J. Matthew Griffis
Major Studio 1/ CCLab
2012-12-05

Final Project Documentation

Origin

After working with writing in my first Studio project and heavy programming in my second (along with increasingly sophisticated programming exercises in CCLab), I decided in some early discussions with Ted Byfield a couple weeks ago to focus more on the aesthetic side of things for my third and final project. I consider visual design to be my glaring weakness, as I know little about color theory, typography and other elements of graphic design, and am not confident in making visual art. I'd neglected those elements in my previous work and wished to explore them more this time around; while I don't expect to every develop a passion for kerning, I recognize the value of having some understanding of such.

Development and evolution of concept

The original plan post-Ted talks was to create a series of three smaller projects, each focusing on a different aspect of aesthetics - the first could be an exploration of color theory, the second a closer look at typography, and the third perhaps an emphasis on drawing. Beyond this I didn't really have a clear idea of how each would take shape.

Idea 1: Fight!

Gradually I decided to do something with color palettes for the first project, but what? A montage, poster or other visual composition of that sort struck me as boring. Then, riding on the subway to go to class the morning of Thursday, November 29th, I was struck with a bizarre but engaging idea: why not make a fighting game with color palettes as the characters, duking it out in arenas I drew myself? I could research different color palettes to be the fighters (project 1), design and illustrate the levels (project 2), then make the game itself (project 3 and unifier). The game would support two players on the same computer and each would choose a color palette, which would be represented as the standard rectangular palette of side-by-side colored squares, but they'd be floating above the ground in the arena. Each would have special moves such as typical fighting-game moves (e.g. uppercut or fireball), or perhaps even moves themed around their colors, and they'd duke it out until one emerged victorious. Then perhaps there would be a horoscope-style ending screen that tells the victorious player about their personality based on the color palette they chose.

I floated this idea in class that morning, and at first people weren't sure what to make of it, but they embraced its hilarity and made suggestions such as theming the palettes around art world luminaries (perhaps not even naming them so the player must guess who is who), using shapes in addition to color to reflect the nature of the painter in question, and emphasizing the characters over the mechanics (Ted recommended even just having the palettes run into each other instead of trying to develop an actual fighting engine; he also decried the personality-analysis trope).

Idea 2: Jump!

Meanwhile for CCLab I hoped to produce a final project in openFrameworks, which we'd just started working with. In a recent class example we'd worked with vectors and given elements movement in the z-axis to simulate a 3D effect. I realized I could use this to make a 3D game. I envisioned a game in which the player controls a character navigating an environment and jumping from platform to platform, similar to the games I'd created previously for both classes, except this one would be in 3D and the action would move into the screen instead of from side to side.

However, in class on Friday, November 30th our guest professor encouraged us to consider a final project that would encompass the needs of multiple classes, for the sake of both time management - given how close the end of the semester is - and also polish, since we could make a single project to a higher standard than we could two. I presented my 3D game idea but realized it was pretty generic in my conception, and I wasn't sure how to make it unique, a problem that plagues at least the initial appearance of my earlier game *How to Play*. Given my new consciousness of the proximity of the end, so to speak, and my lack of passion about my separate ideas, I decided to reevaluate the situation to see if I could come up with a single idea that would engage me more and meet the needs of both classes.

Idea 3: Combine! (and radically revise.)

I knew that I still wanted to work in openFrameworks and still wanted to tackle 3D, and I also knew I didn't want to disregard the emphasis on aesthetics, so the question became how to marry those elements. I'd planned to make the fighting game in Processing and it would be 2D, so I didn't think it would work very well in this new context. Similarly I couldn't conceive of a 3D platform game featuring an art-infused mechanic that I could achieve in the time available.

Within the last few days, I was struck with inspiration - I could develop a sort of 3D museum environment in openFrameworks, in which the player moves an avatar down a single room, at the end of which is a blank canvas. By interacting with certain objects, the player causes different Old Master paintings to appear on the canvas. The nature of these objects I have yet to determine - ideas so far include balls of paint colored according to the appropriate palette and objects shaped and colored appropriate to the painter. Just so with the nature of the interaction - perhaps the player picks them up and drops them in front of the canvas, or kicks them into it like a soccer ball.

In this concept, the product is less a game and more an interactive piece, which is OK, but I may try to find a way to incorporate game elements because I like them. I'm a big fan of the video game series *The Legend of Zelda*, and one of the recurring objects in the games and lore is the Master Sword. I'm also a big fan of puns, and the opportunity to produce the Old Master Sword is too good to pass up. What would this do? Who knows? I'll find a way.

Making it

Yesterday (2012-12-04) I proved most architectural construction times are greatly exaggerated by building a museum in one day. Take that, Frank Gehry! More precisely - and working entirely with placeholder visuals for the sake of establishing proof of concept - I created a sphere in openFrameworks to represent the player, then made a box to be the canvas on which the paintings would appear, then found a picture of the Mona Lisa and put it on the front of the box. I gave the player-sphere movement left and right and forward and back and exploited optical illusion to give the illusion of movement forward while preserving the size of the player-sphere onscreen by moving the canvas and its painting closer to the player-sphere whenever the player presses the forward key (and the reverse for the backward key, of course). I also positioned the sphere to appear to be level with the base of the canvas (so it looks like they're both standing on the same floor).

The basics being established, I began to construct the walls. Initially I struggled with how to do this. I'm new to openFrameworks and I wanted to draw rectangles oriented in the z-axis rather than x and y like rectangles typically are. I tried rotating but that interfered with the movement on the z-axis necessary to scale the objects as the player-sphere moves forward and backward. Eventually I decided to use the function to draw any shape, which did the trick because I just had to specify the vertices of the object (which could be located anywhere, including the z-axis) and OF would do the rest. I built the floor this way, and after positioning it and ensuring the scaling worked correctly with movement, I added an element of gravity and collision detection to let the player-sphere rest on the floor (rather than just position it where the floor would be). Then I added a jumping mechanic because jumping is fun.

I spent some time making the code more readable by renaming lengthy positioning descriptors as their own variables with more intuitive (and shorter) names, which paid off because then I was easily able to construct the remaining surfaces (using different colors for each surface to show where one ends and the next begins), with the result that by the end of the night, I had a fully-contained 3D environment in which the player-sphere could move and jump right through the Mona Lisa if desired. And, hey, who wouldn't desire that?

So far today I have started researching additional paintings and have already added Mondrian to the cache. Then I figured it was probably time to start with this documentation, so here we are.

2012-12-10

I updated the colors to something a little less appalling and have added in WASD controls in addition to the arrow controls - this will facilitate playing the game with either hand and also allow easy use of the mouse, which will be necessary for using the new grappling hook item I am envisioning. Grappling hook, you say?

Yes, I have decided to give this game an art theft motif, as that seems to be what's bouncing around my head lately. I envision a small series of rooms functioning as a training/simulation facility where novice thieves can learn the tricks of the trade. The game opens with a video of me giving an introduction to the player in the role of novice thief, after which play begins. Perhaps the first room involves precision jumping to avoid tripping moving laser grids, the second room requires use of a grappling hook to cross a great distance (perhaps with a humorous note about why the player couldn't use that in the laser room), and the third room features a confrontation with a guard. By successfully navigating the obstacles in each room the player can steal the painting waiting at the end. I have various ideas to increase the complexity such as permitting the player to fail but docking them portions of the painting or other such scoring penalties. I'd like to incorporate a room with spikes since that's a classic of platforming, but what kind of a museum is this, anyway?

[*Update*] I suppose I shouldn't be surprised, but it took me the entire day to get the very first obstacle, moving laser grid, to work properly. Frankly I'm very happy it's working at all. For the most part my difficulty was twofold: not knowing the syntax of openFrameworks, and (more troublesomely) not really understanding how various functions and methods were...functioning and methoding. For example, I had relatively little trouble getting the lasers to show up and even make them slide across the screen (that is, stretching across the floor to form the grid), which looks great, but I ran smack into a brick wall trying to get them to move on their own (in addition to the movement relative to the player that every non-player element has, providing the illusion of player movement). After a tremendous amount of frustration and trying all sorts of things, I finally realized it was because I was using a single variable where I needed an array of multiple variables, I needed to move code around, and various other deep tech explanations that aren't really very interesting here. Suffice to say that it does work at last, though I have to iron out some kinks in the collision between player and laser. I also want to add lasers that run back-to-front (the current batch goes side-to-side), but it should work exactly the same way as the ones I already have, just on different axes. Still, given the time constraints I may have to satisfy myself with a laser grid training room project.

2012-12-11

I had an idea to fix the collision issues, which was simply to increase the range of z-positions that would trigger a collision, and it works beautifully. One problem easily solved. Now to introduce some delay before the lasers start moving for visual panache.

[*Update*] Success! The lasers now stretch across the screen, pause for a second, then start moving. Looks great. Time to add lasers going the perpendicular direction. This *should* be relatively easy. I've said that before, though.

[*Update*] I had to redo how I provided the information for the lasers in order to enable drawing them in a different orientation using the same laser object, but that was easy enough (and now the code is cleaner - always a good outcome). I have succeeded in drawing perpendicular lasers! Now to make them move.

[*Update*] Total success! The greater understanding I achieved after all that struggle yesterday allowed me to easily implement the second, perpendicular set of lasers using the same methodology. The code is now more optimized too, so that if I should desire to add additional lasers going in other directions I could easily do so. I have modified the room's dimensions to be proportionate to the number of lasers. It's pretty tough to get to that painting without tripping a laser, even for me, so the game is probably too hard. But it may be just this one level for the time being so that's probably OK. Now I need to decide how to proceed.

[*Update*] While I could spend an infinite amount of time coming up with and refining new mechanics, the original goal of this project was to work more with aesthetics and focus on polish beyond the technical, so that's what I'm doing. I have an idea for making a more distinct protagonist than a humble sphere (which I have yet to attempt), but in the meantime I've added windows with textures attached that change based on movement - it's not quite the effect I wanted (stars scrolling in the background), because I don't really understand how the mesh technique works and am simply going off the basic introduction given to me by Patricio, but it does give the impression of windows or possibly screens, so it make the environment feel a little more dynamic and interesting. I added windowsills as well and changed the colors to make the room seem dim like it's night (which the outside texture implies). I even used OF's native ability to restrict the range of displayed colors to limit the colors of the painting to those tending more towards the grayscale, with the convincing result that the painting appears to be under dim light - no image editing required. The lasers look great against the darker background.

[*Update*] I've simulated a pulsing effect by switching the lasers' color on a timer. It's almost imperceptible, which is just about right to look natural. I also added a restart command, which I ought to have done much sooner than this.

[*Update*] I realized that the windowsills looked more like the bottoms of the windows were recessed into the wall...so I capitalized on that by applying "windowsills" to three sides of each window and now the windows appear to be recessed. I also added skylights with their own faked recession into the ceiling.

[*Update*] I have further modified the colors - it turns out everything looks amazing and the windows look much more convincing in almost-total blackness. I'm not sure whether I'm imagining the pulsing effect as actually being perceptible, but I'll leave it be for now. I am having an unusual problem with the restart command, which is mapped to a button - the painting disappears when the player collides with a laser (good) and on occasion when restarting the painting will not reappear (bad). After troubleshooting I've identified the issue as only occurring when the player hits the restart button while colliding with a laser. I'm honestly not sure how to fix this; the various strategies I've tried haven't worked, and this is clearly a quirk of OF's inner workings, since the logic is sound.

However, I've just had a *brilliant* idea for a total cheat that not only gets around the problem but even facilitates the narrative of a simulation environment. Since the restart works correctly when the player is not in contact with a laser, I will put a button at the entrance to the room and require the player to be in proximity to that button (safely out of lasers' way) before the restart will work. There will be text on screen and some kind of prompt that indicates the player must return to the entrance to restart.

What's brilliant about this is that it also fixes another problem I'd been thinking about, which is how to handle music. I want to introduce a music track, and I'd prefer to do so within the context of an object in-game. I had the idea of a radio you can turn on. However one of the recurring problems with playing a music track is that every time the player starts the game over, the music starts over, and if the player is restarting frequently they quickly get sick of the music's intro. I played the game *Hexagon* recently and it very cleverly addresses this problem by starting the music at a random point at each restart, which not only avoids annoying repetitiveness but also gives the player the psychological illusion of having made some sort of progress (not the case when they hear the same three-bar opening over and over again). But how to do this in my game? Well, by removing the true restart command entirely. Instead, when the player hits the button at the entrance, it reboots the painting, but the game can keep running, which means the music can, too. It's a seamless experience. I love it! Now what music to use?

[*Update*] I've chosen not one but TWO excellent musical tracks and integrated them into the environment, cued by button press. I wanted a way to fade out one track and start the other, but I couldn't find built-in functionality for that. Pfft, they call this an obstacle? I wrote a crossfade function myself! It works beautifully. Music: success. Now I just need to draw the radio in the environment and make the button cues proximity-sensitive.

[*Update*] I have drawn two control panels, one for playing music and the other for reloading the painting. Both prompt the player with onscreen, context-sensitive text when the player approaches close enough to use the control panel, and it is only then that the buttons to trigger the respective actions work. I must stop for the night but I should make a list of things I most wish to do before wrapping this project:

1) Must make the player-sphere more interesting. I don't think this will work very well in first-person and I certainly don't have time to attempt any kind of 3D model or proper graphic, but I can spruce up the sphere in comical ways.

2) Draw lights on the ceiling and lighter squares on the floor. This will be relatively easy and should look very cool.

3) More sound effects. Would be nice to have something for the lasers at least.

4) Win condition. Right now there's no way to actually grab the painting and win. End screen?

5) More text. Need instructions on how to move and prompts to reload when a laser's been triggered. Something on the frame under the painting?

6) Recognize when collected painting without music and prompt to try again with music?

7) I would really like to make a short introductory video as part of the “training camp” motif - this could be very funny. Record the video and put it in at the beginning.

8) Should there be some kind of bibliography? Maybe not. There’s not a lot of reference here other than what’s cited in the code commentary.

2012-12-12

Immediately I got distracted from my list above by drawing concentric rectangles around the front of the room to give a kind of “entering the Holodeck” effect. It does look pretty good, though. Then I got back on track by adding the lights to the ceiling. They’re just colored squares and wouldn’t look very convincing, except that the positioning and the coloring help, and then I drew lighter patches in the appropriate places on the floor, and the results look quite good.

[*Update*] I have spruced up the player’s avatar as planned. It’s still a sphere, but I added additional spheres to represent arms and legs. I set the arms to glow green when the player is in proximity to the painting and could steal it (at least once I implement that part). The “character” still looks stiff and not really in keeping with the rest of the aesthetic, but now it’s got some personality. I’d love to make the feet animate, but that would take more time than I’ve got. What I *did* do was add a cape, which swings upward when the player jumps. That looks hilarious and really helps add some dynamism to the avatar, even though the feet don’t move. I also attempted to add goggles by drawing a curve around the “head,” but I couldn’t get the curve to work properly, so for the sake of time I gave up on the goggles. I need to tweak the colors but otherwise the avatar is looking much better.

[*Update*] I tweaked the colors. We'll see how the purple cape goes over. I intend to make fun of it. I also refined the detection of whether the player has successfully navigated to the painting. Now players must be within the actual bounds of the painting (in terms of x and y; they must be very close in the z-axis) and the painting must still be displayed (i.e. they cannot have tripped a laser). If they meet these conditions, the arms glow green, and once I make it possible to steal the painting, they can do so.

[*Update*] Lots to report. Tripping the laser causes a failure message to be displayed on the frame where the painting used to be. The player can now steal the painting. Doing so successfully draws a success message on the frame, and also generates an overall "you win"-type message that slides onscreen, then after a set period of time prompts the player either to quit or restart. The lasers disappear when the painting is successfully stolen. The restart command allows the player to restart after success as well as failure, and in the case of success the lasers are drawn as though from scratch (stretching across the screen once more). All necessary timers and other variables revert to their original values as well so it's completely functional each time. All of which is tech-y talk for saying that you can win the game now, and it's polished. I'd love to draw a crude picture of the avatar on the frame when the player wins, something in the vein of "Burr Ghul wuz here"; we'll see if time allows it.

[*Update*] I added more text. The movement instructions appear onscreen at all times, and when the player triggers a laser, a message appears taunting the player and indicating the console can be used to restart.

[*Update*] I added a looped sound effect for the lasers as though they're humming. Also one for colliding with a laser, and one for grabbing the painting. Game has music and sound. Only a little bit left to do.

[*Update*] I added a system that only lets the player win if music is playing and sasses the player otherwise, because no game is complete without sassing. This is everything on my list complete! With two exceptions, that is. One is the bibliography, but honestly I'm not sure what I'd put on it as this has been more or less the genesis of my own mind over an intensive several days. I've mentioned people in the code commentary and in this documentation who helped me or provided resources (whether they knew it or not, in the case of the sound effects I "borrowed" for this project); I think that is sufficient. The other exception, which I'm sad not to be able to implement before presenting this tomorrow, is the introductory video where I provide a welcome to training camp. Unfortunately there just isn't time as I would need to write the script, record the video, and then integrate it into the experience, all of which is manageable but not before tomorrow. In any case, the project stands on its own, though it occurs to me just now that there's no actual prompt to tell the players what the goal is. Then again, that may be better. If they see a laser grid with a view focused on a painting at the end of a dimly-lit hallway, they're probably going to guess what's up. I will wrap this here and see if I can find a way to put the application file up for people to play! In the following section, I list some ideas I've had over the course of development that I have neither had the time to implement nor described as goals in the documentation above. There's some good stuff there; lots to do with this project if I should choose to continue it (I'd probably need to get my own sound effects...). Thanks for reading.

Ideas (not yet implemented or listed elsewhere)

2012-12-12 Scoring system - time to finish plus multiplier based on laser speed. Ranking system?

2012-12-11 button to increase laser speed? // Multiple levels with same layout - could challenge player to get to the painting multiple ways, e.g. have to jump over x-number of lasers, have to not jump but rather stay in one square (although that wouldn't work because the squares eventually reverse direction...)

2012-12-10 art heist - each room challenges you to steal a painting by overcoming a different obstacle; ~~one you have to jump over moving lasers~~ (I did this one), two you have to jump over spikes (?); three you have to use a grappling hook to get over a pit; four you have to do battle with a guard, possibly by reflecting his attacks back at him (like a beam gun?). guard has a humorous "gonna getcha" speech bubble poking out of his head and staying there and moving with him. // depending on your success, you may recover only pieces of the painting (bits get chopped off; can I do this using mesh? at the end you get scored according to how many paintings you've recovered and what state they're in. // ironic twist - you then get to vandalize them ala Ted's examples? // should this game be first-person or third-person? // hidden paintings? // find the old master sword behind a painting that you steal? // name of protagonist is "Burr Ghul" - too much? // rug on the floor?

2012-12-05 multiple canvases? // minigames when you interact with the paintings // paintings come out of the canvas and attack by bouncing up and down about the room - use the Old Master Sword to smite them?