# Implementation Report – Step2

For step 2, I did not have too much work to do as I had decided on not hardcoding any Rides in step 1 as this would allow for future rides to be added in seamlessly. It was important to me to build a system that would require minimal, if any, code changes should new features or rides be added in the future. This meant that for step 2 all I had to do was create instances of rides for all the rides listed in the project specification.

in terms of the Decision Tree, it was a case of making use of the previous tree querier I had built, all that was required to be added was the new tree generator, which set the new set of choices and their appropriate results. I decided on using a kind of binary code which would have set a ride as 1 if the ride was still “alive” (met the criteria of the visitor), and 0 was represent that that ride was dead. Once we had our final binary value we could then pass it through a loop to extract the “winning” rides for that specific visitor.

I used various arrayLists in order to collate individual visitors ride choices, first by grouping them all together, then by filtering out rides that weren’t suitable for all parties, and then removing the duplicate values.