platform & Graphics API independent
- Direct import of common mesh and texture formats
- Shader support
- Character animation
- Particle effects, billboards, light maps, shadows, environment mapping
- Easy physics integration
- Open source!



Startup Sequence

```
Main() {  
    // startup engine ...  
    // load scene & models ...  
    // add camera, turn on lights ...  
  
    // draw everything  
    while(device->run() && driver) {  
        driver->beginScene(true, true,  
            video::SColor(255,0,0,255));  
        scenemgr->drawAll();  
        driver->endScene();  
    }  
  
    return 0;  
}
```

Loading models



- Some of supported formats

.3ds, .obj, .x, .ms3d, .mesh, .bsp, .md2 ...

- IAnimatedMesh* mesh =
 smgr->getMesh("../..../media/sydney.md2");
IAnimatedMeshSceneNode* node =
 smgr->addAnimatedMeshSceneNode(mesh);
if (node)
{
 node->setMaterialFlag(EMF_LIGHTING, false);
 node->setFrameLoop(0, 310);
 node->setMaterialTexture(0,
 driver->getTexture("../..../media/sydney.bmp"));
}



Adding camera

- addCameraSceneNode()
- addCameraSceneNodeMaya()
- addCameraSceneNodeFPS(
 - ISceneNode * *parent* = 0,
 - f32 *rotateSpeed* = 100.0f,
 - f32 *moveSpeed* = 500.0f,
 - s32 *id* = -1,
 - SKeyMap * *keyMapArray* = 0,
 - s32 *keyMapSize* = 0,
 - bool *noVerticalMovement* = false,
 - f32 *jumpSpeed* = 0.f)

Loading Quake3 maps

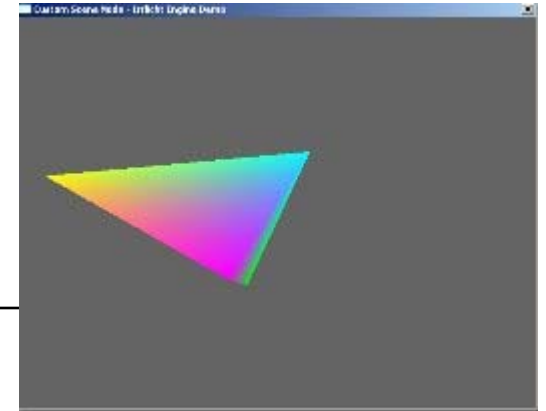


```
scene::IAnimatedMesh* mesh =  
    smgr->getMesh("20kdm2.bsp");  
scene::ISceneNode* node = 0;
```

```
if (mesh) {  
    node = smgr->addOctTreeSceneNode(mesh->getMesh(0));  
}
```

```
if (node) {  
    node->setPosition(core::vector3df(-1300,-144,-1249));  
}
```

Custom SceneNode's



- `class CSampleSceneNode : public scene::ISceneNode { ... };`
- `core::aabbox3d<f32> Box;`
`video::S3DVertex Vertices[4];`
`video::SMaterial Material;`
- `irr::video::S3DVertex::S3DVertex (`
`const core::vector3df & pos,`
`const core::vector3df & normal,`
`SColor color,`
`const core::vector2d< f32 > & tcoords);`
- `u16 indices[] = { 0,2,3, 2,1,3, 1,0,3, 2,0,1 };`
`driver->drawIndexedTriangleList(Vertices, 4, indices, 4);`

User Input

```
class MyEventReceiver : public IEventReceiver {
public:
    virtual bool OnEvent(SEvent event) {
// event types: GUIEvent, KeyInput, LogEvent, MouseInput, UserEvent
        if (event.EventType == irr::EET_KEY_INPUT_EVENT) {
            switch(event.KeyInput.Key) {
                case KEY_KEY_W:
                    ...
                    break;
            }
        }
    };
    ...
    MyEventReceiver myEventReceiver;
    device = createDevice(..., &myEventReceiver);
    ...
}
```

Character Animation



```
scene::IAnimatedMeshSceneNode* animMS =
    smgr->addAnimatedMeshSceneNode(smgr->getMesh("../sydney.md2"));
scene::ISceneNodeAnimator* anim =
    smgr->createFlyStraightAnimator(core::vector3df(...),
                                   core::vector3df(...), 10000, true);

animMS->addAnimator(anim);
anim->drop();
animMS->setMaterialFlag(video::EMF_LIGHTING, false);
animMS->setFrameLoop(320, 360); anim->setAnimationSpeed(30);
animMS->setRotation(core::vector3df(...));
animMS->setMaterialTexture(0, driver->getTexture("../media/sydney.bmp"));
```


2D Graphics

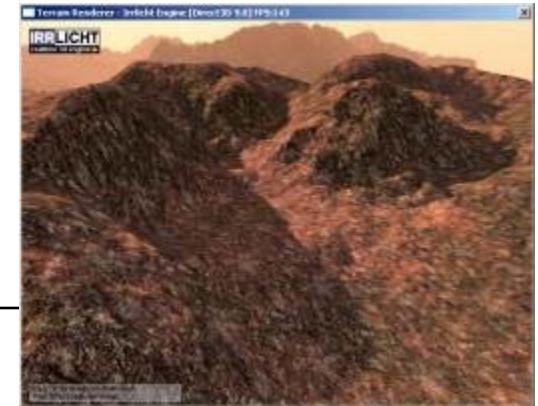
```
video::ITexture* images = driver->getTexture(...);
driver->makeColorKeyTexture(images, core::position2d<s32>(0,0));
core::rect<s32> frames[NUM_FRAMES]; // fill in rect coordinates
...
while(device->run() && driver) {
    static int currentFrame = 0;
    u32 time = device->getTimer()->getTime();
    if (time/33 % 2) {
        currentFrame = (++currentFrame)%NUM_FRAMES;
    }
    driver->draw2DImage(images, core::position2d<s32>(...),
                       frames[currentFrame], 0,
                       video::SColor(255,255,255,255), true);
}
```

Particle Systems



```
// fire particle system
scene::IParticleSystemSceneNode* ps = 0;
ps = smgr->addParticleSystemSceneNode(false);
ps->setPosition(...);
scene::IParticleEmitter* em = ps->createBoxEmitter( ...);
ps->setEmitter(em);
scene::IParticleAffector* paf = ps-
    >createFadeOutParticleAffector(...);
ps->addAffector(paf);
ps->setMaterialTexture(...);
```

Terrain Rendering



```
scene::ITerrainSceneNode* terrain =  
    smgr->addTerrainSceneNode(...);  
terrain->setScale(...);  
terrain->setMaterialFlag(...);  
terrain->setMaterialTexture(...);  
terrain->setMaterialType(...);  
terrain->scaleTexture(1.0f, 20.0f);
```



For more details...

- Go through the tutorials under the “examples” directory
- Check out the [API](#) and [forums](#)



Language bindings and add-ons

- Java/PureBasic/Python ports
- irrEdit scene editor
- irrKlang 3D audio library
- Exporters/importers etc

Go to [Add-ons site](#)



References

- <http://irrlicht.sourceforge.net/>