

Musicrib

DESIGN DOCUMENT

By Overdance for the 2025 Pirate Software Game Jam

Introduction	1
Game Summary Pitch	1
Inspiration	1
Player Experience	3
Platform	3
Development Software	3
Genre	4
Target Audience	4
Concept	4
Gameplay overview	4
Theme Interpretation (You Are The Weapon)	4
Primary Mechanics	5
Secondary Mechanics	6
Art	7
Design	7
Audio	7
Music	7
Sound Effects	7
Game Experience	8
UI	8
Controls	8
Development Timeline	8



Introduction

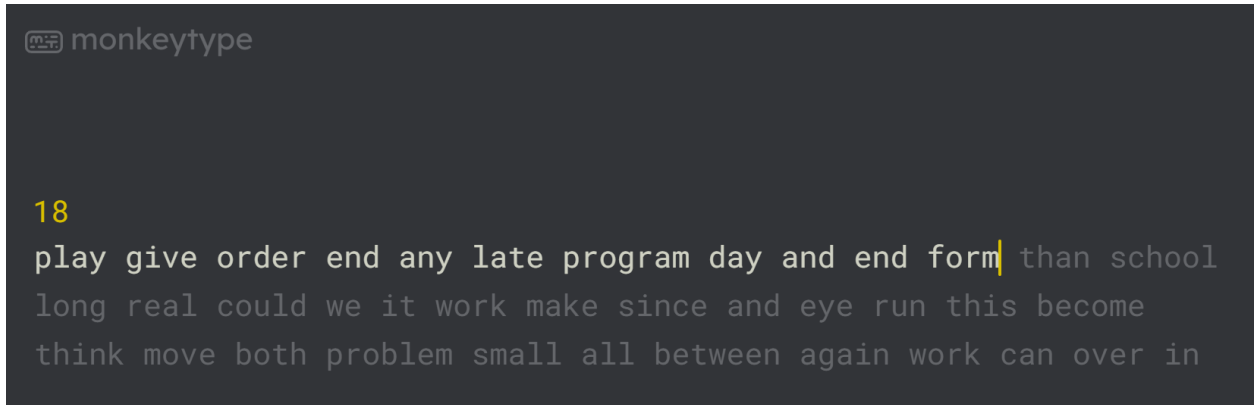
Game Summary Pitch

Musicrib is a one of a kind VSRTG (Vertical scrolling, rhythm typing game). Where you type the text according to the rhythm, displayed by the falling boxes.

Inspiration

MonkeyType

Monkeytype provides a browser based typing trainer, which aims to **track and improve over time your typing speed and accuracy.**



Osu! mania

Osu! Mania is a VSRG which is a part of a bigger rhythm game - osu!. Osu! Mania itself has one of the sweatiest communities. (if not the most) of VSRG out there. With it you can **improve your rhythm accuracy.**

Player Experience

The prototype contains 10 levels (including tutorial) with varying difficulty. The interface was made to be **extendable and customizable** with the possibility of downloading or creating new maps. As well as community made skins. (Not available for prototype runned on browser)

Compared to other rhythm games. Musicribe is able to tell various stories, with the possibility of typing. Nine levels intend to tell a story about You - a weapon.

Platform

The prototype is done in Pygame, developed to be released on windows PC. However if there would be forming a community of active players and mappers like in osu. I would rewrite it in C# on osu framework which could possibly help with optimization and releasing it on apple OS and linux.

Development Software

- pygame-ce 2.4.1 (SDL 2.28.5, Python 3.12.1)
- Photopea for graphics
- Bitwig 3.2.8 for SFX and music made by me (Overdance)

Genre

Singleplayer, VSRG, TypinGame

Target Audience

With a clear interface and easy to learn, hard to master gameplay. It is marketed to **rhythm game players** who are up for new challenges. As well as veteran typing game players interested in broadening their proficiency.

Concept

Gameplay overview

The player will have to type in rhythm which **requires manual and rhythm accuracy**. As in any rhythm game, the player would need to hone his accuracy in lower difficulties before being able to challenge higher ones.

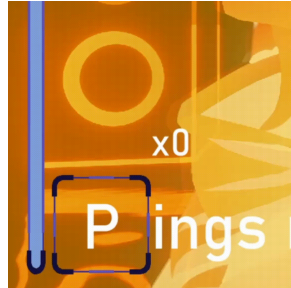
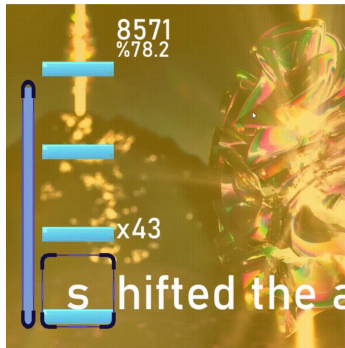
Theme Interpretation (You Are The Weapon)

'You are a Weapon' interpretation - Is told through the story in those nine levels.

Within the context of this rhythm game. The theme was fulfilled through the storytelling in typing and art. Initially the plot is introduced to the player as him being just a material in a womb of the second temple in a higher fantasy world. To be fully born and forged into a weapon, the player would have to absorb the data the womb provides to it. ***In the last level it is revealed that all of it was just an allegory to real life.***

Primary Mechanics

Mechanic	Prototype showcase (Art is final, custom skins may be superior)
<u>Tap</u> Press a letter in a square when the falling box reaches the bottom of the square. Adds combo.	

Mechanic	Prototype showcase (Art is final, custom skins may be superior)
<u>Slider</u> Hold and release according to the length of the box in the back.	
Mechanic	
<u>Missing</u> Missing resets your combo and damages your health. If the health bar goes all the way down then it's a game over. Letters are not bound to the rhythm object. So if you miss the letter character stays. (you can change it in settings)	

Notice that all primary mechanics change color depending on the correctness of the rhythm and letter.

Secondary Mechanics

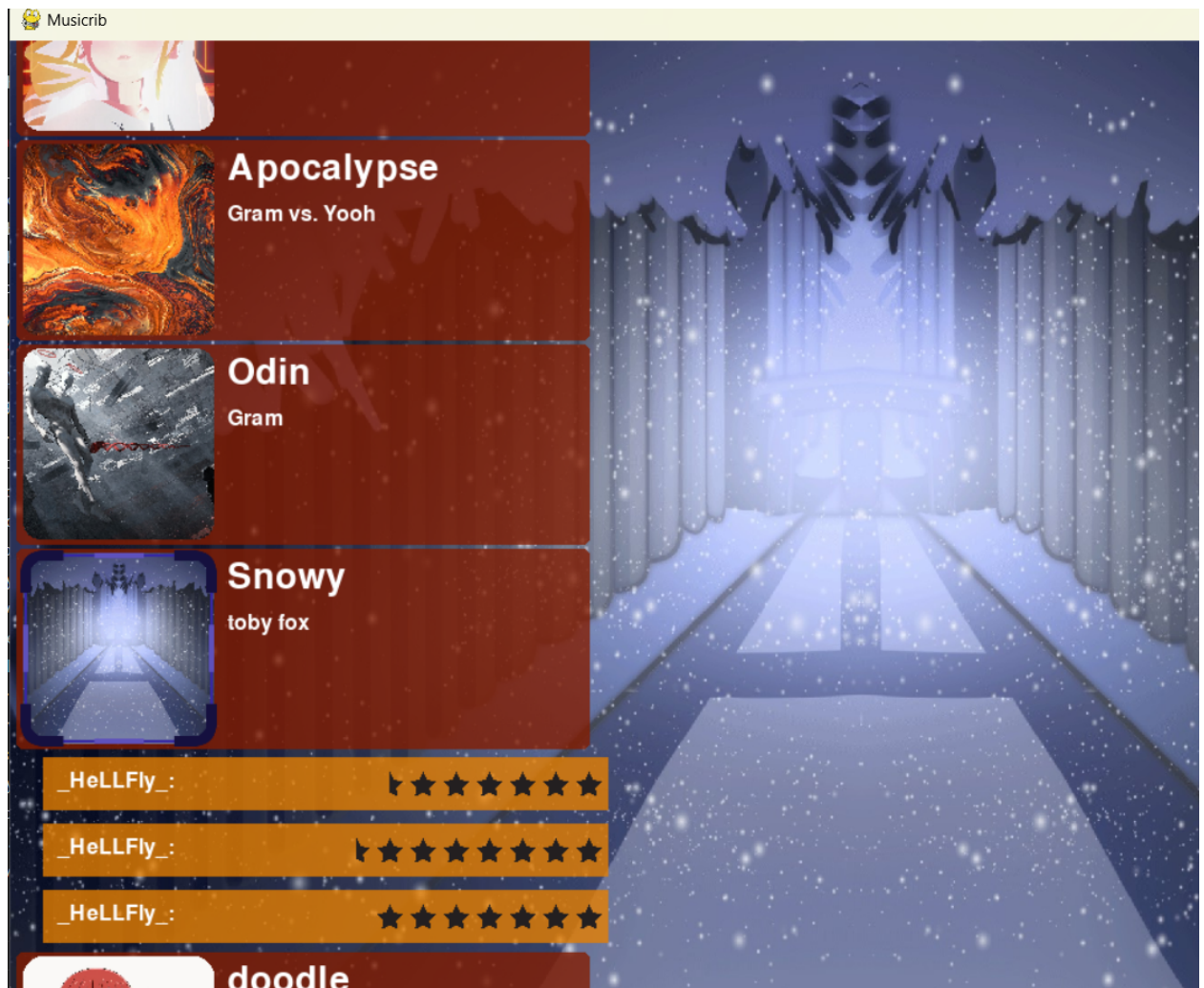
Mechanic	Mockup (They were not the priority during development so there is a possibility they didn't make it into the final prototype)
----------	--

<u>Rush</u> Press a given sequence of letters as many times as you can. Doesn't give a score but stacks combo multiplier. But be careful, the combo will reset if you type it wrong.	They weren't the priority during development. So if this text is still here. It means that they didn't make it into the final game jam prototype.
<u>Overdance</u> Press as many letters in a given time frame as you can. For every press there is a set 10 score.	

Art

Design

Modern and sleek design is intended to deliver all necessary information about the gameplay without distracting or overwhelming the player. As well as be comfortable and readable for a new to the game player.



Audio

Music

As the game is music based, most of the songs in it are upbeat for the faster experience of “enhancing” or “the flow” which is characteristic and sought after experience for rhythm game players. However, like most rhythm games, it has the potential to satisfy every music taste.

Sound Effects

The sound effects were designed to polish and brand the player experience in a unique pleasant way.

Game Experience

UI

Intuitive and easy to grasp.

Controls

Keyboard, (Optional Mouse in UI)

Development Timeline

MINIMUM VIABLE PRODUCT

#	Assignment	Type	Status	Finish By	Notes
0	Overall structure (think of it as an engine)	Coding ▾	Finished ▾	17 sty ...	The first attempt to make this game was in March 2024 in osu framework. In pygame, I've started working since June.
1	Design Document	Other ▾	Finished ▾	Jan 20...	English isn't my primary language. It may not be correct but it's on a communicative level.
2	Main menu theme	Audio ▾	Finished ▾	Jan 20...	Something short but iconic like in osu.

#	Assignment	Type	Status	Finish By	Notes
3	Slider	Coding ▾	Finished ▾	Jan 20...	The tap, but holding. Post note: HAHA MULTIPROCESSINGG GO BRRRRRRRRRRRRRRR RRRRRRR..
4	UI / Main menu	Coding ▾	In progress ▾	Jan 21...	Button UI, screen transition, title screen
5	Tutorial level theme	Audio ▾	In progress ▾	Jan 21...	Should be out of the box.
6	Rush & Frenzy	Coding ▾	Not started ▾	Jan 22...	New logic to conductor, typer and file converter.
7	Modifiers Settings	Coding ▾	Finished ▾	Jan 23...	Only 4
8	Loading screen	Coding ▾	Fire on gl... ▾	Jan 24...	Because it actually lags. HELLo pygame.
9	Special effects	Art ▾	Not started ▾	Jan 25...	Miss, combo, death animations
10	Art for menu and tutorial	Art ▾	In progress ▾	Jan 25...	Something interesting
11	Sound effects	Audio ▾	In progress ▾	Jan 25...	UI blips. + typing sound option in settings
12	Level design (1-5)	Other ▾	Not started ▾	Jan 27...	Create levels 1 through 5
13	Level design (6-11)	Other ▾	Not started ▾	Jan 28...	Create levels 6 through 10
14	Any extra polish	Other ▾	Not started ▾	Jun 29...	
15	SUBMIT	Coding ▾	Not started ▾	Jan 29...	Create Itch Page and upload. XD PYGAME

BEYOND (if ahead of schedule / extra time)

Extra levels	Other ▾	Not sta... ▾	
Settings Menu	Coding ▾	Not sta... ▾	Volume slider, fullscreen toggle

These are personal logs I had during game jam:

I may have over scoped. Like... A lot. I managed to catch a heartbreak in the time I should've been programming. xD.

So. The game itself in it's current prototype stage does not fit within the strict rules of this game jam:

The maps themselves were imported from osu along with the music and background images. Within downloadable Musicrib you can convert any .osu file into a playable musicrib beatmap. (Don't try to do it in browser)(afternote: I've removed it from browser)

The only music I have a commercial license for is my own Musicrib theme in the main menu.

Some of the maps text was written by AI.

However. If I didn't work alone and had someone to team up with. I'm 100% sure the game would fit within those rules. I know I may be bitching about it right now but I would humbly ask the judges to be more lenient with it.

Fun fact: this game uses MULTIPROCESSING 😊

And if you didn't guess it. Multiprocessing along with the pygame - It's like setting a fire on the glaciers.

Side note: You can't and should not try to set up fire on glaciers. The fire will go out and the glacier will melt. Yes. At the beginning it may work:

The tap notes are rendered by multiprocessing. But when I tried to do the same with sliders. They became longer and bugged. I wasted 2 days on this shit trying with asyncio, threading and multiprocessing until github copilot started begging me to stop asking about multiprocessing for pygame and just do it on one thread. But I added slider handling for mappers. So if the map is too heavy they can just turn it off for optimization.

Should've learned C# or C with frameworks. At least I would start a fire on a NORMAL PLAIN GROUND!

YOU WOULDN'T HAVE PROBLEMS WITH DEPLOYING IT IN WEB. Like. What?!?!?! Why is deploying a web version of the game in pygame so hard? Why is there no errors that I can fix but a black screen? THERE'S ONLY 17 HOURS TILL THE ENDDDD

14 Now. Finally. The game itself works. With threading... asyncio for pygbag... and multiprocessing. EVERY FRIKING POSSIBLE LIABRARY for doing multiple tasks at the same time. HMMMMMM... I WONDER WHY DOES PYGBAG SHOWS ME BLACKSCREEN

8 I didn't sleep tonight. But it was worth it. I've finally made a web version. But I had to cut out half of the code. The removed parts were all optimization ones. So now it's laggy, buggy and bonkers.