**Project 1: Interactive Design**

In this project, I made a game using Processing 3. With it I was able to create an interactive environment that increased in difficulty and also forced the user to learn and build skill.

**Coding:**

In order to define how the “balls” were to be made and how they would interact with the user, I made a class called “Barrel” (my original plan was to use barrels in the game rather than balls, but it was easier to do the size stuff with circles) and in it defined how the bouncing objects would interact with the user and how they would react when meeting certain conditions (such as hitting the floor or a wall). This class also contained physics parameters that controlled speed, direction, acceleration, and if the balls would fall through the floor and start at 0.

The main body of coding was where I had the most trouble. I was alright with creating objects (I set the balls to appear at random coordinates within the box and away from the borders, and that wasn’t too hard) and with importing images. Everything else, however, was a pain in the neck, especially figuring out how to add another object at the end of each “turn”. I didn’t know about the “*object.property”*  thing, and it took quite a bit of searching before I even came to that. I learned how to make objects from a class into an array, and I tried different ways of arranging it in order to get the desired effects. A lot of times, the code just wouldn’t run (returning a NullObject something or the other, or an EmptyArray something), and sometimes it would run, but just not the way I wanted it to (showing no balls at all; giving me 100 balls at once, making the ball keep appearing randomly all over the screen…).

For a while, I tried to solve this problem by using append, and later by trying to find a way to check if one of the properties of the class was true for all of the objects at once. However, I eventually figured it out by using an “if” statement inside a “for” loop, and another if statement outside that, all inside another for loop. That took a lot longer than it should have, and I had to make use of the *object.length* variable.

In my code, I also added images for the background and for the objects that the user interacts with.

When a user clicks on one of the balls, it returns a true statement for that ball. This true statement is checked for all the balls on the screen. If that statement is true for all of them, then the screen resets on setup, and the whole thing starts again as a new level. I added a counter for the level in the upper right corner, and instructions at the top of the screen for how to play the game.

**Interaction:**

The interaction between the user and the game is of several types. First of all, I wanted there to be an element of difficulty for the user. As they progress in the game, they also increase the difficulty of what they are doing. The challenge increases as the difficulty increases, hopefully bringing the player in. I also wanted there to be an element of skill building, where the user gets better and better at clicking their target as it zooms past them, as well as doing it quickly enough that they all are clicked before they hit the bottom.

Of course, part of the experience is also frustration; not only to get to the next level, but also for the fact that *there is no end.* This is, to some degree, supposed to frustrate the player until they quit the game or actually *do* pass the level. Maybe some time I’ll make a stopping point, in which the player gets some kind of a reward as a result of passing, say, the fifth level, but as for now I’m satisfied with the interaction annoying the user.

**Personal Observations:**

There were several things I observed personally while playing the game. First, I noticed that, while the first two levels are fairly easy, all the levels after that are extremely hard, even with my boost of “air” whenever a user clicks on one of the objects. Perhaps I could add more time to the “air” after a certain number of objects are on the screen (or not). I believe, however, that frustration is as much a part of the interaction as enjoying the graphics or in honing skill to pass on to new levels. It would be quite the accomplishment for someone to get past five at a time, and that is what I am aiming for.

**Future Changes:**

In the future, I think I would like to make the screen larger. The instructions at the beginning may also disappear after the first two or three levels. I would also like to change the graphics to something more “fun,” like pop can labels bouncing in front of a soda vending machine, or maybe something else equally as fun. I may also add sound effects, especially to parts of it such as when the balls are bouncing through the floor, or when I click on one of the balls. If I go with the pop idea, I would like to have a background sound for the game which is essentially white noise where a bunch of people are talking, but I’m not sure how that will turn out. My idea for the game was based partially off of a Wii game where the player shoots the cans and they bounce up and eventually disappear. However, I didn’t want them to be that easy, and I wanted the user to have multiple chances to get it right. I also got the idea for this game partially from playing jacks when I was in early grade school. I was pretty good at it (if I do say so myself), and it would be fun to do something similar to that sometime. In jacks, the object of the game is to bounce a rubber ball, and before the ball touches the ground again to sweep up first one jack, then two, then three, all until they are all swept up (at least that’s how I played it). If you messed up, it was another person’s turn. Obviously we aren’t picking up jacks in this game, and in my opinion this game is a lot harder. However, it could be an idea for a future game, and it wouldn’t be too difficult to adapt this game into something similar to jacks.

**Conclusion:**

I learned a lot through making this interactive piece of coding. Not only did I make an array of objects, but I also figured out how to check if there was a certain property in all of them, and how to pass on to another “level” (which I figured out completely on my own). I was able to create an environment where a user could do something to an object in their playing window, and the object would respond back. The user was able to increase difficulty as they passed certain levels, and thus was also able to improve their skills as they added more and more balls. I noticed that frustration is a huge component of my game, as is often the case in games with increasing difficulty or in which a certain skill set has to be developed. In the future I may use what I have learned to make more similar games. In short, I’m glad I was able to do this project and am happy with how it turned out.