Department of Computer Science, Clemson University
CpSc 4160 – Data-driven 2D Game Development with C^{++}11
Syllabus: January 10, 2018

Potential Video Game Topics

1. Introduction
2. Drawing with textures
3. blitting
4. Data-driven programming with XML
5. Colorkeys and transparency
6. Loading image formats
7. gimp: image processing
8. Inkscape: vector graphics
9. Sprite animation
10. Game time and timers
11. Sprite clipping
12. Backgrounds & virtual worlds
13. Tile editors and Platform games
14. Bouncing sprites
15. Sound & Music
16. Parallax scrolling
17. Painter's Algorithm
18. Explosions & memory management
19. Object Pool
20. Artificial intelligence
21. Processing input: keyboard, mouse, joystick
22. Adding sound
23. Collisions, collision detection
24. Particle Systems
25. Adding text
26. Pausing the game
27. Game levels
28. L-Systems

Language Topics

1. **Basics:** I/O, data types, iteration: for, ranged for, while; control structures: if/else and switch; short-circuit evaluation, functions, The 4 parameter transmission modes, C-strings, command-line parameters, intro to files, constexpr, nullptr,
2. **The C++ Class:** Constructors: default, conversion and copy. Class instantiation, constructor initialization vs assignment, when are constructors called, which one! destructors and when they are required. Also, the class members that are supplied by default, those that you should supply. Shallow vs deep copy, orthodox canonical class form, functions that C++ silently writes, overloading functions, overloading *operators*, overloadly assignment, the output operator and others. friend functions (functions that are members of a class vs part of a class), nomenclature and programming style, overloading a class for binary arithmetic, make files, dynamic vs static storage, Writing a string class. Deep copy vs shallow copy. Dynamic memory allocation. Comparing C string with C^{++} strings. iterators, the stack class, the template stack class, exceptions, the linked list class,
3. **The standard library:** Containers: `vector`, `list`, and `map`. Standard library iterators and algorithms for containers. Insertion & deletion. Sorting, searching, function objects, lambda functions.
4. **Inheritance:** What are the kinds of inheritance, what are the kinds of functions that can be involved in inheritance (virtual, purely virtual, non-virtual), public inheritance vs private inheritance, when should inheritance be used, what are the problems with inheritance, casting down the inheritance hierarchy, the 4 kinds of cast in C++, passing parameters to a base class.
5. **Design Patterns:** Model-View-Controller, Singleton, Composite, Factory Method, Flyweight, Observer, Strategy, Command, and Visitor.
6. **Generic Programming:** Template fns & classes.