

Documentation

Group/Game Name:

Crazy Towers

Brief description of implementation:

Crazy Towers is a Tower Defense Game in which the player defends against Attackers by placing towers on a grid. The game was implemented using OpenGL and the GCG-framework was used as a basis for our implementation.

The models for the towers, the map (world-plane), the attackers, and the mountains surrounding the map were created in Blender, using Blenders Modelling tools. Furthermore textures were also created for the attackers and the plane. All of the models and materials were exported as .obj (wavefront) and imported into our game using Assimp (see src/Object.h).

Textures from the GCG-Framework were also used for the Tower, Arrow and Mountain models (rock, tiles and rock respectively). During gameplay the player can interact with the world using a grid. The current selected cell is marked in green while WASD will move the selection (up, left, down, right) while also showing the range of the towers from the current selection (green circle around the grid-cell).

A user-interface is also provided which shows the current amount of money (money-symbol on the upper left) and 3 buttons with different types of towers. A click on one of the tower symbols will place the according tower on the currently selected grid cell. The gray tower button will place an arrow tower (base-unit), the yellow tower button will place a temple, which generates money and the red tower button will place a missile tower (stronger but more expensive). The camera is being controlled by the mouse, using click-drag and the mouse wheel for zooming.

In the game world two Portals can be seen. From the red portal attackers will spawn and they will start to move towards the lila/blue one along the path on the world-plane. When one Attacker reaches the end portal the game is over and a game-over message will be displayed on the screen. The portals use simple box-geometry for the arches and the brick-texture from GCG, while the portal itself uses a plane which is being modified by a wave shader during runtime for the "wobbling" effect.

A simple point light is being used to illuminate the world, which also moves during gameplay for additional visual effects. For the "skybox" a gradient texture is used which is being modified/generated during runtime.

When an Attacker reaches zero hitpoints the model is being deleted while a particle effect is being displayed, showing an explosion-like effect to signify that this attacker is defeated.

Pressing the "m" key will display the world with normal mapping.

There are 3 types of attackers. A base unit (normal fish), an angler fish (faster but less hp) and a tank-fish (more hp but slower). Various combinations of these Attackers will spawn during 3 rounds of gameplay. When all of the attackers from the 3 rounds are defeated, the game is won!

Additional libraries:

[Assimp](#)

Gameplay:

Mandatory:

- 3D Geometry:
 - Towers, attackers, world objects created in Blender
 - Loaded into the game using Assimp
- Playable:
 - Placing towers with the UI
 - Interactions (tower placement)
 - Win/lose conditions
- Min 60 FPS and Framerate Independence:
 - Deltatime was used for framerate independent drawing
 - Simplicity of models allows min of 60 FPS
 - Tested on NVIDIA GeForce 1060
- Win/Lose Condition:
 - All waves cleared/attacker reaches portal
- Intuitive controls:
 - WASD for grid movement
 - Mouse-click on GUI for tower placement
- Intuitive Camera:
 - Mouse-drag for camera movement
 - Mouse-wheel for zoom
- Illumination model:
 - Single point light source acts as sun
 - All the objects have materials
 - Normal vectors are present
 - Phong model from GCG
- Textures:
 - Some from GCG (tiles, bricks)
 - Self-drawn (world-plane, attackers)
- Moving Objects:
 - Attackers
 - Arrows/missiles
- Adjustable Parameters:
 - Normal Mapping by pressing "m"

Optional:

- Collision Detection (Basic Physics):
 - Collision of arrow or missile with an Attacker
 - Using simple range calculation

- Advanced Gameplay:
 - Tower placement only possible on buildable fields (not path)
 - Different types of towers with different attributes
 - Score/economy aspect
- Heads-up Display:
 - UI for game interaction and win/lose state

Effects:

Lighting:

- Lightmap using Separate Textures: STATIC LIGHTMAP TEXTURE ADDED! -> for Map Object (Blender freezes while generating Lightmap for Map-Rocks due to highfaces count)
- Lightmap using In-Game Calculation: Key J to disable/enable
 - Shaders? Key L to enable Camera Spinning
- Shadow Map with PCF:
- Shadow Volumes:

Advanced Modelling:

- CPU Particle System:
 - Attacker Death animation

Animation:

- Vertex Shader Animation:
 - Wave shader for the portal

Texturing:

- Procedural Texture:
 - A gradient texture is being used and calculated for the skybox

Shading:

- Simple Normal Mapping:
 - Activated using the "m" key

Other special features:

Walk-through:

Upon starting the game, place arrow towers (gray) in cells which can reach the attackers path. Place temple towers (yellow) in cells you don't need, i.e. cells which are out of range of the attackers path. Let your money-score increase and place a missile tower. Enjoy the particle effects!