CS2261 Media Device Architecture – Professor Olufisayo Omojokun

Reviewed the syllabus and other class information (posted on the homepage of T-Square):

* Quizzes taken during lecture, about 8 of them
  + Dates announced in class, quiz length won’t be too long (won’t take up whole class). Dates of exams and other projects will also be announced in class.
* In the final project, which will be discussed later in the semester, we will essentially be building a game.
* Professor then reviewed some slides, which are posted under Resources
* Piazza will be set up later
* Lecture includes: slides, live coding, interaction, quizzes
* Lab (TA led) includes: help, assignments, demos, receive quizzes back, bonus game features
* Attendance is only taken during the final project demos
* This class will focus on more low-level coding and we get to see fundamentals. Will program directly on machine. Get to understand more about what’s happening behind code.
* Using GameBoy Advance (GBA)

GBA (GameBoy Advance) Basics:

* John von Neumann: mathematician, physicist, computer scientist
  + Most famous for his model of modern computer architecture.
* What architecture do we use?
  + Components expected in a modern computer
    - Processor – does math, logic, testing (=0)
    - Control Unit (like the conductor, fetches instructions and decides what needs to go where) and Arithmetic Login Unit (ALU, does calculations) in the Processor. Both work together to process instructions.
    - Memory Unit – stores data and programming, CPU loads info and writes memory unit, also where program interfaces with I/O
    - Input and output – interaction with device
  + Many machines follow this basic modern architecture. GameBoy is one of them
* Why are we using the GameBoy in this class?
  + It has no operating system
    - So, we are programming to metal (code directly impacts hardware)
      * There’s not much separating your code and the hardware.
    - Helps you understand what an operating system does
  + It makes C more obvious
    - Fairly low-level language
    - C is legitimately used in programming GameBoys
  + GameBoy is slow (so we get to see limitations in time vs space), understand performance tradeoffs of your programming decisions
  + And it’s fun 😊
* Processor:
  + Arm-7 3 stage pipelined RISC processor
    - Advanced RISC Machine
    - UK-based
    - Just design chips and sell licenses to manufacturers
  + Vary popular processor
    - PDAs, cellphones, portable gaming
  + 16.78 MHZ clock speed
  + Datatypes
    - 8 bit – char
    - 16 bit – short
    - 32 bit – long
    - No floating point processing
  + Addresses are 32 bits but does not really have 4GB of memory
* Memory:
  + RAM – random access memory
    - “random” - meaning that you can access any piece of memory just by using the address, Different addresses take the same amount of time to access (no cost between one versus the other), random means more about flexibility
    - Go to any location to read or write
    - Not persistent, not permanent
  + ROM – read only memory
    - Can’t be modified, can only read it and not write it. Is persistent
    - Retains content even when you power off device
* Input: 10 buttons (start, A, B, etc)
* Output: what you see on the screen

End of Class:

* Ran a demo on a GameBoy Advance emulator, ran a .gba file that was called Elements of Darkness
* Will be writing C files representing our games