

Documentation

Administrative

| | |
|------------------------|---|
| Group Name / Game Name | HoleInOne (Group 30) |
| GitHub Link | https://github.com/FlorianStein/cgue21-HoleInOne |
| Students | Florian Steiner, 11909426 Yizhou Cui, 11807884 |
| Genre | Casual / Puzzle |
| Goal | Hit the golf ball to the hole with a limited amount of strokes and while keeping the ball on the field. |

Game Idea and Content

| | |
|--------------------------------|---|
| Gameplay (max. 4 sentences) | Hit the mini golf ball to the goal, with max. 9 strokes allowed, just like a golf game in the real world. To make things a bit more exciting, collect power-ups (heavier, lighterball) on pressure plates and experience different friction on the course. |
| User interaction | Mouse drag, on or near the ball: Shoot direction and strength Mouse drag, elsewhere: Pan camera (arcball around the ball) Tab: Toggle HUD F8: Print out the ball position debugging. F9: Place the ball to the goal coordinate directly for debugging. HUD buttons: Start game, pick/skip level. |
| Scene lighting | The sun (level 1 & 2), or multiple point lights with dimmer sunlight (level 3). |

Implemented Features

| Category | Feature | Description (Usage) |
|-------------------|---|--|
| Optional Gameplay | Collision Detection (4 Points) | Golf ball is colliding with the floor and obstacles. |
| | Advanced Physics (6 Points) | <p>Simulate the physics of the golf ball with the barrier or the floor. Friction and mass changes are demonstrated in some levels.</p> <p>Both points above are achieved with Bullet Physics</p> |
| | Heads-Up Display (4 Points) | <p>Displays the number of strokes used, power ups and menus.</p> <p>It's rendered with an orthogonal projection matrix in screen space.</p> |
| Effects | Shadow Map with PCF (16 Points) | <p>Course, flagpoles etc. cast shadows.</p> <p>The depth is rendered to a shadow map. This framebuffer texture is used in the main shader to get the closest depth. A bias is used to prevent shadow acne and front culling on selected objects are used to counter peter panning.</p> |
| | Cel Shading (4 Points) | <p>Cel shading for achieving a flat comic style look.</p> <p>The diffuse color variable is floor-divided to several levels to achieve the cel shading.</p> |
| | Contours via Edge Detection (12 Points) | <p>Is used for achieving the final look of the game.</p> <p>The normals and depth are rendered into 2 framebuffer textures. "fwidth" is used to generate a gradient and is matched against a threshold.</p> |

Description

The game loads models from OBJ files and uses Bullet Physics to do collision detection. Cel shading with edge detection contours is used to achieve the look. Shadow is applied to the first directional.

The models are generated via blender and exported as an obj. Inside model.h vertices and textures are loaded. The mesh is passed to Bullet Physics and OpenGL.

Due to lang object sizes, the loading time is quite long.

No other sources besides those provided in TUWEL were used. ECG_Framework and Bullet Physics are used as 3rd-party frameworks.