

Python in Game Development

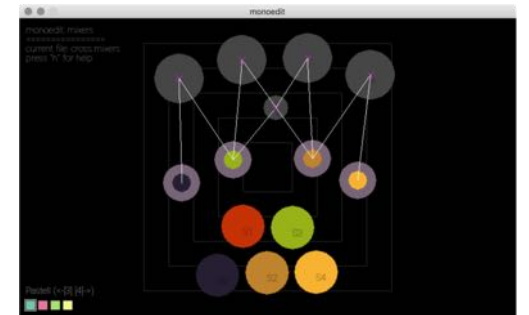
Thomas Perl

m@thp.io · <https://thp.io/> · @thp4

A talk about using Python in game development, from the asset pipeline over editors to fully fledged 3D game engines.

Overview

- Consoles and Mobile
- Physical Party Games
- Tooling: Level Editor
- Full 3D Engine



Overview

- Consoles and Mobile
- Physical Party Games
- Tooling: Level Editor
- Full 3D Engine

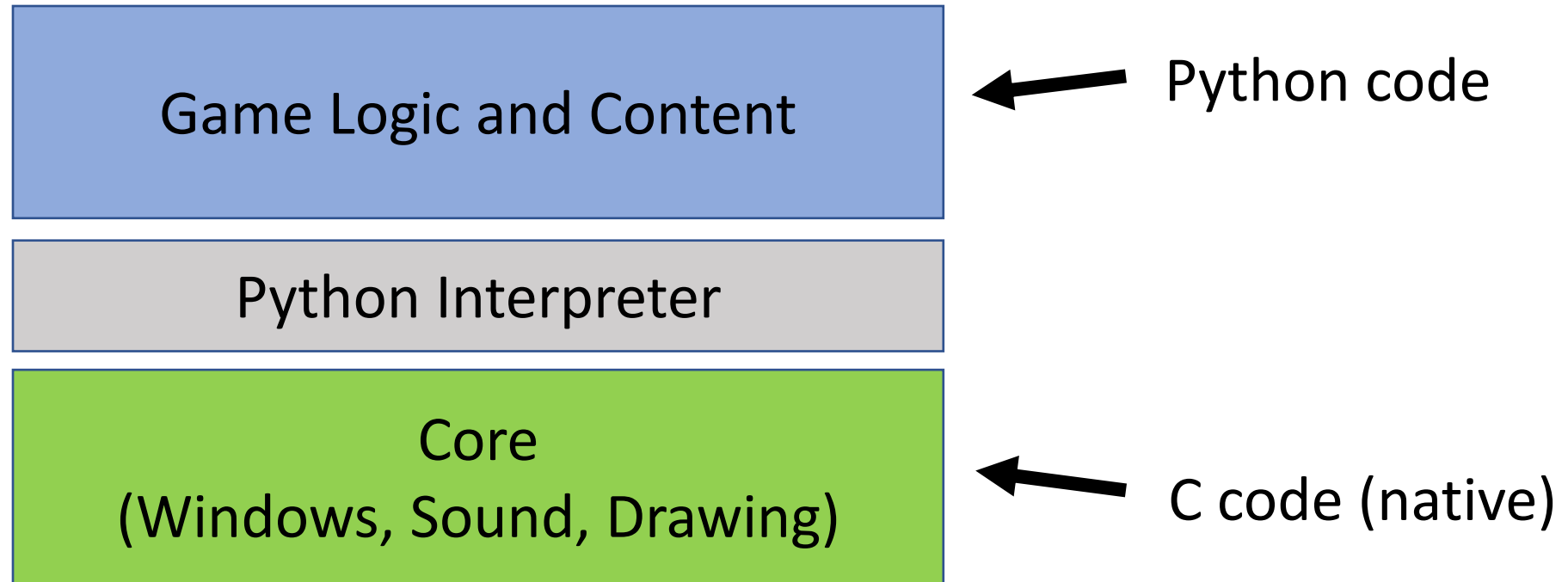


Consoles and Mobile: PyWeek

- We participated in PyWeek 15 (September 2012)
- Task: Write a game in one week from scratch as team
- Topic: "One Way Trip"
- Our entry: "One **Whale** Trip"
- More infos: <https://pyug.at/PyWeek/2012-09>

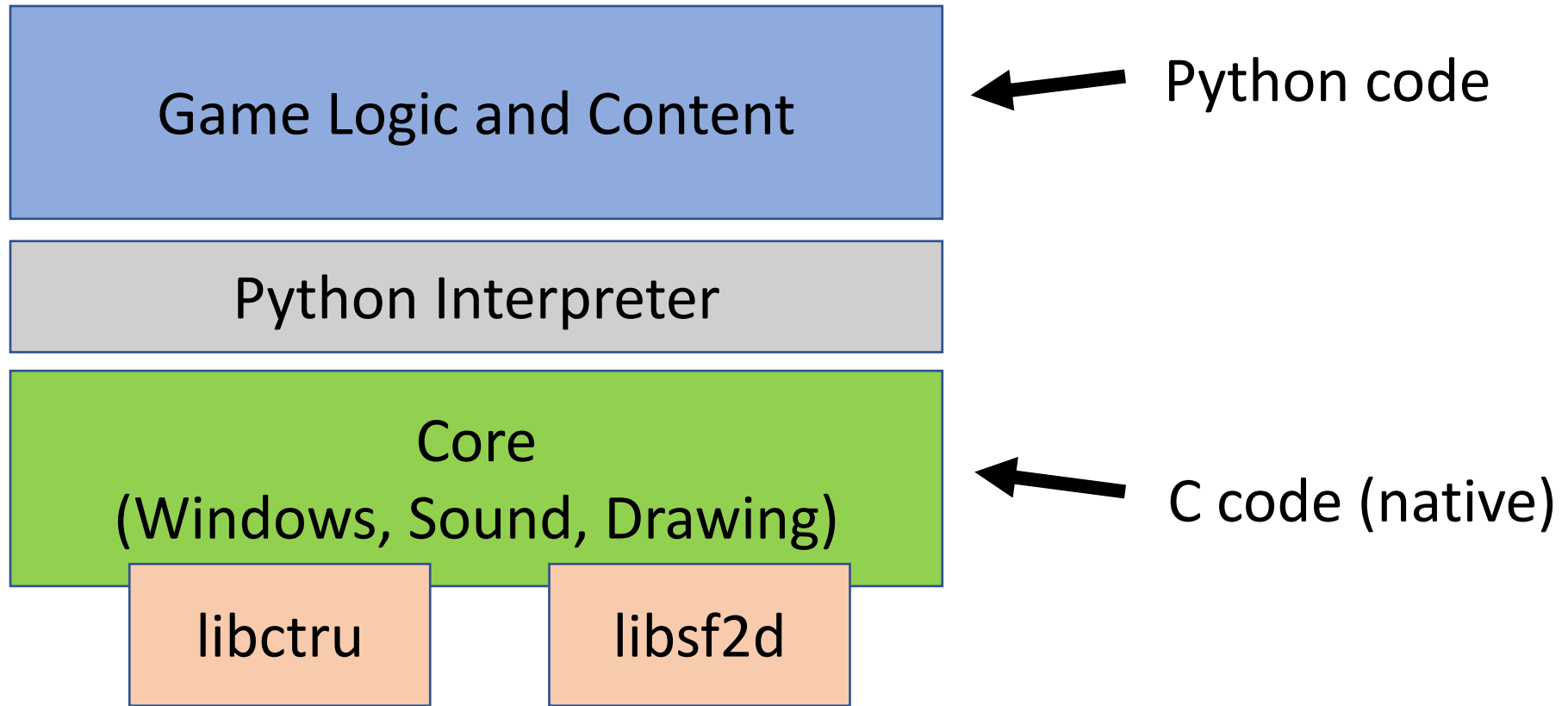


Consoles and Mobile: Architecture

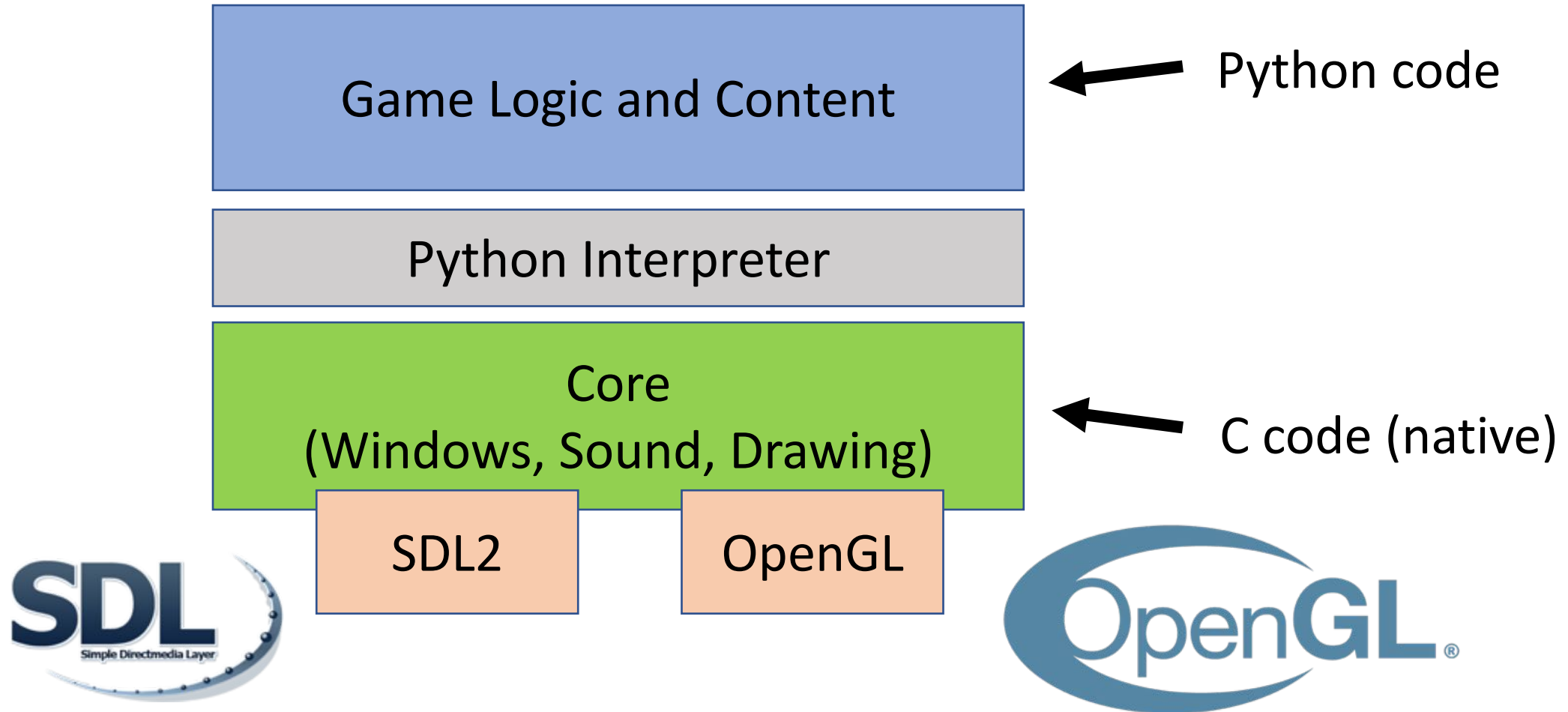


..platform-specific code goes here..

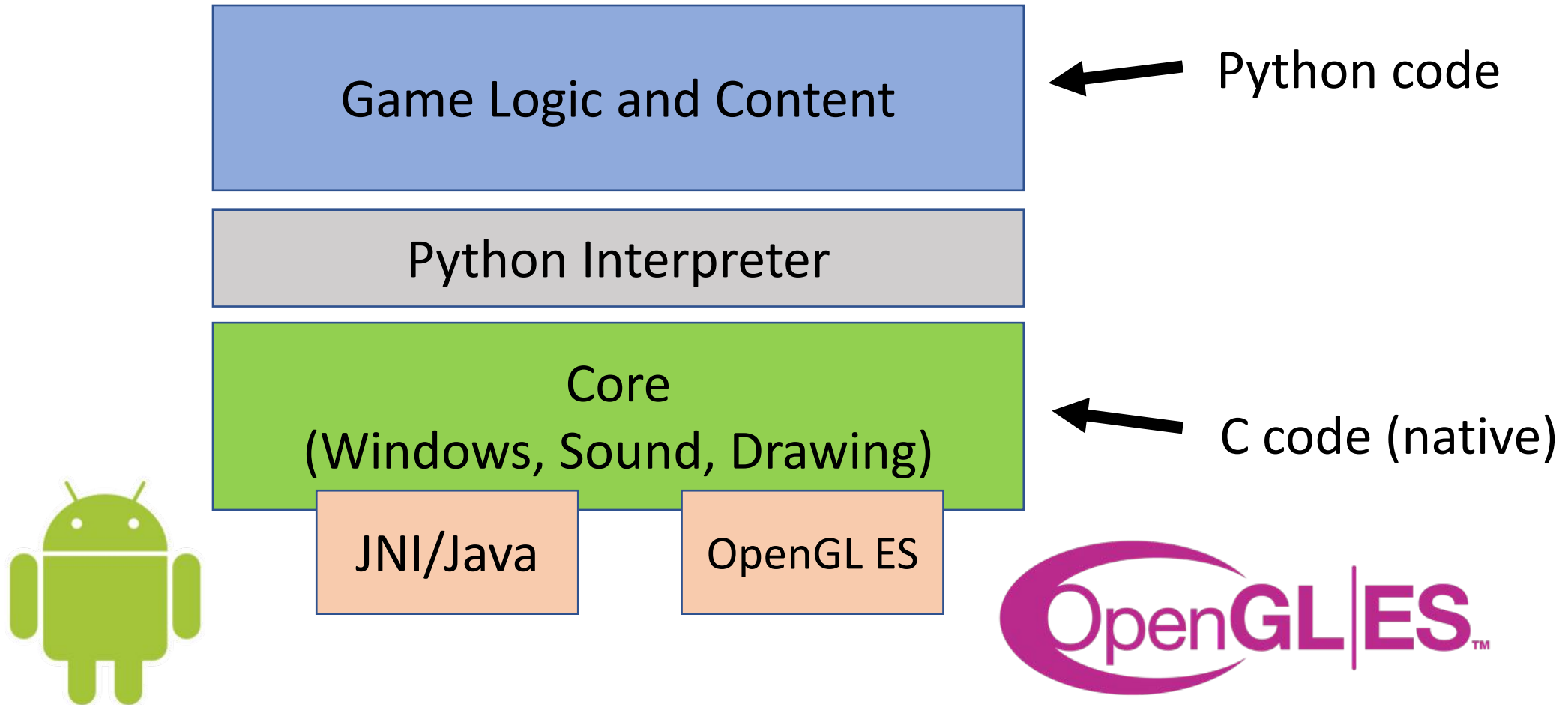
Consoles and Mobile: on 3DS



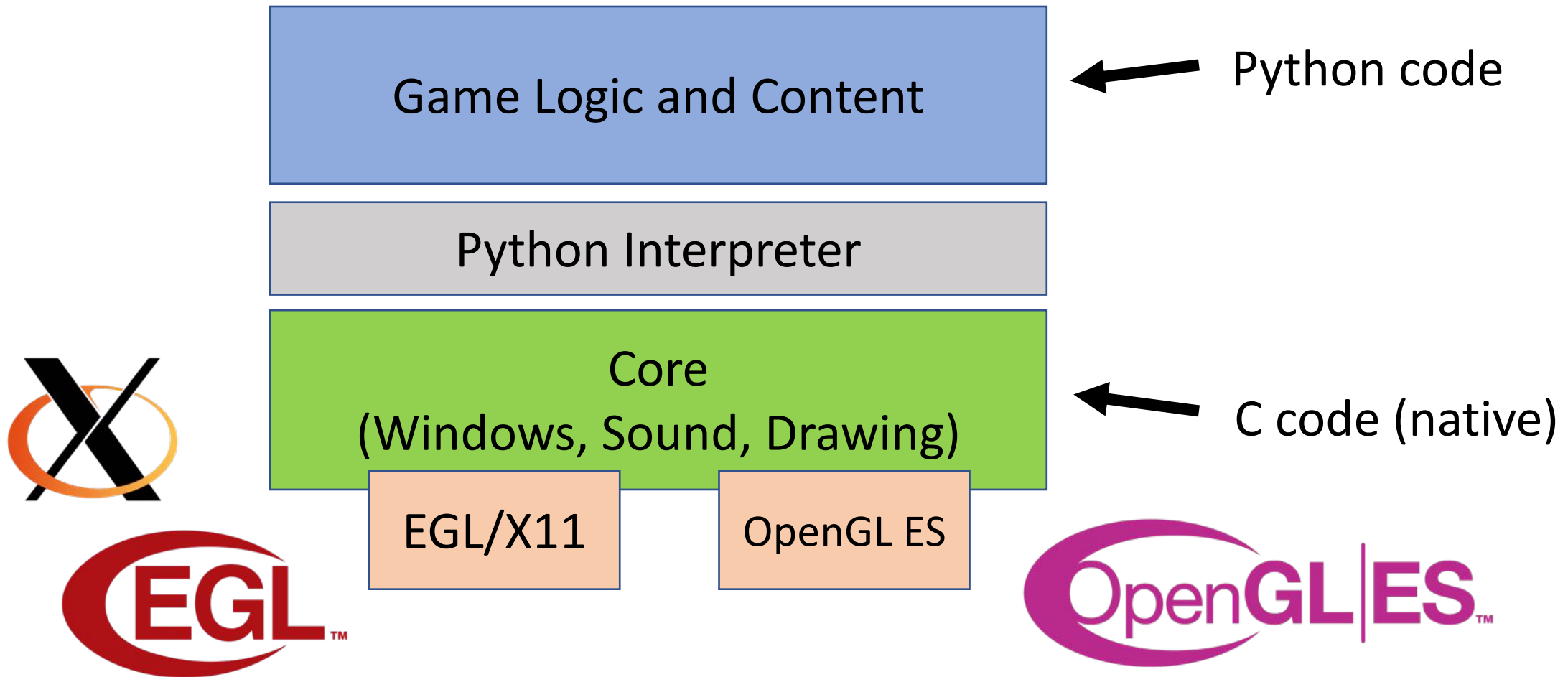
Consoles and Mobile: on Desktop



Consoles and Mobile: on Android



Consoles and Mobile: on Pocket C.H.I.P





Demo

One Whale Trip

Consoles and Mobile: Porting

- After the PyWeek, ported to:
 - OpenGL Renderer
 - Nintendo 3DS Homebrew (libs2d)
 - Pocket C.H.I.P (X11/EGL + OpenGL ES 2.0)
 - Android (OpenGL ES 2)

Consoles and Mobile: 3DS Port

- Using Python 2.7
- Stripped-down build (no dynamic extensions supported)
- Needed to patch the CPython source to get it to build
- Different implementation of "core" modules (rendering using sf2d)



Consoles and Mobile: Pocket C.H.I.P

- Using Python 3
- Debian Linux
- X11 ("X Window System")
- EGL for GL context creation
- OpenGL ES 2.0 for rendering



Consoles and Mobile: Android Port

- Using Python 3
- Python 3 built as static library
- Native library loaded via JNI from Java code
- Native library instantiates Python 3 interpreter
- Sound playback using Android's SoundPool
- Rendering: OpenGL ES 2 directly from native code



One Whale Trip

Download it now for
free from Google Play!

Overview

- Consoles and Mobile
- Physical Party Games
- Tooling: Level Editor
- Full 3D Engine



Physical Party Games



- Motion Controllers via Bluetooth
- PS Move API – <https://thp.io/2010/psmove/>
- Edgar Rice Soiree (2012) – PyGame, Python 2
- Parley Who Vertigo (2017) – Custom Engine, Python 3



Parley Who Vertigo



Edgar Rice Soirée



Edgar Rice Soirée



Parley Who Vertigo



Parley Who Vertigo



Parley Who Vertigo: Minigame API

```
minigame.py (~/.src/bit...-vertigo/python) - VIM — vim python/minigame.py...
class MiniGame(object):
    def __init__(self, gameflow):
        self.gameflow = gameflow

    def button_pressed(self, player, button):
        ...

    def status_message(self):
        return ''

    def base_color(self):
        raise NotImplementedError

    def start_game(self):
        raise NotImplementedError

    def update(self):
        raise NotImplementedError

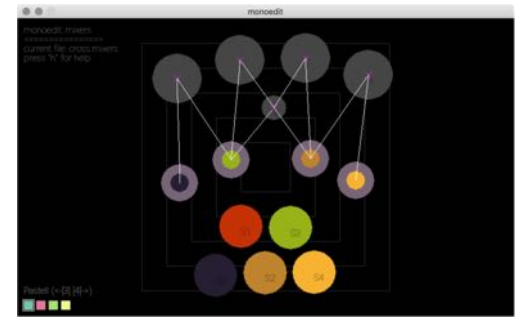
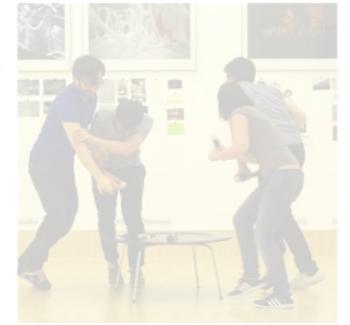
python/minigame.py
-- INSERT --
```

Demo

Python + PS Move API

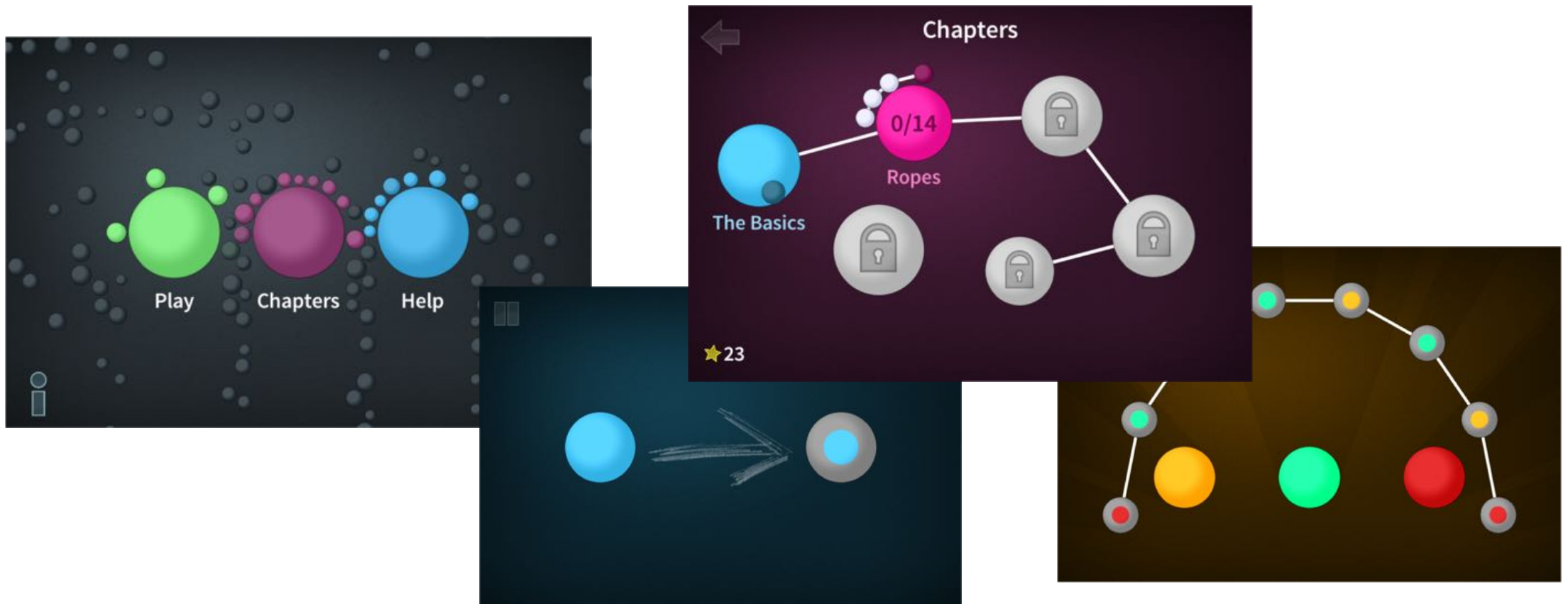
Overview

- Consoles and Mobile
- Physical Party Games
- Tooling: Level Editor
- Full 3D Engine



Tooling: Level Editor

- chro.mono (2013): Mobile game, levels done "in code"

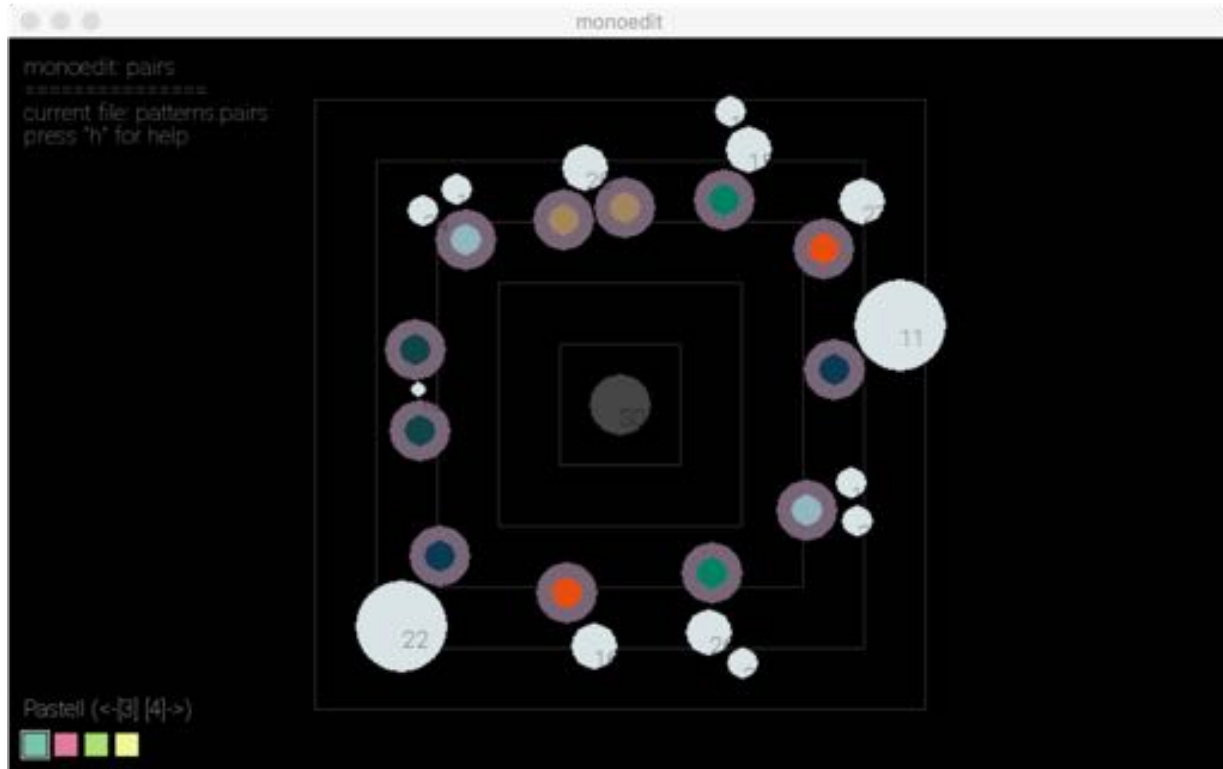


Tooling: Level Editor

- Sequel: "chro.mono 2" (2018)
- Task: Create an easy-to-use level editor
- Three different "content" types:
 - Untangle (remove overlapping lines via dragging)
 - Mixers (mix colors with connected circles)
 - Pairs (colorize pairs via touching elements)
- Game engine doesn't have Python interpreter
- Store "source" level data as text
- Convert to native C structs for loading by game engine



Pairs levels

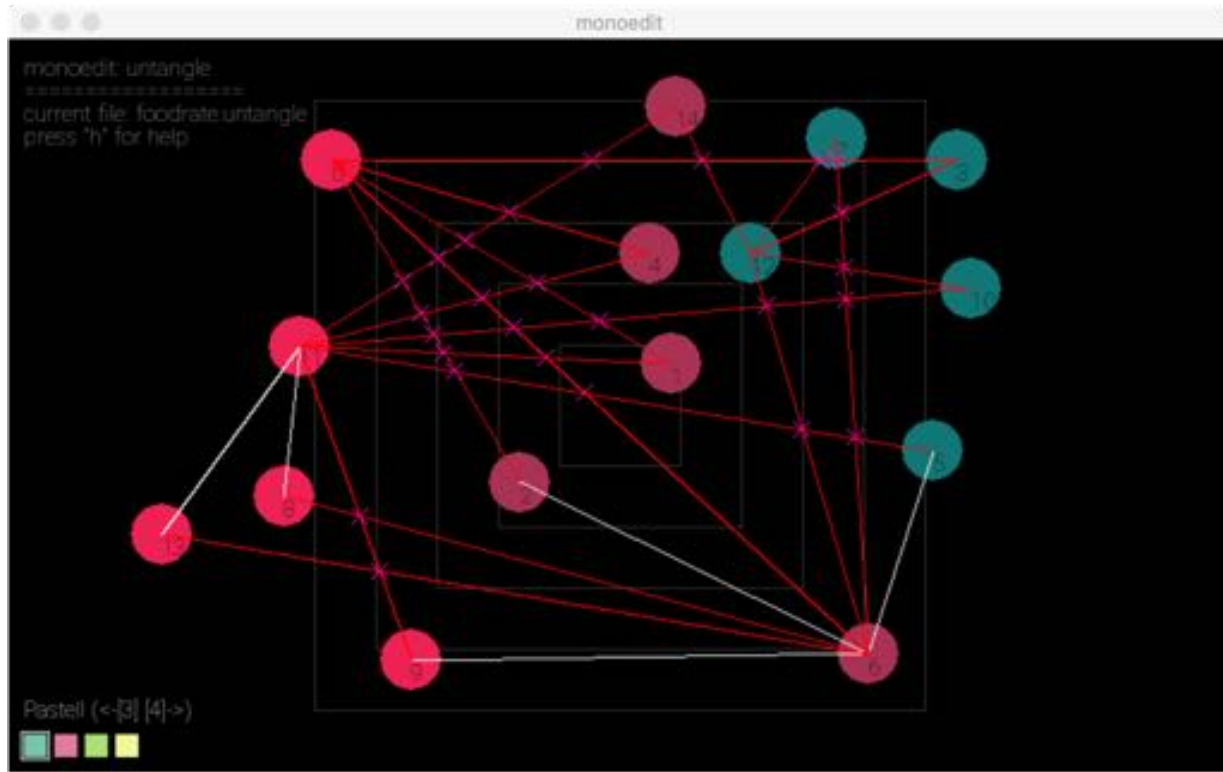


Editor

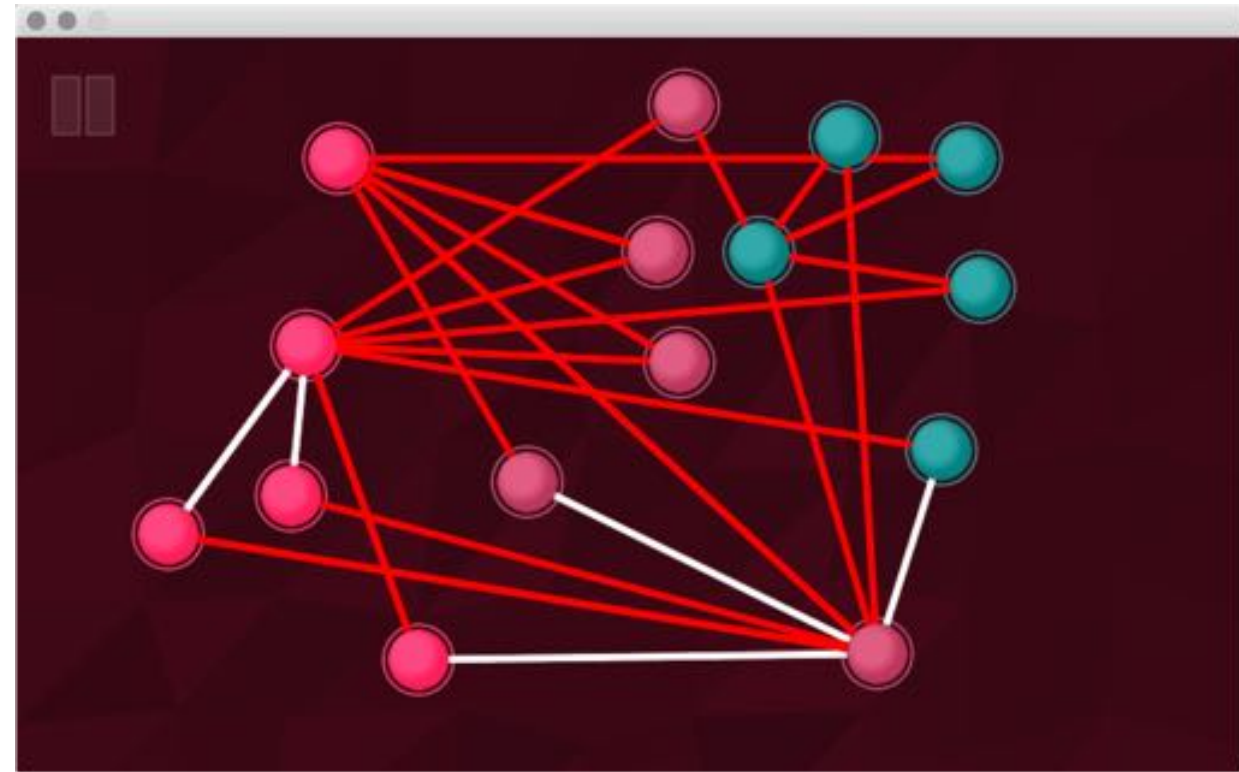


Game

Untangle levels

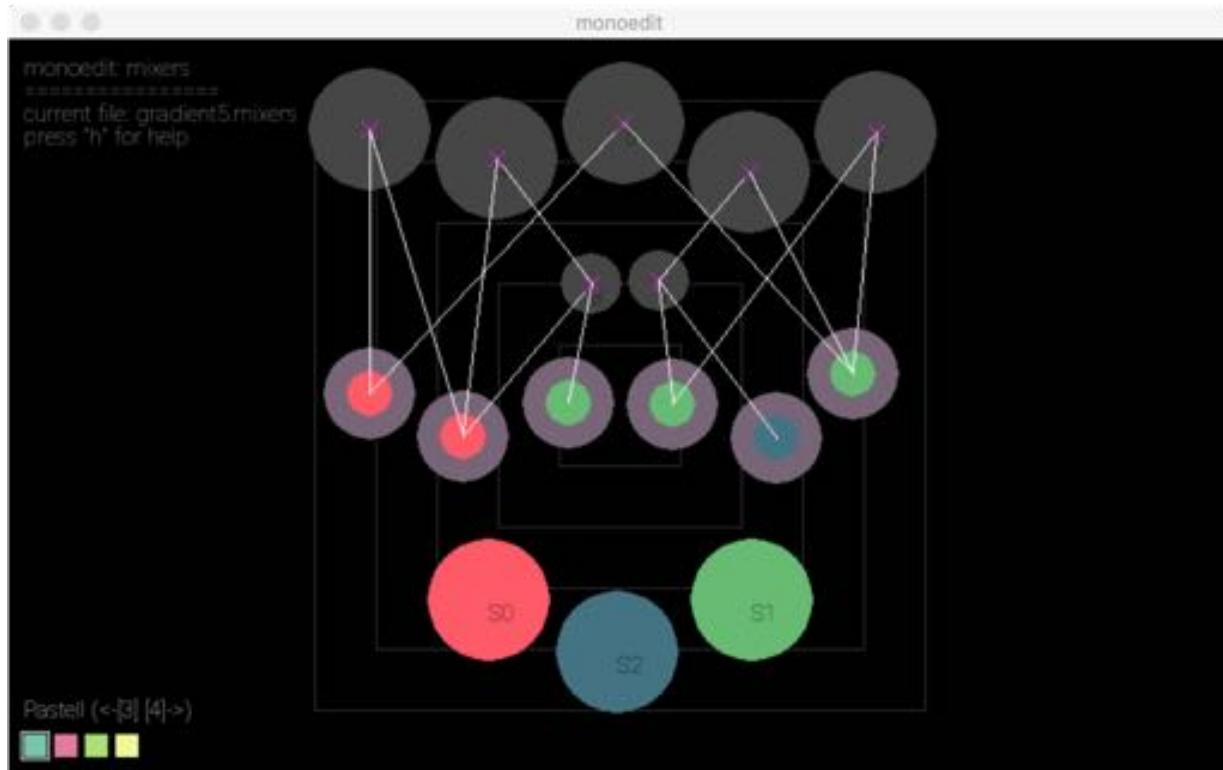


Editor

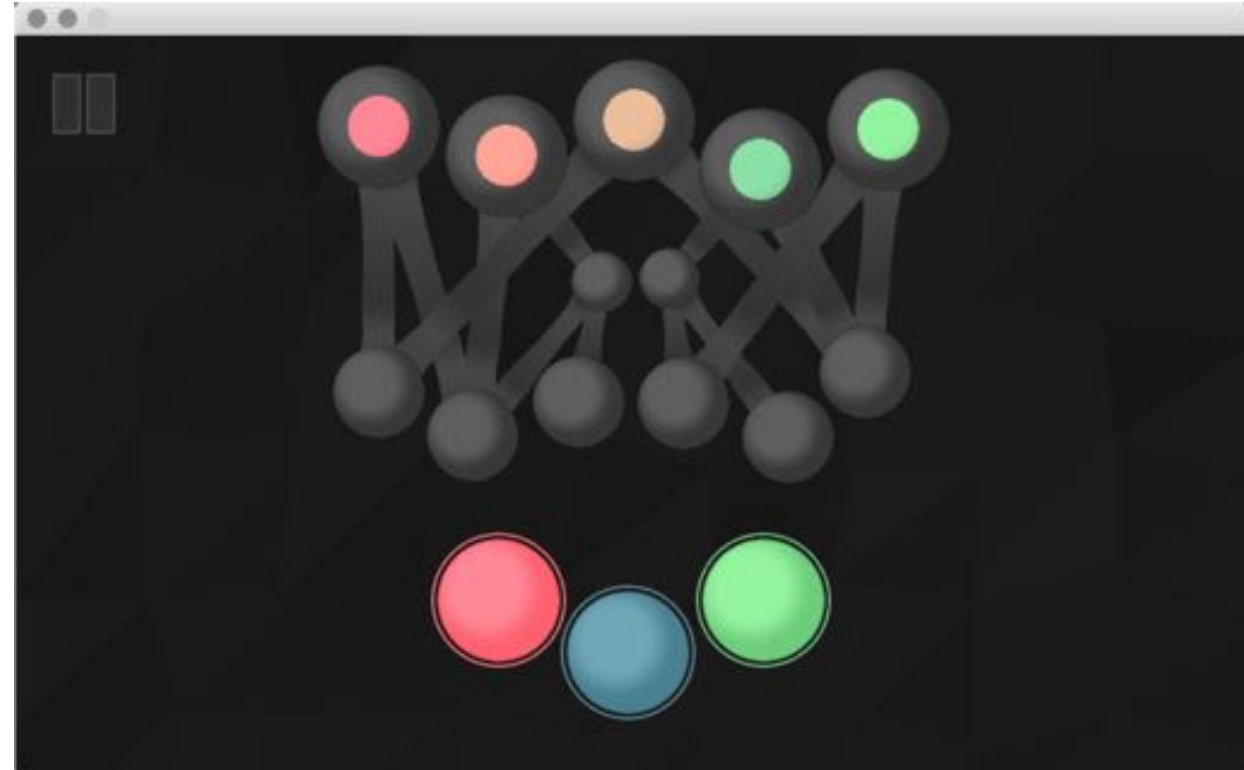


Game

Mixer levels



Editor



Game

Demo

Python Level Editor

Editor implementation



- PyGame for rendering
- Custom text-based format for "level source"
- Python struct module for turning into binary data

```
fp.write(struct.pack('<BfffBBBI', vertex.idx, vertex.pos.x,  
                    vertex.pos.y, vertex.size, vertex.color[0],  
                    vertex.color[1], vertex.color[2],  
                    vertex.calculate_flags()))
```


Level data

Source for a single level (text)

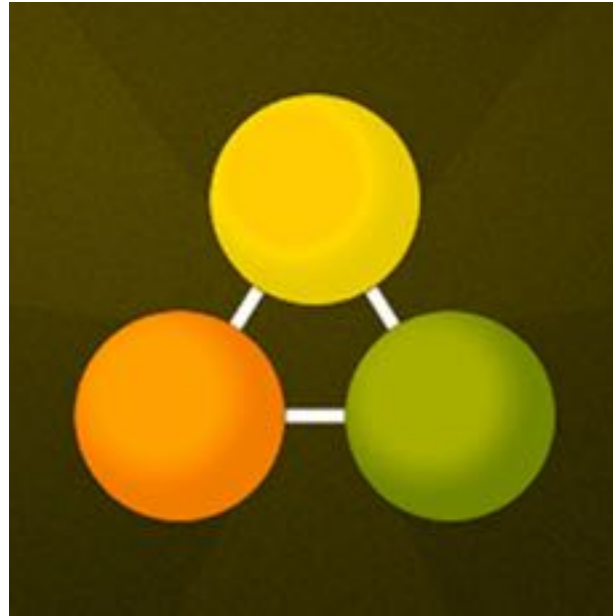
All levels in binary format



```
modthree.pairs (~...tor/content) - VIM -- vim content/...
v 0 395.000 229.000 15.000 30 30 30 ns
v 1 420.000 405.000 35.000 237 36 78 c
v 2 493.000 373.205 35.000 185 38 106 c
v 3 543.205 322.000 35.000 246 102 15 c
v 4 567.000 211.000 35.000 237 36 78 c
v 5 544.205 137.000 35.000 185 38 106 c
v 6 498.000 74.795 35.000 246 102 15 c
v 7 378.000 47.000 35.000 237 36 78 c
v 8 308.000 73.795 35.000 185 38 106 c
v 9 252.795 127.000 35.000 246 102 15 c
v 10 226.000 245.000 35.000 237 36 78 c
v 11 241.795 318.000 35.000 185 38 106 c
v 12 289.000 376.205 35.000 246 102 15 c
v 13 236.000 182.000 10.000 255 0 0 h
v 14 302.000 199.000 10.000 255 0 0 h
v 15 421.000 118.000 10.000 255 0 0 h
v 16 440.000 51.000 10.000 255 0 0 h
v 17 498.000 249.000 10.000 255 0 0 h
v 18 567.000 269.000 10.000 255 0 0 h
v 19 353.000 405.000 10.000 255 0 0 h
v 20 368.000 351.000 10.000 255 0 0 h
e 13 14
e 15 16
e 17 18
e 19 20
~
content/modthree.pairs
-- INSERT --
```

00000000	4d 49 58 52 14 00 45 00	bb 00 62 6f 72 69 6e 67	MIXR..E...boring
00000010	00 63 72 6f 73 73 00 66	69 66 74 65 65 6e 73 68	.cross.fifteensh
00000020	61 64 65 73 00 66 69 72	65 63 72 6f 77 6e 00 67	ades.firecrown.g
00000030	72 61 64 69 65 6e 74 33	00 67 72 61 64 69 65 6e	radient3.gradien
00000040	74 35 00 6a 75 73 74 61	74 72 65 65 00 6c 65 76	t5.justatree.lev
00000050	65 6c 00 6c 65 76 65 6c	31 33 00 6c 65 76 65 6c	el.level13.level
00000060	31 35 00 6c 65 76 65 6c	31 35 61 00 6c 65 76 65	15.level15a.leve
00000070	6c 31 39 00 6c 65 76 65	6c 32 35 00 6c 65 76 65	l19.level25.leve
00000080	6c 32 36 00 6c 65 76 65	6c 32 37 00 6e 65 74 77	l26.level27.netw
00000090	6f 72 6b 00 71 75 61 64	72 61 6e 67 75 6c 61 74	ork.quadrangulat
000000a0	69 6f 6e 00 73 69 6e 67	75 6c 61 6e 67 6c 65 00	ion.singulangle.
000000b0	73 6d 69 6c 65 79 00 74	72 69 61 6e 67 75 6c 61	smiley.triangula
000000c0	74 69 6f 6e 00 00 00 ff	ff 00 78 6c ff 00 b7 ce	tion.....xl....
000000d0	ff 00 ff 00 ff 00 ff ff	ff 02 1f 22 ff 06 48 46"..HF
000000e0	ff 08 16 3d ff 08 d3 3b	ff 0e 41 4a ff 11 77 77	...=...;..AJ..ww
000000f0	ff 19 33 ff ff 1b dd e8	ff 1d 5d 90 ff 25 45 78	..3.....}..%Ex
00000100	ff 26 1e 32 ff 33 66 e5	ff 33 ff 19 ff 34 03 5c	.&.2.3f..3...4.\
00000110	ff 44 74 83 ff 46 91 8a	ff 66 00 00 ff 66 e5 33	.Dt..F...f...f.3
00000120	ff 67 bb 73 ff 78 c4 ad	ff 79 9e bb ff 7d 9f 2f	.g.s.x...y...}/
00000130	ff 7f ab 96 ff 85 0d 4b	ff 88 aa 00 ff 89 1a 25K.....%
00000140	ff 90 be 6d ff 97 23 5f	ff 98 b3 18 ff 99 81 67	...m.#_.....g
00000150	ff a4 0b 5e ff aa 33 55	ff ab e1 75 ff b3 a4 78	...^..3U...u...x
00000160	ff b7 3b 7b ff be d9 56	ff c1 84 2d ff c2 ff cd	...;{...V...-....
00000170	ff c5 32 07 ff c9 7f 99	ff c9 8e a3 ff c9 9d ac	..2.....

Now available on the iOS App Store



Download it now
from the App Store!

chro.mono 2

<https://itunes.apple.com/at/app/chro-mono-2/id1378253641>

Overview

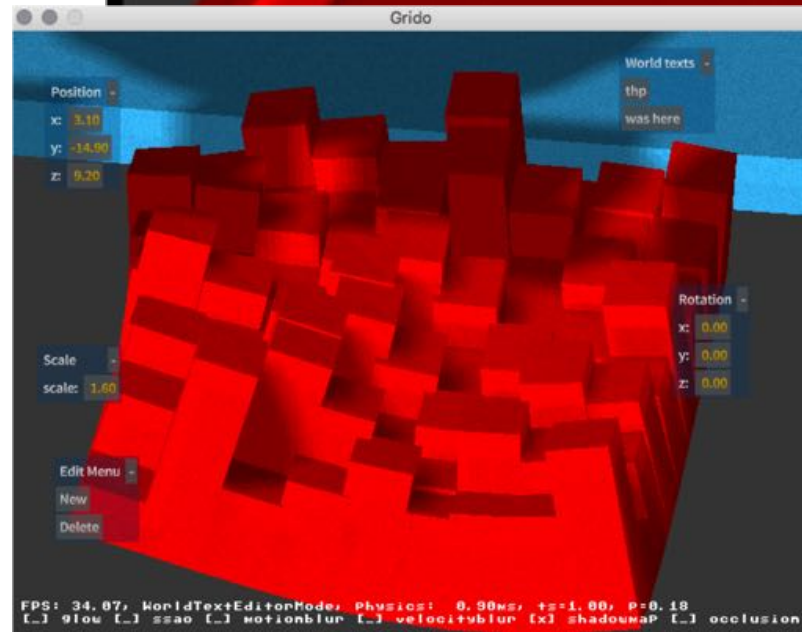
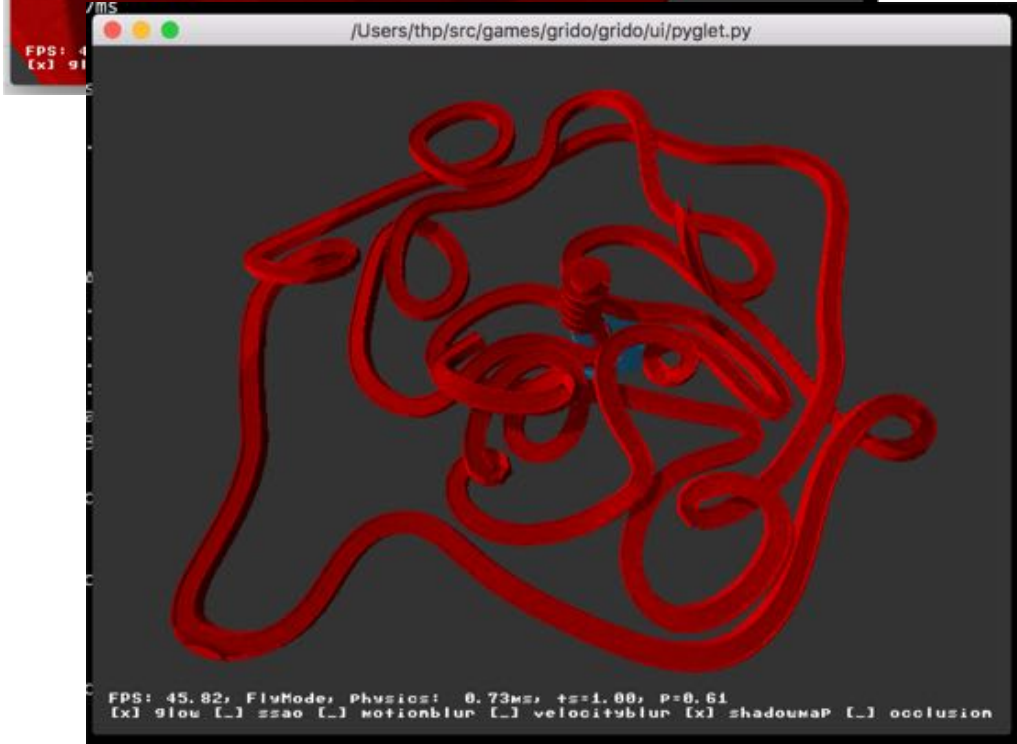
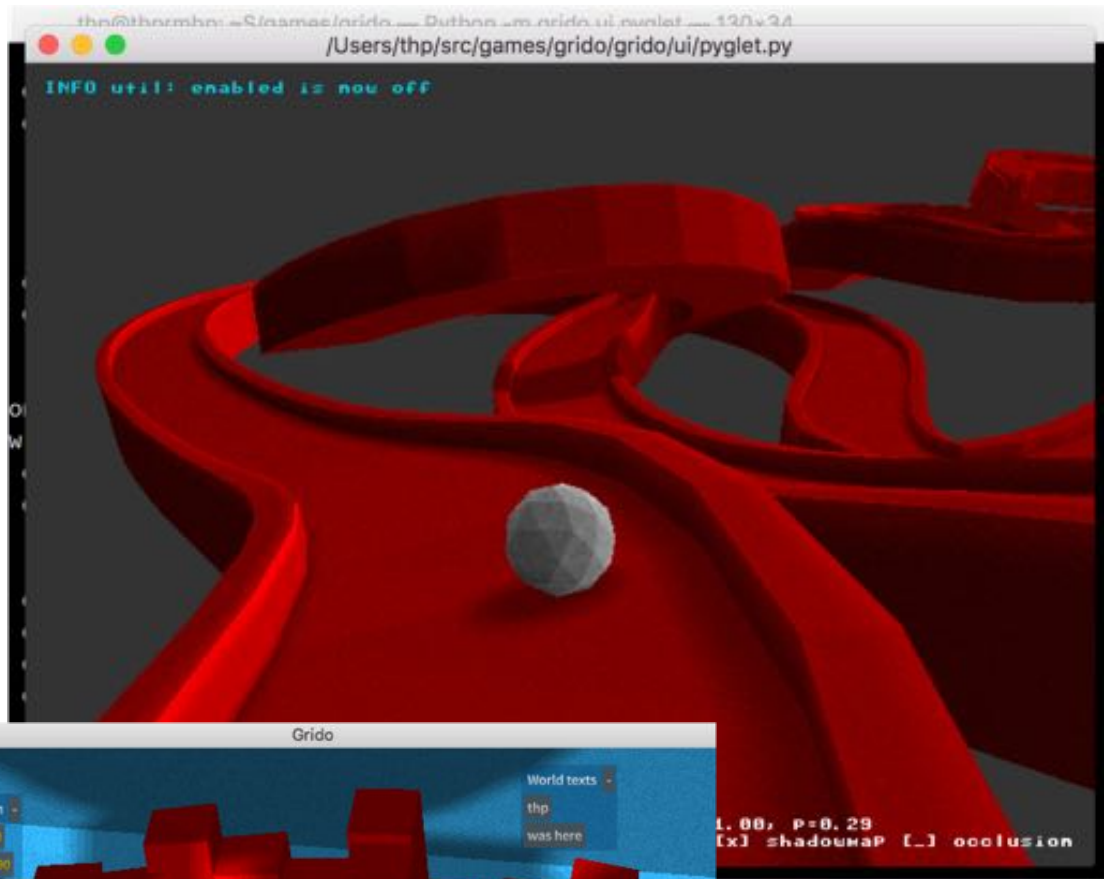
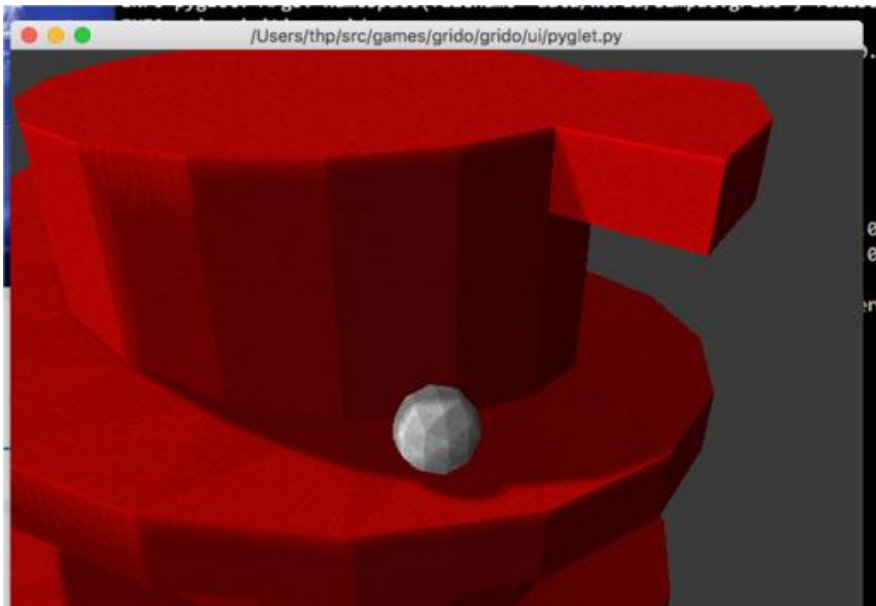
- Consoles and Mobile
- Physical Party Games
- Tooling: Level Editor
- Full 3D Engine

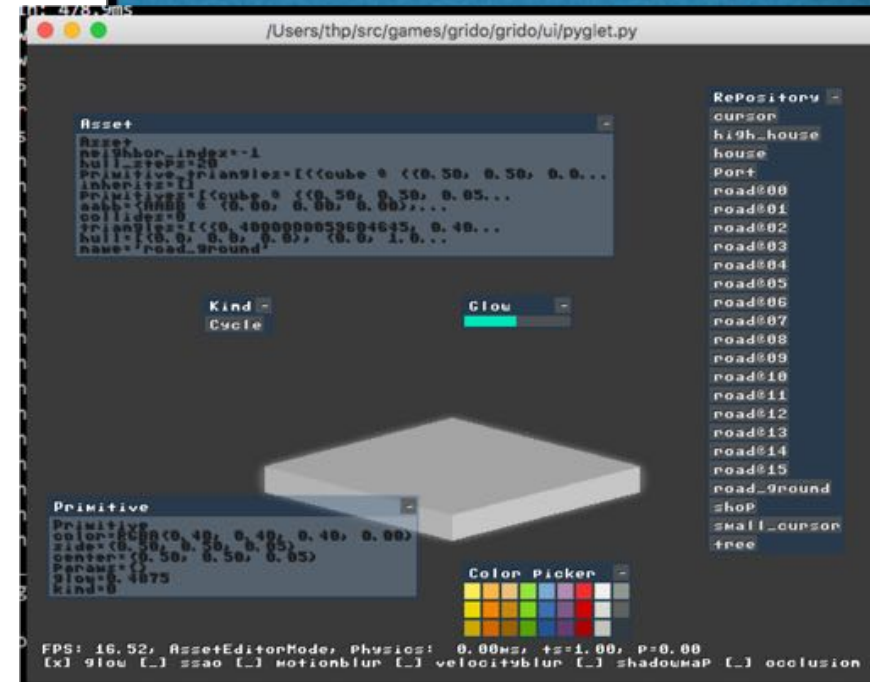
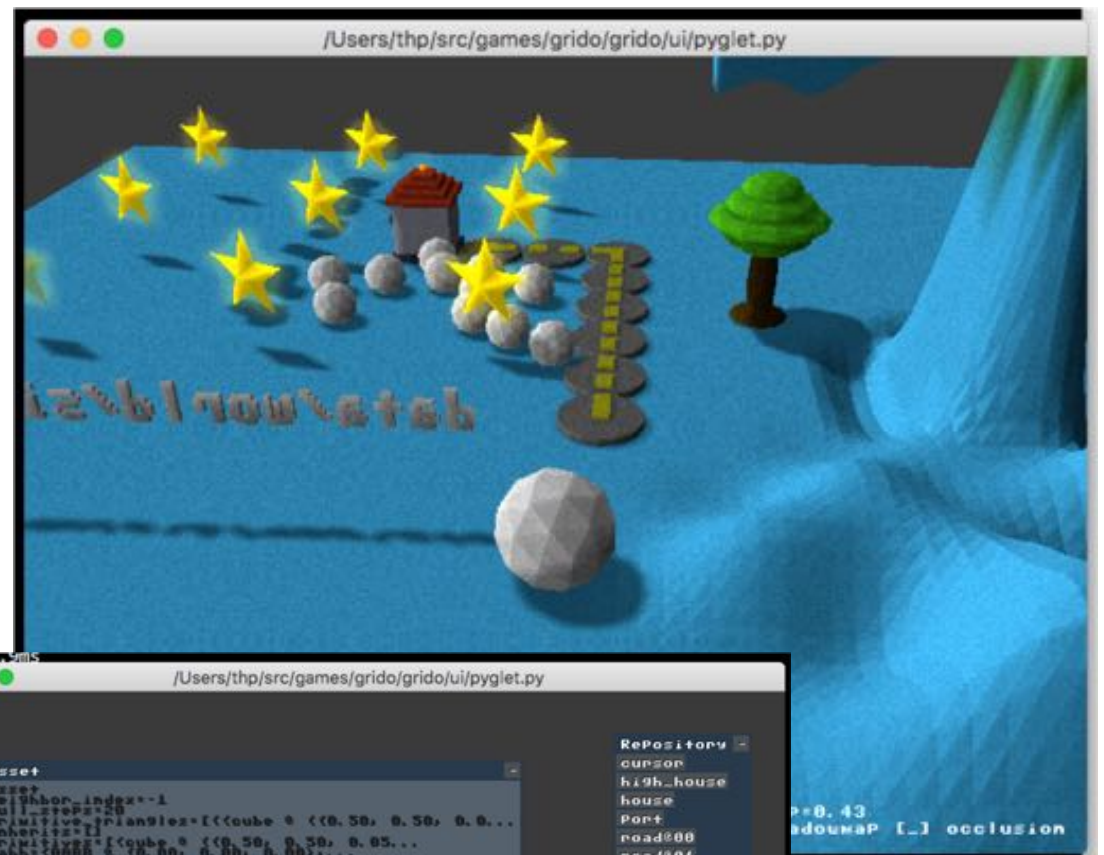
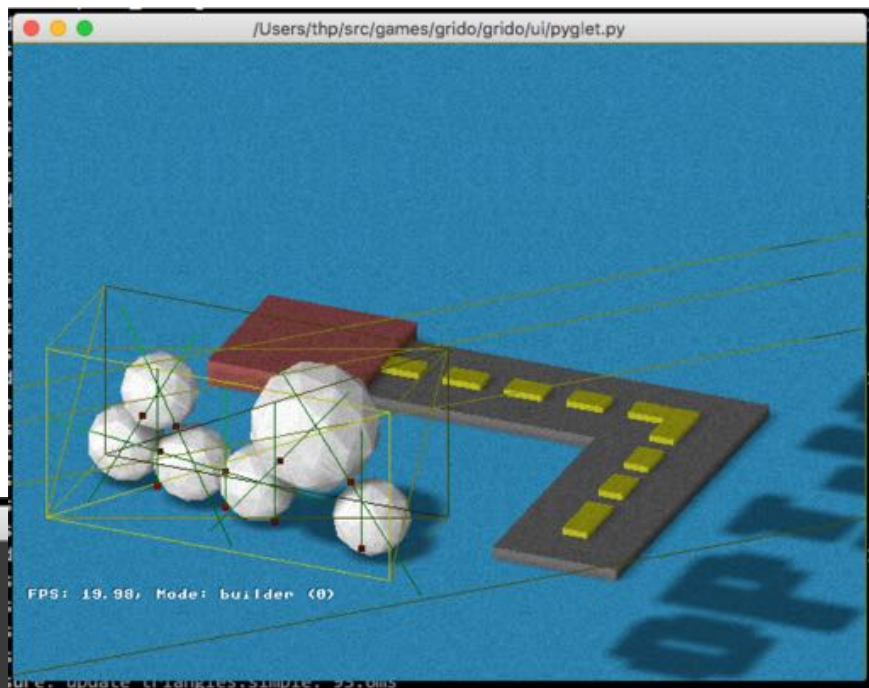


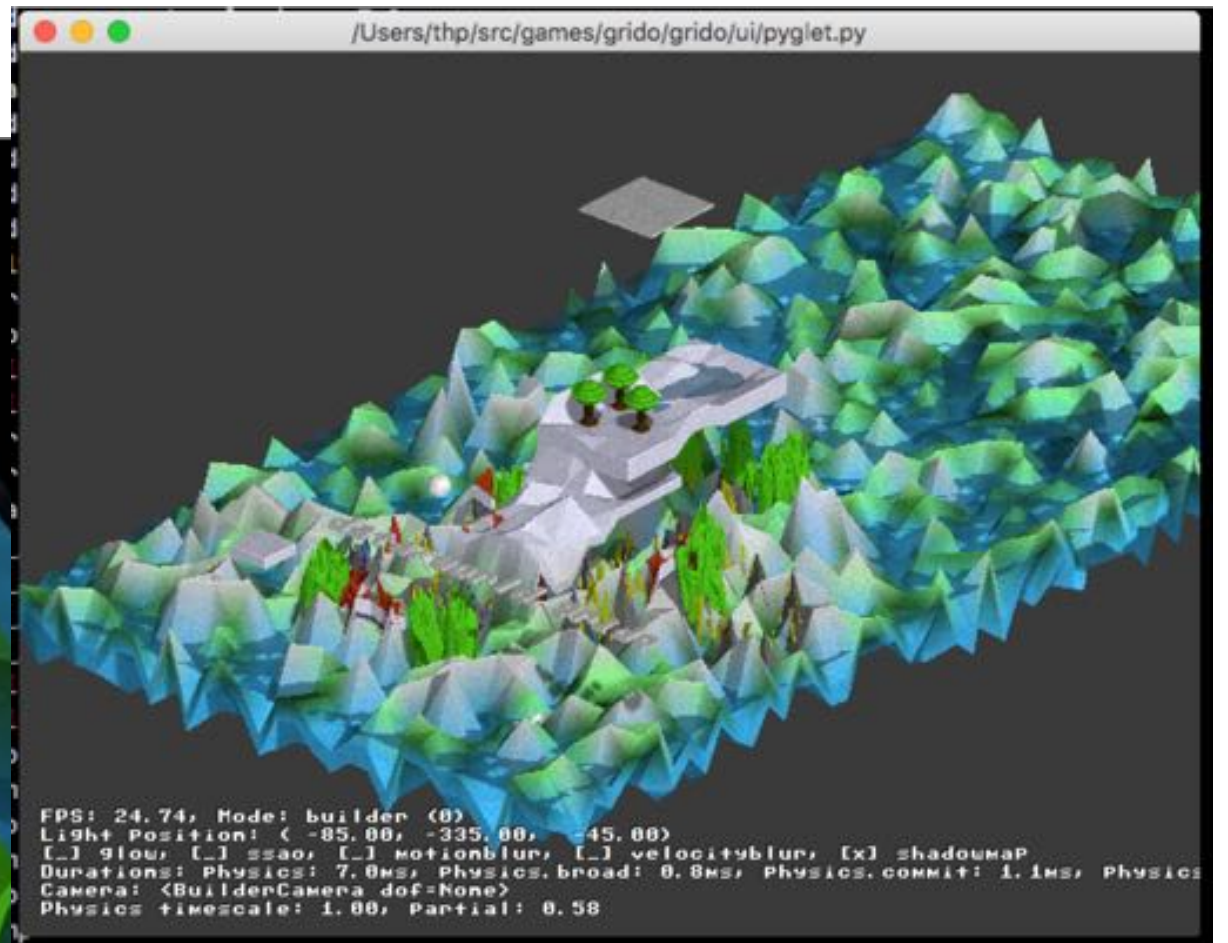
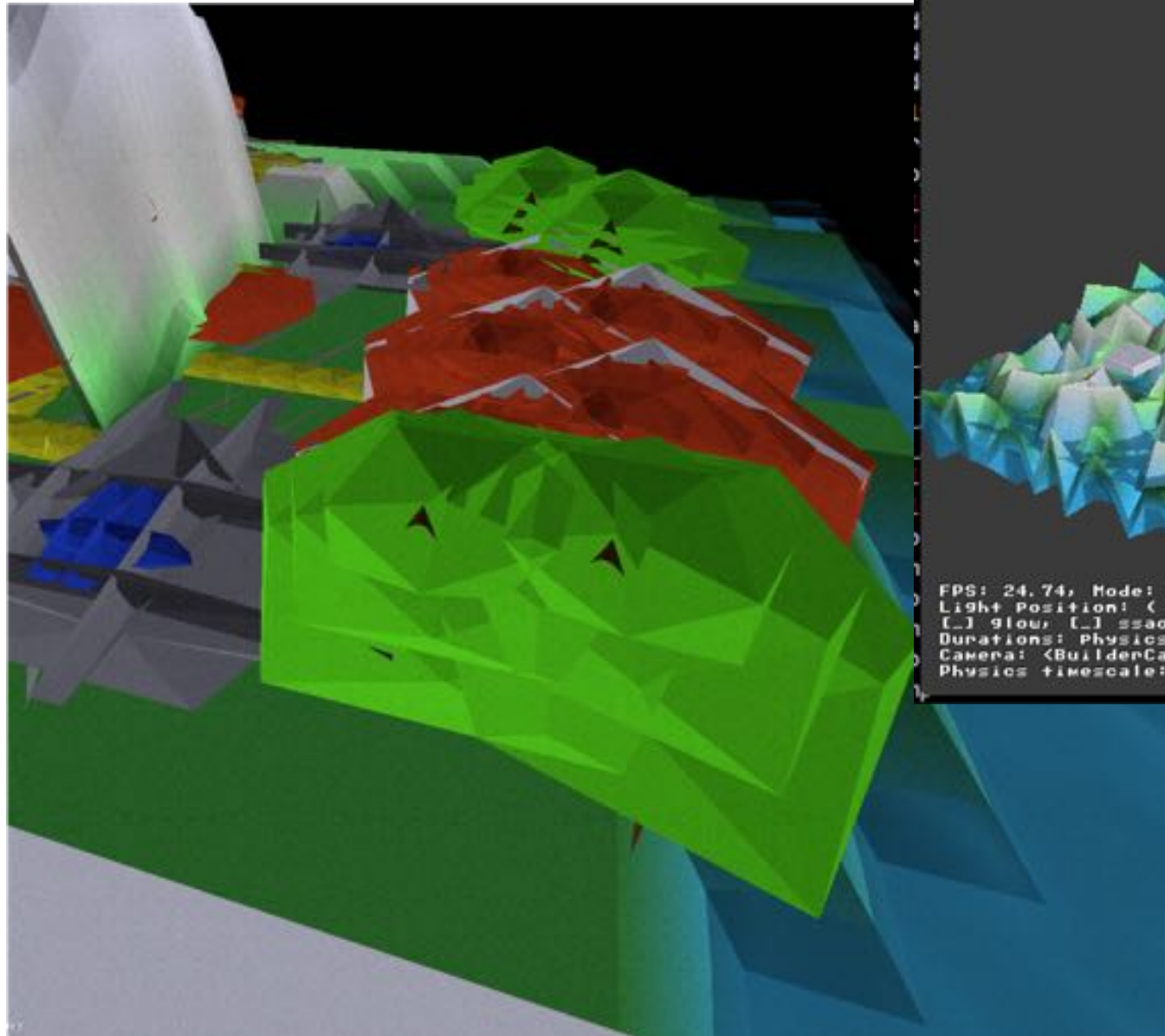
Full 3D Engine

- Try to do "as much as possible" in Python 3
- Physics
- Gamepad Input
- In-Game Editor
- Rendering
- 3D object creation
- Quake-style Python shell
- Modern OpenGL for performance
- Quick iteration









Grido

World texts

data/world/simple.grido

Hello world!

untitled

Scale

scale: 1.00

Rotation

x: -11.60

y: 58.20

z: 15.40

Position

x: 11.00

y: -5.40

z: 0.00

Edit Menu

New

Delete

FPS: 26.93, WorldTextEditorNode, Physics: 2.43ms, vs-1.00, P-0.12
[x] view [x] scene [x] worldAmbient [x] selectEditor [x] shadowMap [x] occlusion

/Users/thp/src/games/grido/grido/ui/pyglet.py

FPS: 26.87, Mode: builder (0)
Light Position: (-85.00, -335.00, -45.00)
[x] glow, [x] dof, [x] ssao, [x] motionblur, [x] velocityblur, [x] shadowMap
Durations: Physics: 7.4ms, Physics.broad: 0.9ms, Physics.commit: 1.2ms, Physics
Physics timescale: 1.00, Partial: 0.50



DepthOfField.distance +

DepthOfField.falloff +

DepthOfField.range +

DepthOfField.kernel +

FPS: 19.85, ThirdPersonMode, Physics: 17.28ms, t=1.00, p=0.27
[x] glow [] ssao [] motionblur [] velocityblur [x] shadowmap [] occlusion

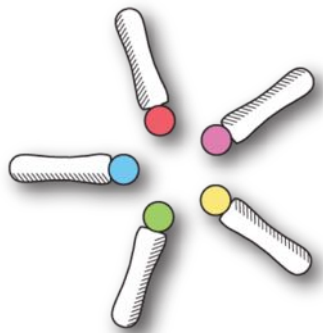
Demo

Python 3D Engine

Python in Game Development

Thomas Perl

m@thp.io · <https://thp.io/> · @thp4



Parley Who Vertigo

Available on
Itch.io

Available on the
iOS App Store



chro.mono 2



One Whale Trip

Available on
Google Play