



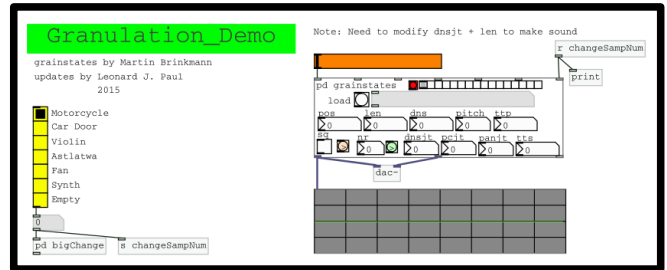
Game Audio Using Pure Data

Leonard J. Paul
School of Video Game Audio

<http://School.VideoGameAudio.com>
@SchoolGameAudio

What is Pure Data ?

- Pure Data(Pd) is visual-based scripting
- Similar to Max/MSP & Reaktor
- Script can be edited interactively
- In development since 1996
- Great for synthesis & generative music
- Cross-platform
- Open source + free





Pure Data for Games

- LibPd: <https://github.com/libpd/libpd>
 - Embed: iOS, Android, PC, etc..
- Magicolo's uPd – LibPd within Unity
- Heavy – Optimized Pd implementation



Heavy

- Compile Pd patches into:
 - Wwise & Unity plugins
 - Plain C/C++ code
 - JavaScript (WebAudio)
- Code is optimized for platforms
- Does not use Pd source code
- EnzienAudio.com



Compiling a Heavy Plugin

Enzien Audio Documentation Pd Objects Issues About release 'r2016.10' lpaul Logout

lpaul / TankEngine No file selected

Nov. 06, 2016 at 00:01:15 UTC in 374ms with release 'r2016.10'.

Notices

Warnings

1. hip~_pd in "_main.pd/hip~_pd" @ (x:47, y:157):
[rzero~] accepts only signal input. Arguments and control connections are ignored.

Files

Type	File	Size	
Pd	TankEngine.NThnD.pd.zip	973 Bytes	
C	TankEngine.NThnD.c.zip	72.2 kB	🔗
JS	TankEngine.NThnD.js.zip	47.9 kB	🔗
Pd External Src	TankEngine.NThnD.pdext.zip	on demand	
Pd External OSX	TankEngine~_pd_darwin	on demand	
Wwise Src	TankEngine.NThnD.wwise.zip	on demand	🔗
Wwise OSX/iOS	TankEngine.NThnD.wwise.zip	on demand	
Wwise Win32	TankEngine.NThnD.wwise.zip	on demand	
Wwise Win64	TankEngine.NThnD.wwise.zip	on demand	
Unity Src	TankEngine.NThnD.unity.zip	100.2 kB	🔗
Unity OSX	TankEngine.NThnD.unity.zip	133.1 kB	🔗
Unity Win32	TankEngine.NThnD.unity.zip	218.1 kB	
Unity Win64	TankEngine.NThnD.unity.zip	204.9 kB	
Unity Android32	TankEngine.NThnD.unity.zip	65.6 kB	
Unity Android64	TankEngine.NThnD.unity.zip	on demand	

Previous Versions

Version	Upload
1	NThnD - 00:01:15 Nov. 06, 2016



Heavy Advantages

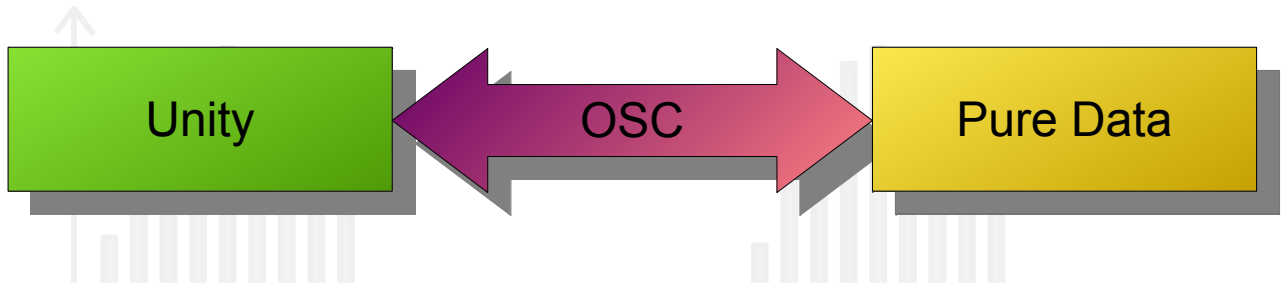
- Compiles optimized code
- Free for non-commercial use
- Online compiler updated automatically
- Access AudioClips in Unity
- Easily export Javascript for online use
- Online library of Pd helpful abstractions¹



Heavy Disadvantages

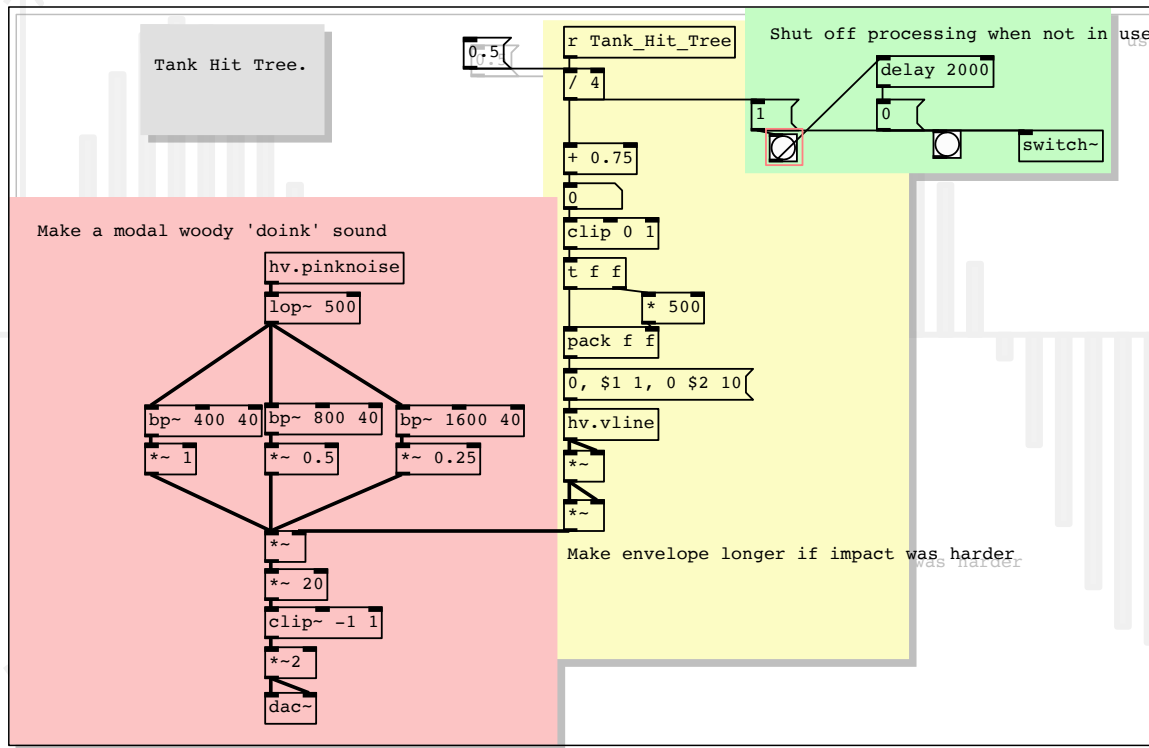
- Not all Pd objects are supported
- Code could be be more robust
- Errors on compilation sometimes vague
- Licensing cost (vs. LibPd)
 - Cost for Indies?

Prototyping: Unity + OSC+ Pd



- Open Sound Control¹ can communicate between programs and machines
- Work interactively with Pd via OSC
- Compile Pd patches in Heavy to finalize
- Contact me at SoVGA.com for Demo project

Sound Effect Using Synthesis



Demo=Unity Tanks +OSC+Pd+Heavy



The image shows a Unity game engine interface with a 3D scene of a tank game. Overlaid on the scene are several Pd software patches. The patches include:

- OSC_Receive**: A patch for receiving OSC messages on port 8001. It listens for a 'bang' message, stops listening, and outputs UDP packets to a 'pd parseTankOSC' object.
- OSC_Send**: A patch for sending OSC messages. It receives a 'bang' message, delays it by 20 units, and sends a 'Round_Start' message to a 'connect' object.
- Tank_Move**: A patch for handling tank movement. It receives a 'Tank_Speed' message and outputs a 'bang' message to a 'main-TankEngine 1' object.
- Tank_SFX**: A patch for handling tank sounds. It receives a 'Tank_Speed' message and outputs a 'bang' message to a 'main-Tank' object.
- Tank_Hit**: A patch for handling tank hits. It receives a 'Tank_Hit' message and outputs a 'bang' message to a 'Group' object.
- Shell_Hit**: A patch for handling shell hits. It receives a 'Shell_Hit' message and outputs a 'bang' message to a 'pd shell_hit' object.

A code editor window shows the following C# code for `EngineAudio()`:

```
92 // ... change the clip to driving and play.  
93 m_MovementAudio.clip = m_EngineDriving;  
94 m_MovementAudio.pitch = Random.Range(m_Orig  
95 m_MovementAudio.Play();  
96 }  
97  
98  
99 // SovGA: Use Heavy or Pure Data engine?  
100 if (m_Use_Heavy) {  
101 // set the speed of the tank engine in Heavy  
102 HeavyScript.Tank_Speed = m_Magnitude;  
103 } else {  
104 // find our OSC Library script  
105 GameObject go = GameObject.FindWithTag ("OSC");  
106 // send a /Tank_Speed OSC message with the playe  
107 go.GetComponent<OSC_SenReceives> ().PlaySoundOSC  
108 // read the m_size variable set by pd engine part  
109 transform.localScale = new Vector3 (1, go.GetCom  
110 }  
111  
112  
113  
114 private void FixedUpdate ()  
115 {
```

Questions

Feel free to ask me ?'s anytime at:

[@SchoolGameAudio](#)

or

[School.VideoGameAudio.com](#)