

**Will Richards - *IMD 3002 - Prof. Chris Joslin***

## Summary of Task:

The goal of assignment 1 is to write a maya script in python to find the intersection point between a circle center and each of its vectors on a cube

## Methods Used:

I found the course notes and tutorial sessions to be incredibly helpful for this assignment. I have included my notes to show my process but this generally follows the discuss ideas in class.

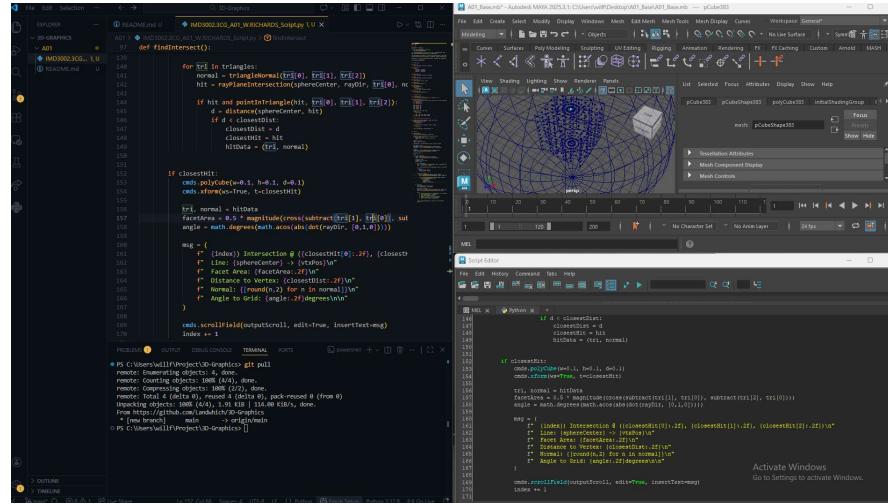
I took the following steps:

1. find circle midpoint using maya selection and xform
2. get all vertices and faces, subdivide target faces into triangles to get the facet normals
3. find the planar intersection and make sure it falls within the triangle

I later realized it wouldn't be so simple as I would have to check this for every target "sub-triangle" to find the shortest distance

## Additional Info

Here is the script working for me:



The screenshot shows the Maya interface with several windows open:

- Script Editor:** Displays the Python script code for finding intersections.
- Outliner:** Shows the project structure and selected objects.
- File Manager:** Shows file paths and versions.
- Maya Viewport:** Displays a 3D scene with a cube subdivided into triangles and a sphere centered at the origin.
- Python Shell:** Displays command-line output from running the script.

```

# This script finds the intersection point between a circle center and each of its vectors on a cube
# It uses Maya's API to select objects and calculate distances

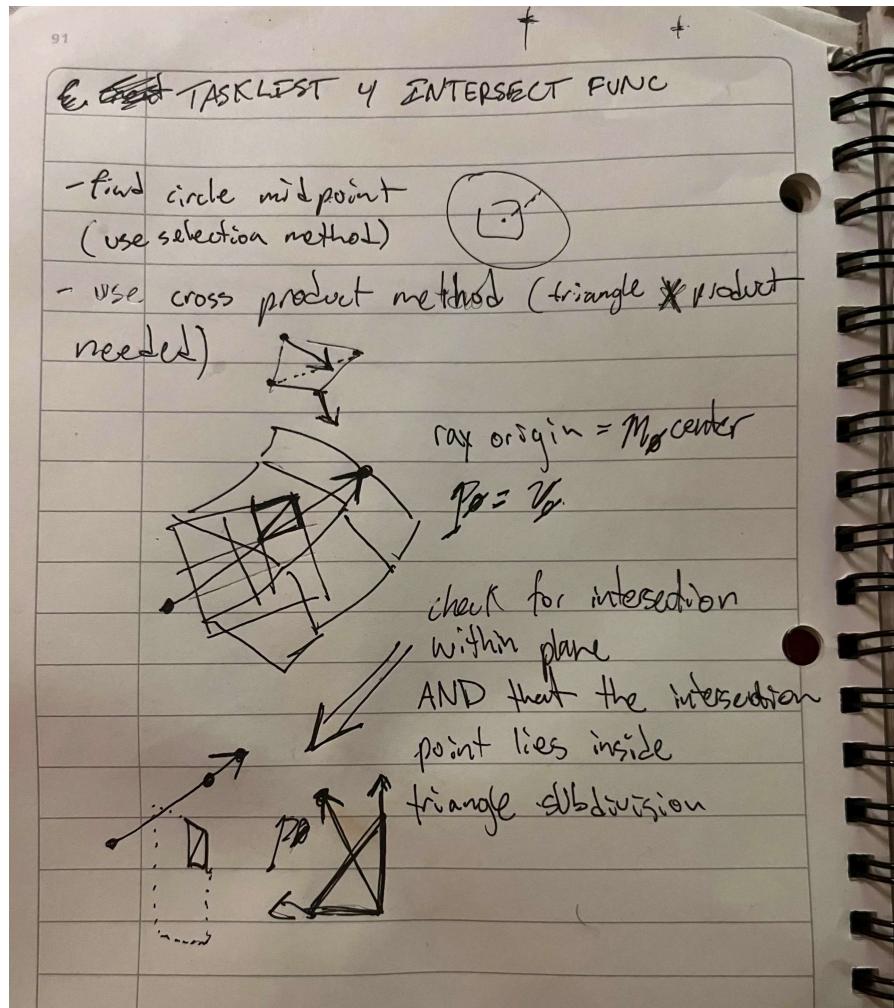
# Import necessary modules
import maya.cmds as cmds
import math

# Set up variables
cube = "pCube003"
sphere = "pSphere001"
center = [0, 0, 0]
radius = 10
epsilon = 0.001

# Function to find intersection
def findIntersection():
    # Get all triangles on the cube
    tris = cmds.ls(cube + ".tri", flatten=True)
    for tri in tris:
        # Get triangle vertices
        v1 = cmds.getAttr(tri + ".v[0]")
        v2 = cmds.getAttr(tri + ".v[1]")
        v3 = cmds.getAttr(tri + ".v[2]")
        # Calculate triangle normal
        normal = cmds.polyNormal(w=0.1, h=0.1, d=0.1)
        # Check if sphere intersects triangle
        if rayPlaneIntersection(sphereCenter, rayDir, tri):
            # Find intersection point
            hit = pointInTriangle(v1, v2, v3, center)
            if hit:
                closestDist = d
                closestHit = hit
                hitNormal = (v1 - center).normal
            else:
                closestDist = radius
                closestHit = None
                hitNormal = None
            # Calculate angle between ray direction and triangle normal
            angle = math.degrees(normal.acos(rayDir))
            if angle < 90:
                # Check if hit is on the triangle
                if hit:
                    # Check if hit is on the triangle
                    if hit in triangle:
                        # Check if hit is on the triangle
                        if hit in triangle:
                            # Check if hit is on the triangle
                            if hit in triangle:
                                # Check if hit is on the triangle
                                if hit in triangle:
                                    # Check if hit is on the triangle
                                    if hit in triangle:
                                        # Check if hit is on the triangle
                                        if hit in triangle:
                                            # Check if hit is on the triangle
                                            if hit in triangle:
                                                # Check if hit is on the triangle
                                                if hit in triangle:
                                                    # Check if hit is on the triangle
                                                    if hit in triangle:
                                                        # Check if hit is on the triangle
                                                        if hit in triangle:
                                                            # Check if hit is on the triangle
                                                            if hit in triangle:
                                                                # Check if hit is on the triangle
                                                                if hit in triangle:
                                                                    # Check if hit is on the triangle
                                                                    if hit in triangle:
                                                                        # Check if hit is on the triangle
                                                                        if hit in triangle:
                                                                            # Check if hit is on the triangle
                                                                            if hit in triangle:
                                                                                # Check if hit is on the triangle
                                                                                if hit in triangle:
                                                                                    # Check if hit is on the triangle
                                                                                    if hit in triangle:
                                                                                        # Check if hit is on the triangle
                                                                                        if hit in triangle:
                                                                                            # Check if hit is on the triangle
                                                                                            if hit in triangle:
                                                                                                # Check if hit is on the triangle
                                                                                                if hit in triangle:
                                                                                                 
# Call the function
findIntersection()

```

Here is my initial process:



## References:

I used the maya documentation and lecture notes for this assignment, as well as tutorial notes for information on maya's object notation