Some notes to use the code

● For move_dot_away function: `move_dot_away("horizontal", camera, axes)`

- Choose the direction : **"horizontal"**
  or **"vertical"**

- Need to change dist_to_fish to a list of distance from fish to the dot.

```
ruut_p = 30 #ruutus of the dot
dist_to_fish = [150, 200, 300, 400, 500, 550] #distance from the first dot to the fish
dist_to_plane = 200 #distance from the vertical (spherical) plane to the fish
dist_btw_eye = 80 # distance between the two eyes
```

- In vertical case and horizontal case: dist_to_fish is the distance below:
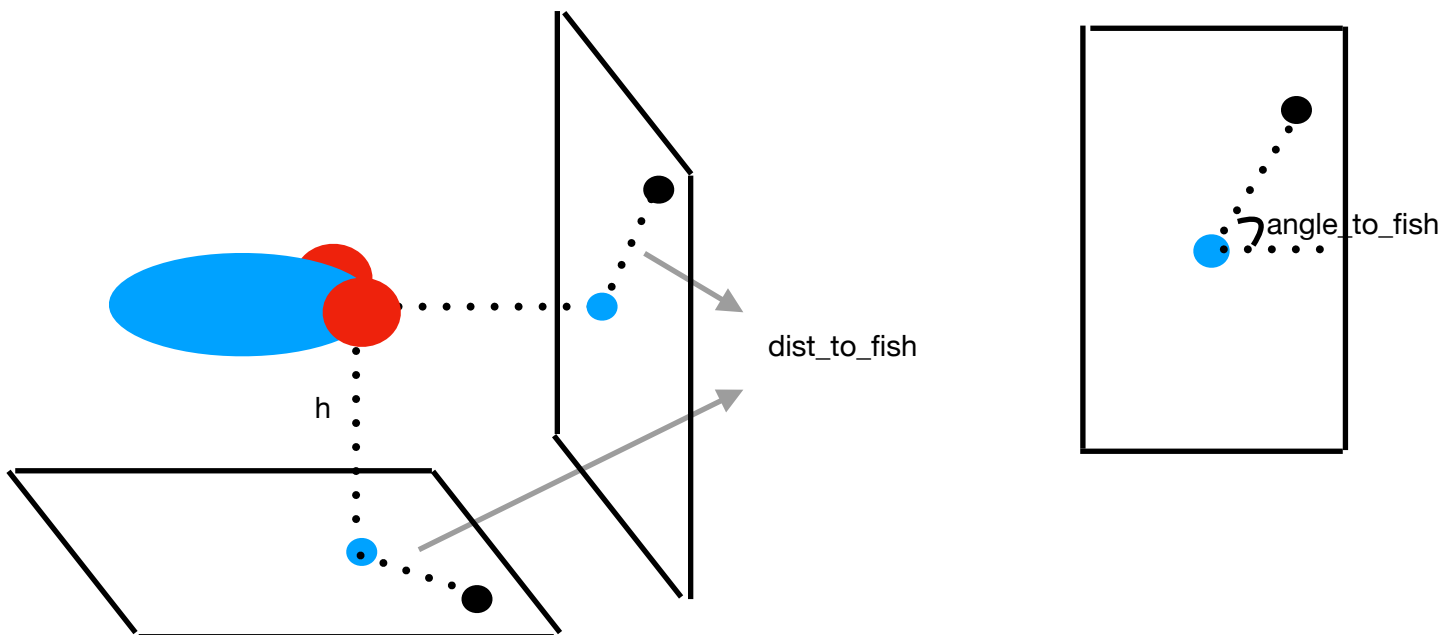
- In spherical case: dist_to_plane is the distance from the fish's center between two eyes to the sphere surround it like below:
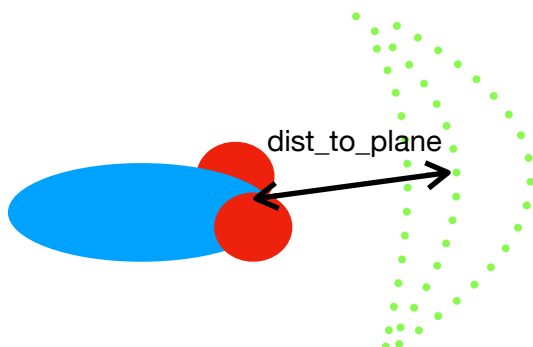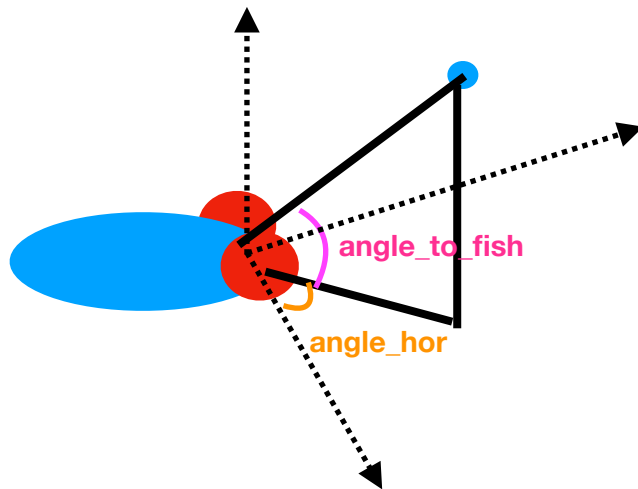

- In spherical case we don't consider the dot moves away from the fish but only consider the case the dot moves around the fish.

1

- For move_dot_around function:

`move_dot_around_fish("spherical", camera, axes)`

- In horizontal and vertical case:

• change dist_to_fish to a constant number

• use a list of different angle_to_fish

```
dist_to_fish = 150 #
dist_to_plane = 250
dist_btw_eye = 80 #
```

- In spherical case:

• Depends on how we want the dot to move, in spherical case, we can keep the angle_to_fish constant and change angle_hor or keep angle_hor constant and change angle_to_fish

- About function to find image's coordinate:

`find_new_coordinate_list(points,num_points, list_of_input, eye_rad, True)`

USE coordinate with the retina's center at the origin or USE the usual xyz coordinate

- To try ellipse dot*, we change the values of minor_p and major_p:

```
minor_p = 30 #minor axis of the dot
major_p = 60 #major axis of the dot
```

(*for spherical case, there is still only 1 case - a circle dot (in this case, the radius of the dot = major_p)

2