Colton Gowans Time Log - CPSC 599 F24

Date	Time spent	Doing what
Sept. 9th	1hr	Read manual (skipping BASIC stuff (no pun intended))
Sept 10th	2hr	Same as above, and trying basic BASIC programs from the manual
Sept 13th	3hr	Setting up workflow: installing dasm, crafting a makefile, using emulator on servers, finally running own prog!
Sept 14th	2hr	Messing around and writing assembly progs that mostly just printed out chars
Sept 16th	2hr	Looked at audio, made a few progs that play different audio
Sept 20th	2hr	Made prog that puts custom characters into graphics memory
Sept 22nd	10hr	Started term proj pitch video
Sept 23rd	3hr	Finished term proj pitch video
Oct 2nd	3hr	Set up assignment 2, worked on custom character programming
Oct 5th	6hr	Created a custom character generator in the form of a python gui application. Also made custom level editor in same form
Oct 6th	8hr	Refined level editor (import/export + etc.). Created representation of title screen.
Oct 8th	10hr	Worked on input handling, fixed issues with level editor, fixed export to optimized asm code, started on cursor movement
Oct 9th	6hr	Finalized title screen, finished picking up and placing portals. Improved input checking and cursor movement logic
Oct 10th	4hr	Fully fully completed title screen and added new characters. Cleaned up code. Addressed import issues in level editor. Commented stuff
Oct 25th	1hr	Setting up repo and stuff for A2

Oct 27th	8hr	Setting up more stuff, porting files from A1 that will be useful, figuring out how exomizer works
Oct 29th	2hr	Refining exomizer approach
Oct 30th	8hr	Tried huffman encoding, fell back on RLE
Oct 31st	2hr	Cleaning up stuff and doing everything left for A2 submission
Nov 3rd	1hr	Setting up repo for final project
Nov 4th	5hr	Ported over code from other assignments, got title screen working
Nov 5th	6hr	Making level template + level data and drawing it to screen
Nov 6th	6hr	Added every other feature that exists at WiP state. Commented + cleaned up everything for submission
Nov 10th	4hr	Added a proper level template drawn from encoded bin
Nov 11th	10hr	Tons of fixes/features into screen editor such as better exporting as asm code and converting screen files to json, organized asm src code, added constants and zero page file
Nov 12th	1hr	Polish the exporting screen as asm code, can tweak any encoded bin that gets drawn easily
Nov 13th	5hr	Lots of work regarding the character table and smoothening out the process all the way from making it to using in screen editor to exporting for use in game
Nov 19th	5hr	Lots of work for exporting a screen as level data, working on making that very dynamic and easy to change levels and swap them out
Nov 23rd	2hr	Refactored codebase to have better naming schemes and more consistent naming
Nov 24th	6hr	Can load in levels that are stored as char_code,x,y, as well as making some super awesome helper funcs for utility (f_convert_xy_to_screen_mem)
Nov 25th	4hr	Cursor draws properly, and collides with walls!
Nov 26th	2hr	Laser shooter/receptor added to collision check, getting ready to add everything else in a smooth and seamless and bloated way

Nov 28th	3hr	Can interact with level objects to pick-up and place down (some bugs to iron out regarding covered char and collision with it)
Dec 1st	7hr	Finished off interacting with level objects, they get properly drawn in like every possible case, and never accidentally deleted by cursor. Also discovered how to use ROM characters alongside RAM characters, so remade titlescreen!
Dec 2nd	6hr	Drew every ROM character pixel for pixel… Also added new custom sprites since more space! Optimized use of ROM characters too
Dec 3rd	5hr	Tons of laser drawing logic
Dec 4th	8hr	Lasers get drawn from laser shooters somewhat!!! Many bugs spotted but it works!!! Made reflectors work with the lasers too!!!
Dec 5th	4hr	Fixed lots of sprite related collision issues and added rest of collision sprites, fixed some colour stuff with cursor and updated titlescreen
Dec 6th	10hr	Added portals to the game ! Laser collisions have wayyy less bugs, collided characters draw perfect like 95% of cases, reduced memory footprint of the codebase a lot, added portal collisions (still bugged a bit :( ), efficiently colour the laser path green when level is won, added text on screen after beating a level, Changed all levels and placed some in unused space of RAM char table