**Project Statement**

My first project, called UNTITLED MOONCAKE PROJECT, explores a Sisyphean interface in a way that is somewhat contrary to the more nihilistic aspect of both Camus’ essay and the Greek figure’s punitive predicament. The project rather approaches the concept of an enjoyable hell, a repetitive, pointless act that people could hypothetically be enthralled to do *ad infinitum*. This type of mental “prison” is more interesting to explore to this coder than physical types of constrain due to how they permeate human behavior in ways that often gets caught in every self’s blind spots. But instead of unclesaming people into guilt on their trouble with expressing constant reflection, the experience wants to create positive feelings thanks to a fictional creature called Mooncake from Netflix’s *Dead Space*. Behind the character’s *kawaii* appearanceis a planet-destroying force of irony, though the project doesn’t (dare I say yet?) include this characteristic and instead strips Mooncake to its bare essence: the sound it produces.

This light-hearted approach to the concept was concentrated on producing simple generative jQuery element, “bubbles”, that the player pops in front of Mooncake and provoke one of his iconic reactions. The simplicity of the system’s core mechanics – constant generation of floating bubbles within the window, bubbles showing the current onomatopoeia and popping shortly upon click, Mooncake reacting or waiting – allowed me to put some time in the aesthetic side of things, shaping Mooncake’s expressions through Photoshop thanks to stills from the show, as well as extracting and doing minor editing on his sounds. All in all, I feel like what I had in mind translated pretty accurately, minus moments where animations still overlap each other. Doing a project again that dynamically produces html elements was a good way to review the logic behind the html document tree and how to affect it effectively through JQuery. My last attempt at it had been content-heavy and easy to get lost in the repetitive lines, but in this case the bit of code can be iterated and is thus concise and easy for me to go over and fully comprehend. I had some difficulties due to JQuery’s non-animation nature to get things timed properly but I think now it works mostly well thanks to timing the current reaction gif to the current audio’s end.

There is place for improvement and I definitely want to polish this soon. I want to clean the pixelization of all gifs, fix some of the animation, and shift the background to a p5 canvas to regulate the animation more smoothly. Another definite change is to make the eyelids separate from the face to draw the pupils under them, constrained to the sclera, and make it follow the mouse around to make Mooncake’s face more alive. I also wanted to make a popping animation for the bubble for once the text fades out; the idea would’ve been to delete the bubble div and put a gif image in its stead that would animate the exploding div. Another way I am not sure I will go but which sounded cool was to make him become increasingly happier/under the influence (…of soap?) the more bubbles are burst. The latter is sort of silly but don’t know, there was a charm to it and the thought came naturally while working on the “hmmm” animation. Another way this could go in is the opposite, with Mooncake becoming angrier the more the player “pokes” him, until he reaches his final form and “obliterates” the player, ending the simulation. This one is definitely more attractive, simply for the “state” changing aspect I enjoy in creating interactive stuff. But that is beyond the scope of the class…