Binary Math Game

Summary:
This project was developed for “CME 332: Real Time Computing” at the University of Saskatchewan. We were told we were to develop a simple real-time operating system in µC/OS-II that could be ran on Altera’s DE2-115 FPGAs via first programming it with the Media Computer shim in Quartus. I had the idea of making our project into a video game, which was agreed upon, on the condition that it was an educational video game.

In this game, the player is presented with a simple math question, i.e. “What is 4+17?”, and they are tasked with calculating the answer and submitting it in binary. The DE2-115 FPGA features several toggle switches at the bottom of the board, each representing a binary value that is 2^n (n being the nth switch, n = [0 . . . 18]). Since the answer is 21, or $(10101)_2$, the answer would be to set switches 4, 2 and 0 (because $2^4 + 2^2 + 2^0 = 21$), and then press the submit button. The questions become more complex, with later levels including multiplication, division and negative values, and a scoring system.

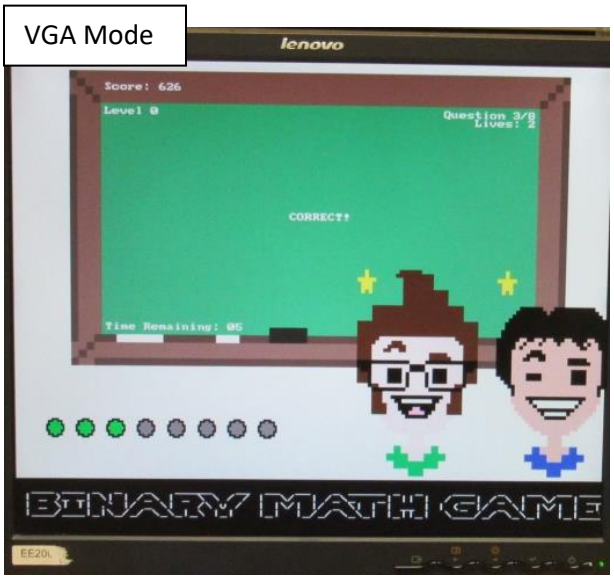
The game features two different boot-up modes; one that can be played self-contained on the board (using the on-board seven-segment displays and 16x2 character LCD screen), and the other which connects to an external monitor to give the game a full graphical interface (VGA mode). Additional programming allowed VGA mode to have a title screen and a high score table.

My work on this involved cloning and modifying existing semaphores and mailboxes for use with VGA mode, creating animations to play on-screen, and learning how to interface with the DE2-115’s video buffer in order to display the visuals (something not covered in the course teachings).

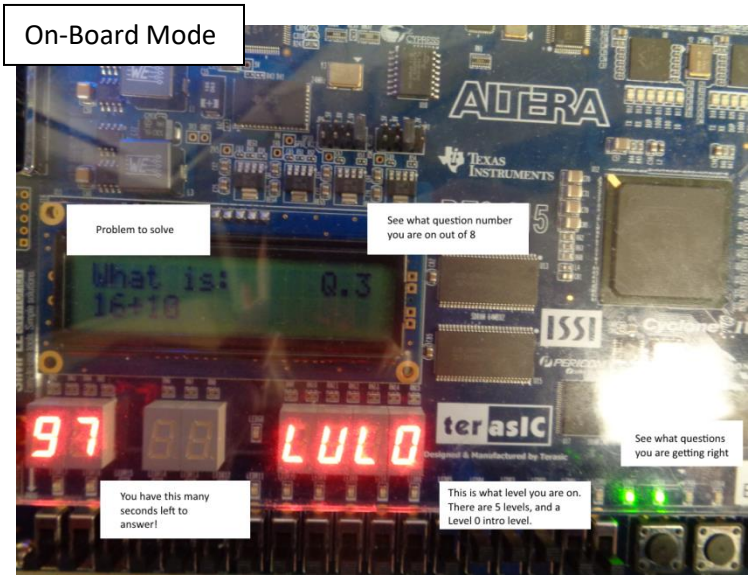
An early-stage port for the Nintendo 64 is in development as of January 2022. It is expected to be completed within the year, and will be console-compatible, able to be ran on unmodified hardware.

Screenshots:

VGA Mode



On-Board Mode



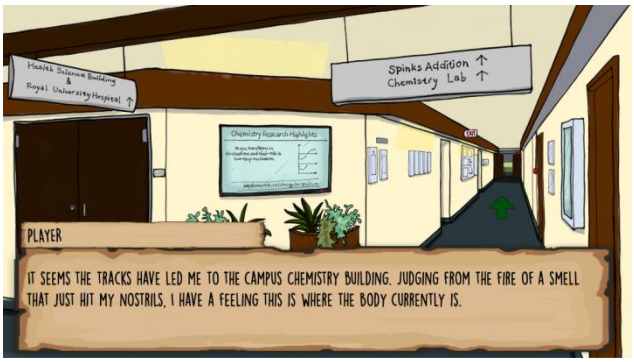
Link:
https://github.com/SpencerLBrown/Portfolio/tree/main/Binary%20Math%20Game

2 Steps Back

Summary:
This game was developed for “CMPT 406: Advanced Game Mechanics” at the University of Saskatchewan as the course’s main project. It is a visual-novel murder mystery game that takes place on the U of S campus. The game plays similar to the “Phoenix Wright: Ace Attorney” series.

I was primarily in charge of writing scripts that handled the main character’s inventory, which was used to handle progression flags at some points in the game, as well as creating the “phone UI” in the game. In addition to this, I was heavily involved in debugging others’ work and doing beta testing to identify and resolve any bugs found (progression flags not triggering, softlocks, being able to repeat segments that weren’t intended to be repeatable, ensuring elements were visually-consistent, etc.).

Screenshots:



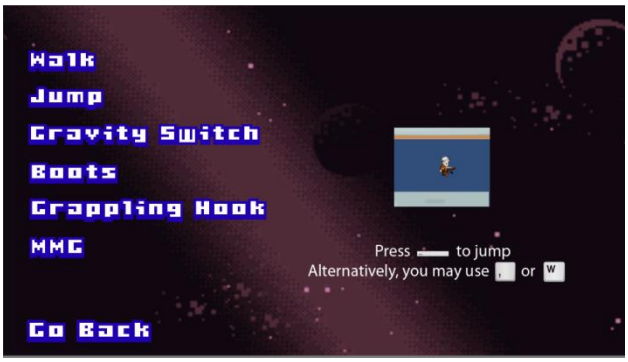
Link:
https://github.com/SpencerLBrown/Portfolio/tree/main/Two%20Steps%20Back

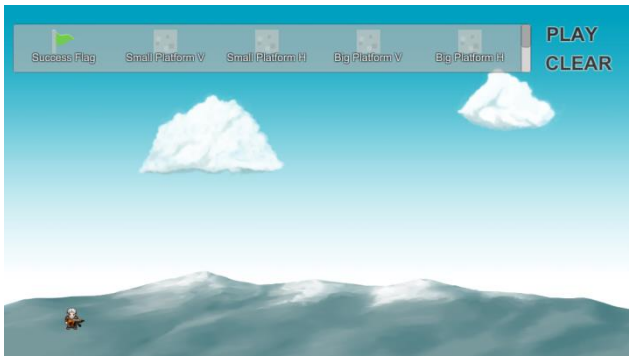
Gravitas

Summary:
This game was developed for “CMPT 306: Game Mechanics” at the University of Saskatchewan as the course’s main project. It was a 2D puzzle platformer game with gameplay elements similar to that of “Portal”.

My primary task was to handle debugging all of the scripts in the game that handled physics, matter-manipulating gun (MMG) controls, as well as writing scripts that handled audio control, the title screen/menu options, and an interactive control scheme menu. In addition to this, I designed the game’s logo, created animations for the title screen (3D parallax scrolling background), and handled audio design, composing some of the songs and sound effects present in the game.

Screenshots:





Link:
https://github.com/SpencerLBrown/Portfolio/tree/main/Gravitas

Design Guidelines for Vestibular Disorders

Summary:
This guide was written as my final submission for "CMPT 480: Accessible Computing" at the University of Saskatchewan. In this class, we were to evaluate a wide range of accessibility issues, as well as programs designed to assist with accessibility issues in the computing field, and find areas that were lacking or did not fully address the issues.

For our final project, we were to investigate a topic related to one covered in class, and discuss it in greater detail. I chose to investigate the guidelines given in ISO/CD 9241-394 "Ergonomics of human-system interaction — — Part 394: Ergonomic requirements for reducing undesirable biomedical effects of visually induced motion sickness during watching electronic images" and expand on them. I found that these ISO guidelines were lacking, focusing mostly on image rotation (pitch, roll, yaw).

I decided to expand on these guidelines to address accessibility issues for users who suffer from vestibular (inner-ear) disorders, as in Canada alone, approximately 35% of the population over the age of 40 have suffered from this disorder at some point (including someone that I know personally).

Through an investigation of peer-reviewed medical journals, new stories, and various other sources, I found that there was no evidence that pointed towards specific triggers, instead finding that there existed many common triggers from various types of visual and auditory stimuli.

After a careful analysis of these common triggers, as well as current accessibility programs that could exacerbate issues, I constructed 18 visual guidelines and 8 auditory guidelines that a developer could follow, should they need the program, operating system or webpage they are designing to be more user-friendly towards individuals who suffer from vestibular disorders.

Link:
https://github.com/SpencerLBrown/Portfolio/tree/main/Design%20Guidelines%20for%20Vestibular%20Disorders

Note: Some elements of the document have been edited or redacted to remove unnecessary academic details and to protect the identity of those involved.

Banjo-Tooie in Italiano

Summary:
This is a console-compatible text dialogue and graphics overhaul to Banjo-Tooie for the Nintendo 64, the overall goal being to translate everything in the game from English to Italian. Expected project completion is Q3 2022.

I happened upon this project when I saw a user in the modding community wanting to make this project a reality (as the original game, released in 2000, never received an official Italian translation), but it was a large amount of work; thousands of lines of dialogue to translate, as well as hundreds of graphics, all needing to be edited to remove the original text, then recreate it with the translation, finally reimporting everything into the original memory address, requiring everything to fit within the size constraints.

I offered my time to this project, as I am a fan of this game, because I knew I had the skills needed to recreate the visual style of the graphics so they were indistinguishable when compared to the originals, and because it gave me opportunity to learn more about how text, texture data, palette data, etc. were stored in some of these older Nintendo games.

Screenshots:

Banjo-Kazooie in Italiano

Summary:
This is a console-compatible text dialogue and graphics overhaul to Banjo-Kazooie for the Nintendo 64, the overall goal being to translate everything in the game from English to Italian.

This project is my own undertaking, having been inspired by the “Banjo-Tooie in Italiano” project. I found that an unofficial translation did exist already for this game, but it did not translate any of the in-game graphics that featured text into Italian.

I contacted the user who led the Banjo-Tooie translation and gave him a list of what I needed translated into Italian and set to work recreating all of the graphics. Since there were similarities between this and Banjo-Tooie for how graphical data was stored in the rom file, it allowed me to work more efficiently, as there was less problem solving involved.

Screenshots: