

Hui Gao

CS108

Design Document Update 1 - Cameras

Not much has changed with regards to the design of the camera system. The Decorator pattern is used because each type of camera is essentially a combination of varying zoom or scroll functions. For example, if we had a “side-scroll” scroll camera and wrap it with a “tracking” zoom camera that centers around the average of the sprites, we would end up with a camera style that would be like SSB where close combat is zoomed in, long range is zoomed out, and the camera can follow the action as players move across the level. By implementing a few types of scroll or zoom cameras, we can wrap a default camera (fixed point, with no scroll and no zoom) and end up with the various kinds of camera modes available.

In terms of responding to gameplay, the camera should be interacting with the fighter sprites, whether it is centered on a particular player or revolving around the center of the action. However there will be times where we want special or manual zooms, for instance if a player pauses, the menu can zoom onto his character until the appropriate button is pressed again, or if a powerful powerup is picked up, that can be zoomed in on for a second or so.

A hard problem to consider is that when we zoom in and out, we need to scale the background as well as all of the sprites. This means that for each render/update the camera would need access to the CombatInstance’s state. One possible solution is to wrap each Sprite/Background and override their render methods with a method that uses the Affine Transform to scale the images to the correct size/resolution, before calling the superclass’s render method.

To be honest my conceptual grasp of this design is still extremely fuzzy, but I hope to gain clarity and insight after my meeting with Nathan this weekend.