ADC21

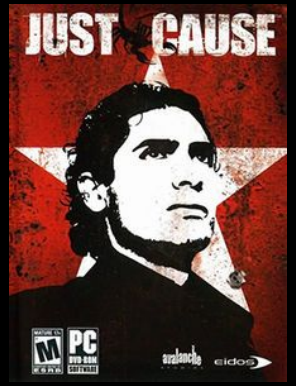
GAME AUDIO:

DATA & CONTEXT FOR RICHER NARRATIVES IN SOUNDTRACKS

XAN WILLIAMS & DOMINIC VEGA

Avalanche
Studios Group

Avalanche Studios Group


JUST CAUSE
PC
avalanche
eidos


JUST CAUSE 2
MATURE 17+
eidos
SQUARE ENIX

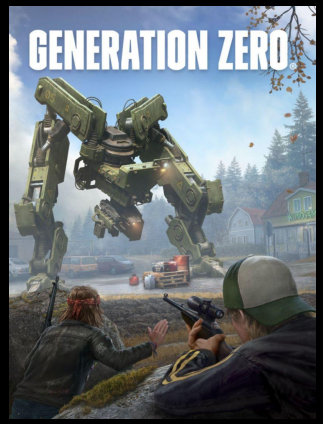

JUST CAUSE 3
PC
SQUARE ENIX

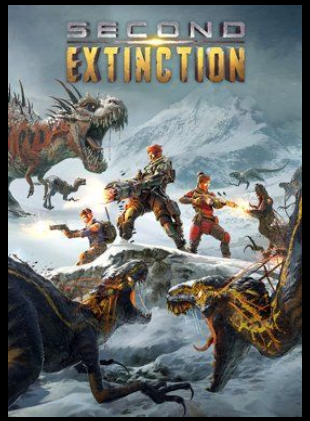

PC
DIGITAL
JUST CAUSE 4
18


MAD MAX
18
PROVISOIRE


RAGE 2

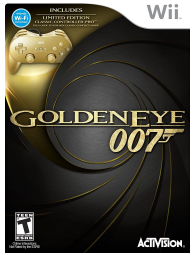

PC DVD
theHunter
CALL OF THE WILD
T
MAXIMUM GAMES


GENERATION ZERO


SECOND
EXTINCTION

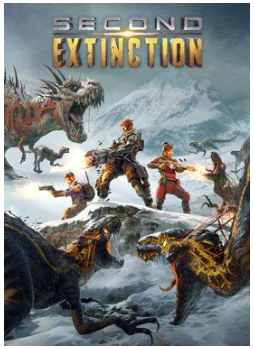
Eurocom Entertainment Software

(2007 - 2012)

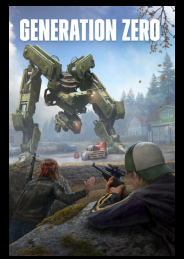


Avalanche Studios Group

(2012 - Present)



(as part of main project team)



(feature sound design & tech sound support)

Avalanche Studios Group

Avalanche
Studios Group

Niclas Frohagen

Principal Sound Programmer

About The Talk

Who’s it for?

If you're new to music design

What we want to tell you...

About the talk

Talk with programmers like we would in the studio

Show some of the methods we use to turn data into narrative

How we think about narrative state

...and why

About the talk

What we want to tell you...

1. See Something in the game

What we want to tell you...

1. See Something in the game
2. We want to put sound to it

What we want to tell you...

1. See Something in the game
2. We want to put sound to it
3. We work out how


CRANE

The Foreshadowing Example

Video: Dino Controller & Foreshadowing Example

The Foreshadowing Example

Avalanche
Studios Group

The Data Bit

Types of Data

The Data Bit

Types of Data

(As categorised by me)

Types of Data

Stuff we **gather**
Stuff we **derive**

Stuff we gather

Types of Data

Go find it!

For example,
Time of Day

Video: Passing Data (ARC)
(time_of_day)

Stuff we gather

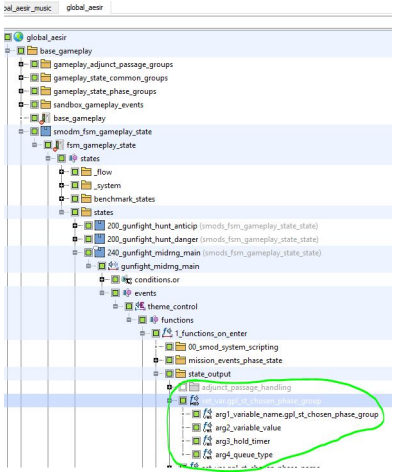
GameSoundUpdate.cpp - OpenG

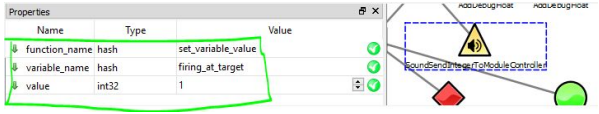
Not secure | opengrok/xref/shoebill-main/source/project/Game/Sound/GameSoundUpdate.cpp#2151

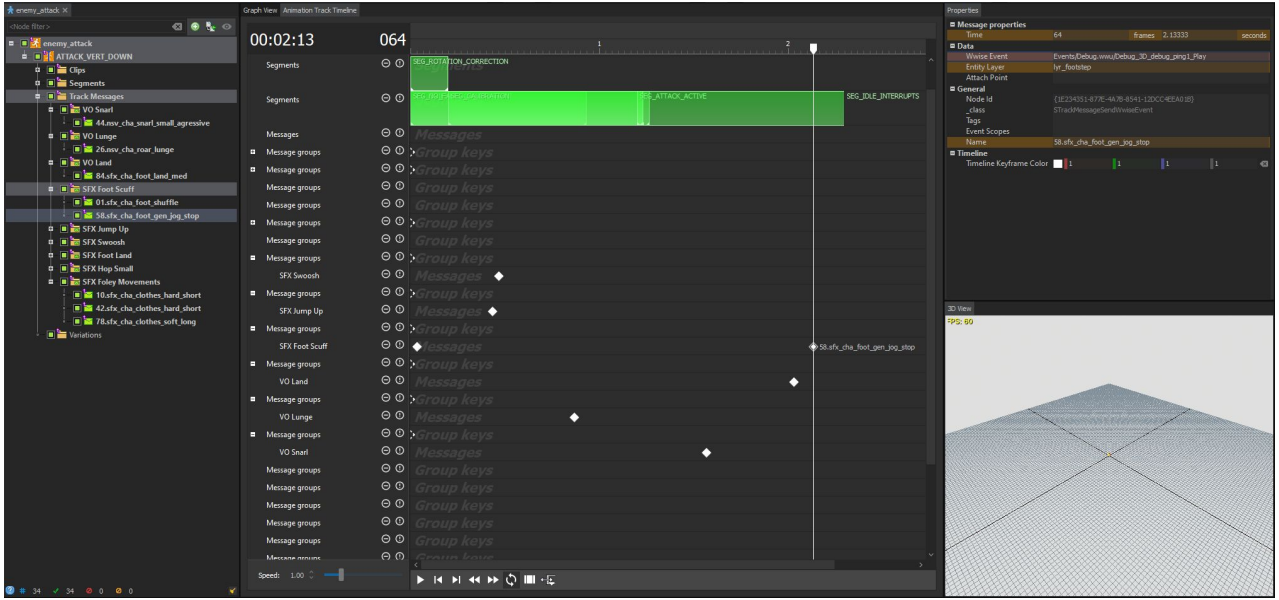
xref: /shoebill-main/source/project/Game/Sound/GameSoundUpdate.cpp

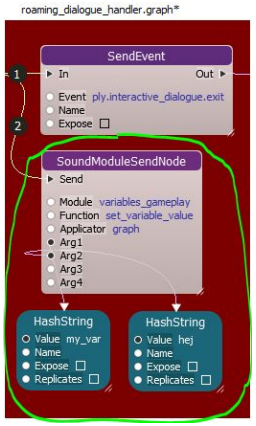
Home | History | Annotate | Line# | Scopes# | Navigate# | Raw | Download Search only in /shoebill-main/so

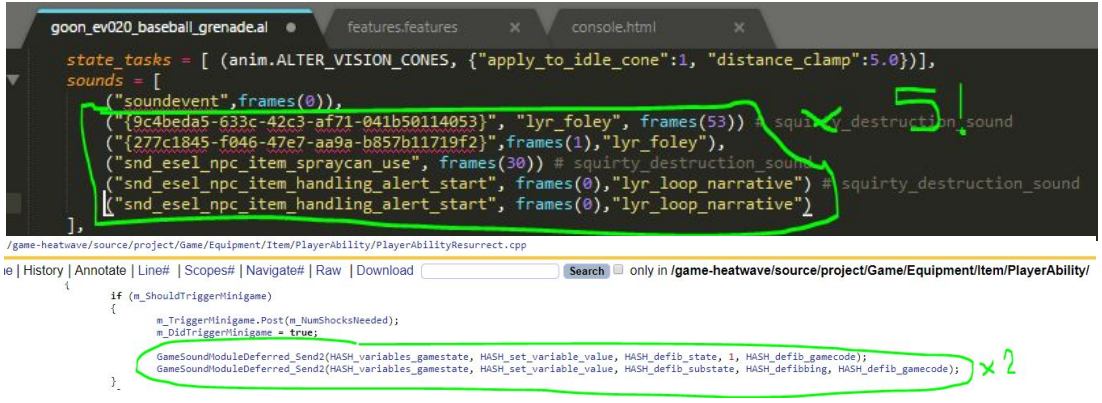
```
2126
2127    #define GAME_SOUND_MODULE_BINDINGS(module) \
2128            static CSoundModule* _##module = 0; \
2129            if (!_##module) \
2130                    _##module = SoundModule_GetModuleFromNameHash(ctx, HASH_##module); \
2131            if (_##module)
2132
2133            void UpdateSoundModuleBindings()
2134            {
2135                    SSoundModuleContext* ctx = GameSoundModule_GetContext();
2136                    if (ctx)
2137                    {
2138                            static CHashString applicator("update_game_world_conditions");
2139
2140                            GAME_SOUND_MODULE_BINDINGS(variables_gameworld)
2141                            {
2142                                    // update time and update speed
2143                                    {
2144                                            ASSERT(CLandscapeManager::Instance != 0);
2145                                            const float time = CWorldTime::Instance->GetCurrentTimeOfDay();
2146                                            ASSERT(Base::CClock::Instance != 0);
2147                                            const float gamespeed = Base::CClock::Instance->GetUpdateSpeed();
2148
2149                                            GameSoundModuleDeferred_Send4(HASH_variables_gameworld,
2150                                                    HASH_set_variable_values,
2151                                                    HASH_time_of_day, time,
2152                                                    HASH_gamespeed, gamespeed,
2153                                                    applicator.get(),
2154                                                    E_SOUND_MODULE_DEFERRED_URGENCY_LEVEL_MEDIUM);
2155                                    }
2156
2157                                    // update rain
2158                                    {
2159                                            ASSERT(CLandscapeManager::Instance != 0);
2160                                            const float rain = CLandscapeManager::Instance->GetPrecipitationDensity(PRECIPITATION_TYPE_
2161                                            GameSoundModuleDeferred_Send2(HASH_variables_gameworld, HASH_set_variable_value, HASH_preci
```


global_aesir
base_gameplay
gameplay_adjunct_passage_groups
gameplay_state_common_groups
gameplay_state_phase_groups
sandbox_gameplay_events
base_gameplay
smodm_fsm_gameplay_state
fsm_gameplay_state
states
_flow
_system
benchmark_states
states
200_gunfight_hunt_anticip (smods_fsm_gameplay_state_state)
200_gunfight_hunt_danger (smods_fsm_gameplay_state_state)
240_gunfight_midmg_main (smods_fsm_gameplay_state_state)
gunfight_midmg_main
conditions.or
events
theme_control
functions
1_functions_on_enter
00_smod_system_scripting
mission_events_phase_state
state_output
adjunct_passage_handling
arg1_variable_name.gpl_st_chosen_phase_group
arg2_variable_value
arg3_hold_timer
arg4_queue_type

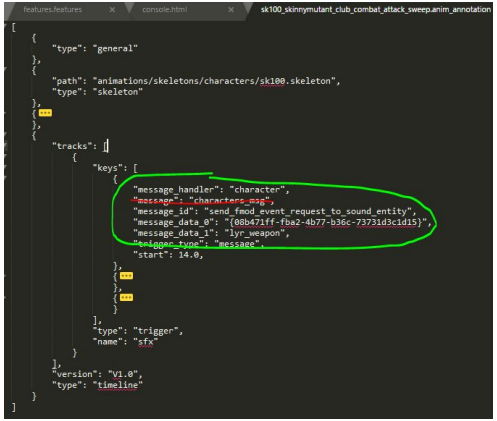

Properties
Name
Type
Value
function_name
hash
set_variable_value
variable_name
hash
firing_at_target
value
int32
1
AddDebugFloat
AddDebugFloat
SoundSendIntegerToModuleController


enemy_attack
ATTACK_VERT_DOWN
Clips
Segments
Track Messages
VO Snarl
44.nsv_cha_snarl_small_agressive
VO Lunge
26.nsv_cha_roar_lunge
VO Land
84.sfx_cha_foot_land_med
SFX Foot Scuff
01.sfx_cha_foot_shuffle
58.sfx_cha_foot_gen_jog_stop
SFX Jump Up
SFX Swoosh
SFX Foot Land
SFX Hop Small
SFX Foley Movements
10.sfx_cha_clothes_hard_short
42.sfx_cha_clothes_hard_short
78.sfx_cha_clothes_soft_long
Variations
Graph View
Animation Track Timeline
00:02:13
064
Segments
SEG_ROTATION_CORRECTION
SEG_ATTACK_ACTIVE
SEG_IDLE_INTERRUPTS
Messages
Group keys
SFX Swoosh
SFX Jump Up
SFX Foot Scuff
58.sfx_cha_foot_gen_jog_stop
VO Land
VO Lunge
VO Snarl
Speed: 1.00
Properties
Message properties
Time
64
frames
2.13333
seconds
Data
Wwise Event
Events/Debug.wwu/Debug_3D_debug_ping1_Play
Entity Layer
lyr_footstep
Attach Point
General
Node Id
{1E234351-077E-4A7B-8541-12DCC4EEA01B}
_class
STrackMessageSendWwiseEvent
Tags
Event Scopes
Name
58.sfx_cha_foot_gen_jog_stop
Timeline
Timeline Keyframe Color
3D View
FPS: 60


roaming_dialogue_handler.graph*
SendEvent
In
Out
Event ply.interactive_dialogue.exit
Name
Expose
SoundModuleSendNode
Send
Module variables_gameplay
Function set_variable_value
Applicator graph
Arg1
Arg2
Arg3
Arg4
HashString
Value my_var
Name
Expose
Replicates
HashString
Value hej
Name
Expose
Replicates


goon_ev020_baseball_grenade.ai
features.features
console.html
state_tasks = [(anim.ALTER_VISION_CONES, {"apply_to_idle_cone":1, "distance_clamp":5.0})],
sounds = [
("soundevent",frames(0)),
("{9c4beda5-633c-42c3-af71-041b50114053}", "lyr_foley", frames(53)) # squirty_destruction_sound
("{277c1845-f046-47e7-aa9a-b857b11719f2}",frames(1),"lyr_foley"),
("snd_esel_npc_item_spraycan_use", frames(30)) # squirty_destruction_sound
("snd_esel_npc_item_handling_alert_start", frames(0),"lyr_loop_narrative") # squirty_destruction_sound
("snd_esel_npc_item_handling_alert_start", frames(0),"lyr_loop_narrative")
],
x 5!

/game-heatwave/source/project/Game/Equipment/Item/PlayerAbility/PlayerAbilityResurrect.cpp
History | Annotate | Line# | Scopes# | Navigate# | Raw | Download
Search
only in /game-heatwave/source/project/Game/Equipment/Item/PlayerAbility/
if (m_ShouldTriggerMinigame)
m_TriggerMinigame.Post(m_NumShocksNeeded);
m_DidTriggerMinigame = true;
GameSoundModuleDeferred_Send2(HASH_variables_gamestate, HASH_set_variable_value, HASH_defib_state, 1, HASH_defib_gamecode);
GameSoundModuleDeferred_Send2(HASH_variables_gamestate, HASH_set_variable_value, HASH_defib_substate, HASH_defibbing, HASH_defib_gamecode);
x 2


features.features
console.html
sk100_skinnymutant_club_combat_attack_sweep.anim_annotation
"type": "general"
"path": "animations/skeletons/characters/sk100.skeleton",
"type": "skeleton"
"tracks":
"keys":
"message_handler": "character",
"message": "characters.msg",
"message_id": "send_fmod_event_request_to_sound_entity",
"message_data_0": "{08b471ff-fba2-4b77-b36c-73731d3c1d15}",
"message_data_1": "lyr_weapon",
"message_type": "message",
"start": 14.0,
"type": "trigger",
"name": "sfx"
"version": "V1.0",
"type": "timeline"

Balancing Requests

Crash Bug
vs.
3 day task

Crash Bug
vs.
30 minute task

Single line to push data to us

Stuff we gather

=

Smallest time per task possible

Stuff we gather

🚫 “Expensive” tasks.

If it’s only for us, then it’s an Audio task.

Avalanche
Studios Group

Stuff we derive

Types of Data

Why?

Single source of truth?

Absolutely.

But...

Many perspectives

Avalanche Studios Group

The Player's sense of Combat

🚫 The omniscient combat state

Many perspectives

Bringing us to...

The State Bit

(stuff we derive, cont.)

The “State Machine”

The State Bit

The “State Machine”

How?

Video: “State Machine”

The “State Machine”

States are self-evaluating

The “State Machine”

TRUE Priority 2
TRUE Priority 5
FALSE Priority 8

The “State Machine”

~~TRUE~~ ~~Priority 2~~

TRUE Priority 5

~~FALSE~~ ~~Priority 8~~

The “State Machine”

Variables > Value

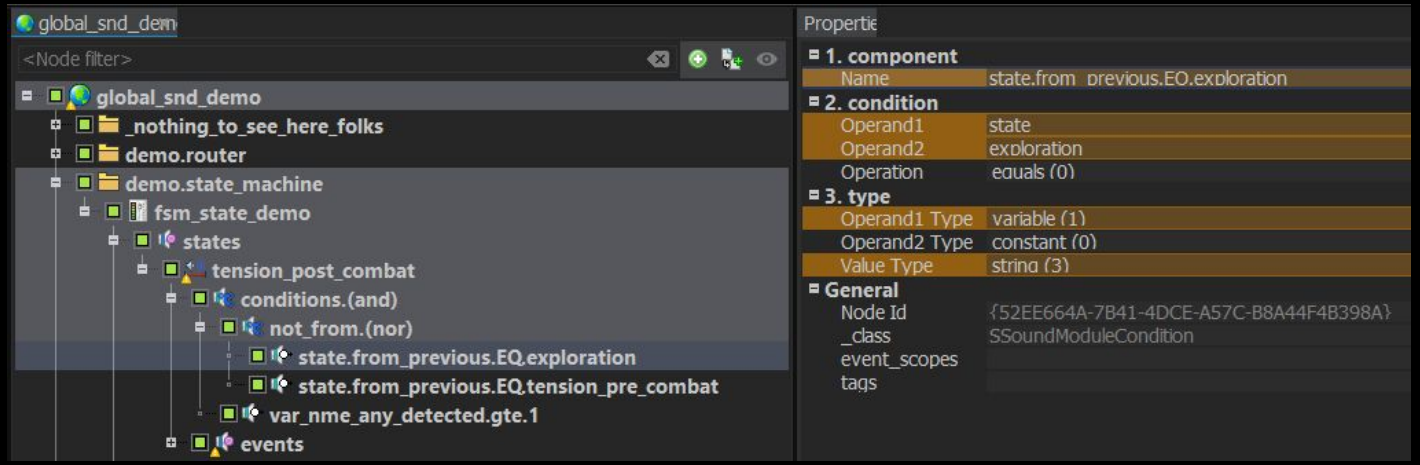
...

```
router - variables_demo (using filter=[;])
+ pause counter : -1
+ variables:
  + time_of_dawn: value=6.000 (previous=5.000) (max=6.000)
  + time_of_day: value=17.000 (previous=15.000) (max=17.000)
  + time_of_dusk: value=18.000 (previous=14.500) (max=18.000)
```

The “State Machine”

➡️ extra data **derived**

➡️ **better** logic



The “State Machine”

IF

(enemies > 1
& state.previous == combat)

Then

post_combat_tension 🎶

The “State Machine”

Logic helps arrange consecutive state

Logic helps contextualise data

gunfight_combat

What is kind of scripting is this..?

The State Bit

What is this..?

Good **Question** 🤔🤷

Decision Tree?

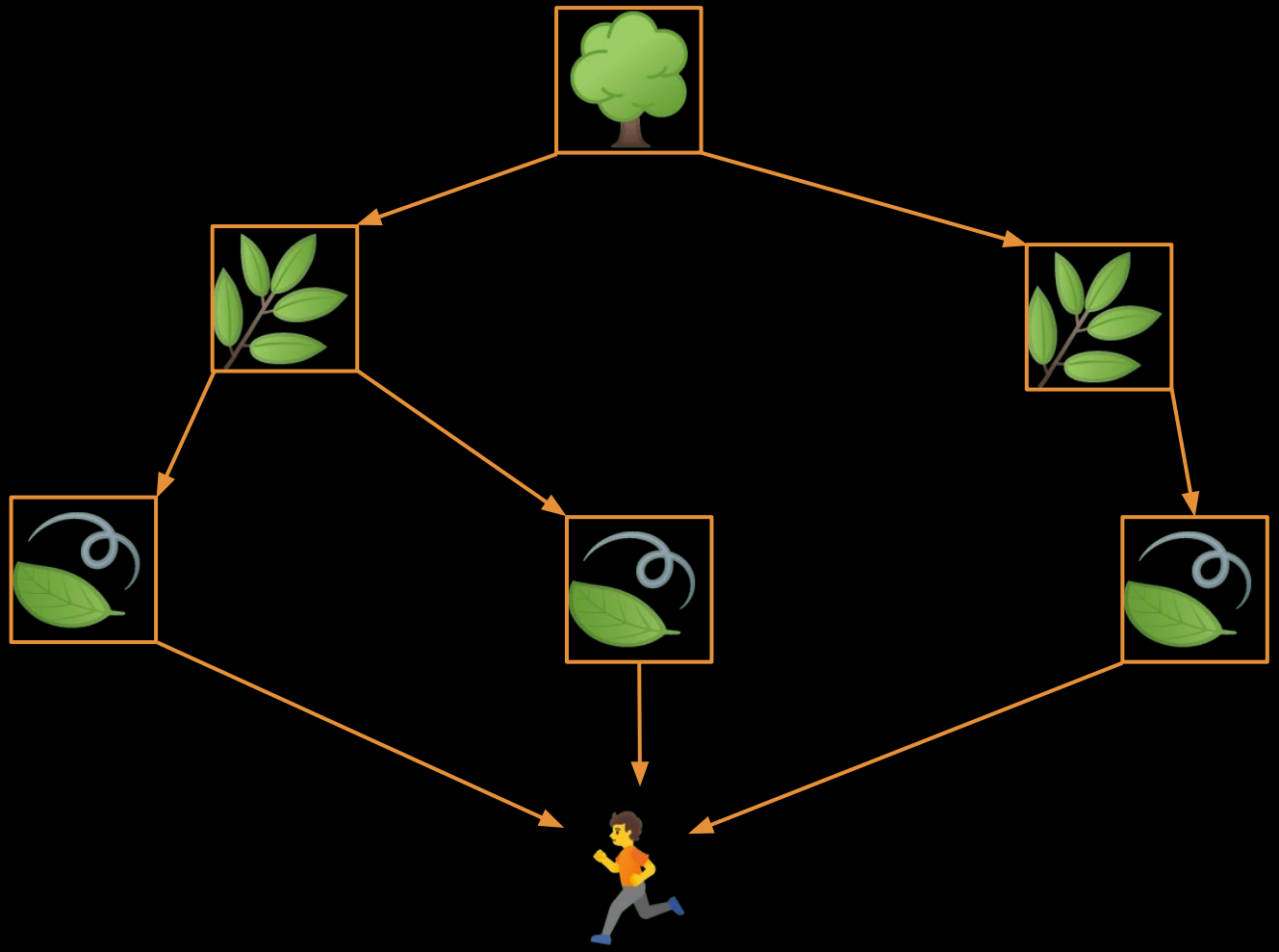
The State Bit

Multiple state machines are used together

What type of decision tree ?

https://web.stanford.edu/class/cs123/lectures/CS123_lec08_HFSM_BT.pdf

2.4 Hierarchical Finite State Machine (HFSM) & Behavior Tree (BT)



Decision Tree

- ✔ Maintainability
- ✔ Scalability
- ✔ Reusability

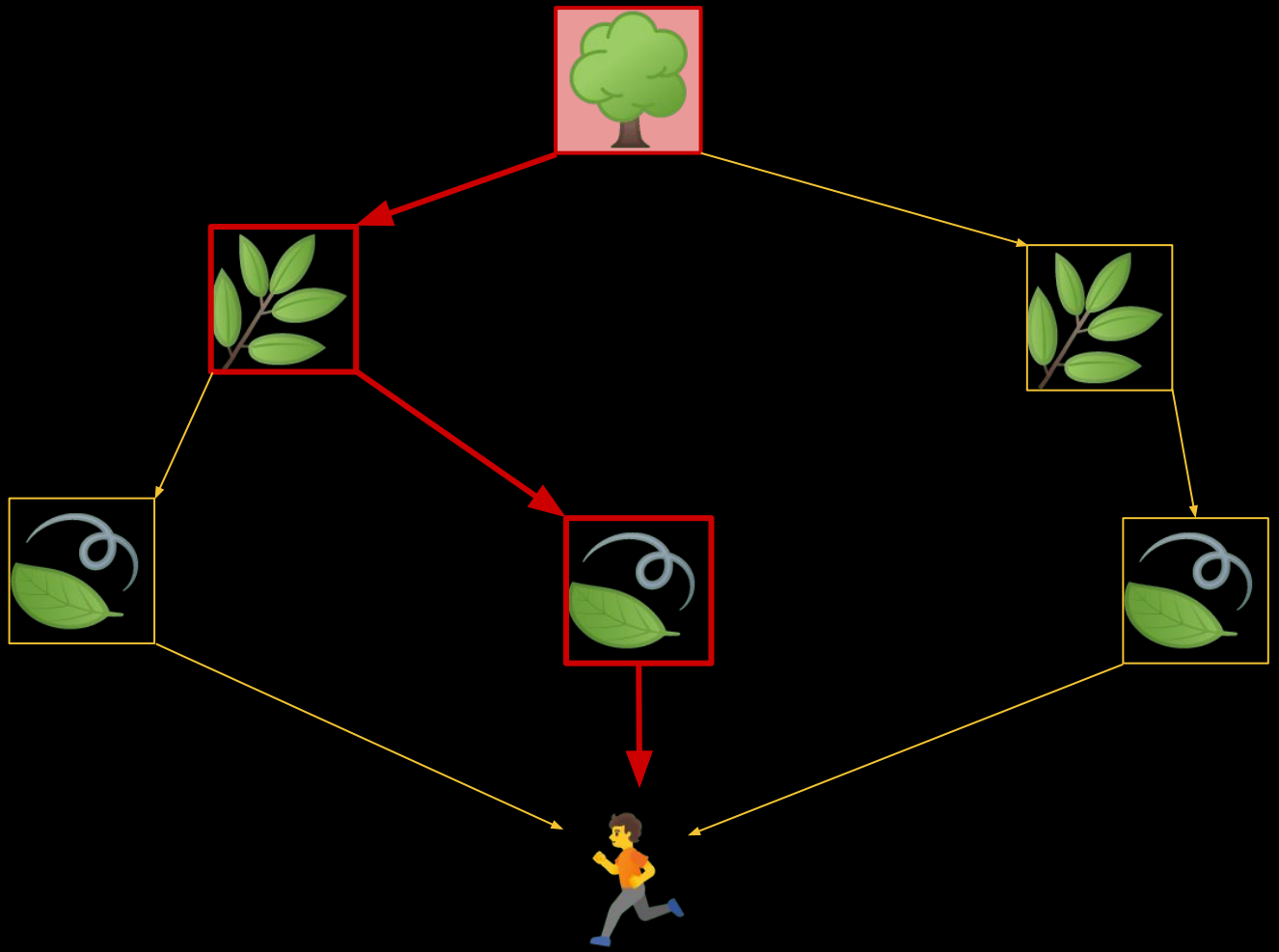
❌Transitions

(no leaving leaf nodes)

https://web.stanford.edu/class/cs123/lectures/CS123_lec08_HFSM_BT.pdf

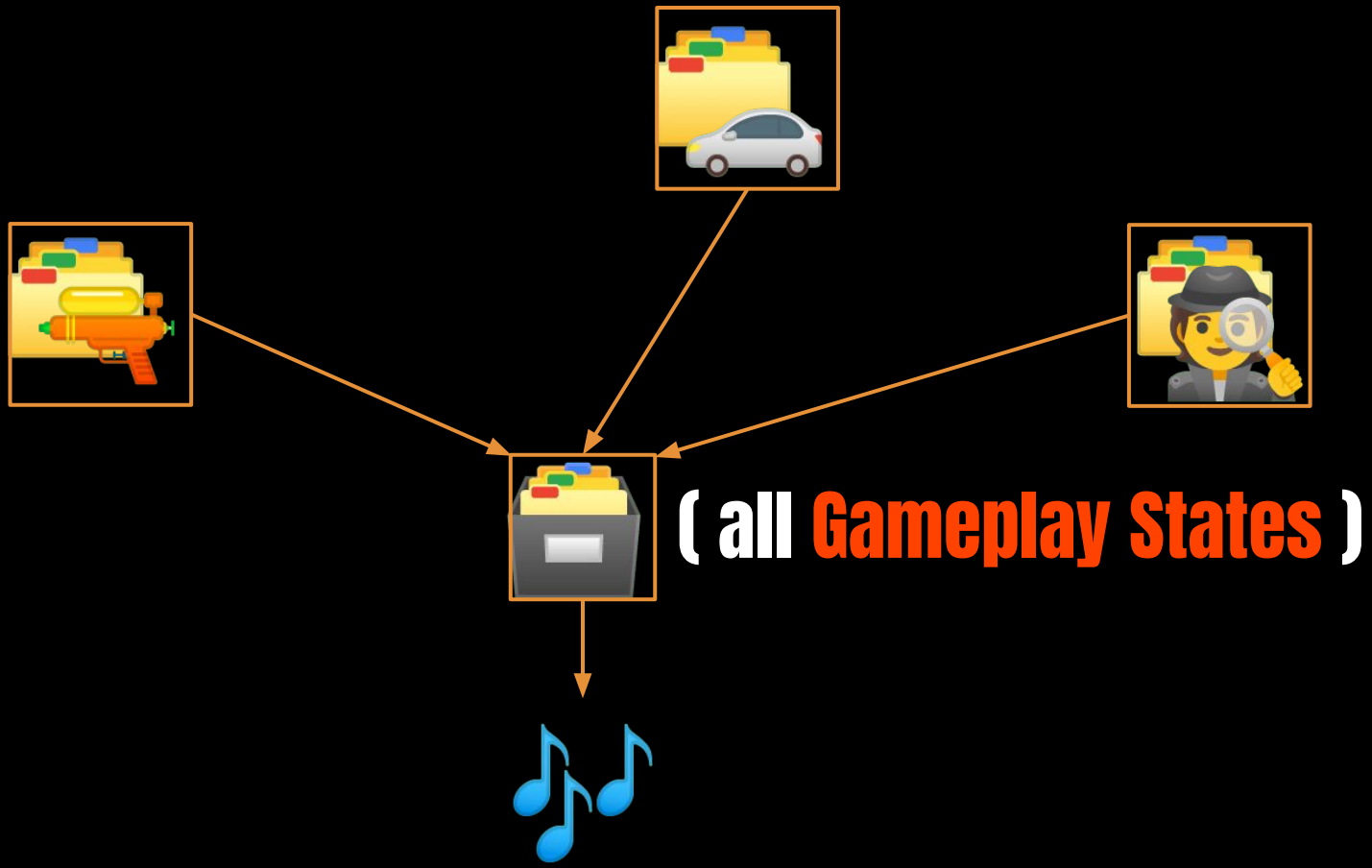
Tree Traversal Issue

- Always start from root node
- This isn't a very efficient way to do things, especially when the behavior tree gets deeper as its developed and expanded during development.
- Store any currently processing nodes so they can be ticked directly within the behavior tree engine rather than per tick traversal of the entire tree

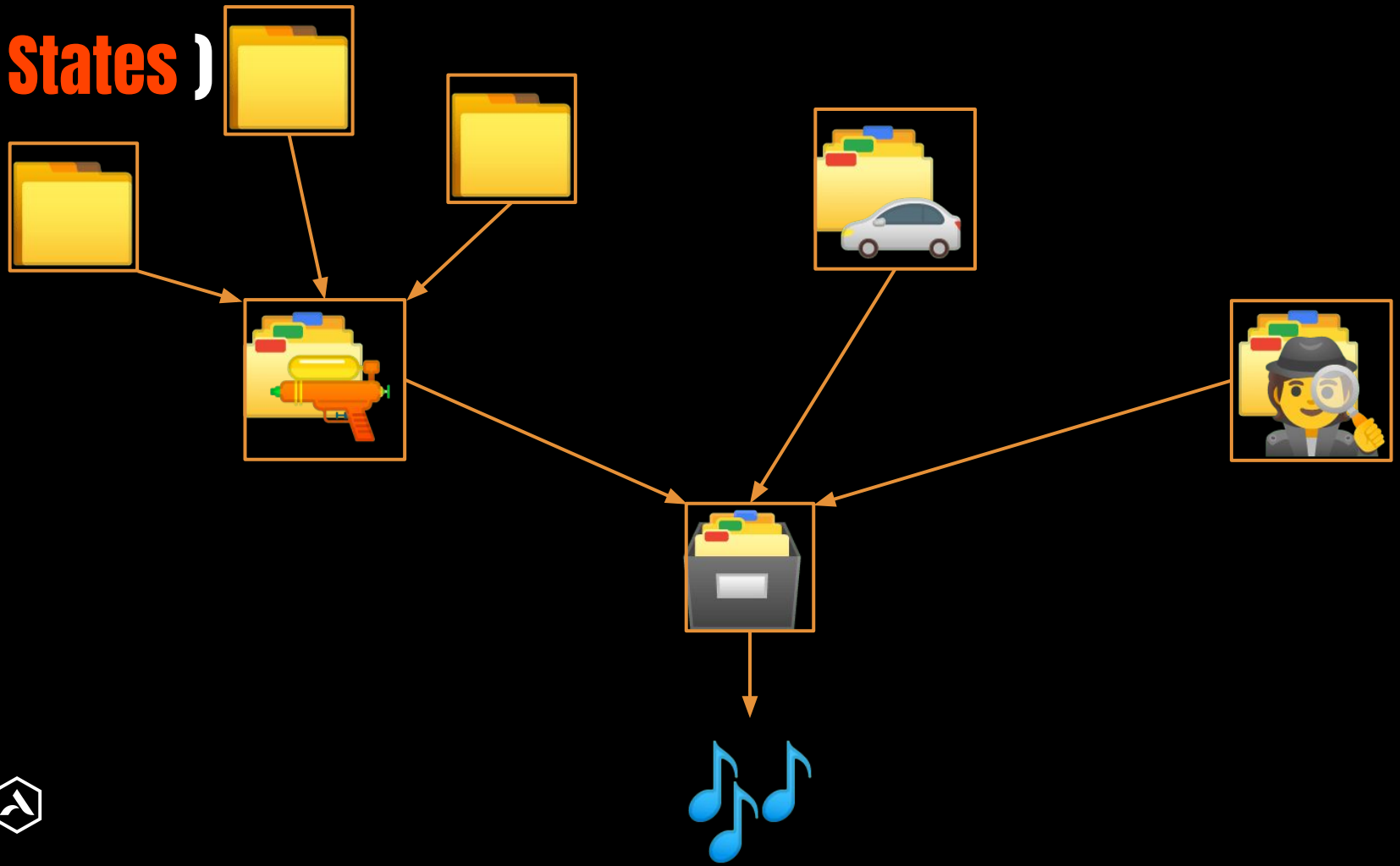


Similar but different

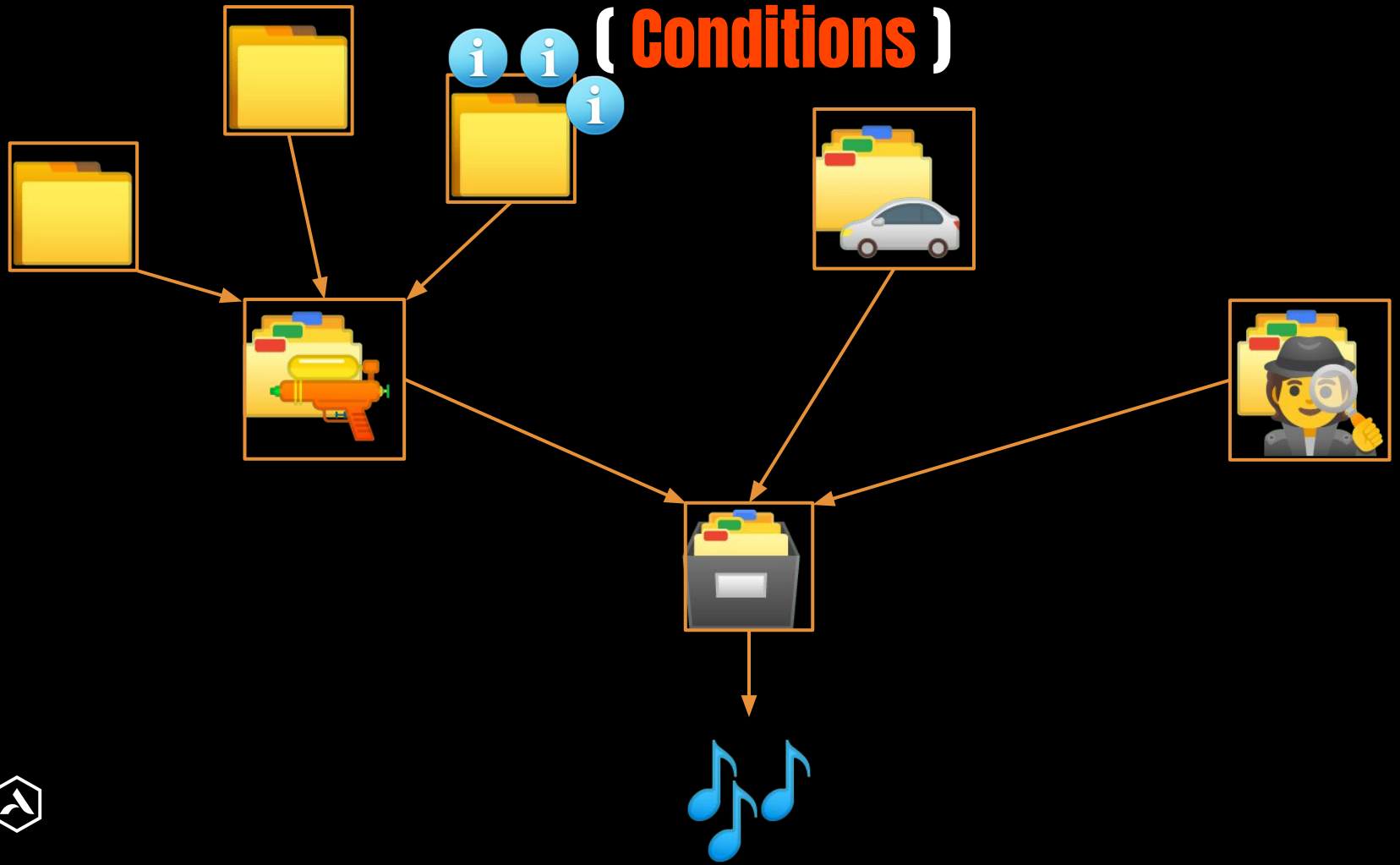
(Gameplay Type State Machines)


(all Gameplay States)

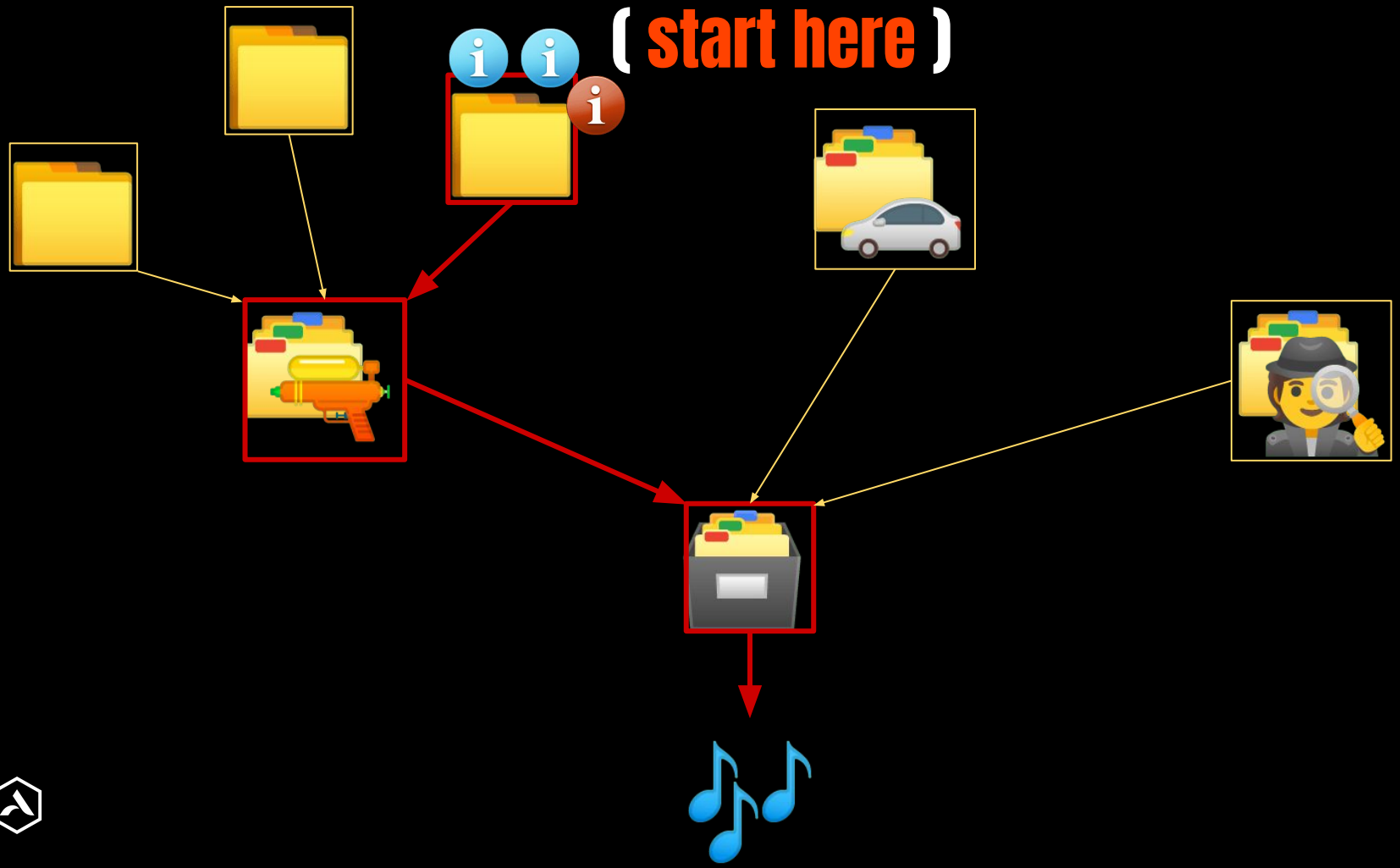
(States)



(Conditions)



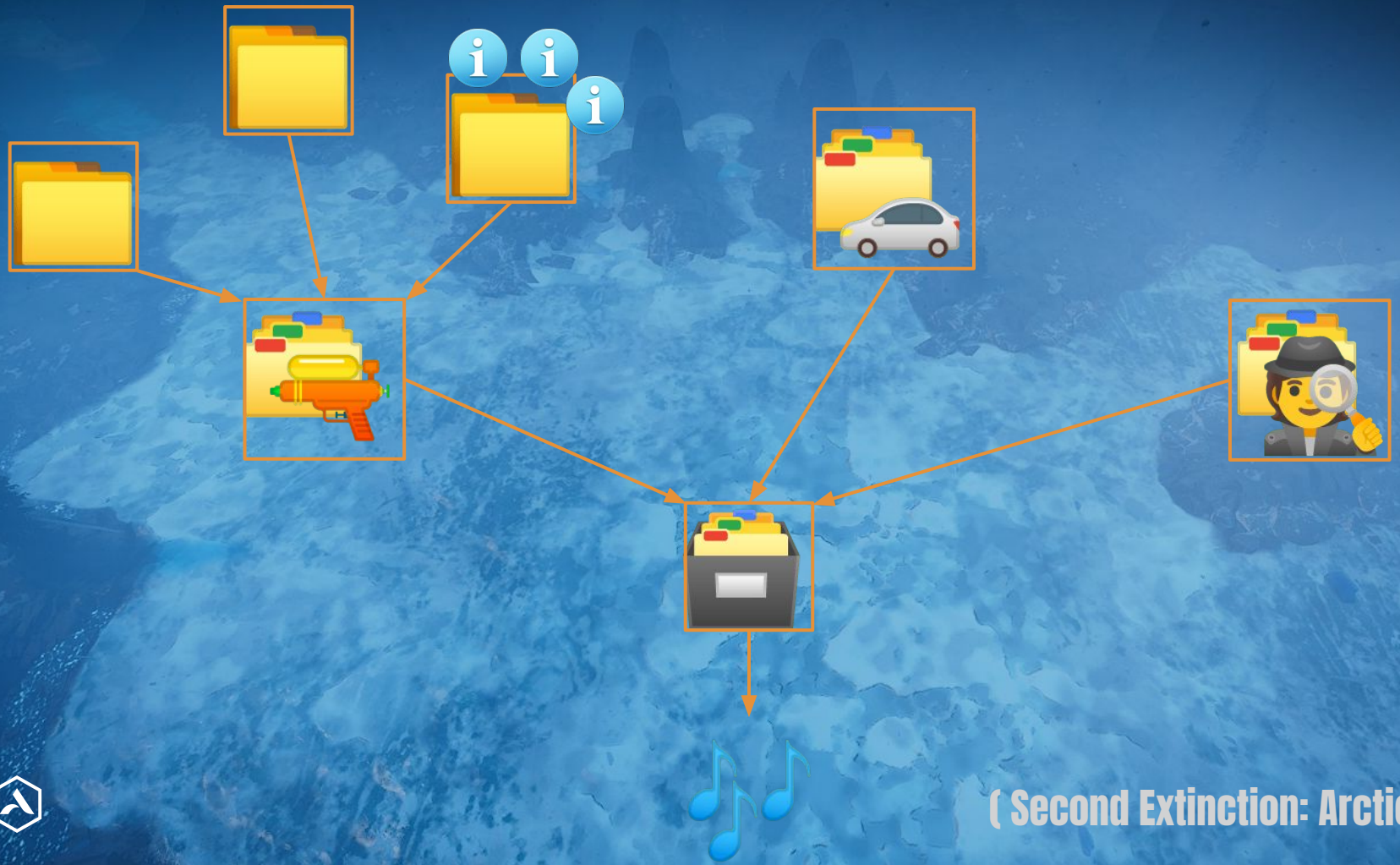
Execution?


(start here)

Decision Tree

🌳 **Root** ❌

🌊 Data Tributaries ✔️



(Second Extinction: Arctic River)

Inheritance

The State Bit

https://web.stanford.edu/class/cs123/lectures/CS123_lec08_HFSM_BT.pdf

Behavior Inheritance

- HFSM combines hierarchy with programming-by-difference, which is otherwise known in software as *inheritance*.
- As class inheritance allows subclasses to *adapt* to new environments, behavioral inheritance allows substates to *mutate* by adding new behavior or by overriding existing behavior.
- State nesting introduces another fundamental type of inheritance, called *behavioral inheritance*

Templating Activities & Logic

Behavioural Inheritance

smodm_gpl_st_activity_owenc

<Node filter>

- smodm_gpl_st_activity_owenc
  - smodm_gpl_st_activity_gunfight (smodm_gpl_st_activity_gunfight)
    - smods_fsm_gpl_state_group_base
      - debug_visualisation (smodc_graph_fsm_setup)
      - gpl_st_activity_owenc
        - states
          - 000.none (smods_gpl_st_active_state)
          - 110.gfafter.(gfdanger).nme_area_unoccupied (smods_gpl_st_active_state)
          - 150.gfafter (smods_gpl_st_active_state)
          - 175.gfafter.camp_complete_screen (smods_gpl_st_active_state)
          - **200.gfdanger.base (smods_gpl_st_active_state)**
          - 210.gfdanger.nme_area_occupied (smods_gpl_st_active_state)
          - **300.gfanticip.base (smods_gpl_st_active_state)**
          - 310.gfanticip.nme_area (smods_gpl_st_active_state)
          - **450.gfcomb (smods_gpl_st_active_state)**
            - gfcomb
              - conditions.OR
                - continuous.detect_any_presence.AND
                - enemy_types.OR
                - presence_of_onfoot_enemies.and
                - presence_of_vehicle_enemies.and (smods_cond_activity_g
                - enemies.vehicles.(AND)
                - enemies.or
                - player_slow (smods_cond_validation_enemy_vs_trai
                - inclusion.validation_enemy_vs_transport.OR
                - state.EQ.this_(gfcomb)
              - enter.from_any.OR
              - lovar_debug_state.eq.this_(gfcomb)
            - events
          - 500.gfclimax (smods_gpl_st_active_state)
          - 999.blocked (smods_gpl_st_active_state)
        - transitions
        - variables

Properties

Transform

Attach Max Tilt	0
Attach To Ground	☐
Attach To Mesh	☐
Attach To Water	☐
Transform	0 0 0 / 0 0 0
Transform Rotation	0 0 0
Transform Translation	0 0 0

General

Event Scopes	
Name	450.gfcomb
Node Id	{6F8EED58-82B0-4823-B344-5ED9258
Tags	
_class	_entityinstance

State

Module Name	gpl_st_activity_owenc
Priority	450
State Name	gfcomb

State Selection Output

Use Group Variable	yes (1)
Use State Variable	no (0)
state_phase_group_module	gpl_st_activity_gunfight
state_phase_group_value	gfcomb
state_phase_group_variable	activity_owenc

State Timers

Use State Timers	count (1)

ai

Navmesh Merge	None (0)

fmod_bank

Add Referenced Fmod Banks	☑

random

Random Mode	None (0)
Random Probability	1

Multiple state machines talk to each other

Hierarchical FSM

Inheritance

- ✔ Maintainability
- ✔ Scalability
- ✔ Reusability

✔ Transitions

(leaving leaf nodes)

Inheritance

❌ Visual:
(Readability)
(Learnable 🌎🧠)

Avalanche
Studios Group

Good Question: *"You breezed past Transitions"*

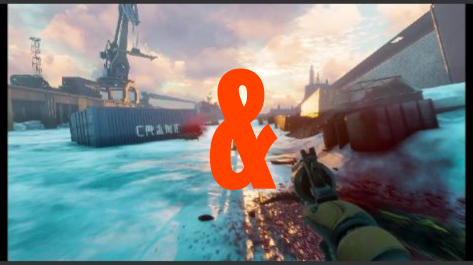
(and other scripting tricks)

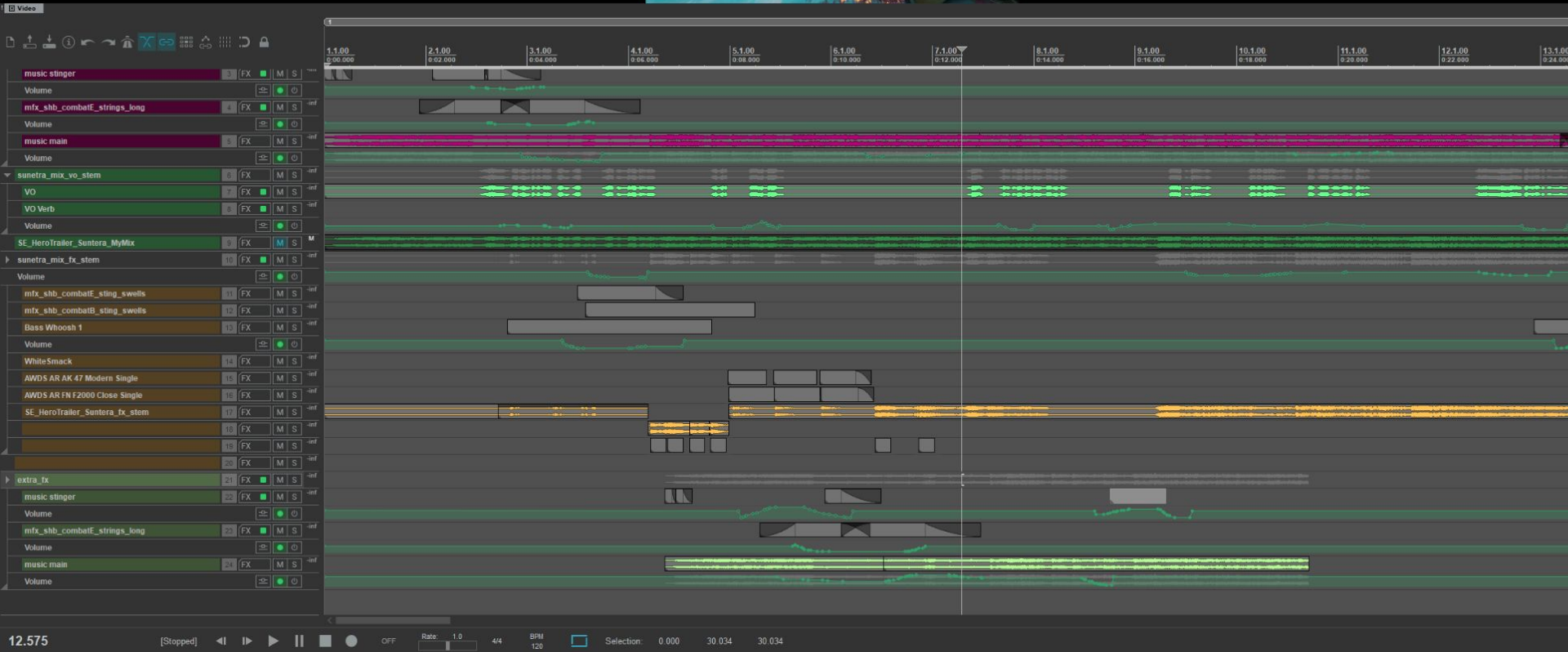
Q & A Bait 🎣

State as Context, Automation

The State Bit

Tracklaying & Automation





State as Context, Automation

Tracklaying ➡️ **Triggering**

(sequencing & editing) (implementation)

Automation ➡️ **State**

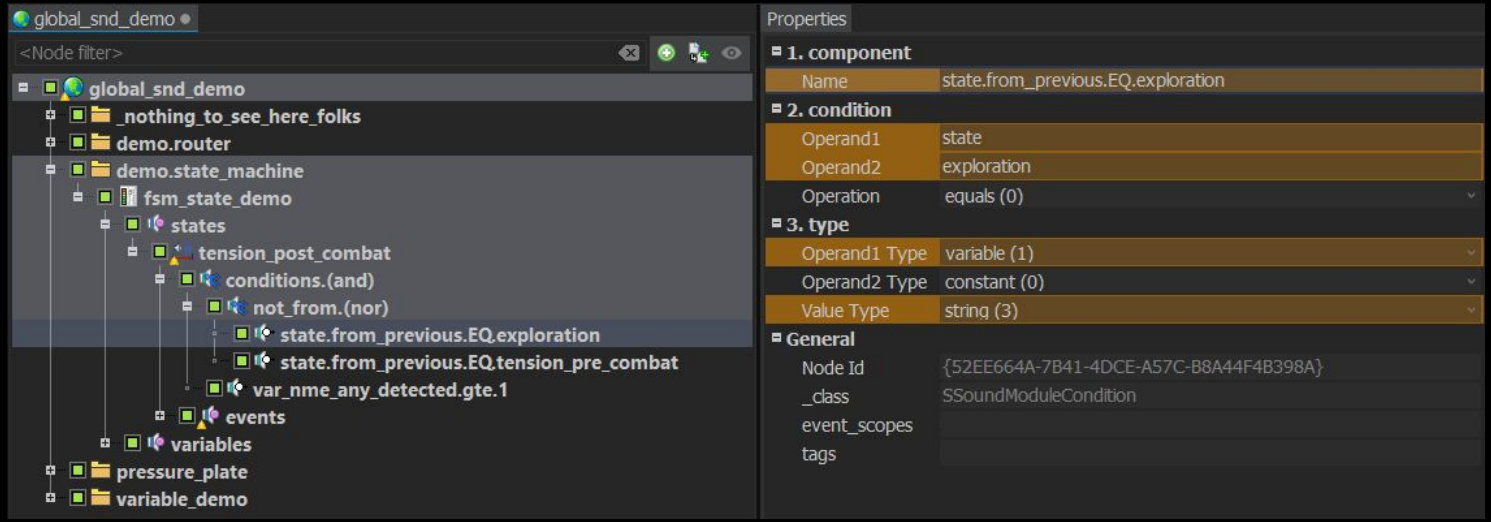
(mixing / mastering) (culling / mixing / modulation)

Sequences of Regions and Automation Curves

Narrative Context for Sounds

State as Context, Automation

pssst
remember the pre- and post- combat tension?



State as Context, Automation

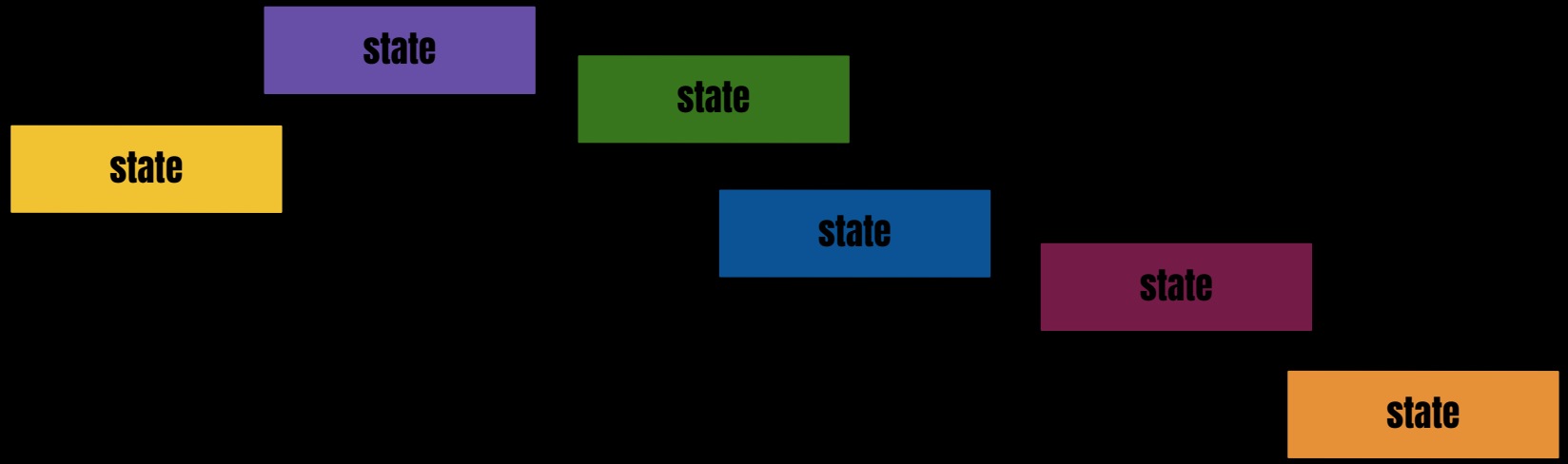
$$\frac{\text{Narrative}}{\text{Time}}$$

State as Context, Automation

State as Context, Automation

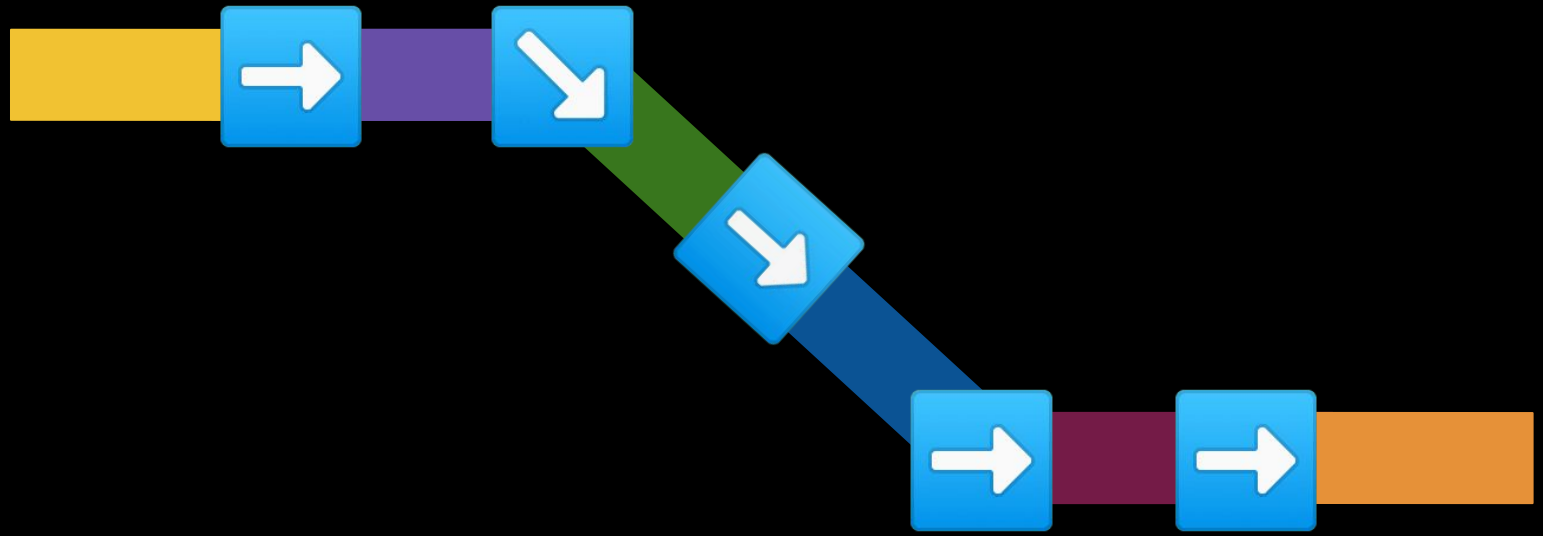
⏳ Beginning ❓ Middle ❓ End ⏳

State as Context, Automation



⏳ Consecutive State ❓

State as Context, Automation



⏳ Consecutive State ➡️

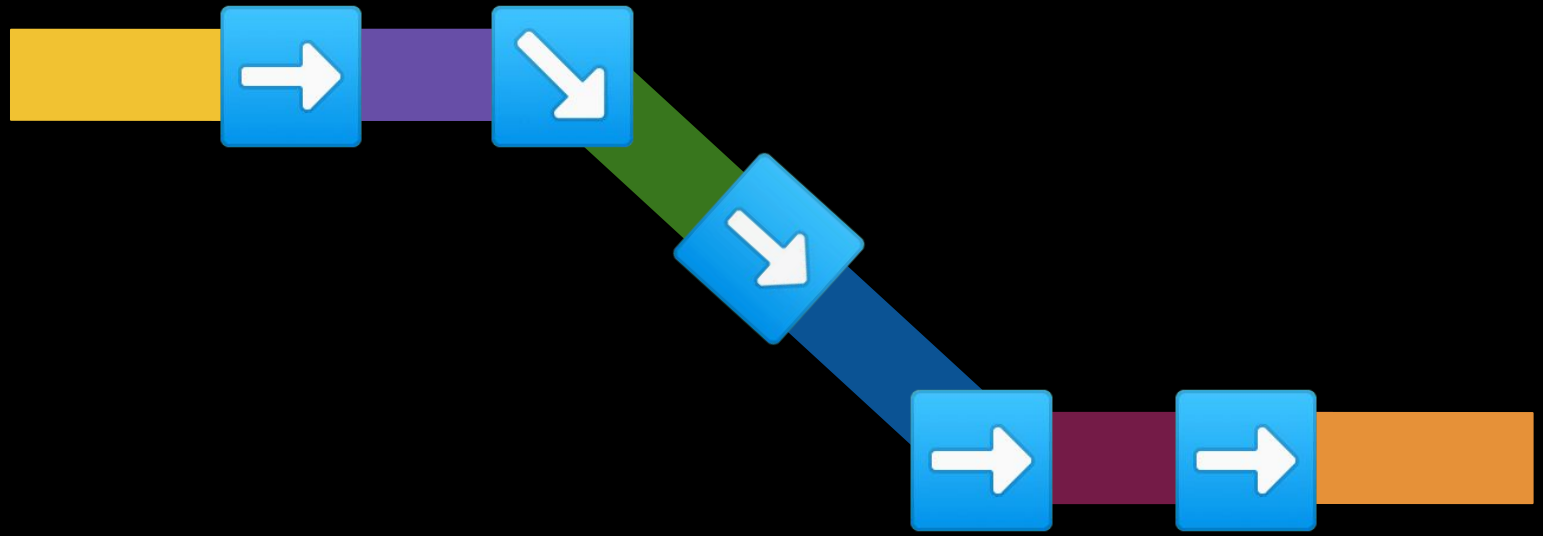
State as Context, Automation

⏳ Beginning ↘️

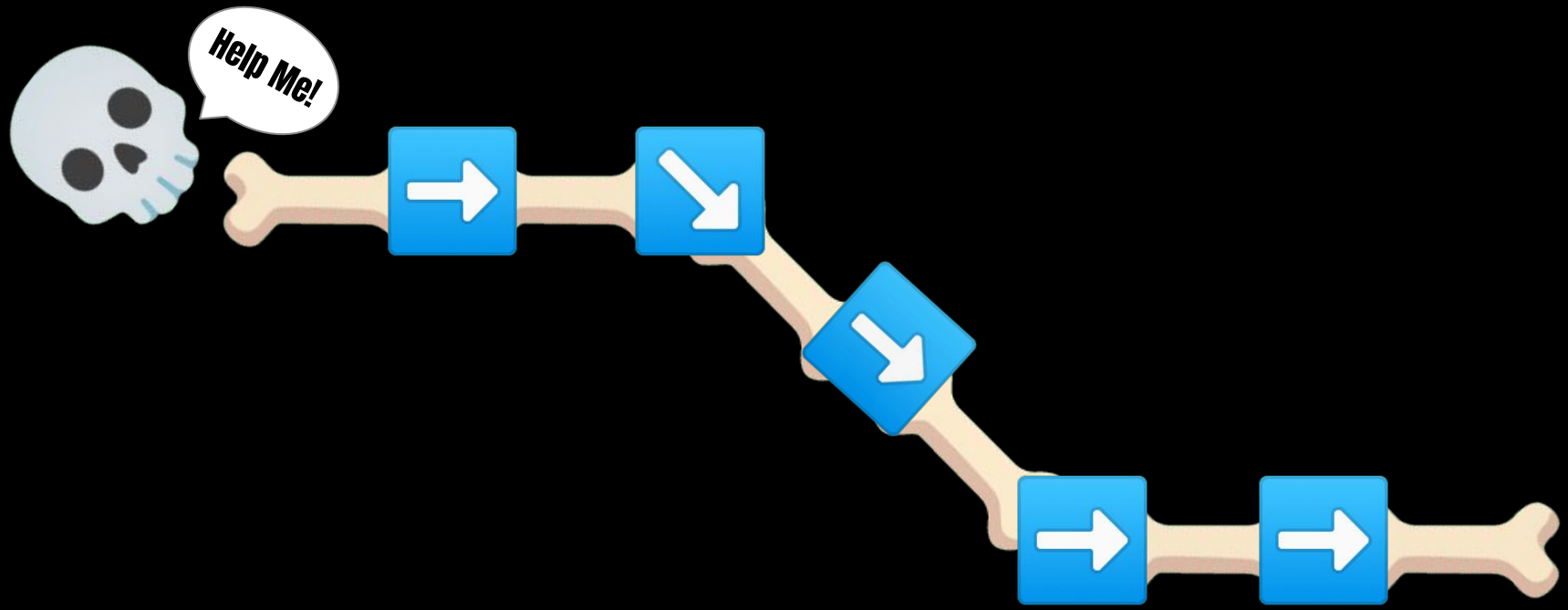
Middle ↘️

End ⏳

State as Context, Automation



⏳ Consecutive State ➡️



Narrative Backbone

State as Context, Automation

States are
the **Tracks** and **Regions**
we put **Automation** on

The Building Blocks 🧱

The State Bit

Stuff we gather

(Game Data)

(time_of_day:12.00)

(awareness:combat)

(death_screen:FALSE)

(Game Data)

(time_of_day:12.00)

(awareness:combat)

(death_screen:FALSE)

Stuff we derive

(Game Data)

(time_of_day:12.00)

(awareness:combat)

(death_screen:FALSE)

is:

& awareness=combat
& detected=TRUE
& distance < 50m

Stuff we derive

is:

& awareness=combat
& detected=TRUE
& distance < 100m

(Automated Variables)

(Game Data)

(time_of_day:12.00)

(awareness:combat)

(death_screen:FALSE)

is:

& awareness=combat
& detected=TRUE
& distance < 50m

is:

& awareness=combat
& detected=TRUE
& distance < 100m

P3

state_combat

P2

state_tension

state_combat P3
raptors >= 2
& t-rex > 0
& death_screen=FALSE

(state_combat:TRUE)

(Automated Variables)

(State Machine)

(Game Data)

(time_of_day:12.00)

(awareness:combat)

(death_screen:FALSE)

is:

& awareness=combat
& detected=TRUE
& distance < 50m

is:

& awareness=combat
& detected=TRUE
& distance < 100m

P3

state_combat

P2

state_tension

state_combat P3
raptors >= 2
& t-rex > 0
& death_screen=FALSE

(state_combat:TRUE)

(Automated Variables)

(State Machine)

Avalanche
Studios Group

Full Project Example

RAGE 2

INSANITY RULES

Avalanche Studios Group

RAGE 2 id

RAGE 2 id

Insanity Rules...

...Really?

Ok then 🤷‍♂️

RAGE 2 id

Rage 2: Music Design

Full Project Example

Don’t Panic!

We just have to solve:

Rage 2: Music Design

⏳ Beginning ➡️ Middle ➡️ End ⏳

In an Open World...

The demands of the Game's Design

The wants of the Music Design

Rage 2: Music Design

Game Design ↘️

🥪 Me 🤯

↖️ Music Design

Game Design 🔀

🗺️ Mapping 🗺️

↘️ Music Design

Rage 2: Music Design

💡 Mad Max ➡️ 💡 Rage 2

Rage 2: Music Design

⏳ Beginning ➡️ Middle ➡️ End ⏳

Music Design: States for Gameplay Narrative Progression

exploration 🔀

danger/anticipation ➡️

combat_intro 🔀

combat_bad/good ➡️

combat_climax 🔀

danger/aftermath 🔁

x2
Mad Max

RAGE 2 id

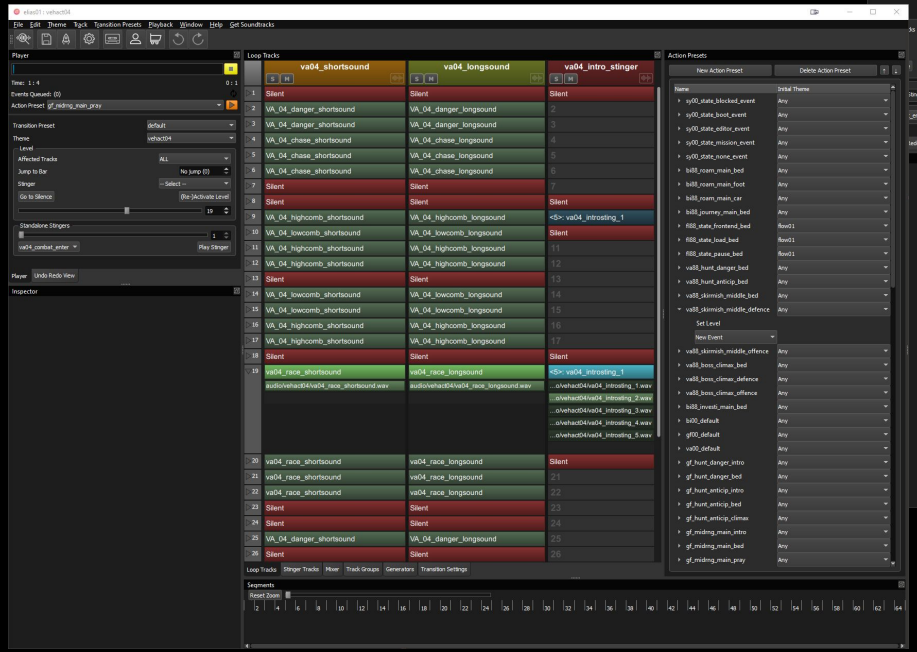
Music Design: States for Interrupting Moments

- ⚠️ health_critical / repair
- ⚠️ death
- ⚠️ overdrive
- ⚠️ combat performance (good / bad)
- ⚠️ etc.

Music Design: States for Gameplay Types (Activities)

open_world_combat, encampment_combat, race, boss, defense_tower, vehicle_combat, mutant_bash_tv, exploration, settlement, frontend, etc.

Rage 2: Music Design



Theme Templates (Elias)

RAGE 2 id

Good Question: *"That's a lot of music boxes to fill..."*

Q & A Bait 🎣

Rage 2: Implementation

Project Example

Rage 2: Implementation

1 Activity

==

1 State Machine

Rage 2: Implementation

encampment_combat

enc_danger
enc_anticipation
enc_combat_intro
enc_combat_bad
enc_combat_good
enc_combat_climax
enc_aftermath

vehicle_combat

veh_danger
veh_anticipation
veh_combat_intro
veh_combat_bad
veh_combat_good
veh_combat_climax
veh_aftermath

open_world_combat

owc_danger
owc_anticipation
owc_combat_intro
owc_combat_bad
owc_combat_good
owc_combat_climax
owc_aftermath

Activity
Inheritance

Rage 2 Open World Encounter Activity State Machine

smodm_gpl_st_activity_owenc

<Node filter>

- smodm_gpl_st_activity_owenc
  - smodm_gpl_st_activity_gunfight (smodm_gpl_st_activity_gunfight)
    - smods_fsm_gpl_state_group_base
      - debug_visualisation (smodc_graph_fsm_setup)
      - gpl_st_activity_owenc
        - states
          - 000.none (smods_gpl_st_active_state)
          - 110.gfafter.(gfdanger).nme_area_unoccupied (smods_gpl_st_active_state)
          - 150.gfafter (smods_gpl_st_active_state)
          - 175.gfafter.camp_complete_screen (smods_gpl_st_active_state)
          - 200.gfdanger.base (smods_gpl_st_active_state)
          - 210.gfdanger.nme_area_occupied (smods_gpl_st_active_state)
          - 300.gfanticip.base (smods_gpl_st_active_state)
          - 310.gfanticip.nme_area (smods_gpl_st_active_state)
          - 450.gfcomb (smods_gpl_st_active_state)
            - gfcomb
              - conditions.OR
                - continuous.detect_any_presence.AND
                - enemy_types.OR
                - presence_of_onfoot_enemies.and
                - presence_of_vehicle_enemies.and (smods_cond_activity_g
                - enemies.vehicles.(AND)
                - enemies.or
                - player_slow (smods_cond_validation_enemy_vs_trai
                - inclusion.validation_enemy_vs_transport.OR
                - state.EQ.this_(gfcomb)
              - enter.from_any.OR
                - lovar_debug_state.eq.this_(gfcomb)
              - events
          - 500.gfclimax (smods_gpl_st_active_state)
          - 999.blocked (smods_gpl_st_active_state)
        - transitions
        - variables

Properties

Transform	
Attach Max Tilt	0
Attach To Ground	☐
Attach To Mesh	☐
Attach To Water	☐
Transform	0 0 0 / 0 0 0
Transform Rotation	0 0 0
Transform Translation	0 0 0
General	
Event Scopes	
Name	450.gfcomb
Node Id	{6F8EED58-82B0-4823-B344-5ED9258
Tags	
_class	_entityinstance
State	
Module Name	gpl_st_activity_owenc
Priority	450
State Name	gfcomb
State Selection Output	
Use Group Variable	yes (1)
Use State Variable	no (0)
state_phase_group_module	gpl_st_activity_gunfight
state_phase_group_value	gfcomb
state_phase_group_variable	activity_owenc
State Timers	
Use State Timers	count (1)
ai	
Navmesh Merge	None (0)
fmod_bank	
Add Referenced Fmod Banks	☑
random	
Random Mode	None (0)
Random Probability	1

Rage 2: Implementation

base_combat 👴

bas_danger
bas_anticipation
bas_combat_intro
bas_combat_bad
bas_combat_good

bas_aftermath

encampment_combat 👧

bas_danger
bas_anticipation
*enc*_combat_intro (NEW)
*enc*_combat_bad (NEW)
*enc*_combat_good (NEW)
*enc*_*combat*_*climax* (added ➕)
bas_aftermath

open_world_combat 👧

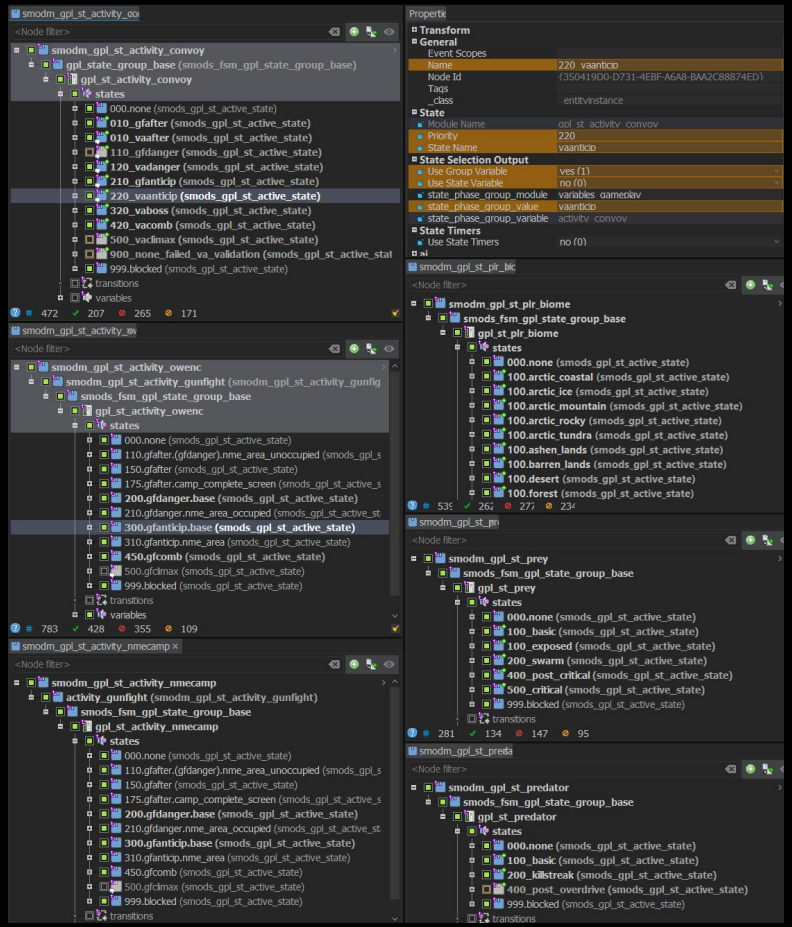
owc_danger (NEW)
~~bas_anticipation~~ (removed ❌)
bas_combat_intro
bas_combat_bad
bas_combat_good

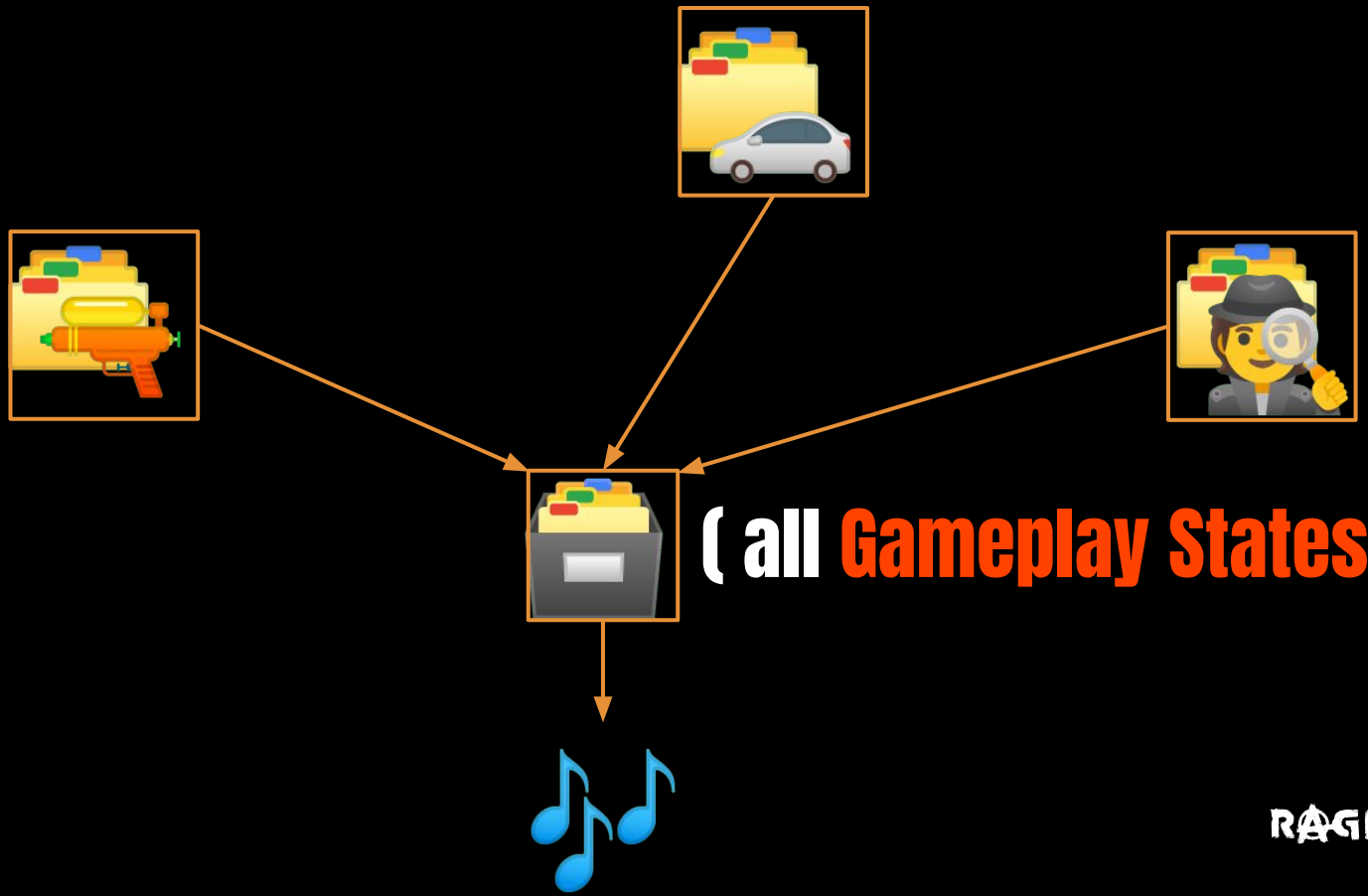
~~bas_aftermath~~ (removed ❌)

RAGE 2 id

Activities resolved by Gameplay State Machine

Rage 2 Multiple Activity State Machines




(all Gameplay States)

Rage 2: Implementation

encampment_combat

enc_danger
enc_anticipation
enc_combat_intro
enc_combat_bad
enc_combat_good
enc_combat_climax
enc_aftermath

vehicle_combat

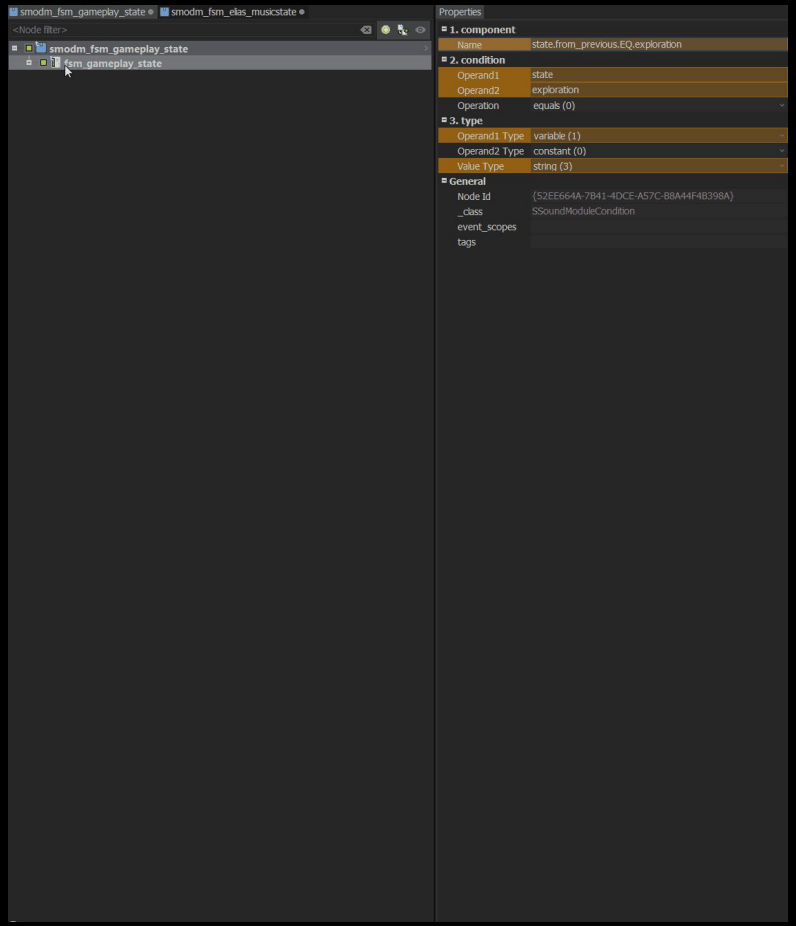
veh_danger
veh_anticipation
veh_combat_intro
veh_combat_bad
veh_combat_good
veh_combat_climax
veh_aftermath

gameplay_state

enc_danger
veh_danger
enc_anticipation
veh_anticipation
enc_combat_bad
veh_combat_bad
enc_combat_good
veh_combat_good
enc_combat_intro
veh_combat_intro
enc_combat_climax
veh_combat_climax
enc_aftermath==veh_aftermath

Low (Priority) High

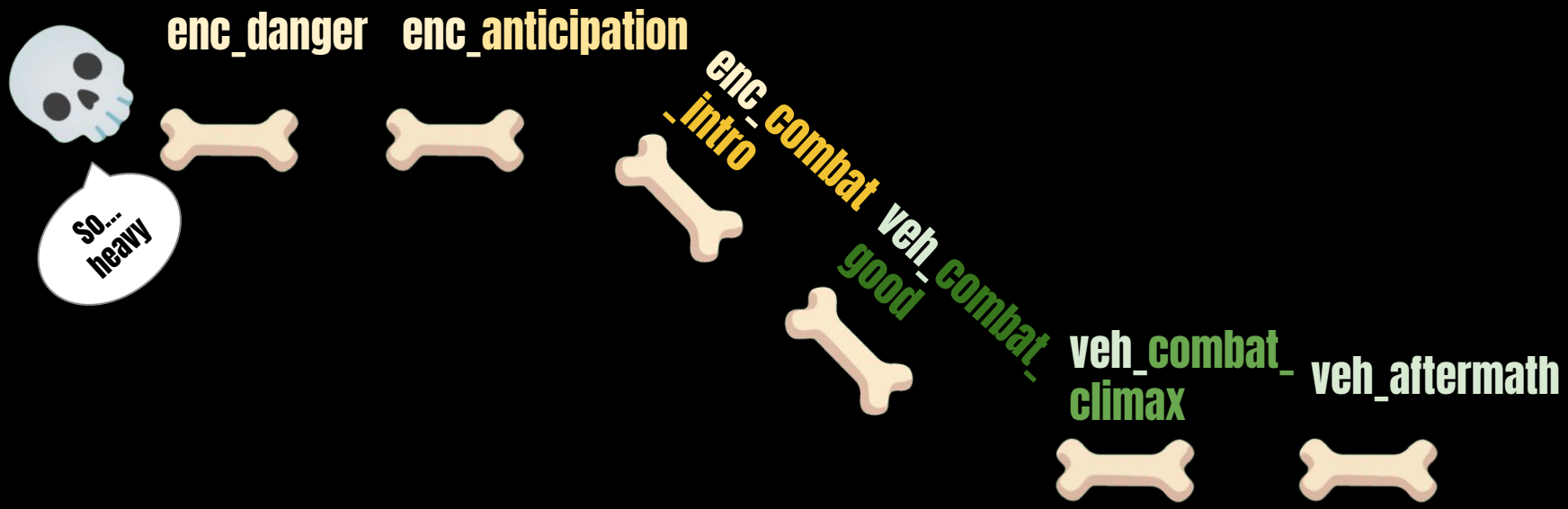
Rage 2 (all) Gameplay States



Good Question: *"Wait... Wasn't that two state machines?"*

Q & A Bait 🎣

Rage 2: Implementation



⏳ Consecutive State ➡️

Rage 2: Implementation

Game ⤵️

Activities ⤵️

Narrative State ⤵️

Middleware

Rage 2: Implementation

Game ⤵️

Activities ⤵️

(Gameplay State Machine) ⤵️

Middleware

(Gameplay State Machine)

=

Rage 2: Implementation

Narrative Sections

danger / anticipation
combat_intro
combat_bad / _good
combat_climax
aftermath

×

Interrupts

⚠️ health_critical / repair
⚠️ death
⚠️ overdrive
⚠️ combat performance
⚠️ etc.

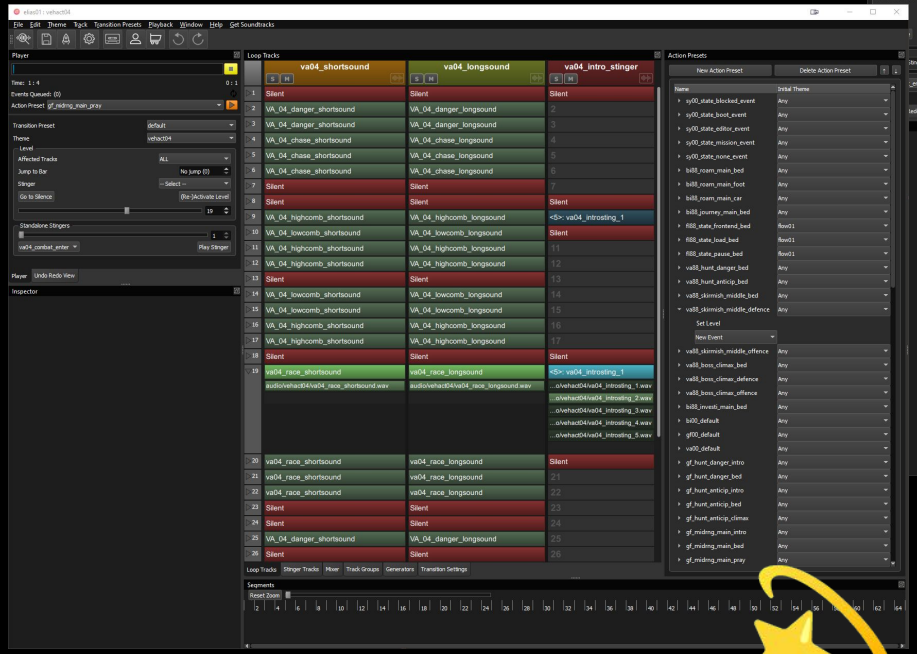
×

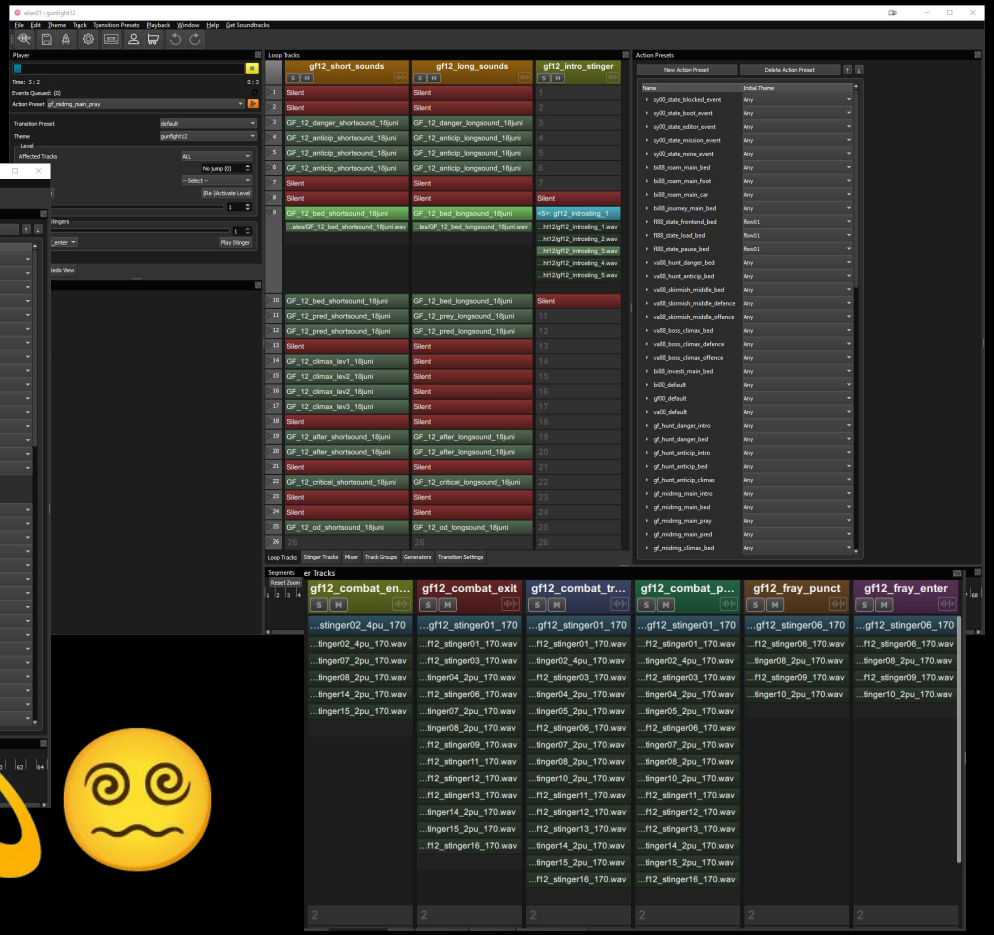
Activities

open_world_combat, encampment_combat, race, boss, defense_tower, vehicle_combat, mutant_bash_tv, exploration, settlement, frontend, etc.

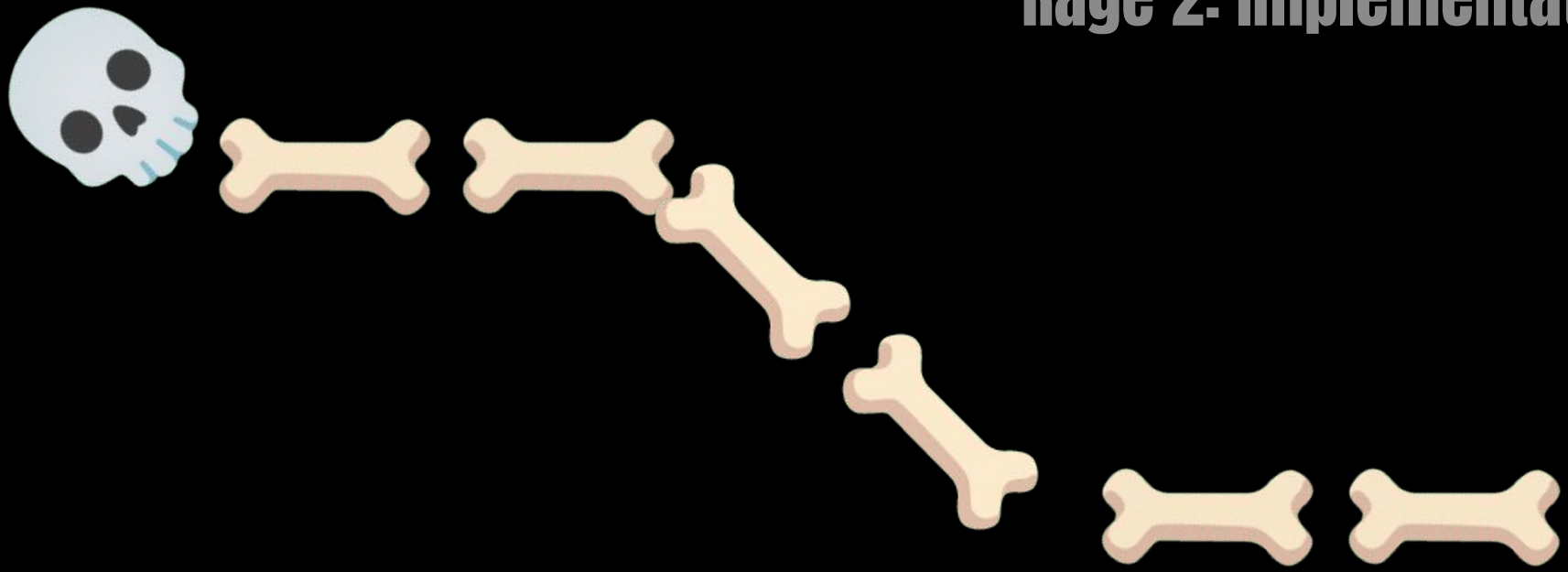
=

Rage 2: Implementation

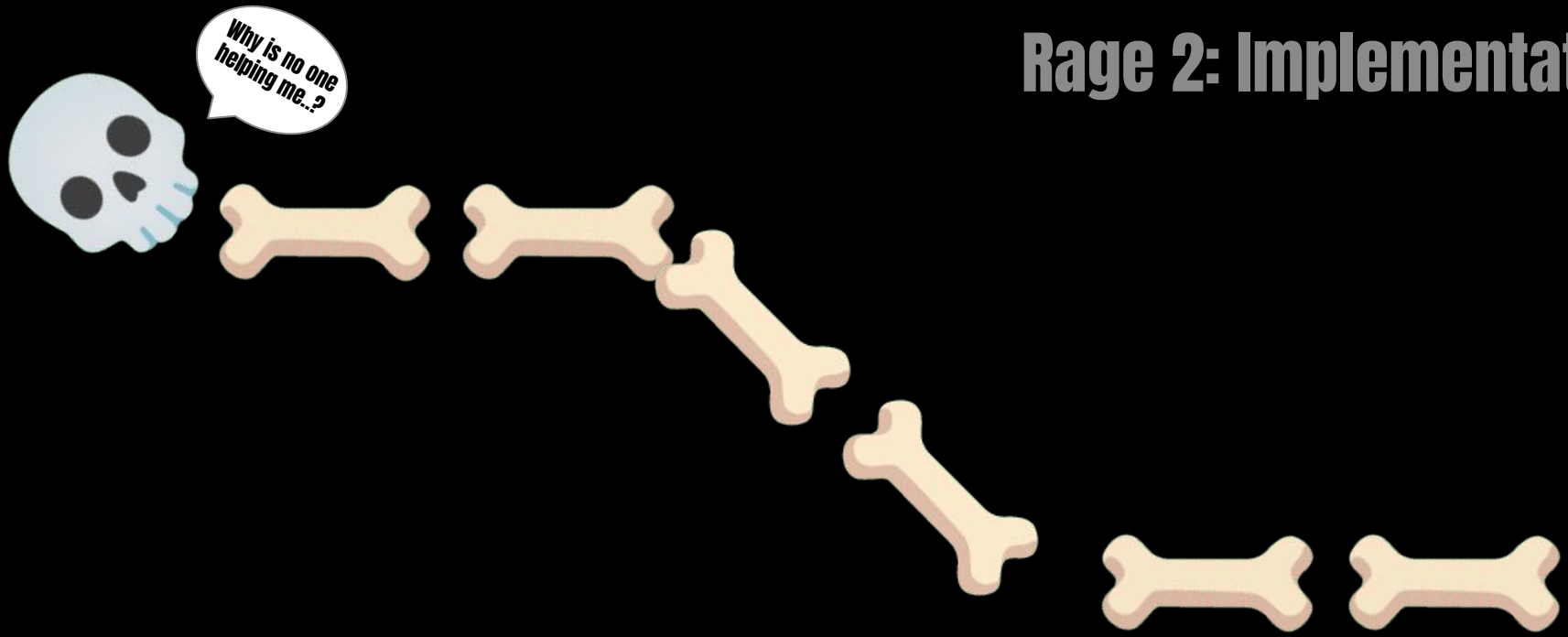




😅 But after all that,



the Narrative Backbone



the Narrative Backbone is in place ✅

Principles Learned

There'll never be a "System"

Principles Learned

Principles

https://splice.com/blog/interactive-music-system-video-games/

Something to keep in mind when building an interactive music system for a game is that **there's no magic bullet to creating the system**. Just because a particular strategy worked really well for one game, it doesn't mean that it'll be right for your game in the same exact way. That being said, **basic principles derived from one game mechanic can easily be translated to another**. Learning transfer is one of the most important things to keep in mind when doing video game music research.

- **Ronny Mraz**

System vs. Paradigm

A **System** is a collection of organised things; a whole composed of relationships among its members.

A **Paradigm** is an example serving as a model or pattern; a template.

Standardise your Principles

System

Learn solutions for common problems

Just like solving for

Sound Design

Programming

Technical Sound Design?

Music Design

Avalanche
Studios Group

Find analogies you can work with

Principles Learned

Principles Learned: Analogies you can work with

Like,

Principles Learned: Analogies you can work with

“Activities as Context”

Sounds simple, but...

Principles Learned: Analogies you can work with

Ideas take Iteration

Principles Learned: Analogies you can work with

💡 Mad Max ➡️ 💡 Rage 2

⬆️

⌛

Find what works for your game

And your brain

🧠

Principles Learned: Analogies you can work with

Game Design ↘️

Me 😅

↖️ Music Design

Principles Learned: Analogies you can work with

Reuse them

Principles Learned: Analogies you can work with

exploration🔀

danger/anticipation➡️

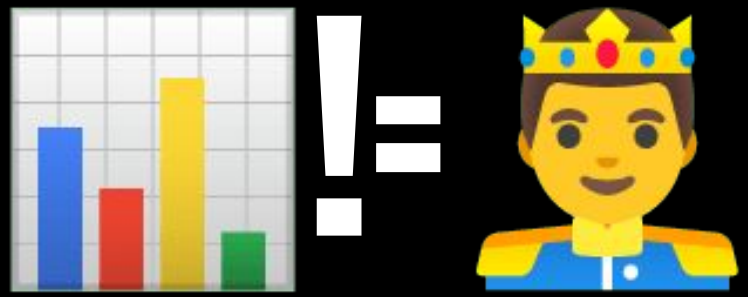
combat_intro🔀

combat_bad/good➡️

combat_climax🔀

danger/aftermath🔁

Principles Learned: Analogies you can work with



Principles Learned: Analogies you can work with

In other words,

Principles Learned: Analogies you can work with

Use Names,
not Numbers

Avalanche
Studios Group

What to expose

Principles Learned

Principles Learned: What to expose

Gameplay ➡️ Numbers ➡️ States

Principles Learned: What to expose

Gameplay 💉 Numbers? 💉 States?

Rage 2 Combat Ranges

:show gameplay fsm

Gameplay & Music States

show music fsm:

show theme:

gameplay_mission_manual: music_elias_theme_manual: section_intro_manual:

gunfight_climax_scale: choose a level | choose_a_level | reset | resend

fsm_gameplay_state: choose a state | state_name | reset

fsm_elias_musicstate: | state_name | reset

:show

Combat Ranges

ranges_manual:

	min:	max:
range_awareness_npc	0.0	40.0
range_awareness_omni	0.0	40.0
range_awareness_plr	0.0	40.0
range_local_kill	0.0	40.0
range_threat_npc	0.0	40.0
range_threat_plr	0.0	40.0

Principles Learned: What to expose

↘️ **Numbers:**

Difficult to sculpt general rule

Principles Learned: What to expose

↘️ **Numbers:**

Difficult to sculpt general rule

Principles Learned: What to expose

↘️ **Numbers:**

Time consuming to script specific contexts

Principles Learned: What to expose

↘️ **States:**

I know what I want

Principles Learned: What to expose

↘️ States:

I know what I want

Principles Learned: What to expose

↘️ States:

I want it now

Principles Learned: What to expose

↘️ States:

Let me set it

Principles Learned: What to expose

Gameplay

↘️ Numbers ❌

↘️ States ✔️

Principles Learned: What to expose

Gameplay

↘️ Numbers ❌

↘️ Names ✔️

Second Extinction Music State Machine

smodm_fsm_music

<Node filter>

- smodm_fsm_music
  - game_design_music_choice
    - game_design_music_choice.set.combat_high
    - game_design_music_choice.set.combat_low
    - game_design_music_choice.set.exploration
    - game_design_music_choice.tension_post_combat
    - game_design_music_choice.tension_pre_comb
  - fsm_music
    - states
      - 10.exploration (smods_fsm_shoebill_music_state)
      - 20.tension_pre combat (smods_fsm_shoebill_music_state)
      - 30.tension_post_combat (smods_fsm_shoebill_music_state)
      - 40.combat_low (smods_fsm_shoebill_music_state)
        - combat_low
          - conditions.(or)
            - music_logic.(and).music_choice.not_overriden
              - music_choice.not_overriden (scon_game_design_musi
              - music_logic.(or)
              - svar_game_design_music_choice.eq.this_state
          - events
      - 50.combat_high (smods_fsm_shoebill_music_state)
    - variables
      - from.game_design_scripting
        - svar_game_design_music_choice
      - from.variables_gameplay
      - to.variables_gameplay
      - state
      - state_age
      - state_type

Properties

1. component	
Name	svar_game_design_music_choice.eq.this_state
2. condition	
Operand1	svar_game_design_music_choice
Operand2	combat_low
Operation	equals (0)
3. type	
Operand1 Type	variable (1)
Operand2 Type	constant (0)
Value Type	string (3)
General	
Node Id	{BEADE607-9D9C-4BD9-92F6-79BF4B3107B0}
_class	SSoundModuleCondition
event_scopes	
tags	

Principles Learned: What to expose

☝️

(state logic override)

Avalanche
Studios Group

It’s not all Music Design...

Principles Learned

The wants of the Music Design

Principles Learned: It's not all Music Design

The demands of the Game's Design

Principles Learned: It's not all Music Design

Some of it is Narrative State Design

Principles Learned: It’s not all Music Design

It can be used for so much more...

Other uses for states

Mixing (snapshots)

Other uses for states

Triggering Sounds

Other uses for states

Affect sounds

(modulate / select)

Other uses for states

Gamestate / Game Flow
(UI screens e.g. frontend)

Other uses for states

Represent sections of scripted mission flow

Defining Key Locations

Other uses for states

Feed or trigger culling rules

Other uses for states

Feed / trigger mix rules
(e.g. sidechain ducking)

Other uses for states

Loading resources

Avalanche
Studios Group

The Golden Rule:
Infer Player Knowledge *and* know the Game's Design Intention

Principles Learned

You want to know:

What the Omniscient Game Director knows.

Where the player thinks they are.

(how they could / should feel about that)

The Golden Rule

What the player thinks they are doing.

(player story / game story)

The Golden Rule

Location

Action

The Golden Rule

📊➡️✒️

Find data that helps you derive the player’s experience

Music / Sound
Can tell you what you <u>know</u>
(Reinforce)

The Golden Rule

(Player Knowledge)

Music / Sound
Can tell you what you <u>don't</u>
(Allude)

The Golden Rule

(the Game’s Design Intention)

Neither is better

The Golden Rule

Use **both**

Avalanche
Studios Group

Closing Thought

Closing Thought

Closing Thought

Closing Thought

Closing Thought

=

Avalanche
Studios Group

Avalanche Careers

Thanks for listening

dominic.vega
@avalanchestudios
.com

avalanchestudios.com/careers

xan.williams
@avalanchestudios
.com

Avalanche Talks

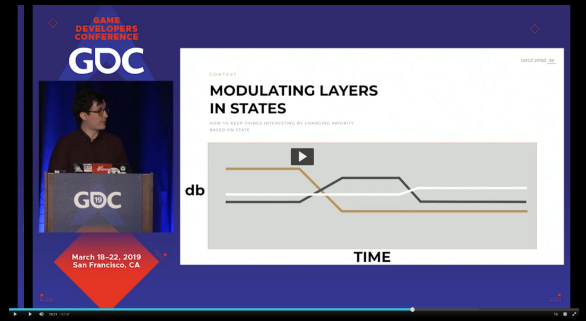
Thanks for listening

https://www.gdcvault.com/play/1025999/Building-a-Mixing-Sandbox-for

Building a Mixing Sandbox for 'Just Cause 4'

Dominic Vega

(scripting mixing in FMOD)



https://www.gdcvault.com/play/1026000/Music-in-a-Sandbox-The

Music in a Sandbox: The Dynamic Music of 'Just Cause 4'

Ronny Mraz

(what it looks like using data to script music in FMOD)

Avalanche
Studios Group

Avalanche
Studios Group