Programming Project #4
CpSc 8700: Data Driven Software Development
Computer Science Department
Clemson University
Building Depth;
Sorting and Searching
Brian Malloy, PhD
November 2, 2015

In order to receive credit for this assignment, your solution folder must be compressed and copied to the web handin bucket by 8 AM, Thursday, November 12^{th}, 2015. If you cannot make this deadline, you can receive 90% of the grade by submitting your assignment within a week of the due date.

For this assignment, you will begin to build an actual game that includes a HUD and a player object. In addition, you will create the illusion of depth in your game world using parallax scrolling and Painter's Algorithm. A summary of the requirements for this phase of the project include:

1. Use background layers to Incorporate parallax scrolling into your game. You'll need at least two background images. Note: World::factor is backwards, and the code changes s/ take about 1 minute.
2. Incorporate a player object into your animation (encapsulated/class); use asdw to control the player.
3. Write a Hud class that uses aaline, a similar line drawing routine, or some other scheme to provide a Heads Up Display, HUD, that appears/disappears when the game begins and when the F1 key is pressed (see Figure 1). Your game should begin with the HUD shown in the upper left corner, and then disappear after 2 seconds. The player should be able to toggle the HUD by pressing F1. An advantage of using aaline is that your HUD will be reconfigurable by modifying the XML specification.
   The HUD should be a rectangular shape within which you display the average fps, elapsed seconds, and information about how to move your player object so that the TA and I can test your game. Optionally, you might also want to add a health meter to your animation that appears or disappears at strategic times in the game. you can find examples of aaline, a HUD, and a health bar, in the repo at:

   `8700-2015assets/examples/sdl/huds`

4. Use Painter's algorithm to draw some of the objects in your animation. You will need to derive a strategy for storing the *painted* objects in your animation, and you may need to write a function object to sort the vector that holds the objects involved in Painter's algorithm.
5. **video** (10 points) (1) Do not generate frames, but F4 does; (2) filename has format <**username**>.{*nnnn*}.*bmp* (where **username** is Clemson userid); (3) set frameMax to 300;
   (4) Your name printed clearly (font color/size) at the top center of the screen.
6. Include, as part of your submission for **this** project, a paragraph that describes the game that you would like to build for your final project. Include details about actions in the game, your sprites, how you will keep score, and how the game will conclude. Include the paragraph, your sprite source, etc., as part of your README file.

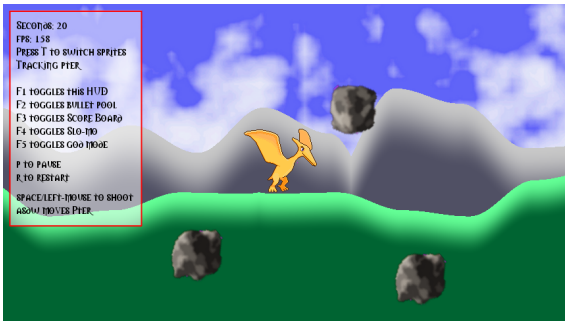

Figure 1: **Heads Up Display**. HUD example in upper left corner.

7. If you would like to work synergistically with a partner, than this project is a good starting point for your collaboration. Please be aware that I will expect a little more from those working as pairs.

As you build your solution for this project, try to use proper C^{++} techniques, and to exploit good object oriented principles. For example, your goal should be to write classes that "take care of themselves." In particular, when you build your player class, place the code to actually move the player in the Player class itself, where you can also eliminate *keyboard bounce*. Of course the Manager class will intercept key strokes and send messages to the Player class about the direction of movement; but the Player class itself should handle the movement and eliminate *keyboard bounce*.

As usual, include an ASCII README file with your solution and submit your project using the web handin command at `handin.cs.clemson.edu`

The Light at the End: Project #5 will entail incorporating projectiles and shooting, collision detection, explosions, sound, and music. The final project, Project #6, will entail incorporating Artificial Intelligence (AI) into some of the NPCs, and a menu system. For the remaining projects, we will use the tracker framework and you will **not** be changing to a new framework. In addition, you may make any modifications or extensions to the tracker framework to accomodate the game that you intend to build.

Extra for Experts

- Incorporate some interesting player movement(s).
- Draw some interesting sprites; take authorship.