



DS ENGINE

TEAM ROAD KILLER

kmh

mph



THE PRODUCT

THE PRODUCT

What We Have

- Game Loop
- Rendering
- Scene Graph
- Physics
- 3D Audio
- Input

What We Don't Have

- Multicore
- Memory Management
- File Representation
- Partitioning
- Resource Manager
- ...

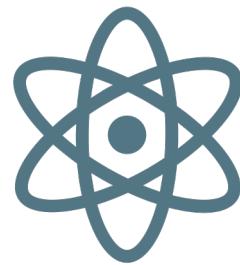
PROGRESSION

- Idea
- Preparation
- Architecture Design
- Fundamental Functionalities
- Working in Parallel
 - Rendering
 - Input
 - Physics
 - Audio
- Application

THE IDEA



Rendering



Physics



Audio

THE TECHNOLOGIES

- Coding Environment
 - Visual Studio 2017
 - Visual Studio 2019
 - Azure Pipelines
 - vcpkg
 - Doxygen

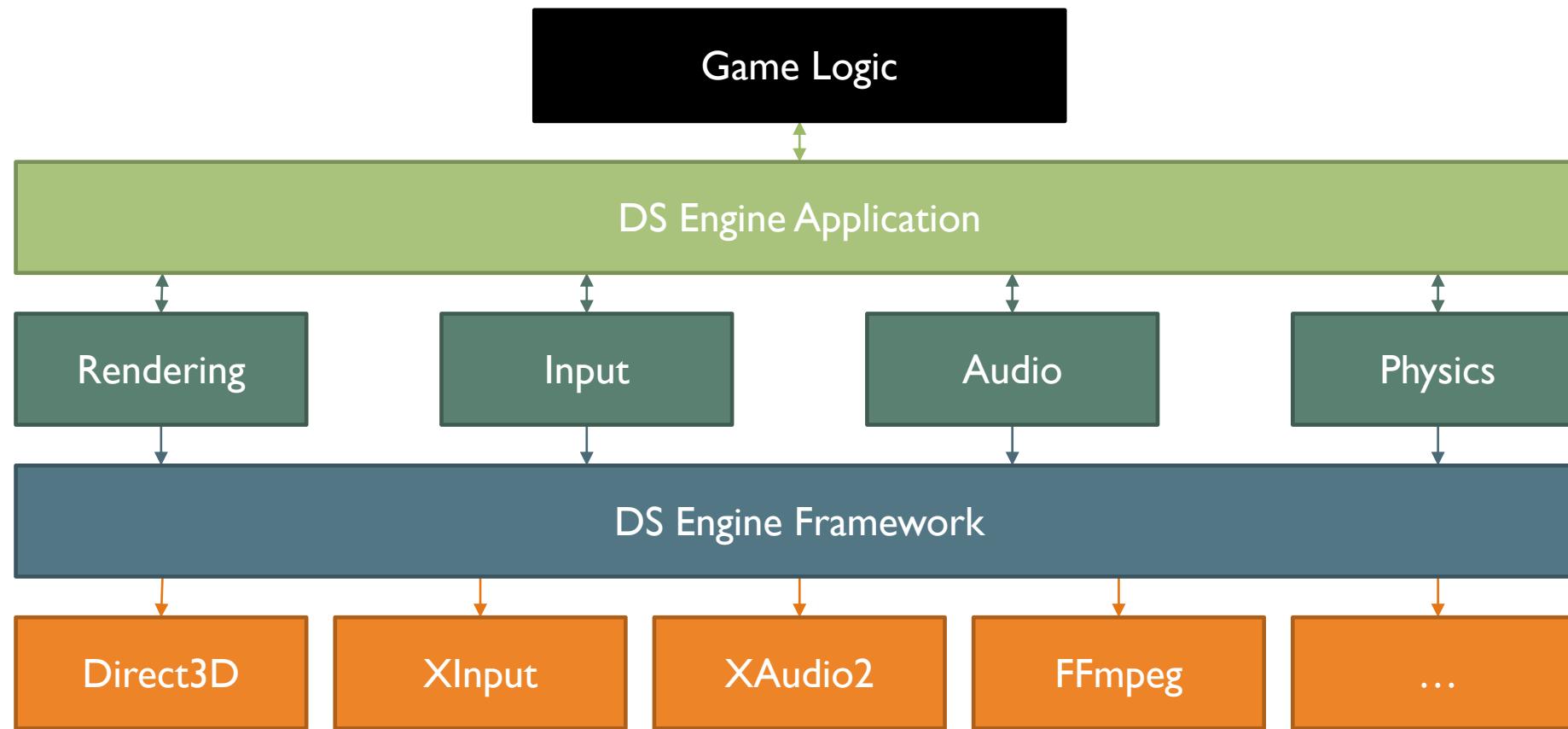
THE TECHNOLOGIES

- Coding Environment
- System Libraries
- DirectX 11
- XAudio2
- XInput

THE TECHNOLOGIES

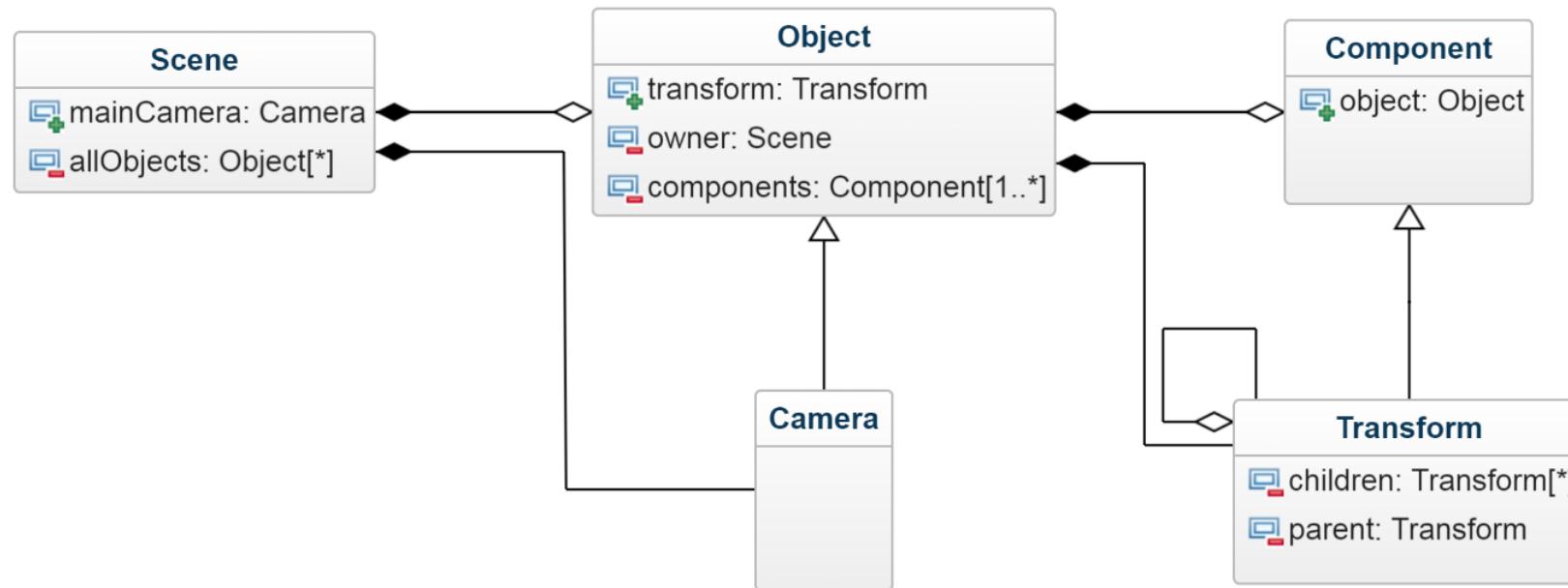
- Coding Environment
- System Libraries
- 3rd Party Libraries
 - Boost
 - FFmpeg
 - Assimp

ARCHITECTURE



SCENE GRAPH

ENTITY-COMPONENT





INPUT SYSTEM

- Unity-like

INPUT SYSTEM

- Unity-like
- Register at application initialization

```
SInput->RegisterInput("Horizontal", "d", "a", "", "", 10.0f, 0.1f, 10.0f, false, Button, MouseX, -1);  
SInput->RegisterInput("Horizontal", "", "", "", "", 10.0f, 0.1f, 10.0f, false, Axis, JoystickLX, -1);
```

```
float horizontal = SInput->GetAxis("Horizontal");
```

INPUT SYSTEM

- Unity-like
- Register at application initialization
- Run time function binding at registration

```
DSJoystickButtonCode buttonCode = GetJoystickButtonCodeFromName(buttonName);

if (buttonCode == RESERVED_0)
{
    getButton [buttonFuncIndex] = [buttonName]() { LOG_ERROR << "Joystick Button \""
                                                << buttonName << "\" Can Not Be Parsed!"; return false; };
    getButtonDown[buttonFuncIndex] = [buttonName]() { LOG_ERROR << "Joystick Button \""
                                                << buttonName << "\" Can Not Be Parsed!"; return false; };
    getButtonUp [buttonFuncIndex] = [buttonName]() { LOG_ERROR << "Joystick Button \""
                                                << buttonName << "\" Can Not Be Parsed!"; return false; };
}
else
{
    getButton [buttonFuncIndex] = std::bind(&DSFXInput::GetButton,      &inputSystem->xInput, buttonCode, joyNum);
    getButtonDown[buttonFuncIndex] = std::bind(&DSFXInput::GetButtonDown, &inputSystem->xInput, buttonCode, joyNum);
    getButtonUp [buttonFuncIndex] = std::bind(&DSFXInput::GetButtonUp,   &inputSystem->xInput, buttonCode, joyNum);
}
```

```
/**  
 * @brief The get button functions.  
 *  
 * 0: positive, 1: altPositive,  
 * 2: negative, 3: altNegative  
 */  
std::function<bool()> getButton[4];
```



INPUT SYSTEM

- Unity-like
- Register at application initialization
- Run time function binding at registration
- High-frequency joystick input polling (250Hz)

RENDERING SYSTEM

- Forward Rendering
- Physically Based Rendering
- User customized material
- Post Processing

PHYSICS SYSTEM

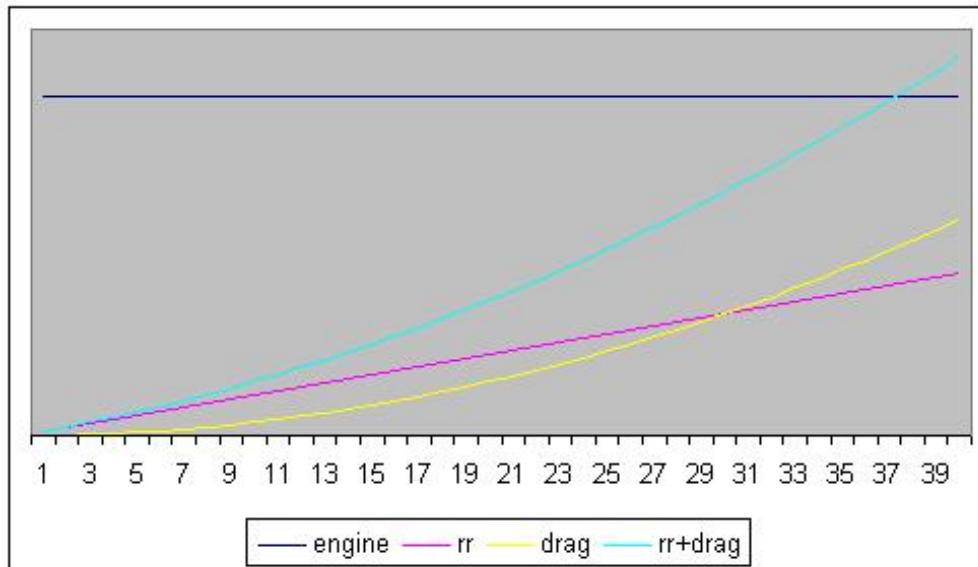
- Sphere Collier
- Box Collider
- Wheel Collider
- Rigid Body
- Ray Cast



CAR PHYSICS

- Straight line physics
- Steering

STRAIGHT LINE PHYSICS

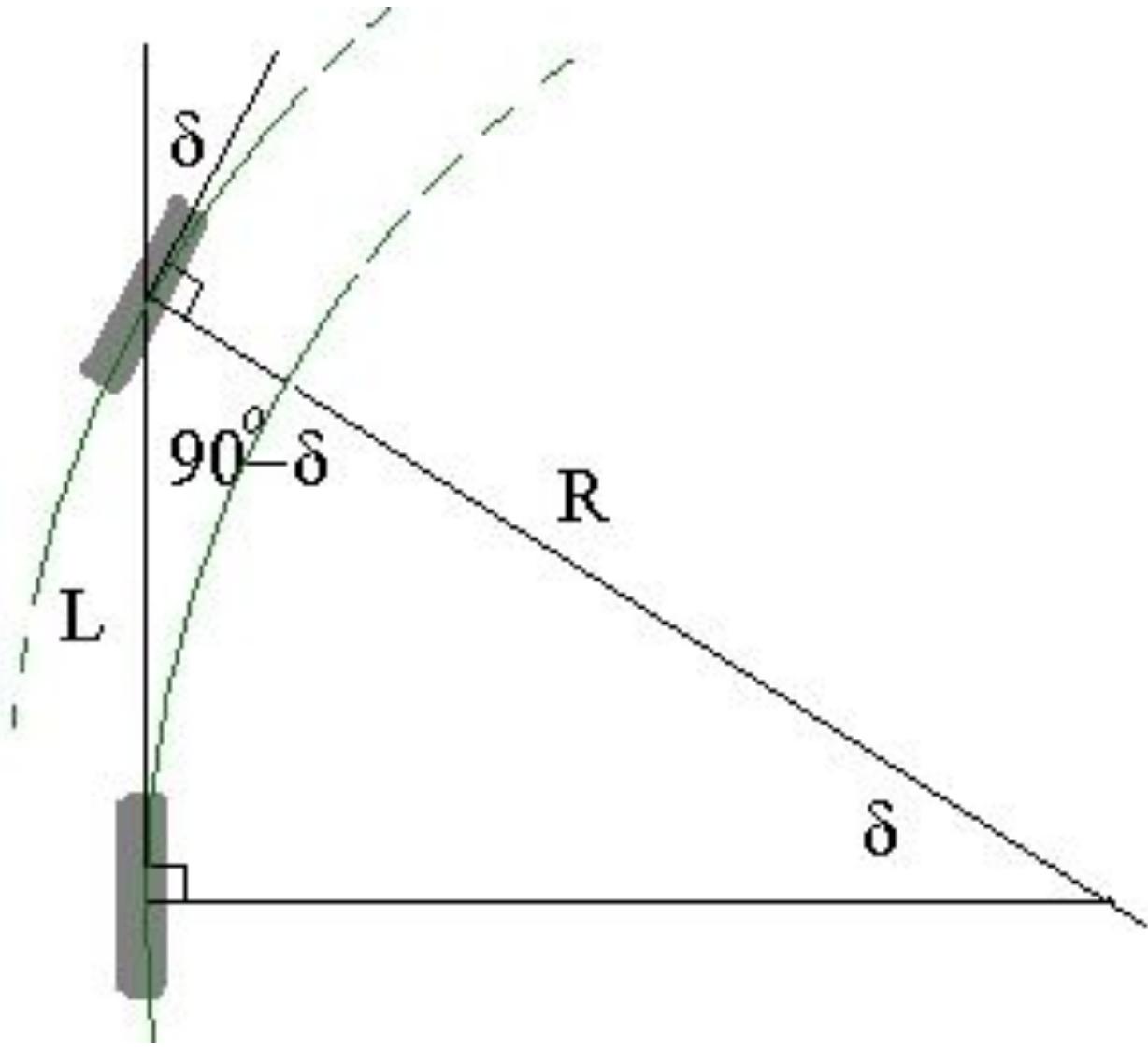


- $F_{long} = \frac{F_{traction}}{F_{braking}} + F_{drag} + F_{rr}$
 - $F_{traction} = u \cdot F_{Engine}$
 - $F = -u \cdot F_{Engine}$
 - $F_{drag} = -C_{drag} * v * |v|$
 - $F_{rr} = -C_{rr} \cdot v$
- $a = \frac{F}{m}$
- $v = v + a \cdot dt$
- $p = p + v \cdot dt$

STEERING

$$\sin \delta = \frac{L}{R} \leftrightarrow R = \frac{L}{\sin \delta}$$

$$\omega = \frac{v}{R}$$

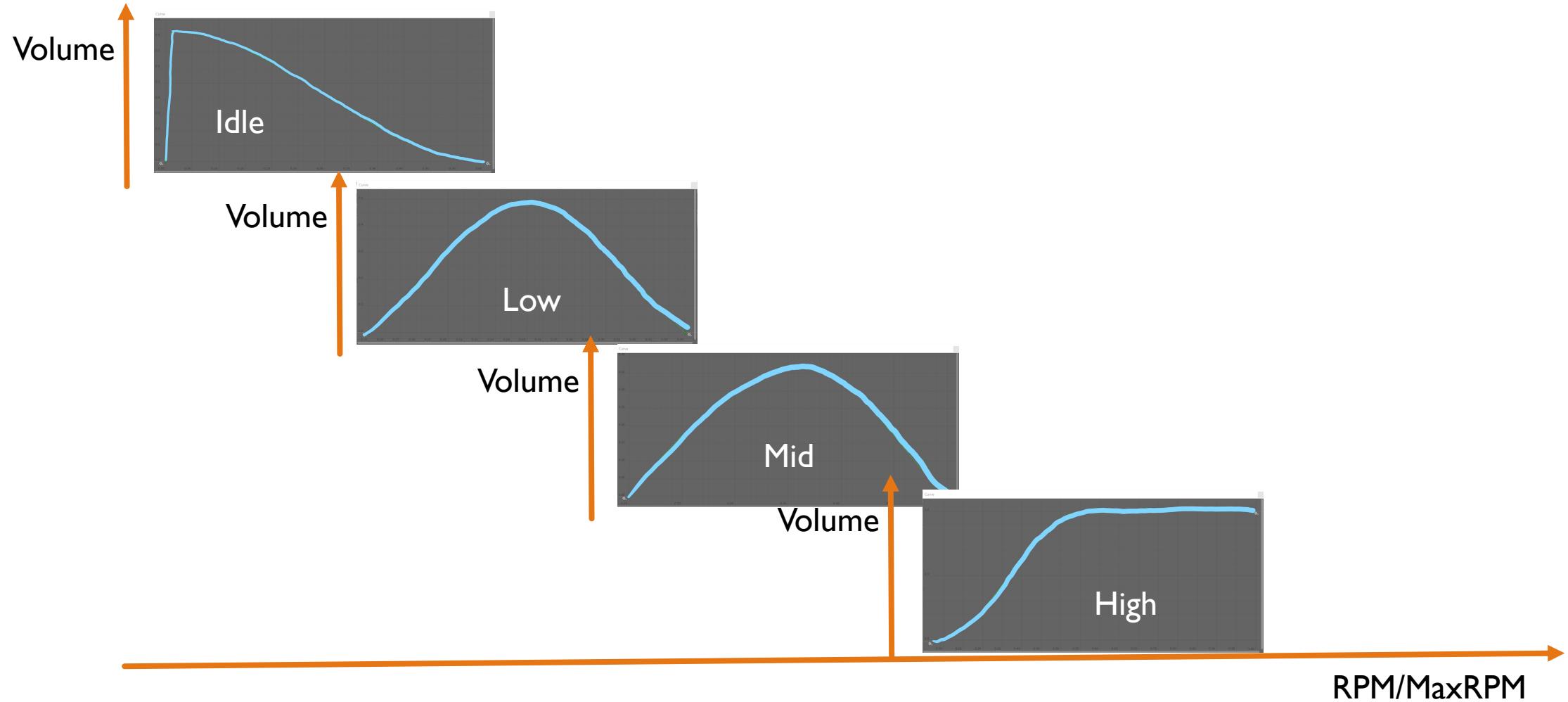




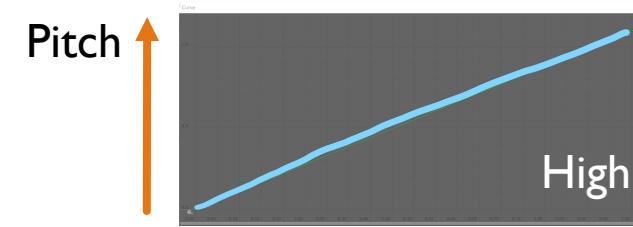
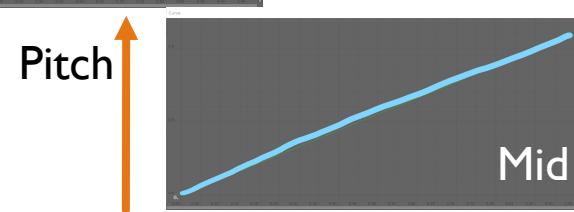
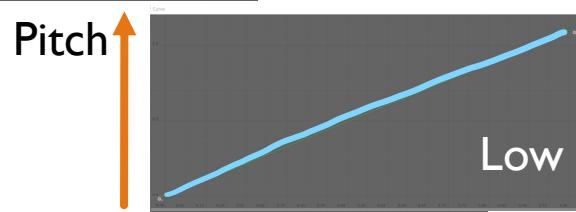
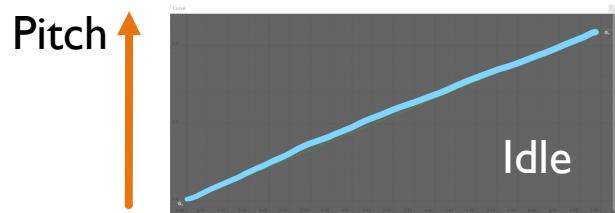
AUDIO SYSTEM

- Unity-like: AudioSource & AudioListener
- FFmpeg for decoding
- XAudio2 for playback
- Streaming from disk
- 3D Audio with X3DAudio

CAR AUDIO



CAR AUDIO



RPM/MaxRPM

POSTMORTEM

✓ Careful research and proper capability assessment

- The project isn't over-scaled

✓ Continuous Integration & Automatic Document Generation

- Teamwork got smoother

✗ We could have a better plan based on each member's specialty

- Might have more work done

✗ We made the decision of using a package manager too soon before researching its cons thoroughly

- More difficult to work because of the limited number of available machines



THANK YOU!

