CLASSIC VIDEO GAME JAM

Tim Wylie
Department of Computer Science
University of Texas - Rio Grande Valley
Edinburg, TX 78539, USA

Abstract

This class is designed as an independent exploration of programming on resource-limited hardware with a focus on a fun application. Students may work alone or in teams.

1 Introduction

The system chosen must be from the 16-bit generation or earlier, e.g., any system up to the Sega Genesis and Super Nintendo. Students are expected to join the Discord and Githubs for the class. Meetings will be every two weeks and at deadlines to give demos of the progress to everyone.

There are eight grades given and the final grade will be an average.

- Proposal (due 2^{nd} week of classes)
- Poster (due at the poster session)
- Final report (due Friday of finals week)
- Final presentation (due finals week)
- Text game (due end of January)
- Fixed-screen game (due end of February)
- Side-scrolling game (due end of March)
- Final game (due end of April)

All items should be in separate folders within your github repository. There is no requirement on the languages or libraries used to program for the system.

1.1 Formal Proposal

This should cover the system chosen, why it was chosen, the libraries to be used, the ide, the emulator, and any other needed software. It should cover the 4 games to be created. Given that the first game is due at the same time, it should cover it in more detail. The other three games can just be ideas/sketches. The proposal should give a timeline over the semester.

1.2 Poster

There will be a poster session roughly halfway through the semester. This should outline the same items as the proposal with more emphasis on the games. At least two of the games should be finished by this time. Further, the poster must be approved by me before it is due and corrections or changes made.

1.3 Final Report

The final writeup is an exercise in technical writing and should be a minumum of 15 pages. It should summarize the entire project including all aspects mentioned in the formal proposal as well as skills learned, obstacles faced and overcome, obstacles faced without solutions, modifications to the original proposal, and a self-reflective summary of how the semester progressed and how the student felt they did. This will also discuss the 4 games in detail with screenshots/etc. and why certain design choices were made.

1.4 The Games

1. The first game is simply to ensure you have your development environment setup and have started learning the system. Requirements are that there is a title screen, and that some simple text-based game follows. Examples may be a number-guessing game, a dice-rolling game, or anything really.

2. This game starts with sprites and some form of gameplay. The game should use a fixed screen environment (the screen world geometry never changes). This was true of most classic arcade games. The requirements are at least one player controlled sprite and at least one independently moving sprite as well as a sound effect. The game should have some sound. Sophisticated examples of these types of games are tetris, pacman, space invaders, frogger, snake, break out, asteroids, pong, etc.

3. For the next game we want to explore a wider world with a character sprite. The requirements are at least one player controlled sprite, at least one independently moving sprite, some feedback sound, and the ability for the character to move through the world causing the background to change. This can be a simple side-scroller, or something like a dungeon crawler where the entire screen refreshes when the player moves off-screen. An examples of a side scroller is Super Mario Brothers and an example of the crawler is Pitfall. It should also include sound effects and music.

4. The last game is open and should demonstrate all the techniques learned to create a final masterpiece.

1.5 Some Tools

- Making Games for the NES - Steven Hugg
- Making Games for the Atari 2600 - Steven Hugg
- Programming Games for Intellivision - Oscar Toledo
- GB ASM - eldred.fr/gb-asm-tutorial
- Gameboy - gameboy.mongenel.com
- AtariAge - forum for programming classic video games (atariage.com)
- SNES Programming Wiki - en.wikibooks.org/wiki/Super_NES_Programming
- Sega Genesis Programming - huguesjohnson.com/programming
- NES Programming Wiki - wiki.nesdev.com/w/index.php/Tools
- 8 Bit Workshop - Browser IDE for several systems (www.8bitworkshop.com)
- GBDK - GameBoy Development Kit (www.loirak.com/gameboy/gbprog.php)