CS212

CS212GBA - Assignment 2

Spring 2008

1 Introduction

In this assignment you will be writing a simple game by implementing a sprite object and a background object. We will provide the framework and the game internals, you will be asked to implement the objects that are outlined in the sprite and background header files. When you have successfully implemented the sprite and background objects you can compile them with the provide source to create a simple, but playable, Mario-like game. It is a goal of this assignment to get familiar with the means by which sprites and backgrounds are controlled on the GBA. The manipulations you do here are fairly straightforward, but take the time to understand them, as you will need to make more a comprehensive sprite manager in the next assignment.

2 Game and Graphics

We have provided an implementation of the game.h header that creates a simple Mario-like game. The source is provided so that you may reference it later on in your own game design, and in the design of your sprite manager, but it you should not modify it for this assignment, as you will only submit your implementations of the sprite and background objects. We will also provide an implementation of the graphics.h header which will control the loading of sprite and tile data into memory. The source will also be available for this, so that students may build upon it in later assignments.

3 To Submit

3.1 sprite.cpp and sprite.h

The sprite object represents a simplified sprite manager. It controls the sprite data in memory, as well as providing an interface for adding and manipulating sprites on the screen. We have provided a fully specified header file for the Sprite object, except for the private members, it is up to you declare them. You'll need to keep track of the various sprite options set, the location and action of the sprite on screen, and various things accounting for the state of sprite memory. When deciding what private members to implement, you should read the specifications of the methods we are asking you to implement for ideas on what members you may benefit from having. The addition of private members is the only change you should make to the header files. Any other changes should be confirmed with a TA first. It is important to recognize how the sprite object is intended to be used. You should somehow keep track of whether sprite data has been initialized yet. In order for any sprite objects to work, the data in memory must be initialized first. So you must begin by implementing the initialize() and commit() functions, so that they may be effectively called from actual object instantiations.

3.1.1 Animation

In the header you can see the setAnimation() and animate() functions, as well as some members referring to frames. As an OPTIONAL (extra credit?) part of this assignment, you can implement animation of sprites. Animation works simply on the GBA - Load your frames into memory in sequence, then increment the sprites' attribute 2 at an appropriate interval.

3.2 background.cpp and background.h

The background object represents a single one layer background. Since the game that you are implementing only needs a single one layer background, we can create a simplistic model for just that one background. We need not design something that is widely extensible, it is more important to be become familiar with the Screen and Character Base Blocks. First start by declaring some private members that represent the screen and character base blocks, so that you can implement initialize. You should recognize that there will be only one instance of the background, and you should enforce this invariant.

3.2.1 Scrolling

In the game, the main sprite can move off the screen, and the 'camera' follows them. This is done by scrolling the background. The visual results of the scrolling are based on the contents of memory, and the settings in the mode line, but the method of scrolling the screen is always the same. There are two registers that control the vertical and horizontal offset of each background. Modify the appropriate register with the desired offset and the hardware takes care of the rest. In the case of this game, the background will loop around so that when a certain point is reached you find yourself back at the beginning of the stage.

4 Notes

There are, of course, many ways to go about implementing the solution to this assignment. If you wish to modify the other files we have distributed, please check with a TA before you do so. This assignment will require you to get very familiar with the GBA hardware, so it is important that you get started early, and if you hit a wall, come to office hours sooner rather than later. Post small questions to the CS212G so that other students may benefit from the answers, and also make sure you check the newsgroup for an answer to your question before you ask it. You will need to make a strong understanding of this material, as you will be doing and expanded version for the next assignment.

4.1 Partners

You can work with one or two partners on this assignment. Make sure that you and your partner(s) have joined together as a group on CMS before submitting.