Game Engine Design



tum.3D

computer graphics & visualization

Assignment 8

- This week: Enemy ships!

Assignment 8

- Simple enemy model
  - Enemies spawn in fixed intervals
  - Enemies fly along straight paths with constant velocity
  - Paths are randomly generated
- Enemy Types & Enemy Instances
  - Enemy Types are defined in config file
    - Name, hitpoints, speed, …
    - Mesh + transformation
  - Enemy Instances are generated at runtime
    - Enemy Type name
    - Position + velocity
    - Remaining hitpoints, …

Enemy Types

- Example definition in config file

```
Mesh Amy ...

#EnemyType name hitpoints size speed mesh_name scale rot_x _y _z trans_x _y _z
EnemyType AmyShip 100 10 50 Amy 0.5 0 180 0 0 0 0
```

- Example internal representation (similar to CockpitObjects and GroundObjects)

```
struct EnemyType
{
    int hitpoints;
    ...
};

//Dictionary of EnemyTypes
std::map<std::string,EnemyType> enemyTypes;
```

Enemy Instances

- Example internal representation

```
struct EnemyInstance
{
    ...
    XMVECTOR pos;   //position p in world space
    XMVECTOR vel;   //velocity v in world space
    ...
};

std::list<EnemyInstance> g_EnemyInstances;
```

- Store instances in a list
  - `std::list` → doubly-linked list in C++
  - Allows for efficient removal of arbitrary elements
  - Usage very similar to `std::vector`

Example: std::list

- A list of int

```
std::list<int> l;

//insert elements at front or end of list
l.push_back(13);
l.push_front(5);

//iterate over all list elements
for (auto it=l.begin(); it!=l.end(); it++)
{
    //you have to use a reference ("int&") in order to affect "value" in list
    int  value_copy = *it;
    int& value      = *it;
    //do something with value...
}

//remove all elements of list
l.clear();
```

- Removing single elements requires caution!

Filtering an `std::list`

- Problem: iterate over list and remove certain elements
- Manual `erase()`

```
//remove all elements that are equal to 5
for (auto it=l.begin(); it!=l.end(); it++)
{
    int value = *it;
    if (value == 5) {
        l.erase(it); //WRONG! call to erase invalidates "it"
    }
}
```

WRONG!

Filtering an `std::list`

- Problem: iterate over list and remove certain elements
- Manual `erase()`

```
//remove all elements that are equal to 5
for (auto it=l.begin(); it!=l.end(); ) //no it++ in loop-header!
{
    int value = *it;
    if (value == 5) {
        auto it_remove = it;
        it++; //increment it before call to erase
        l.erase(it_remove);
    }
    else { it++; }
}
```

- Alternative: `remove_if()` method of `std::list`
  - Using a predicate function or class
    http://www.cplusplus.com/reference/stl/list/remove_if/
  - Using lambda expressions (advanced)
    http://msdn.microsoft.com/de-de/library/dd293608

Spawn Timer

- Problem: Spawn enemies every `g_SpawnInterval` seconds
  - Add this to your config
- Spawn enemies in `OnFrameMove()`
  - Called every frame before `OnD3D11FrameRender()`
  - `fElapsedTime` contains time since last frame (in seconds)
    → Use global timer variable

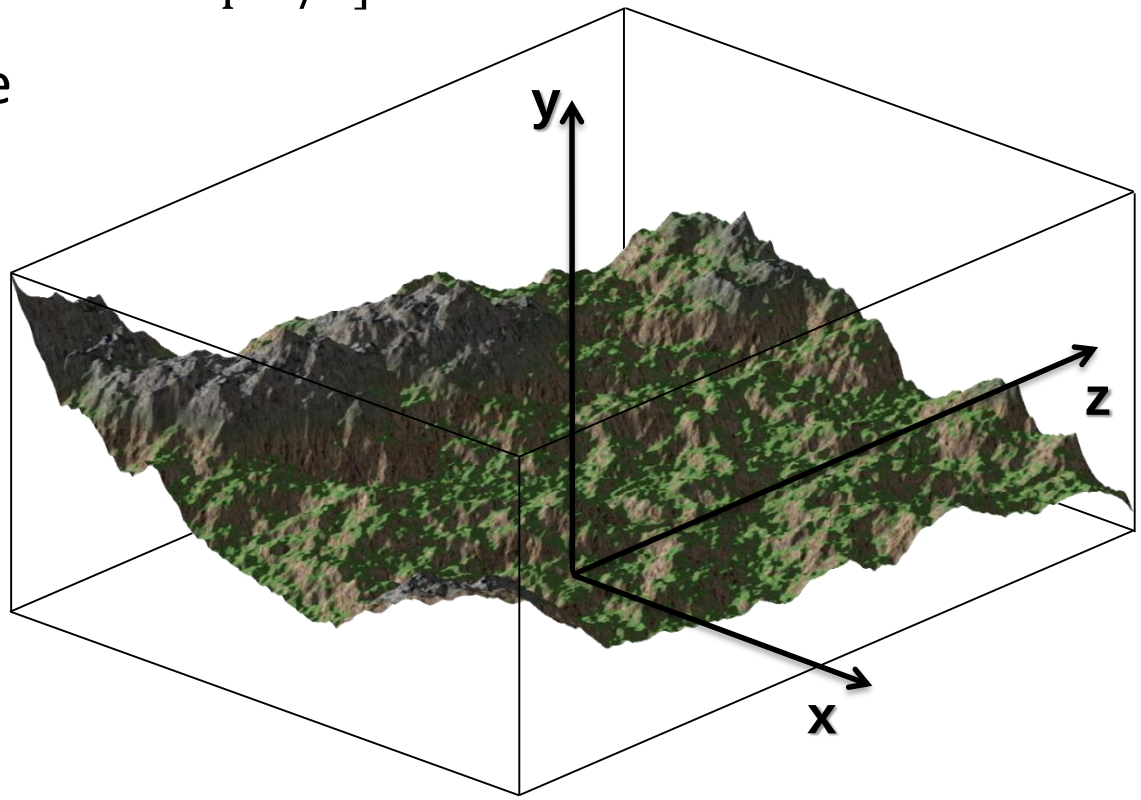
```
float g_SpawnTimer    = 0.0f;

void CALLBACK OnFrameMove( ..., float fElapsedTime, ... )
{
    ...
    g_SpawnTimer -= fElapsedTime;

    if (g_SpawnTimer<0)
    {
        g_SpawnTimer += g_configParser.GetSpawnInterval();
        //spawn new enemy here
    }
    ...
}
```

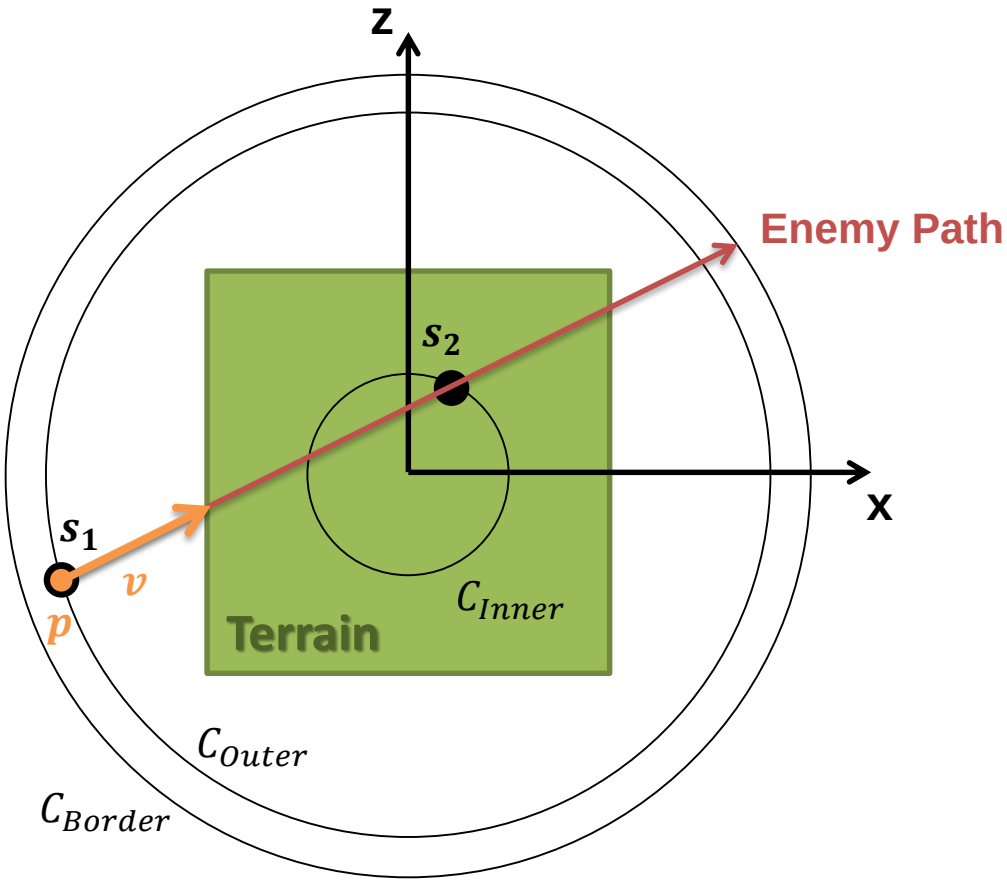
Our World Space

- Position and velocity of enemies is stored in world space
- Terrain range of values
  - $x \in$ [-TerrainWidth/2; TerrainWidth/2]
  - $y \in$ [0; TerrainHeight]
  - $z \in$ [-TerrainDepth/2; TerrainDepth/2]
- Of course, enemies are not restricted to this range of values



Spawning Enemies

- Example enemy spawn model



Top-Down View

Spawning Enemies

- Example enemy spawn model
  - Config file parameters: $h_{min}, h_{max}, speed$
  - $\boldsymbol{s_1}.xz \coloneqq random\ point\ on\ circle\ C_{Outer}$
  - $\boldsymbol{s_2}.xz \coloneqq random\ point\ on\ circle\ C_{Inner}$
  - $\boldsymbol{s_1}.y \coloneqq \boldsymbol{s_2}.y \coloneqq random\ height \in [h_{min}; h_{max}]$
  - Enemy position $\boldsymbol{p} \coloneqq \boldsymbol{s_1}$
  - Enemy velocity $\boldsymbol{v} \coloneqq speed \cdot normalize(\boldsymbol{s_2} - \boldsymbol{s_1})$
  - Remove enemy if outside C_{Border}
- Hint: Random point $\boldsymbol{x}$ on 2D circle with radius r
  - $\alpha \coloneqq random\ number \in [0; 2\pi]$
  - $\boldsymbol{x} \coloneqq r \cdot (\sin(\alpha), \cos(\alpha))$

  (trigonometry in C++: `XM_PI`, `std::sin()`, `std::cos()`, …)

Moving Enemies

- Each enemy instance has a
  - Position $\boldsymbol{p} \in R^3$
  - Velocity $\boldsymbol{v} \in R^3$
- Position update in OnFrameMove()

$$\boldsymbol{p} := \boldsymbol{p} + h \cdot \boldsymbol{v} \quad (h = fElapsedTime)$$

- This is a simple case of projectile motion! (with no gravity and constant velocity)!

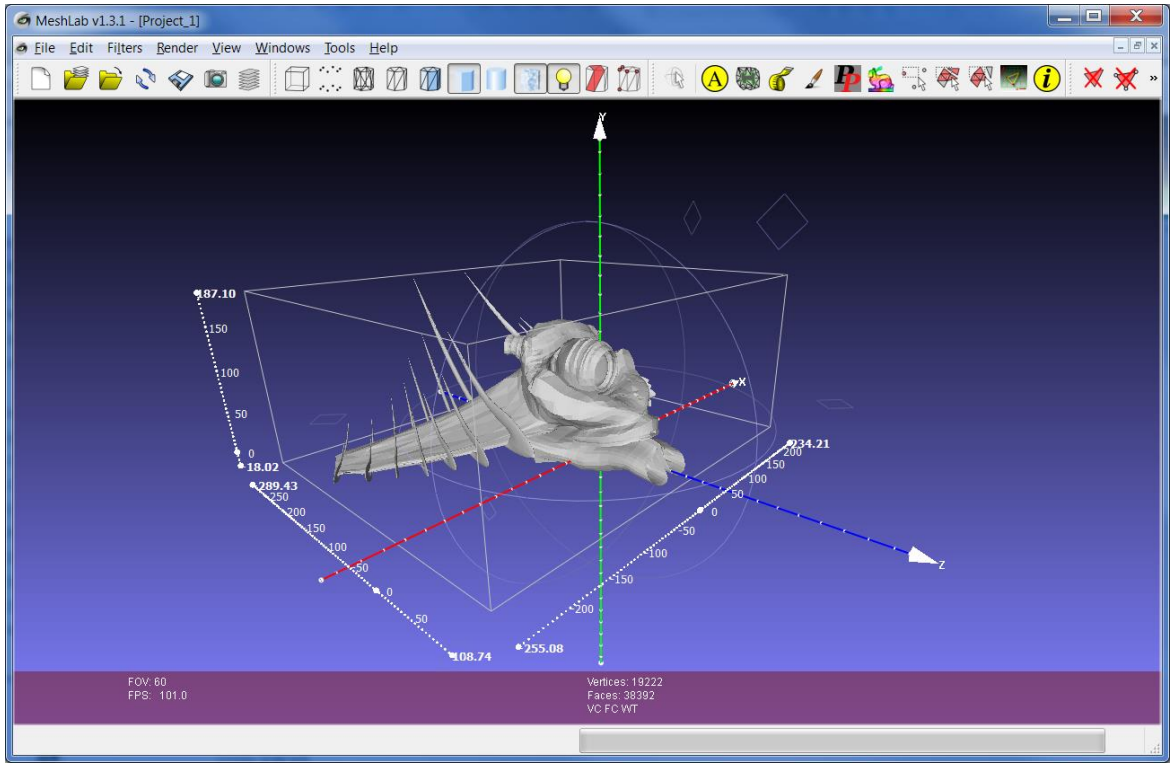
Enemy Transformations

- Cockpit Objects: $M_{WorldView} = M_{Obj} \cdot M_{CamWorld} \cdot M_{CamView}$ (→ Ass. 6)
- Ground Objects: $M_{WorldView} = M_{Obj} \cdot M_{CamView}$ (→ Ass. 7)

- Enemy Instances: $M_{WorldView} = \underbrace{M_{Obj} \cdot M_{Anim}}_{M_{World}} \cdot M_{CamView}$
- M_{Obj} is given by the config file (see slides for Assignment 07!)
- $M_{CamView}$ is the camera view matrix
- M_{Anim}: rotation/translation according to enemy position/velocity

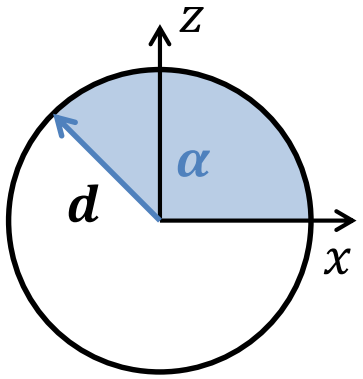
Enemy Transformations

- Object space correction: $M_{Scale} \cdot M_{RotX} \cdot M_{RotY} \cdot M_{RotZ} \cdot M_{Trans}$
- Unify pose of enemy meshes
  - Scale: World space size of enemy ship
  - Rotation: +x ↔ front, +y ↔ up
  - Translation: Ship mesh (roughly) centered at origin (that‘s already the case)



Enemy Transformations

- M_{Anim}: Move enemy mesh to position $\boldsymbol{p}$, facing into direction $\mathbf{d} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$
  - Rotation + translation
  - Assumption: Ships are always aligned parallel to the ground
- Corresponding matrix:
  - M_{Anim} = Rotation around y-Axis with angle α
    … followed by …
    translation by $\boldsymbol{p}$
  - Calculate α using `std::atan2()` with $d.x$ and $d.z$



Questions?