

# Super Archer

*CS 174A Group Project*

**Rohith Venkatesh**

**Rishi Padmanabhan**

**Beomjoo Kim**

# Theme and Story

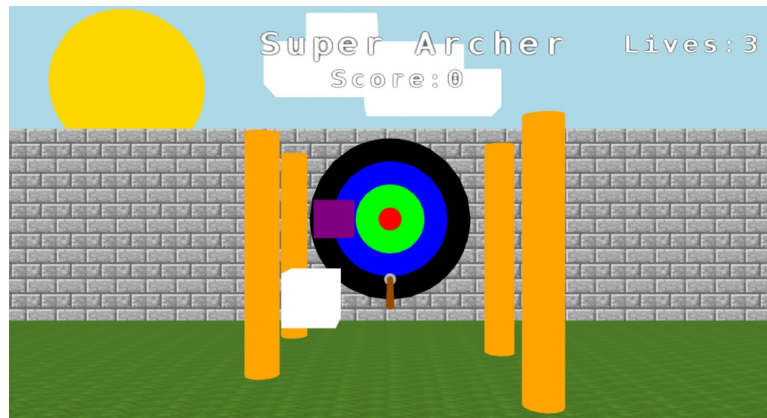
When we first met to discuss the project topic, we agreed that it would be great to create a game, especially a sports-related game. Various suggestions such as bowling, football, and swimming were made, but ultimately, we decided to create an archery game, which we all commonly enjoy.

Generating this idea was mainly inspired by Game Pigeon archery and video games like Assassin's Creed



# Game Mechanics

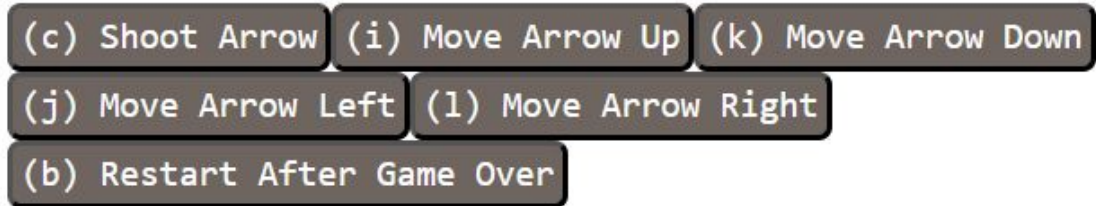
- **Objective:**
  - Shoot an arrow to hit a target.
  - Aim arrow before launch
  - Control an arrow after it is launched.
  - Avoid obstacles and reach the target.
  - Score points based on hit location.
- **Scoring System:**
  - Edge: 1 point
  - Blue ring: 2 points
  - Green ring: 3 points
  - Bullseye: 4 points
  - Unlimited Attempts until lives run out.



# Interactivity

- **Controls:**

- Keyboard controls to control the movement of the arrow.
- (Post-launch control of arrow movement).



- **Scoreboard and Lives Counter:**

- Updates based on hit location and lives left.



# Course Topics Applied

- **Collision Detection:**
  - Detecting when the arrow hits obstacles/target/ground/power-up
- **Anti-Aliasing:**
  - Sampling more points to increase the accuracy of collision detection
- **Bump Mapping:**
  - Adding textures to objects to make them appear more realistic
- **Lighting and Shading:**
  - Ambient lighting and shading for each object used
- **Change of Basis:**
  - For accurate arrow movement and interaction.
- **Geometric Transformations:**
  - Rotation, translation, scaling, and inverse operations.

# Advanced Features

- **Collision Detection:**
  - Track arrow position.
  - Dynamic obstacles moving from left to right in front of the target.
  - Check for collisions with obstacles and target.
- **Physics-Based Simulation**
  - Constant force of gravity acting on the arrow
    - i. in the game, you have to move the arrow upward to apply a force in the opposite direction to resist the gravitational pull
  - Torque applied to the arrow by gravity
- **Bump Mapping**
  - Added textures to the grass and background wall to make the game scene look more realistic

# Challenges and Resolutions

- **Challenges We Faced During the Development Process:**
  - Collision detection accuracy (anti-aliasing)
  - Implementing Gravity
  - Consistent Target Visibility (initially, power-ups sometimes would get in the way of the camera to a point where you would not be able to see the target)
- **Resolutions:**
  - Refined algorithms for collision detection (sampled more points and applied the distance formula)
  - Drew out vector diagrams and implemented it via a rotation (applying a gravitational torque)
  - Added an extra check for the arrow being beyond the power-up impacting the visibility of the power-up (i.e. if the arrow was beyond the power-up, the power-up would disappear)