

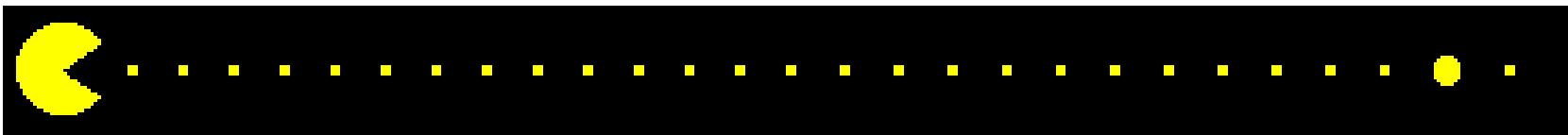
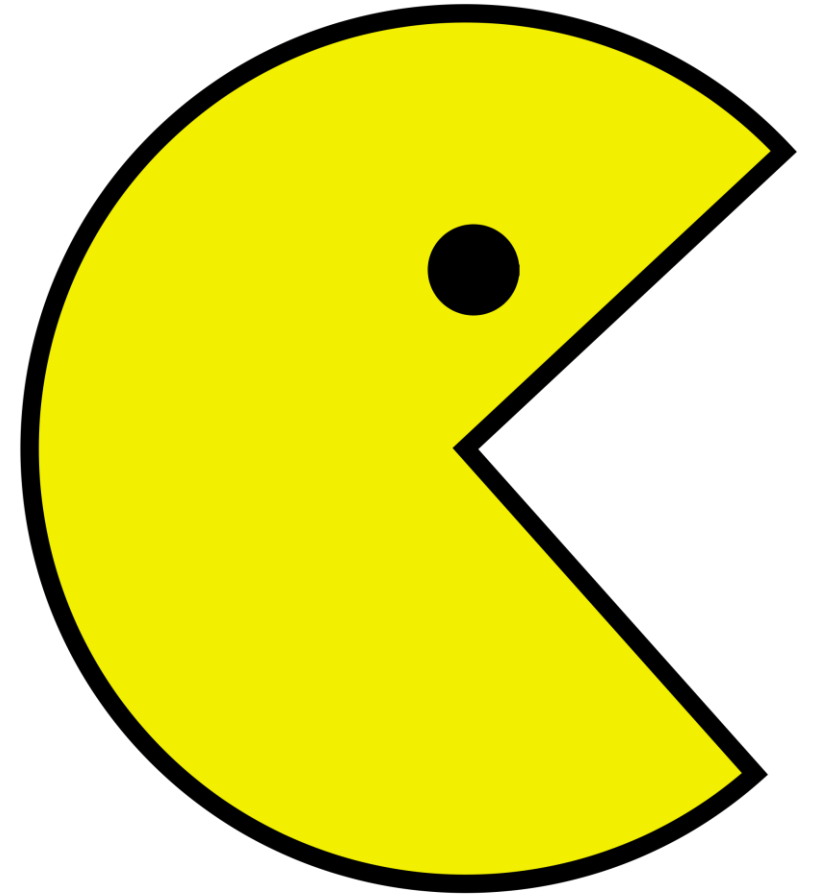
# ME557 Final Presentation

## Pacman 2D/3D

Yuan Fen Chen, Michael Hoefer, Hsiang Sing Naik, Nigel Lee

# Introduction

- Implement a (slightly modified) classic game of Pacman using methods learned in this class.
- User controls Pacman around a maze.
- Goal: Eat every yellow sphere within a given time limit.



# Fun Facts

- Originally known as Puck-man (evolved from Japanese word "paku" which means "chomp").
- Later changed to Pac-man due to Puck having an extreme closeness to a certain explicit English word.
- Each of the four ghosts have names - Inky, Blinky, Pinky & Clyde.
- Tiny dots that he eats are actually cookies.



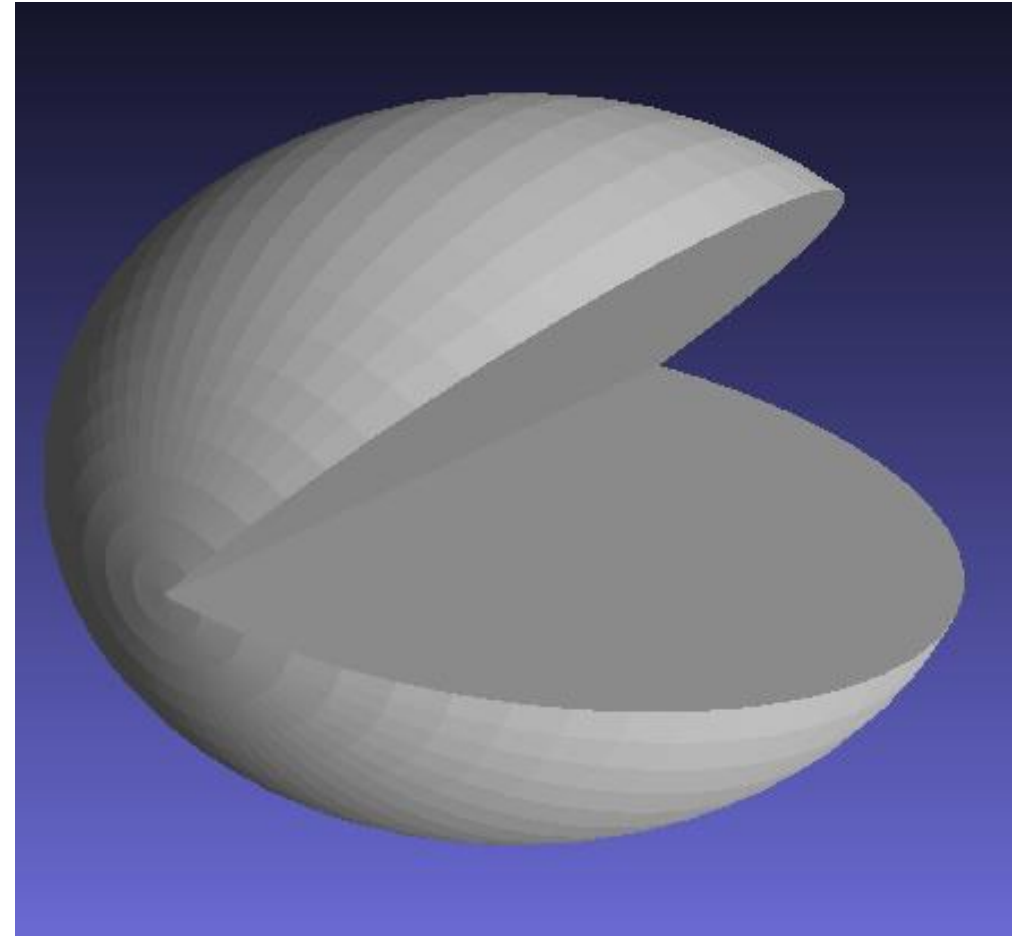
# Objects and behaviors

- Pacman
- Yellow balls
- Red balls
- Diamonds
- Maze



# Pacman

- Large yellow sphere with a mouth
- Eats any balls/diamond that it touches to gain points/bonus times
- created in solidworks, loaded as obj file



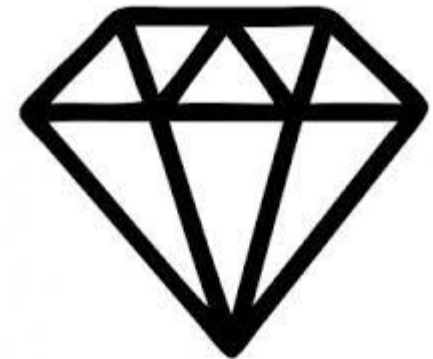
# Colored Balls

- Yellow
  - Needs to be collected to win the game
- Red
  - Gives additional time (1 second)



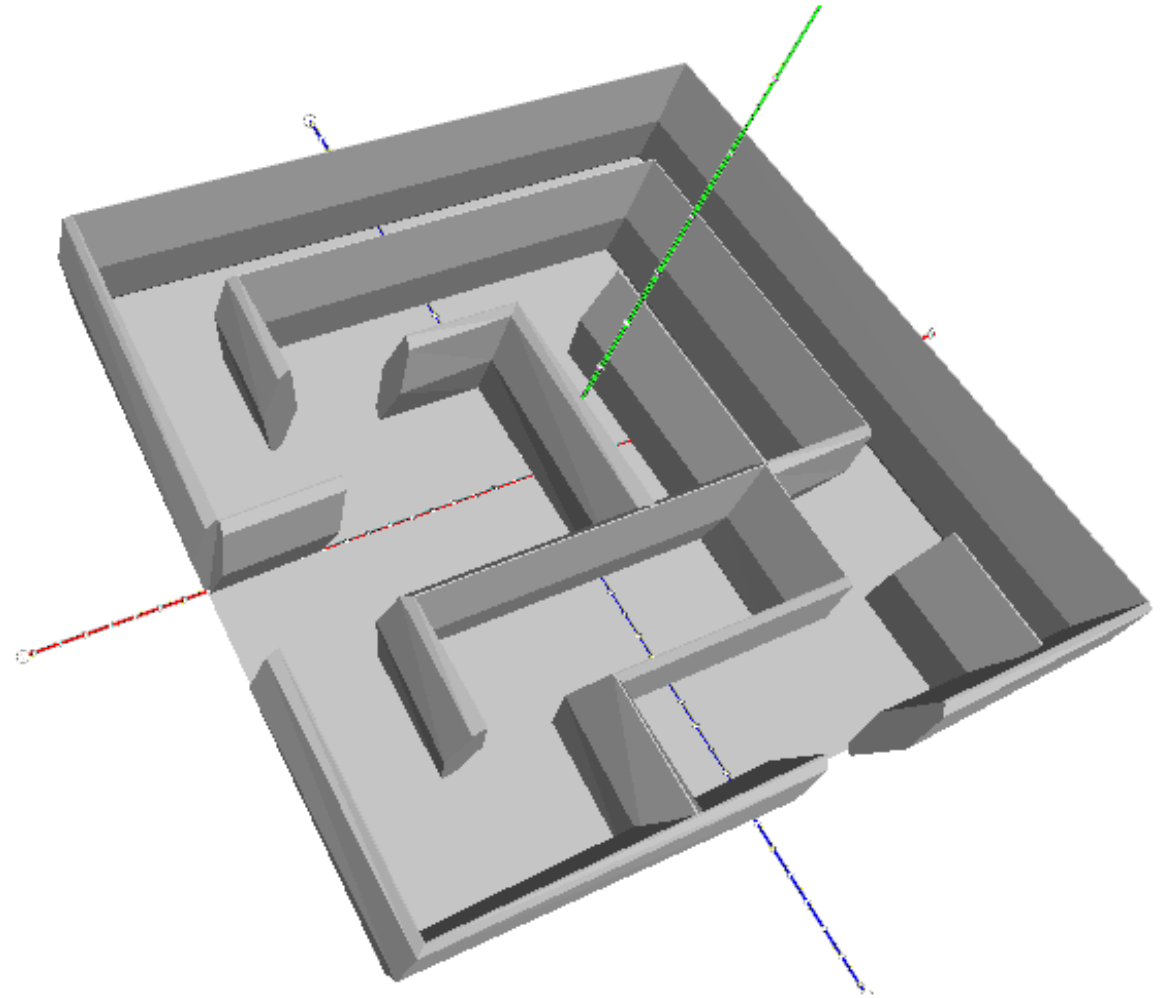
# Disco diamond!

- Adds additional time (10second).
- Changes the lighting of Pacman's world
- Possibilities include changing the color of Pacman, changing the maze color, and changing environment mapping.
- created in OpenGL using multi-primitive.
- Coming soon



# Maze

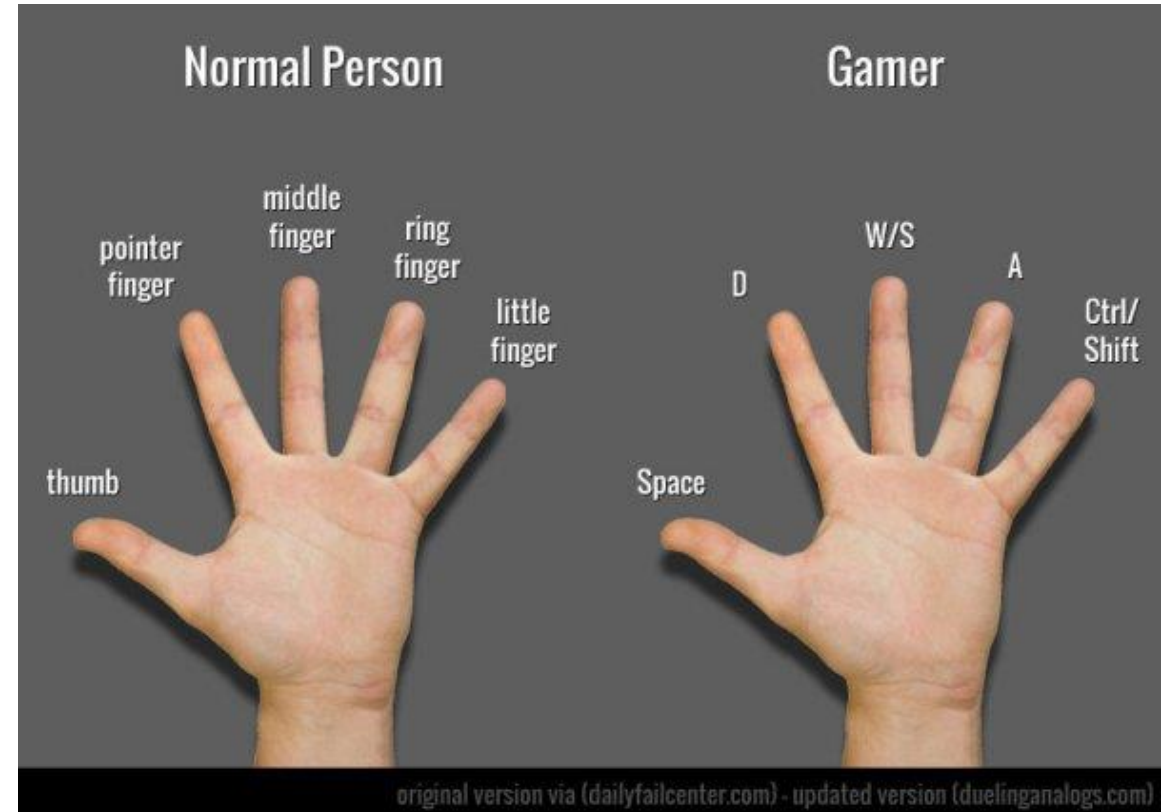
- Ground on which Pacman moves around
- Consists of a floor and multiple walls
- Every area of the maze is accessible through some path





# User interaction

- Keyboard
  - W to move forward
  - A or D to change directions, rotation transformation 90 degrees
  - S key to rotate pacman 180 degrees
  - 1/2 to switch between views(2D/3D)
- Mouse
  - Left click to rotate
  - Right click to zoom in







# Techniques Applied

- Collision Detection/Ray Intersection Test for identifying when a wall hit or sphere is "eaten".
- Quaternions to interpolate Pac-man's movement.
- Multi-Texturing for diamond
- Environment mapping on Pacman
- Triggered Light Sources when a "power up"(diamond) is eaten
- Keyboard and Mouse navigation to control Pac-man.
- Modeled objects using SolidWorks, converted to OBJ files using MeshLab and load OBJ files into framework.
- Modeling the diamond using multi primitives.

# Rubric



	Modeling	Light	Appearance (Material)	Navigation / interaction	Keyframe Animation
1	Single primitive or loaded objects	0 <i>you need a single light</i>	Basic specular + diffuse reflect code	Trackball navigation	Single linear transformation
2	Multiple primitive 	Combined Spot + direct + point light	Texturing	Navigation in 3D	Animation with rotation and translation
3	Complex object surface	Combined light on surface	Multi-Texturing	Navigation with keyboard + mouse	Triggered (keyboard, etc.) animations
4	Hierarchical model	Combined <b>colored</b> light on surface	Environment Mapping 	Navigation along a surface 	Animation with collision detection
5	Hierarchical model, h>2	Triggered light source 	Bump or Displacement Mapping	Navigation with multiple cameras	Multiple animation paths per object

+ Presentation + team assessment

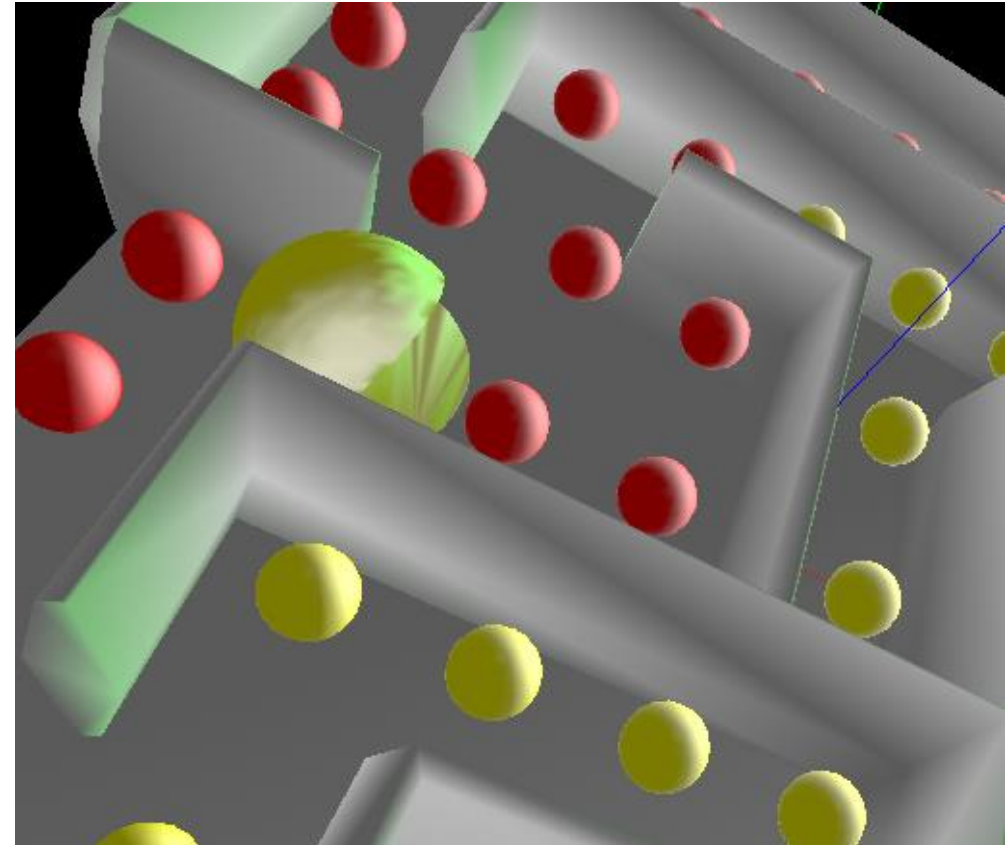


VRAC|HCI

IOWA STATE UNIVERSITY  
OF SCIENCE AND TECHNOLOGY

# Pacman's Motion

- Ray intersection test casts a ray from the center of pacman in the direction that Pacman is facing.
- If the first object detected is a wall, cancel the current move
- If the first object detected is a ball, allow the motion and run the game logic
  - Remove ball
  - Add time
  - Change lighting



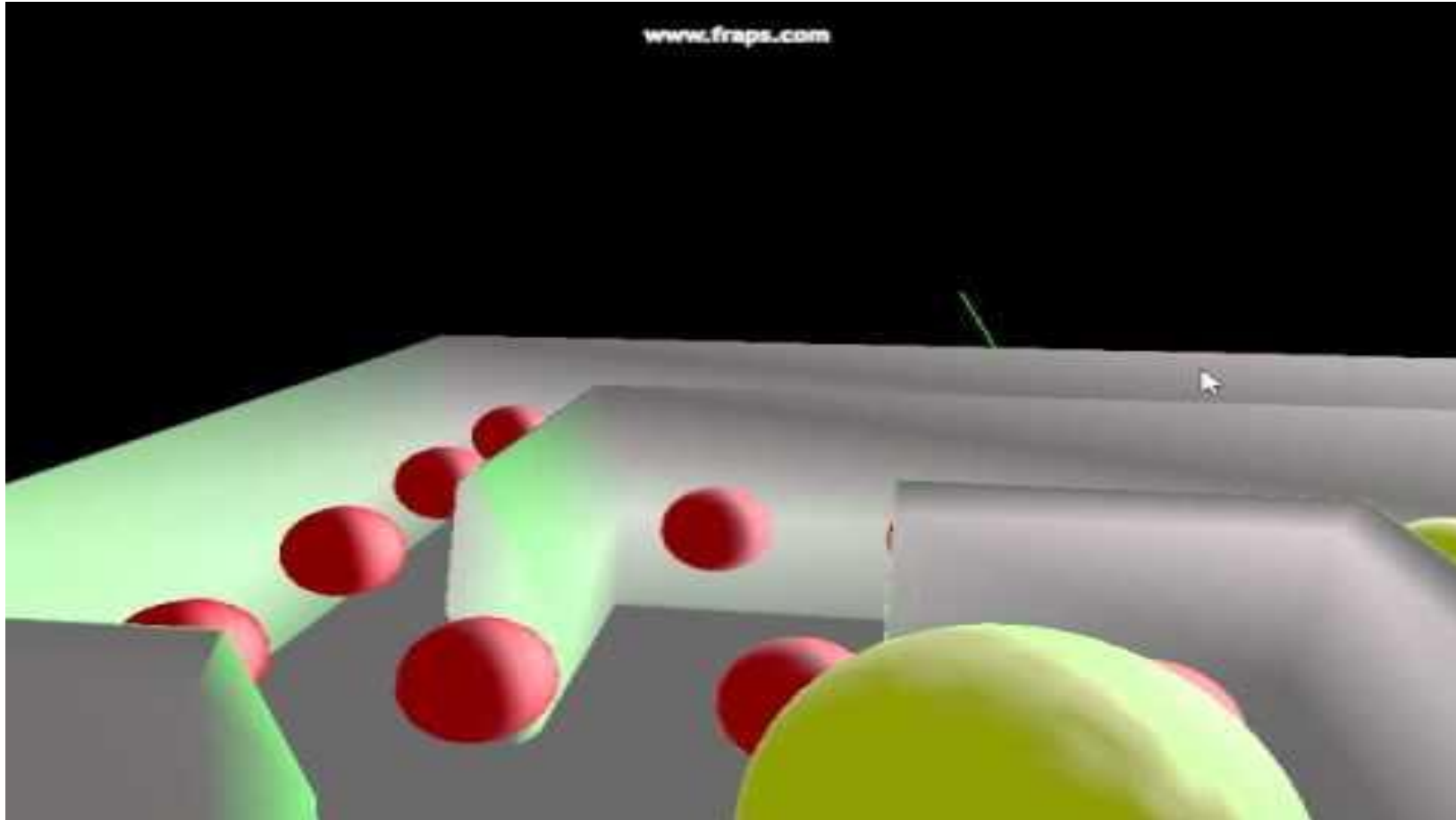
# Challenges Faced

- Pacman navigation - dynamic direction of ray intersection test
- Object modeling - size of pacman must fit the maze
- Determining position of objects within the maze
- Texture mapping on a non-2D surface

# Overall Experience

- Used a combination of working in groups and dividing up tasks to work individually
- Getting started took time, needed to set up a common development environment
- Using a large common code base. A lot of functionality built in, but need to know how to use it.

# Current Demo



Questions