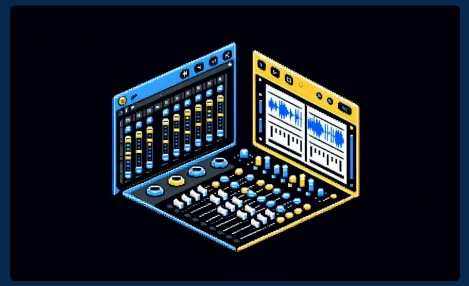
Week 10: Advanced Audio Techniques & Generative Sound

Review of Week 9

Audio Fundamentals

Basic concepts of sound and audio.

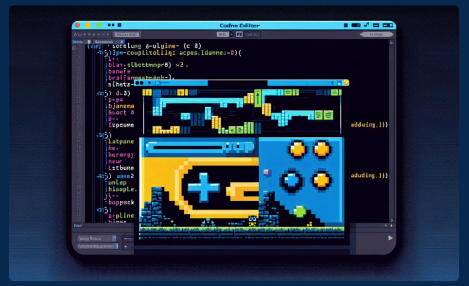


Unity Audio System

Understanding the Unity audio system.

Audio Implementation

Adding audio to Unity game objects.



Scripting Audio

Using code to manipulate audio.

YouTube

How Sounds Get Into Games - Fundamentals Of Game Audio ...

A summary of how sounds get into video games. Providing a general overview with examples and explaining the basics of audio...

Reverb & Reverb Zones

1 **Simulates Sound Reflections**

Different environments

2 **Unity's Audio Reverb Zone**

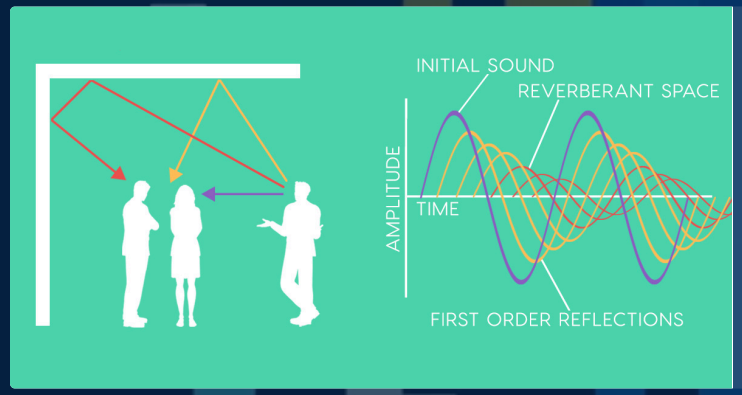
Component for reverb effects

3 **Customizable Parameters**

Room size, reflections, decay time

4 **Multiple Zones**

Different areas in your game



Flypaper

(Artificial) Space Is the Place: A Reverb Technology Primer – ...

Learn about the different types of reverbs, their histories, the effect they have on your recordings, and the best way to achieve them!

Spatialization in Unity: Practical Implementation

1 Why Spatialize?

Creates immersive 3D soundscapes

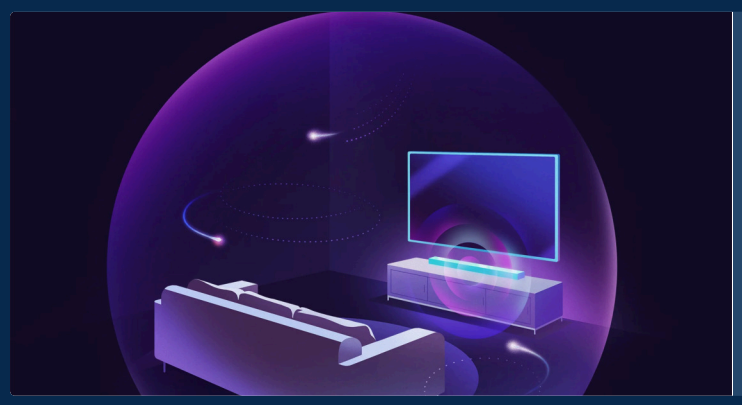
Enhances player orientation and feedback

2 How to Spatialize

- Set Audio Source's "Spatial Blend" to 1 (Full 3D)
- Adjust "3D Sound Settings"
- Choose appropriate "Volume Rolloff"

3 Use Audio Listener

On player/camera



www.dolby.com

What is spatial audio? How it works and how to use it

Learn how to use spatial audio to create cinematic soundscapes that fully surround you. Experience spatial audio in your home with Dolby...

Dynamic Spatialization via C#

```
// Example: Dynamically spatialize based on distance
public class DynamicSpatializer : MonoBehaviour
{
    public AudioSource audioSource;
    public float fullSpatialDistance = 10f;

    void Update()
    {
        float distanceToListener = Vector3.Distance(transform.position, Camera.main.transform.position);
        float spatialBlend = Mathf.Clamp01(distanceToListener / fullSpatialDistance);
        audioSource.spatialBlend = spatialBlend;
    }
}
```

docs.unity3d.com

Unity - Scripting API: AudioSource

Thank you for helping us improve the quality of Unity Documentation. Although we cannot accept all submissions, we do read each...

Audio Mixer & Dynamic Control

Grouping

Organizing audio sources

Effects

Applying effects to groups

Runtime Control

Controlling mix at runtime

Scripting Mixer Control: Code Example

1

MixerController

C# script

2

masterMixer

AudioMixer reference

3

SetMusicVolume

Adjusts music volume

4

SetReverbLevel

Controls reverb amount

```
using UnityEngine;
using UnityEngine.Audio;

public class MixerController : MonoBehaviour
{
    public AudioMixer masterMixer;

    public void SetMusicVolume(float volume)
    {
        masterMixer.SetFloat("MusicVolume", Mathf.Log10(volume) * 20);
    }

    public void SetReverbLevel(float amount)
    {
        masterMixer.SetFloat("ReverbLevel", amount);
    }
}
```

Process of Controlling Mixer Effects

1

Create exposed parameters in the Audio Mixer

Create exposed parameters in the Audio Mixer

2

Reference the Audio Mixer in your script

Reference the Audio Mixer in your script

3

Use SetFloat() to adjust parameters

Use SetFloat() to adjust parameters

4

Connect to UI elements or game events

Connect to UI elements or game events

Generative & Dynamic Music

Algorithmic Composition

Music created by algorithms

Adaptive Music Systems

Music changes based on gameplay

Layered Approach

Dynamic transitions between layers

Real-time Synthesis

Music created and manipulated in real-time

PureData (Pd) Introduction

Visual Programming
Language for audio

Modular Approach
Sound design

Flexible & Powerful
Real-time audio processing

puredata.info

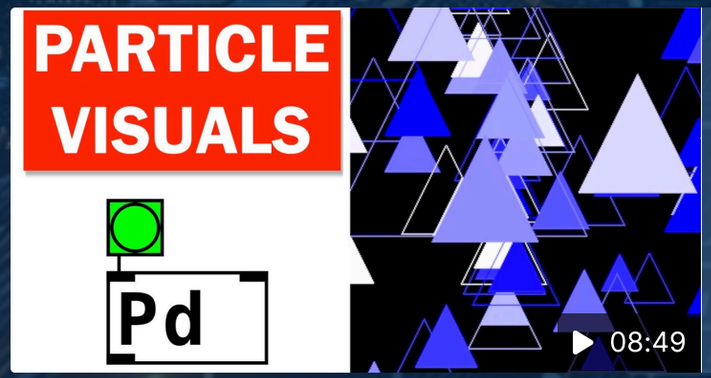
Pure Data — Pd Community Site

This site is a contribution of the IEM to the Pure Data community. Every Pure Data user who wants to contribute is welcome to join this portal and write/contribute some documentation, reports, news, comments and announcing...

plugdata.org

A visual programming environment for audio experimentation, prototyping and education | plugdata

A visual programming environment for audio experimentation, prototyping and education

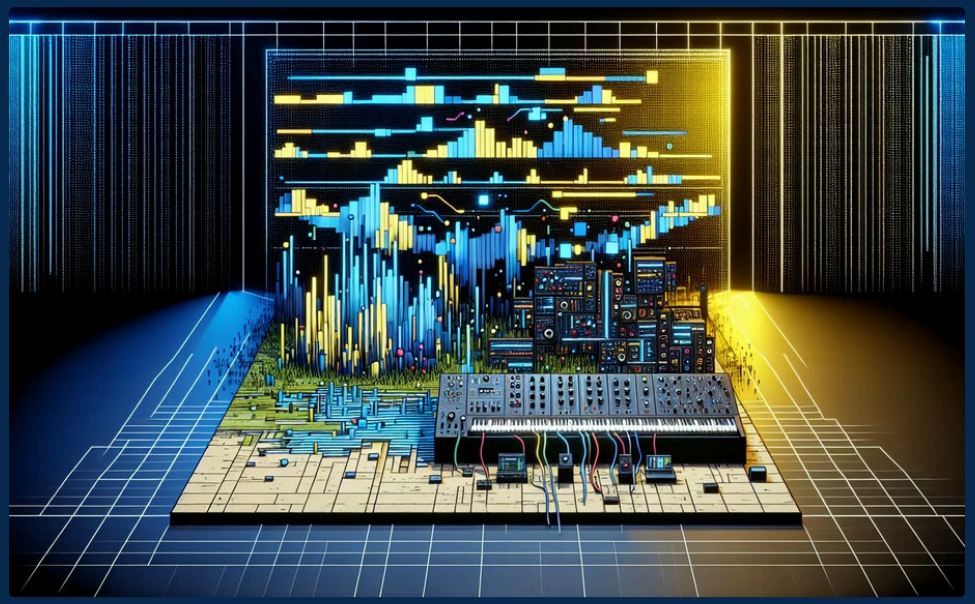


YouTube

Particle Visuals (Pure Data GEM Tutorial)

Let's create and control many shapes at once! Particle objects are great for creating complex live visuals in Pure Data GEM. Example Patchers:...

PureData + Unity: LibPd



LibPd: Embeddable PureData

Allows running Pd patches in Unity

Bridge between visual audio programming and game engine

Allows running Pd patches in Unity

LibPdIntegration/
LibPdIntegration

A libpd wrapper for Unity.

4 Contributors · 17 Issues · 115 Stars · 17 Forks

GitHub

GitHub - LibPdIntegration/LibPdIntegration: A libpd wrapper ...

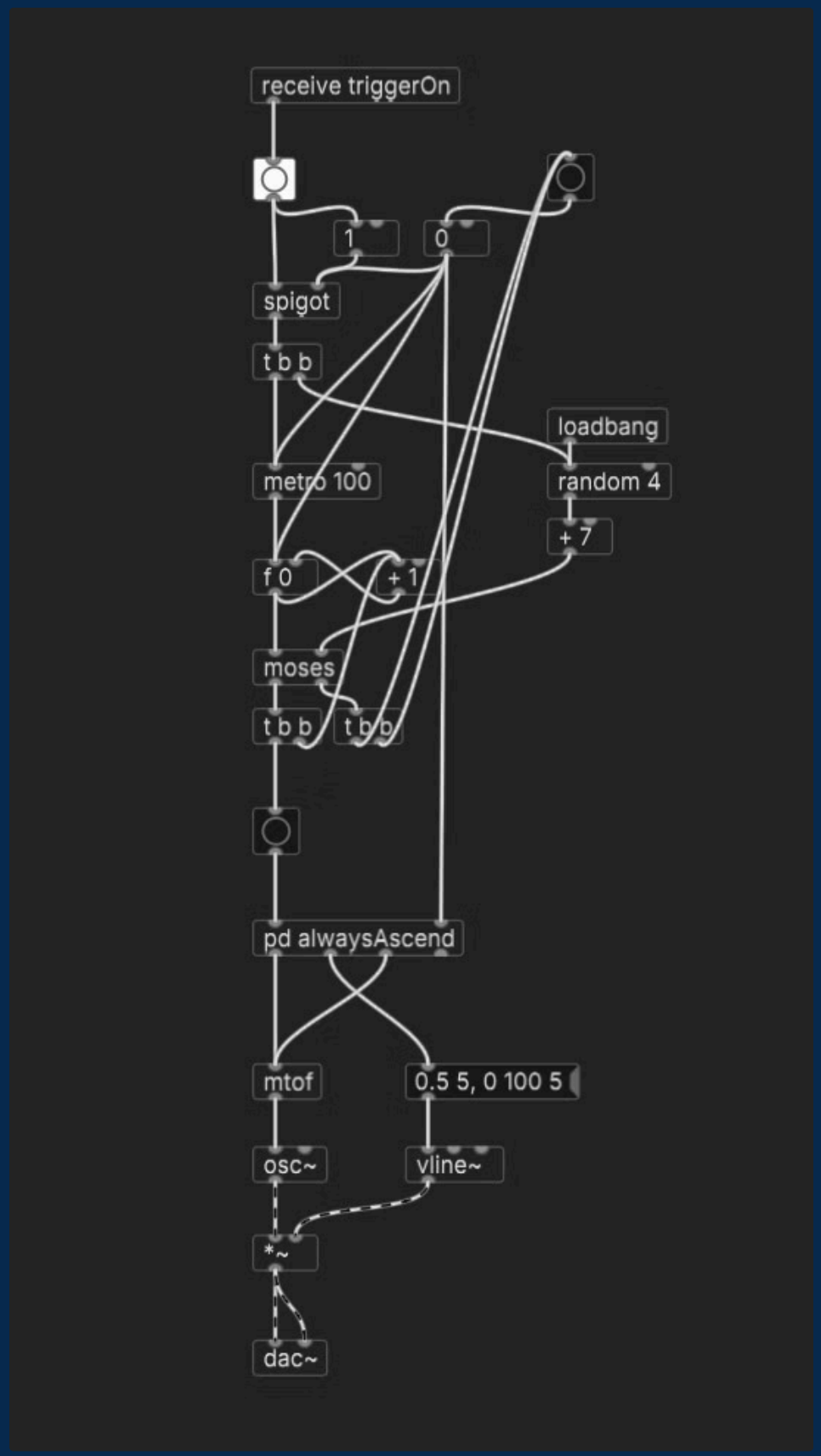
A libpd wrapper for Unity. Contribute to LibPdIntegration/LibPdIntegration development by creating an accoun...

YouTube

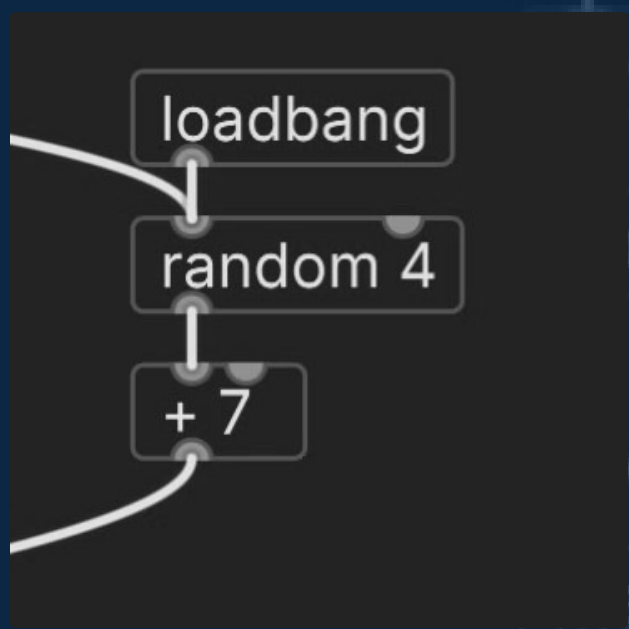
Unity & Pd Tutorial Ep.1: adding libpd to your Unity project

The basic things you need to use Pd in your existing Unity project Get the Unity project here: https://github.com/yannseznec/UnityPdBasics

Generative Music Sequence

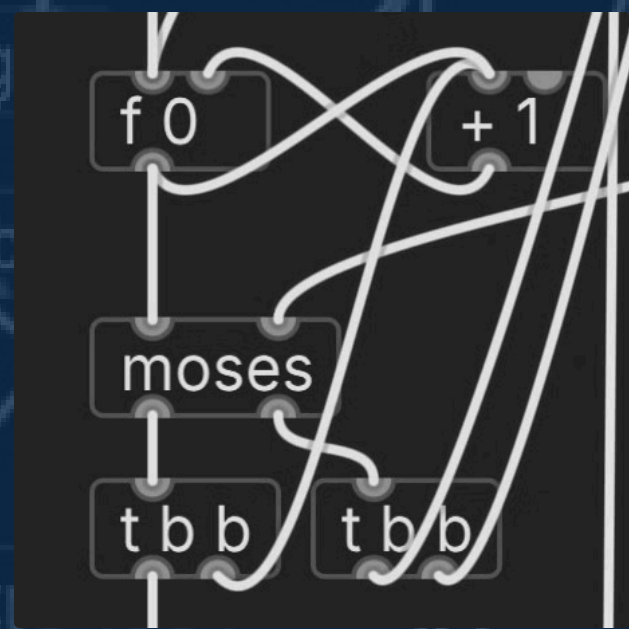


Generative Music Sequence - Explained



Random Note Generation

Creates unpredictable melodies



Counter Mechanism

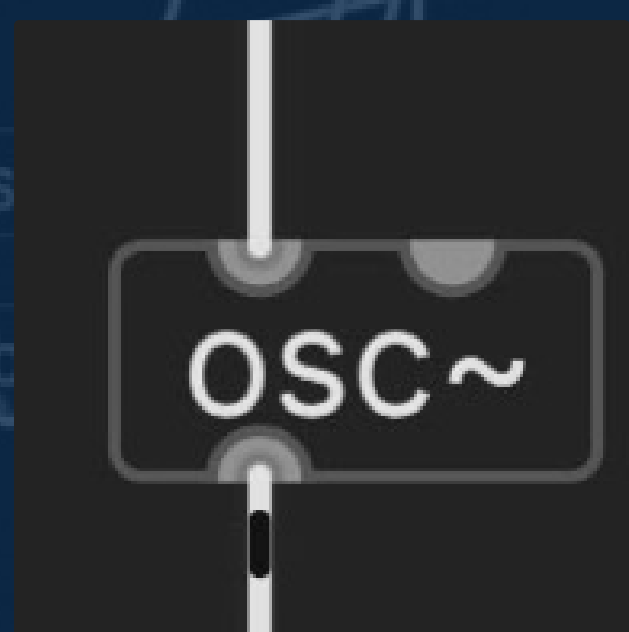
Keeps track of notes played

MIDI to Frequency

Translates notes to playable frequencies

Envelope Generator

Shapes the amplitude for a plucky sound

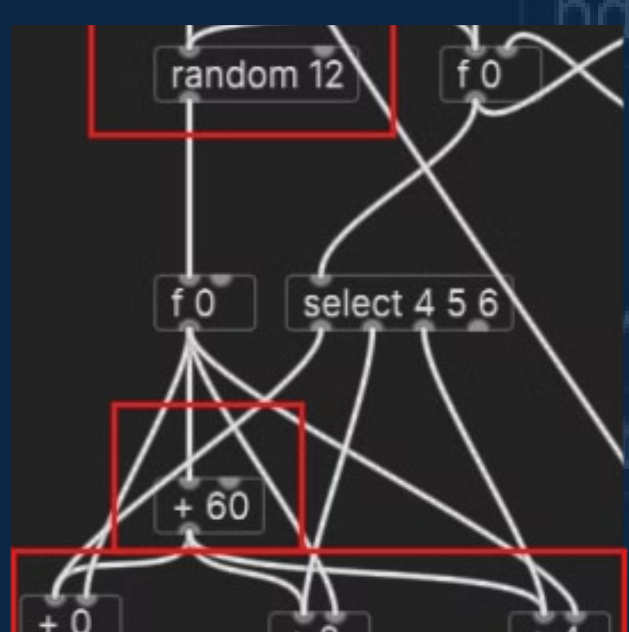


Oscillator

Produces the sound waveform

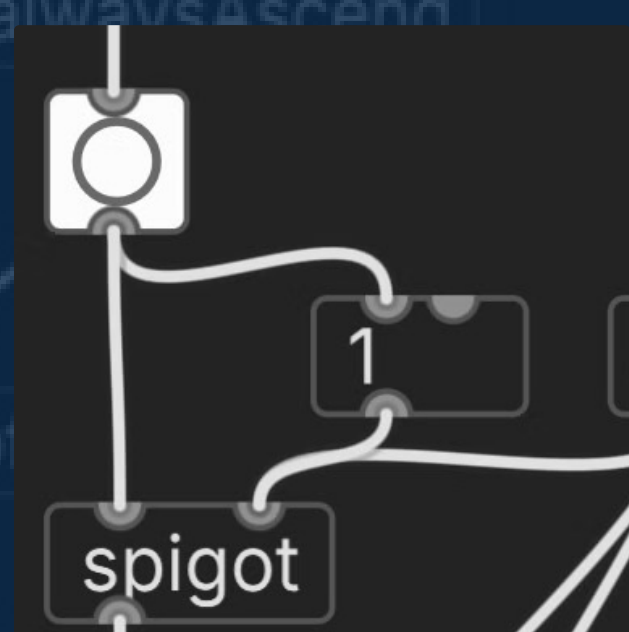
Metro Object

Controls timing and triggering of notes



Transposition

Changes the pitch of the generated notes



Spigot Object

Outputs a stream of random numbers used for note selection

DAC Object

Outputs audio signal via digital audio conversion

Using Pd in Unity: Demo

Final Project Overview

1 **1. Integrated Experience**

Create an interactive game-like experience

2 **2. Showcase Skills**

Demonstrate skills from all five modules

3 **3. Expand Pitch Concept**

Build upon your pitch assignment idea

4 **4. Focus Area**

Choose one: prop, character, or environment

Project Requirements by Module

1

Procedural + Modular Modeling

Implement procedural/modular system

For chosen focus

2

Material + Shader Development

Create at least 2 custom shaders

Shader Graph or HLSL

3

Image + Particle Effects

Develop at least 3 unique effects

Visual effects

4

Lighting + Camera Effects

Implement 3+ distinct lighting sources

Use 2+ image effects or post-processing

5

Game Audio and Implementation

Integrate 2+ audio effects and/or music tracks

Ensure proper spatial audio implementation

Game-Based Interactivity

Player Interaction

Meaningful, real-time interactivity

User Interface

Intuitive and responsive design

Feedback & Response

Clear and timely feedback

Narrative Framing

Contextualize player actions

Deliverables

Playthrough Video
Max 10 minutes

Presentation Video
Max 15 minutes

Unity Project
Zipped folder

Windows/PC Build
Executable file

Postmortem Document
Reflection on project

Final Project Drafts

Instructions for draft presentations.