AI Project Notes

1. Fuzzy Logic.
2. Car that should swerve through traffic.
   1. Car that should stick to the centre.
   2. Randomly spawn traffic for the car to avoid.
   3. Have Pickups for the car to get.
3. Have it avoided stuff like this Mario party minigame. Will have to check if this would be good enough though.
4. Can be done in 2D YEASSSS!!!!



1. You could extend the Genetic Algorithm Jumping example to Shooting an  
   Arrow, where different evolved genes change how the AI handles varying  
   target locations and wind directions.
2. Temple run setup but objects coming towards the agent which it has to avoid.
3. Top-down view of different tracks to follow even.
4. WScilex in the bin file is the application. (https://www.youtube.com/watch?v=QOQv8JtqQgs&list=WL&index=21&t=579s&ab\_channel=MiguelMontielMartinez)

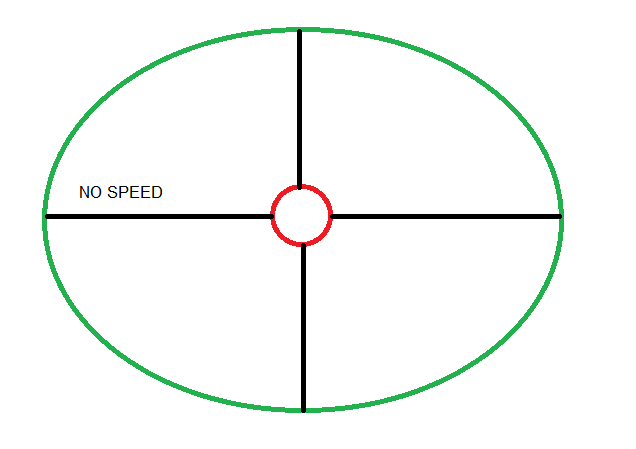


* Need to find out what library the project uses?
  + [https://github.com/davidgrupp/Fuzzy-Logic-Sharp/tree/master/FLS](https://eur02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fdavidgrupp%2FFuzzy-Logic-Sharp%2Ftree%2Fmaster%2FFLS&data=05%7C01%7C2001551%40abertay.ac.uk%7Cd544e78f2ffc4248fd3208db1e3c1218%7Caacb1abaf38f410e9153c16a00ebf4cc%7C0%7C0%7C638137015565752798%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=Atdz0vZXcCQ9lLWMCV%2BYtaqD3pMKCm31irSIss8rQxE%3D&reserved=0)
* Create a GitHub repository for it and change the unity project about to make it your own.
* Figure out if I want to use update or fixed update(Definition of fixed update here https://docs.unity3d.com/ScriptReference/MonoBehaviour.FixedUpdate.html).

# Chart Description automatically generatedWhat I need to do!

Initial Plotting of fuzzy logic for the Initial AI logic

* Change the point the AI moves to on the ground.
  + The fuzzy logic trapezoids are considered the track, with none being the centre!
  + Need to update the positional data according to the position of an object.
    - Add an object to keep track of the centre.
    - Track that objects position.
      * Need to also be able to pass them to the fuzzy logic script.
    - Use that positional value to influence the fuzzy logic shapes.
* Update the position of the centre of the track and let the agent know its new location to move to.
  + Need to make that better currently it takes a wide birth from the object before coming closer. Try shrinking the no movement zone in both x and z axis.



* Have it work in the z axis aswell.
  + Be able to move the camera with wasd.
  + Have the box be able to move in the z-direction aswell.
* Need to add rules to the engine for speed in both the x and the z.
  + Might not even have to do that as what the object currently does is apply force based off that I should maybe try setting it move in accordance with positioning with delta time first.(Nah didn’t work)
  + Need to fix the object overshooting its destination.
  + No need for this. Fuzzy logic needs to be imperfect.
* Have it been able to avoid obstacles.
  + Add rules to the engine such as in the c++ example for it to avoid obstacles.
    - Seemingly is the opposite to the centre track.
      * IF obstacle is to the right move to the left and vice versa.
      * Seemingly each engine may only defuzzify once and that is it, so I am unsure how to have the engines work around this. For now I will use multiple engines.
  + See how difficult it would be to have it avoid multiple obstacles!
* Make the Current multiple movement engines into the one engine!
* Need to show how well the AI is doing so I need to quantify how well it is doing at its job. PRESENTATION IS BIG PART OF MARK!
  + Check for a collision with the cylinder (or area around the cylinder) and add points when it reaches that.
  + If it hits (Or gets too close) obstacles deduct points.
* Need to create an evaluation that goes over how well the fuzzy logic works and all of its method which will be used as a presentation at the end of the project.
  + There is a guide on MLS for this.
    - How long it takes for it to get to the target.
    - Number of obstacle collisions before getting to target.
    - Distance from target after it stops moving.