

PPPB (Pseudo Player Performer Bots)

-2018-

MIDI controllable bots in Garry's Mod for audiovisual performance and composition

IST PLATFORM FOR FLEXIBLE MEDIA LOGBOOK

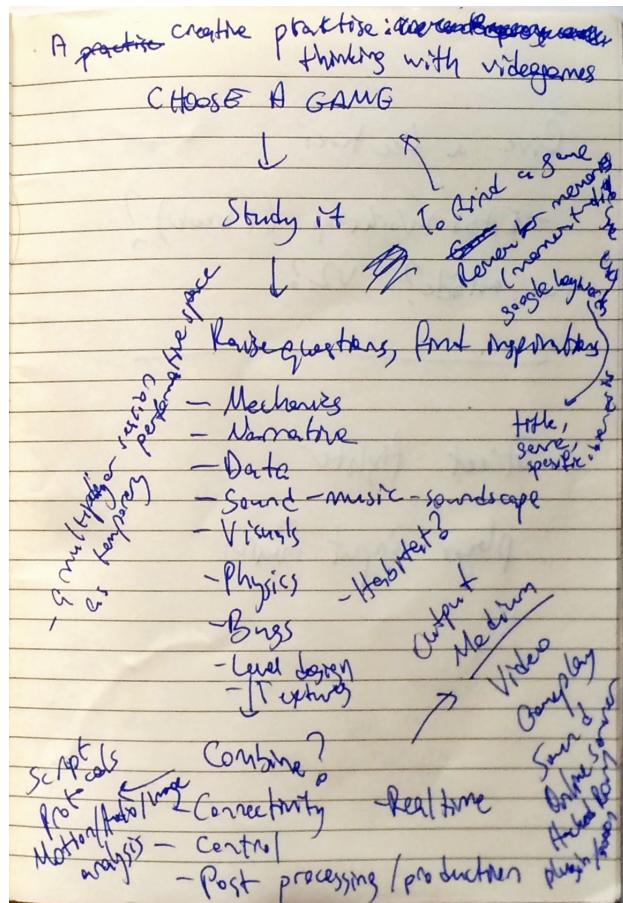
Berk Özdemir

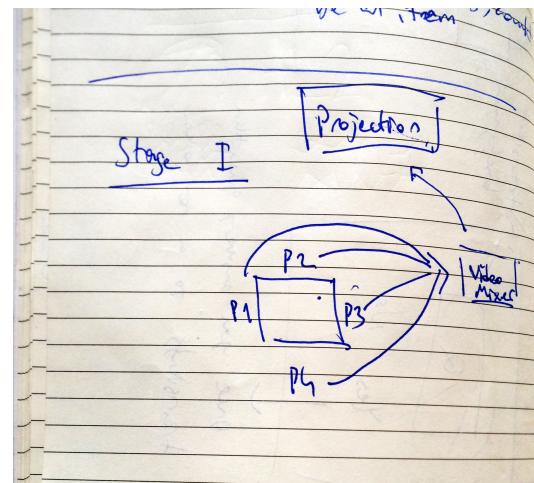
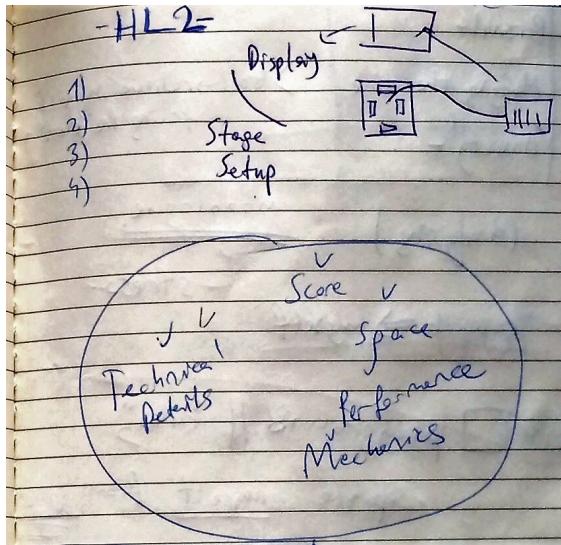
If I choose a videogame as my initial medium to think and work with, how can I turn it into a interactive performative space where I can compose scores for people to perform in? What would be my materials to work with, and how would I deal with time, and game space?

I started working on this project asking these questions. Lately, I've been enjoying a lot using ready-made digital media as my initial raw material to explore their undiscovered affordances to make new tools, art works and audiovisual compositions. I had worked with different media such as videos, websites and browsers as raw digital objects to use in/as audiovisual compositions and instruments, and this time I wanted to give all my focus to a videogame, to explore its new sides, to understand my interactivity and experience with it as a player, and to conceptualize new ideas for art + music projects. This method is very fun; but also usually very challenging; because especially solving the easiest technical problems might be a long and frustrating process, and a new medium such as a videogame demands a lot time investment and dedication for observation / research / experiment to get fruitful results.

I picked "Half-Life" as my choice of game to work with on this project. The Half-Life series was one of the "game changer" games in the videogame industry back in the day (no pun intended). The first Half-Life was an iconic FPS game which introduced the world a new way of storytelling, use of physical space to explore unique solutions to advance in the game, and the most important part; with the modding community, hundreds of mods for Half-Life series are made by fans to provide people a new singleplayer and multiplayer experience (such as Counter-Strike, Team Fortress and dozens more).

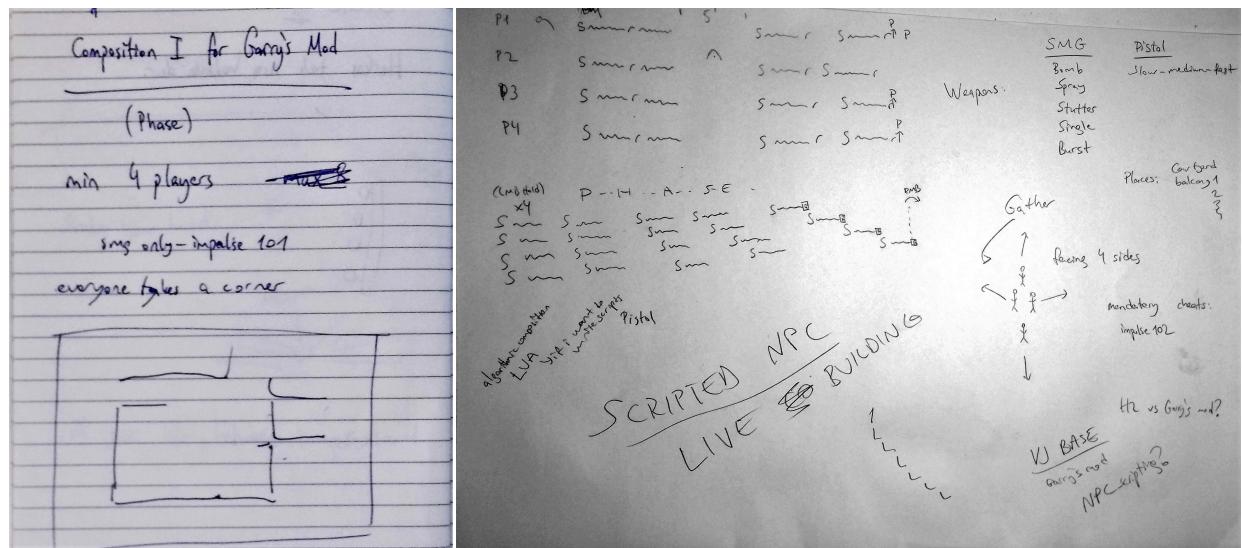
I already had some simple ideas to work with in the multiplayer mode of Half-Life. My first plan was to think of every player in a multiplayer game as a live performer, and write a simple text or graphic score for them to structure an audiovisual composition where people can perform online. I also chose the famous map "crossfire", because I already knew this map very well (because it was almost the only Half-Life map that has been played online in net cafes in 2000s, while I was going there to play with people in my childhood), so I already had some ideas like where to which places to use in the space for the performance.





First idea for performance distribution setup: 4 performers on stage with their laptops, their displays sent to a 4-Channel video mixer to be mixed in real time. I was going to be the person on mixer as the visual-conductor.

I sketched some text - graphic scores, tried to categorize the things the performers can do as a performer (like move, shoot, change weapon, jump etc.), so I could have a simple dictionary of parameters to write my scores. Determining these parameters are very important, because the score is the medium to communicate with the performers to tell how / when / what they need to do in the performance, so it needs to be clear enough for them; otherwise the outcome in the performance might be completely different then what the composer imagines.



Thinking of the score, a new question had also popped up: what if I created a Half-Life multiplayer server myself, and whenever a player joins the game; the server turned them to a performer by giving each player instructions to perform via in-game chat (example: player 1 - go to "A", player 2 - throw bombs while you are jumping on the courtyard, player 3 - chase down player 2 etc.)? Then the server would be a 24/7 working generative piece that could be performed anytime by any group of people.

But there was a big technical problem: I had no idea how to do it. When I made a small research on how to do it, I realised that there might be a chance where I may never figure out to create such a fully functional server in Half-Life as I want. Half-Life's multiplayer documentation was very old, and it wasn't popular among the community anymore, so I couldn't know where to get good information / help.

And also since I hadn't created a score system yet, even if I managed to pass the technical problems, the end result might be a big disappointment.

Half Life server'ı kurmuş/yönetmiş ve modlar konusunda bilgili, teknik konulara hakim bir arkadaş var mı buralarda? Bir projem için fikir yardımı almak istiyorum. Bir hala bilmiyorum bu konuda, kafamdaki soruları nasıl aratıp bilgi bulacağımı çözemedim resmen

Translate Tweet
10:26 PM · Oct 14, 2018 · Twitter Web Client

View Tweet activity

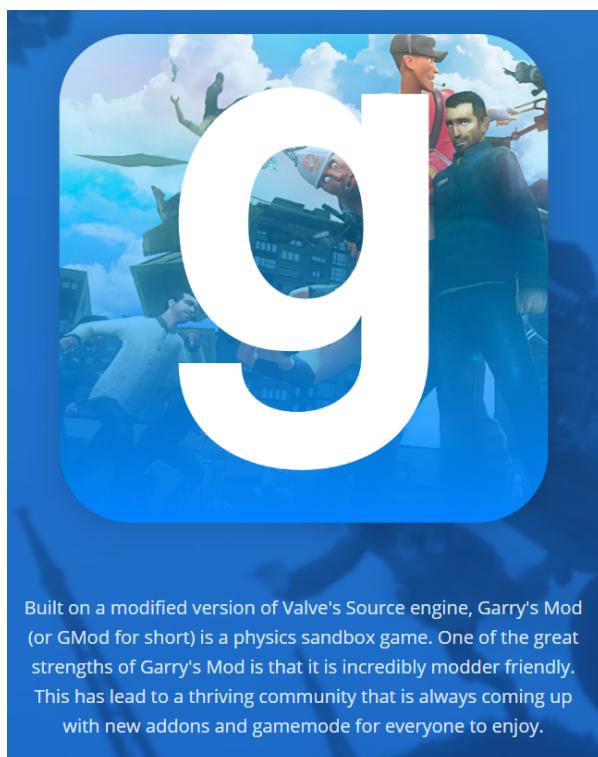
4 Retweets 84 Likes

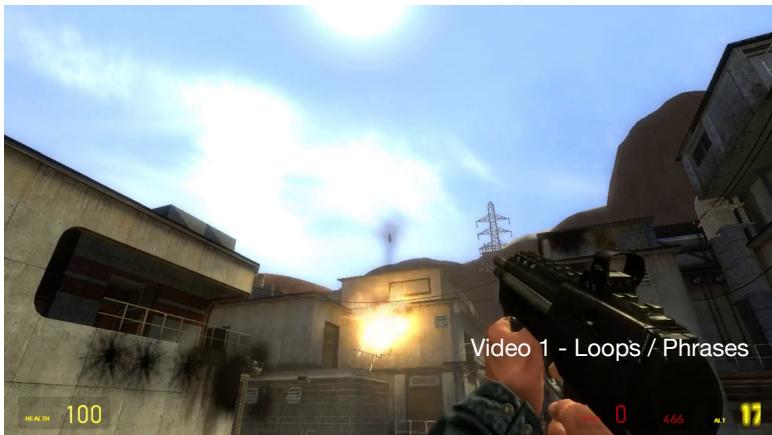
Ifteriş Eroğlu (@linuxgemini) · Oct 15
Replying to @berkozdemir
Abi Digitalocean'daki 5 dolarlık VPS'te vanilla HL, CS1.6 vs hostladım ama Videoyun discordu harici giren olmadı OMEGALUL
Mod (+yönetim) çalıştırma için GoldSrc'de MetaMod + AmxModX, Source'da ise SourceMod + Ulysses Mod (ULX ve ULib). Plugin dili oyun

I asked on Twitter, if such a thing is possible to do.

When I asked people on Twitter to get some help, no one could give me the answers I wanted, but I realised that many people advised me to check "Garry's Mod", if I want to work with scripting. I had known Garry's Mod for a long time, but I hadn't spent time before on exploring it.

They were right. Garry's mod was offering enormous amount of possibilities to create and do new stuff. Besides the scripting capability, there were already thousands of different content made by users like textures, entities, maps, weapons etc. ready to be used, and lots of utility add-on's for server management, GUI etc.





Also, Garry's mod uses Half Life 2's game engine. So it was a very easy decision for me to continue with Garry's Mod.

How do the physics and the space of the game react to my actions and movement? Physical movement in the game as a "human" character is directly depended on what kind of interactivity the game offers to the player; for a FPS game, a player sees the world with the character's eyes, mostly has a weapon holding, and can control the movement (walk, run, crouch, jump), sight (look) and actions (use object, fire weapons etc.).

The communication between the player and the character can be explained briefly:

1-) Player sends physical input to the game with human interface devices (keyboard, mouse)

2-) Game character reacts to input, physical world and entities react to character's output

3-) Player gets visual and sonical feedback of the action

And this process goes on as a feedback loop.

I was thinking a lot about gradual movement and phasing, and I did many improvisations by myself, using weapons and making small changes on my move speed or the view angle. In my opinion, video games are superior to any other medium in the way a user interacts with them; it's very easy to apply specific ideas in gameplay to a performative concept. I could create loops, phases and spatialization only doing simple mouse movements and keystrokes.

I had given up working on a generative server; I think it was a brilliant concept to work on, but I was in doubt on how far I could go with my knowledge in the game. It was going to be a slow process for me to learn how to do it technically, prototype and test my ideas with people. I can work on it in the future, if I can find some people to contribute with me.

Instead of that, while looking for new solutions, I noticed that if I can create characters in game and control them like a real player in real time, I wouldn't have to think about distortion in communication between me "as a composer" and the performer, and I could easily test my ideas with the "pseudo performers" I create.

I made an extensive research to understand how to do it: I downloaded lots of add-ons, scripts and NPC entities, examined them; watched Youtube videos, started learning Lua (Lua is a scripting language, which GMod uses as a platform to allow users to script their content), spent lots of time in Garry's Mod forums, Garry's Mod Wiki, and other documentation websites I found.



Video 2: Spawning many dummy bots and trying different console commands to control them ended with a tragic accident: one poor bot got stuck inside a closet and he couldn't escape since...





Screenshot from Video 3

After I started learning Lua, the first thing I did was to modify some example files of "NextBot NPC" to create NPCs with different behaviours. Although I enjoyed to watch the hilarious visual result when I spawned dozens of them; I wasn't satisfied with the flexibility of control I had on them. I was looking for something more like a "fake player" which would imitate human inputs.



Screenshot from Video 3

In the meantime, I was desperate finding anything working as I wanted, so I posted this thread on Garry's Mod forum:

I was messing around with Gmod for almost 3 weeks for the art project I am working on for my master's studies, and apparently it was way harder than I expected to get the information I need doing my research myself. That's why I wanted to share here what I want to do briefly to get some feedback, advices, and to know if it's possible to do it, and if so, to learn the fastest and most straightforward way to make it possible (otherwise I will waste another month without any progression until my project deadline). Any kind of help is very appreciated!

So here is my problem:

Basically what I want is, to make an time based algorithmic composition using an automated script I will write for dummy models (bots/NPCs/fake players/ whatever you want to call) to execute spesific actions on spesific moments.

For example, after I create a new game, an executed Lua script (or a combination of scripts) will do;

- Create 4 dummy players
- Teleport each of them to spesific positions
- Give them SMG and shit ton of ammo
- Make them look at spesific directions (Player1 to the wall, Player2 to the sky, Player3 to an object near it etc.)
- On second 1, Player 1 starts shooting with LMB for 5 seconds, Player 2 starts spamming RMB (SMG bomb) on second 2 for 10 seconds, Player 3 and 4 shoots a single RMB on second 3 for once, pauses, and shoots another single RMB on second 15

etc.

So my questions are:

What would be the best way to start, which tools or things I need to check out? I checked Bots, NPCs, SNPCs, NextBots, and Expression 2 tool under WireMod. The closest I could go closer to my goal was with NextBot and regular bots. They look like the only way for me to imitate keyboard-mouse inputs; but there still are many problems; I really couldn't figure out how much spesific I can be with my commands, and I still am not sure if it's possible to control each individual player. Most of the commands I found (console commands for instance) controls all the NPCs instead of a spesific one.

How spesific can I be with my commands? Can it be like "Player 2, shoot with pistol 4 times (1 bullet per second), then stop for 5 seconds, shoot with pistol 6 times (2 bullets per second), then spray a magazine with SMG"?

Is it possible to record a game activity of mine as a data (like movement, mouse direction, usage of weapons) and play it back with a dummy model (and edit it)?

And got this response:



Rubat 28

Your best bet if that is all you want to do is to use bots and [GM/StartCommand](#) to control them. You will need to come up with your own timing solution as you control bots by issuing them keyboard events (not really but basically)

Any other method would involve creating systems to imitate player-like behavior which will take more time.

berkozdemir posted:

Is it possible to record a game activity of mine as a data (like movement, mouse direction, usage of weapons) and play it back with a dummy model (and edit it)?

Not without mods, and I am unaware of any such mods existing.

1

This single reply helped me a lot with progressing in my research. I followed the way Rubat advised me, and within 2 days I was able to make my first bot prototypes to play with.



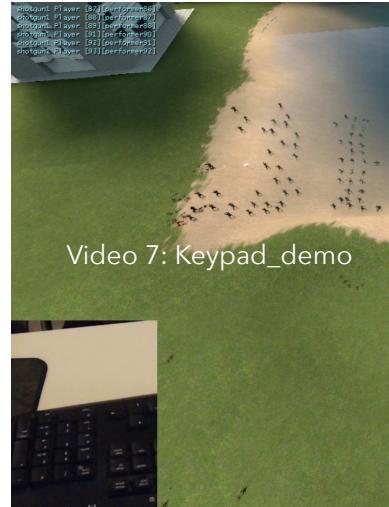
After checking all the example files I could find, I could finally find a way to create a bot, and control it as I wanted. Despite the controls for the bots being very primitive, I was already very happy to play with my new bots.

Video 5: First bot ensemble ->



<- Video 6: Piece 1

I was looking for more interactivity; I mapped each numpad key on my keyboard to different functions to control the bots. I couldn't control the bots individually, so each command applies to all bots at the same time, unless I overload the game with lots of bots, which makes the data transfer between my keyboard and game unstable.



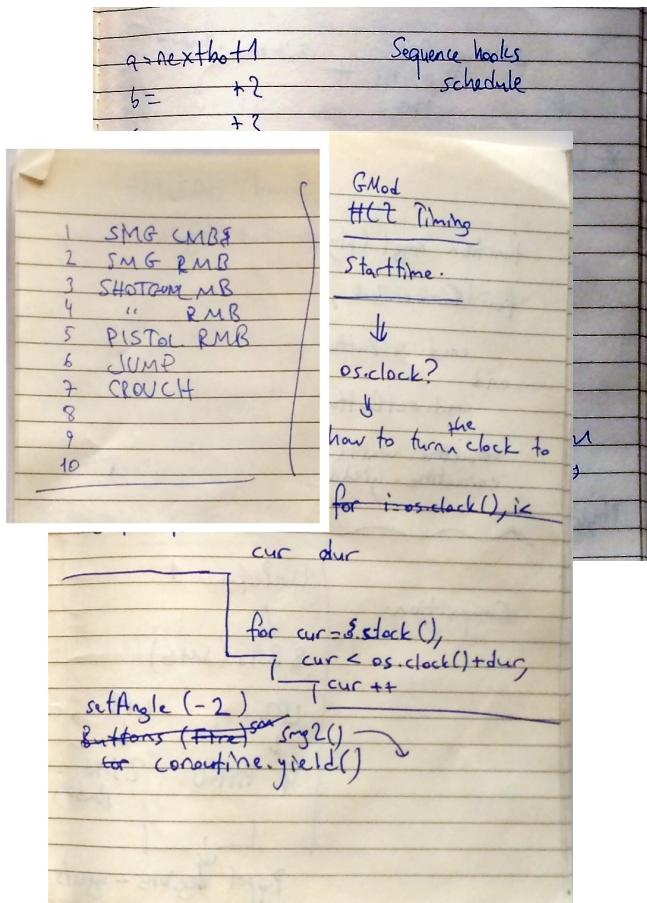
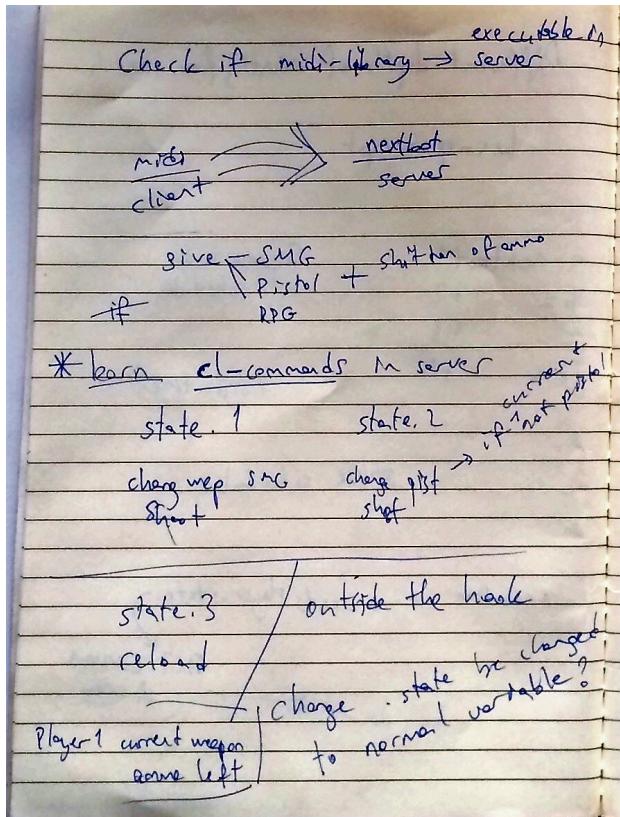
Video 7: Keypad_demo



Geometrically placed bots, at your service, to be played. If you open a multiplayer game in "gm_construct" map and run "berk_whiteroom.lua" file serverside, you can also make your small (or very long) compositions in the white room; check the code and figure out the key mapping! I also made 3 small pieces with it.

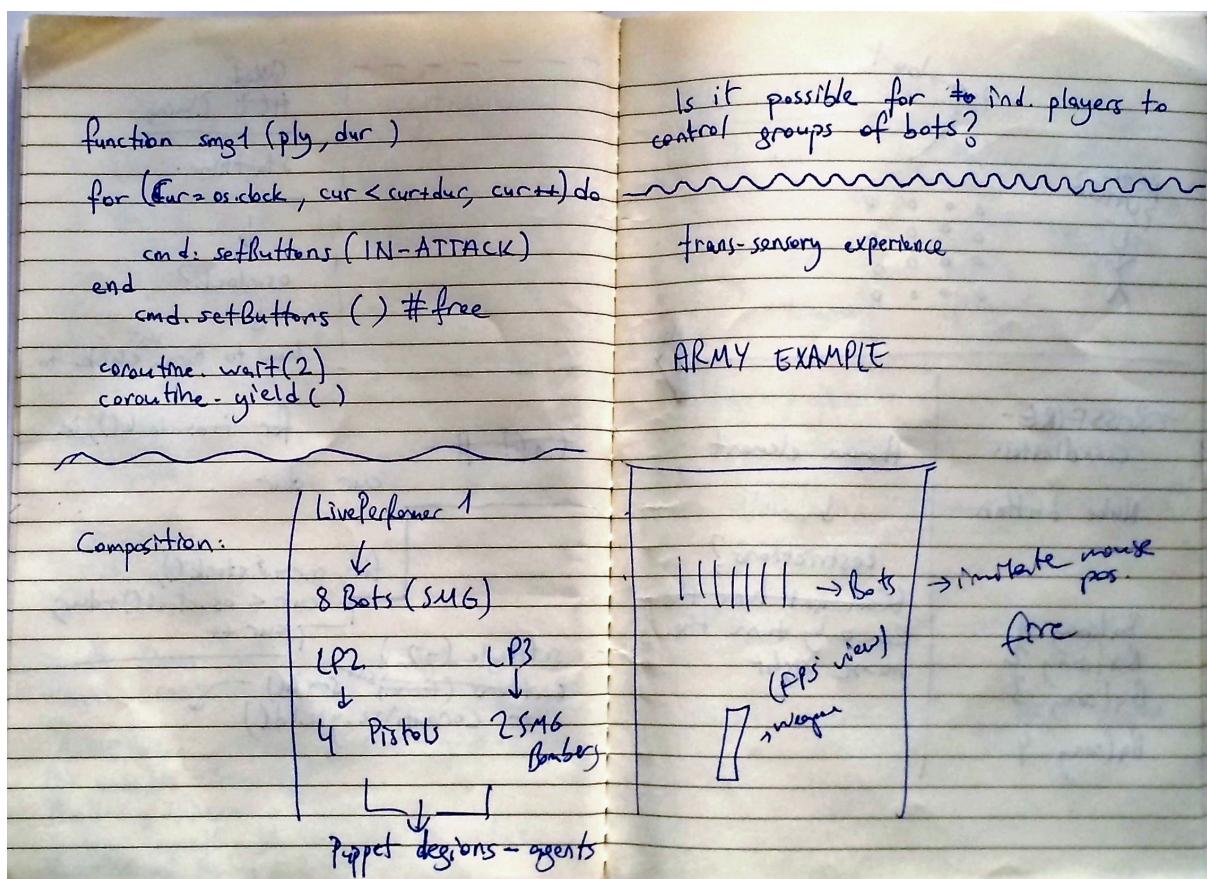
I still had two very critical issues I had to solve: I still couldn't control the bots individually (studying Lua's structure helped me to solve the problem), and there wasn't any way I could find to send time-based commands to sequence bots' actions. Without solving the time issue, I couldn't have full control on the bots, and translate my musical/artistic ideas to their actions.

I had two different possible solutions in my mind:



1- Making a time based system in the code to run the functions sequencally
(coroutines, timers, clock)

2- using a protocol like OSC or MIDI to send commands to the game in real-time.

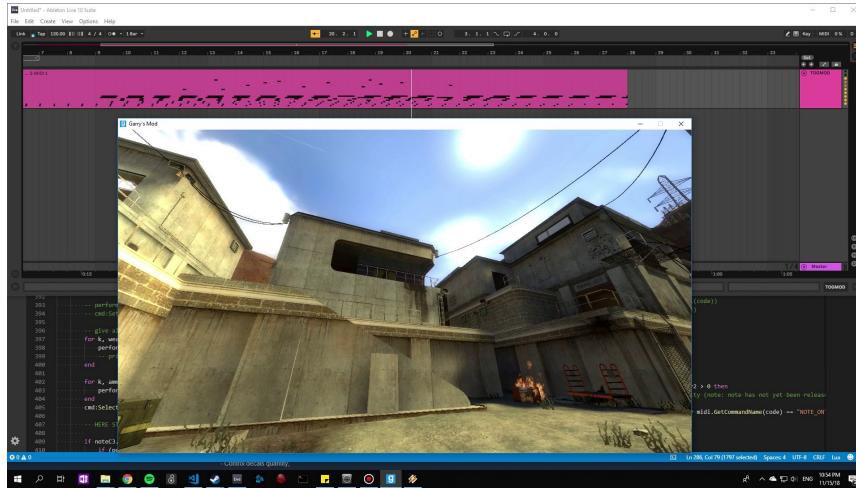


Apperantly, there is a MIDI module already existing for Garry's Mod called gmcl_midi, which allows users to communicate with the videogame using MIDI. As I saw, the only use of this module by the users had been only for musical add-ons to play piano in game. Since my intention to use MIDI was for controlling every individual bot dynamically, I had to structure a new mapping system to assign each possible properties of bots I could control to midi keys and CCs.



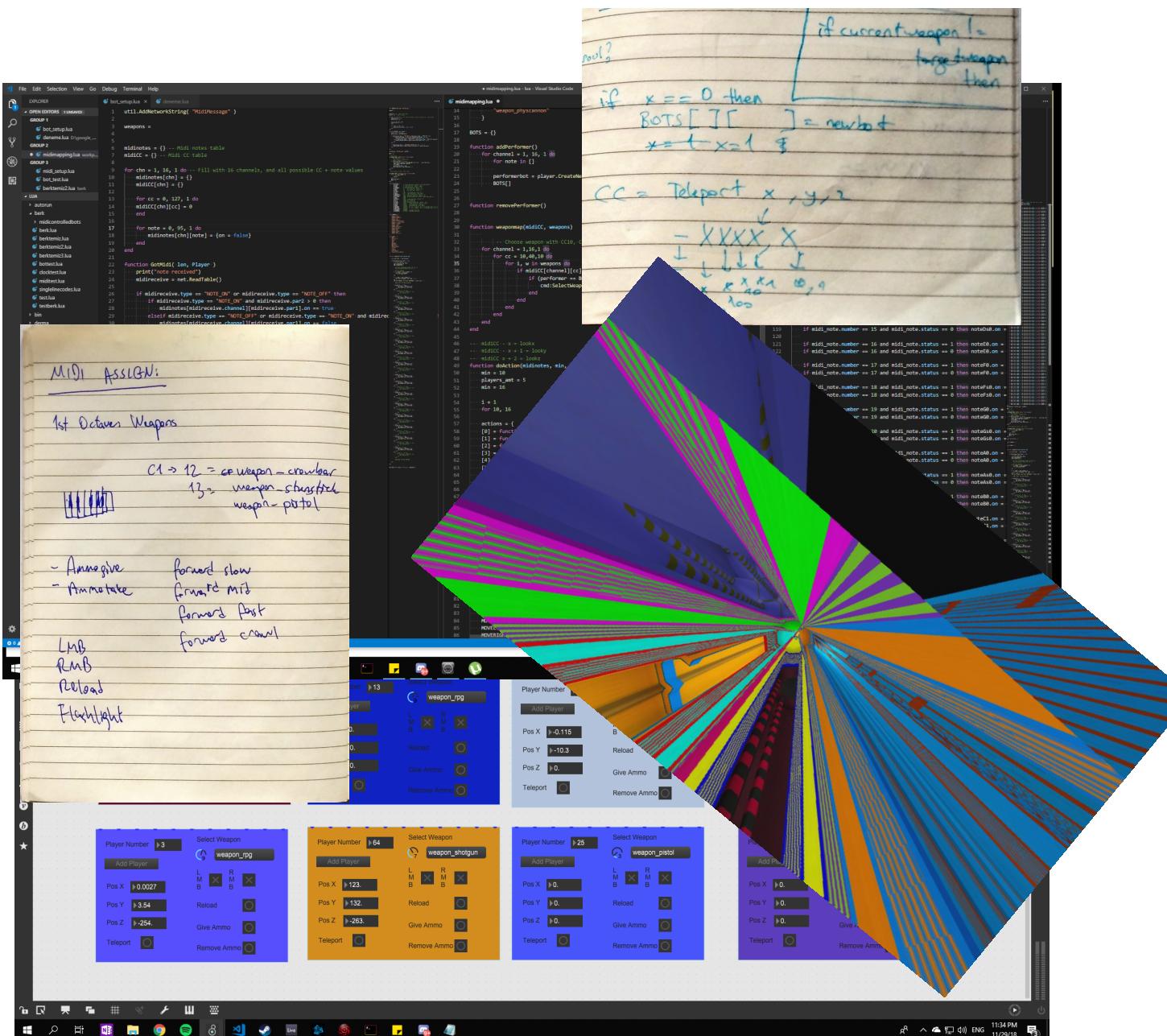
video 9b: midi keyboard test demo

After days of work I managed to get the MIDI module working, I could control several bots individually with different midi keys.



video 9c: short video piece <Awesome Crossfire Solos>

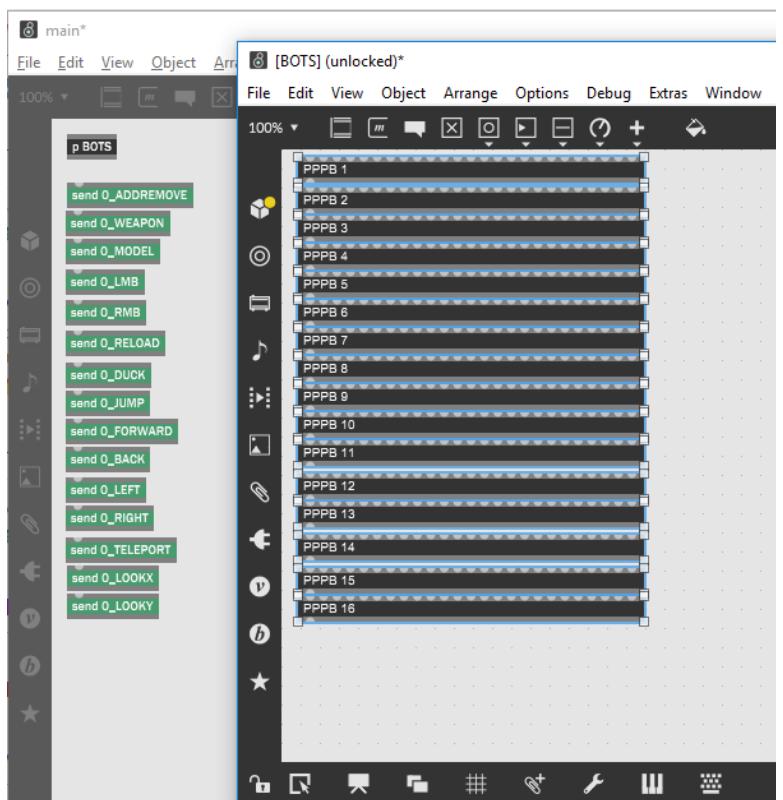
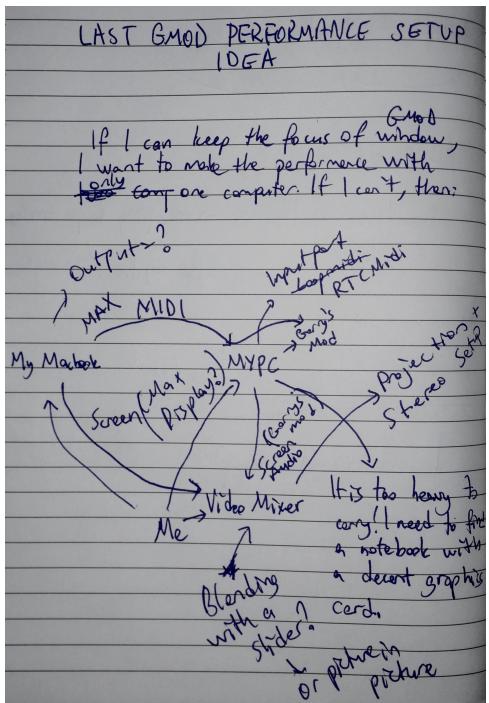
I wrote 2 Lua scripts (one for client, one for server), which allows the user to create and control up to 64 bots individually (theoretically, I don't know how much MIDI information Garry's Mod can handle at the same time).



From now on, I could send MIDI from any device to create and play with bots in real-time. Now, I had to use or create a MIDI instrument to do audiovisual performances and compositions. I'm a Max (cycling74.com) user for a long time, and I decided to create an interface in Max for myself and the people who want to work with bots, because since I made a specific mapping for the MIDI in Garry's Mod, it might be much harder to make something with it without an interface.

I developed a module in Max called "PPPB" which can be instantiated in Max to control a single bot, which is mapped to different MIDI notes and CCs depending of the player number it has been given. This module can be used with the interface, or just be called as an abstraction. I also recorded my experiments during the development of the Max interface, which can be found in the video playlist.

After going through all of this process and making the tools both in Garry's Mod and Max to work with, I started working on my first algorithmic composition in Garry's Mod. I'm already very curious and excited about the result.



This project has been a huge adventure for me: after months of research and experimentation, getting frustrated with failures, making decisions and changing them over and over again, learning a new programming language, coding for sleepless night, I ended up with something I couldn't imagine. My initial plan was to write a composition to be performed in the game by real human performers, and now I have an instrument which anyone can play however they want. I am also sharing the source code of all the scripts / patches I made; and I'm excitedly looking forward for other people contribution. This project still holds a lot of potential, and I can not foresee how far it can go from here.

Here are the links for the github repo and youtube playlist:

Github Link: <https://github.com/vortextemporum/PPPB/>

Youtube Playlist: <https://www.youtube.com/playlist?list=PLL-JW4H9We1zRa491Rav-emuWGOD8SQm1>

