**super().Project Description:** My game is now much more resembling of an arcade game. The track, as opposed to being a set buggy track, is now infinitely generated. The display is also now in 2.5D and the view of the buggy has changed from an aerial view to more of an isometric view, as if the track is being viewed from above and behind the buggy. Additionally, there is no “winning” – instead, the user will try to survive for as long as they can, and score will be kept track of (points will increase for each second alive and for each power-up hit). As opposed to having different modes of difficulty, I may end up making my game harder as time increases.

**super().Structural Plan Update:** This is going to remain pretty largely the same. The main differences are that I will likely have less UI files since I probably won’t be able to implement as much. I might get in a start screen and a game over screen, but I’d imagine that that’d be it. Additionally, due to the differences between different obstacles and power-ups, they all have their individual main classes, as opposed to them being subclasses of general classes.

**super().Algorithmic Plan Update**: My complexity concentrations have slightly changed. I am not trying to approximate the shapes of various sprites perfectly anymore, however I am still trying to get pretty close, and am creating sprites that can be accurately represented by rectangles/ovals. Another area of complexity is now in the 2.5D view that I am creating. In order to do this, I am making the sizes of the yellow lines larger and fatter as they near the view of the buggy. In addition, I am slanting the road inwards, and will determine the slope of the slants created. I will do a lot of math for collisions with the curbs, and will try to make it so that power-ups and flags will move on that same slope and fan out as they reach the near side of the screen.

**super().Timeline Plan**: I am working on this with my mentor, but tentatively, here is the current timeline: I want to get collisions with the track done on Friday at the latest, and have the powerups added and completed by the end of Saturday. I would want to have indicators for lives, powerups and score as well as perfected collisions, by the end of Sunday, and would want to finish up the 2.5D implementation by the end of Monday. That would then leave me two days to implement a final complexity feature that would likely get me to MVP, which is now my realistic goal for the project. I am currently thinking about figuring out how to do a local multiplayer. This would ideally be implemented by Thursday at noon-ish, along with a few UI screens. That would leave me a few hours to do thorough citation of my code, create a video and readme, and submit the project. I want to have the project submitted by 4 pm at the very latest so I don’t miss the final deadline.