

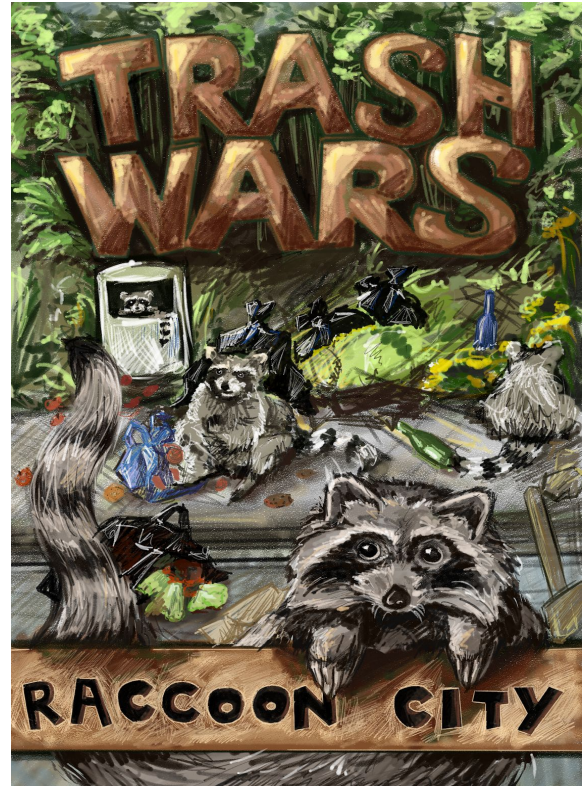
Trash Wars: Raccoon City

Dzenan Omerovic (01226947)

Gabriel Ratschiller (11778247)

Overview

"Trash Wars: Raccoon City" is a FPS/action shooter. The player controls the main character in first person view and the aim is to defend your base. You lose if the raccoons deplete the health of your trashcans which happens only if they manage to get close enough. You can see your remaining health in the console. After you dispatch every raccoon you win the game and you are crowned the ultimate trashman.



Features

Model Import

A model loader is implemented that loads .obj data formats with the help of the Assimp library and stores the models as meshes that can then be used in the game engine.

Keyframe Animation

The raccoons are animated and move their legs with the help of vertex animations. The models have different poses and an animation shader interpolates between them in order to realise a natural leg movement.

CPU Particles

The particle system consists of a large amount of similar quads with different positions and velocities. A buffer stores the particles and they are rendered using instancing.

PhysX

The PhysX library is used for collision detection and the shooting mechanic. Both the player and the raccoons are character controllers which move on a plane.

For the shooting we are using raycasts from the player (camera) position and we set the "RACCOONS" as the only active group for the query. On a hit, the raccoon is immediately destroyed and removed from the game.

Effects

Cel Shading

In the game there are shaded several objects with the cel shading technique. The raccoons, the fence and the trash cans are shaded this way in five gradations. In order to achieve this the diffuse and the specular part of the light are shaded gradually in the fragment shader, i.e. the light is mapped to a number of levels, displaying only few different shades.

- <http://in2gpu.com/2014/06/23/toon-shading-effect-and-simple-contour-detection/>

Contours via Edge Detection

To display edges around the models there is used an edge filter (Sobel Filter) in a post processing step. In the first render pass the visible scene is stored in a Framebuffer, the bufferdata is folded with the kernel and in a second render pass the folded scene is combined with the original scene and drawn with the contours.

- <https://learnopengl.com/Advanced-OpenGL/Framebuffers>
- <https://computergraphics.stackexchange.com/questions/3646/opengl-glsl-sobel-edge-detection-filter>

Vertex Shader Animation

The vertex animation is used to animate the legs of the raccoon. To achieve this the raccoon model is exported with blender in three different positions of the legs. Then the three positions are interpolated in the animation shader using the time as the offset factor.

- https://www.khronos.org/opengl/wiki/Keyframe_Animation
- <http://www.mbsoftworks.sk/tutorials/opengl3/24-animation-pt1-keyframe-md2/>

CPU Particle System

The particle system is used to visualize the "smell" of the trashcans. A particle generator is set into the trashcan objects that spawns new particles that decay over time. Every particle has a life variable and if the life is less than 0 the particle is killed and another particle spawns. A buffer stores and controls the lifespan of every particle and the particles are rendered via instancing.

- <https://learnopengl.com/In-Practice/2D-Game/Particles>
- <http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/billboards/>

Illumination

In the game there is a directional light as main global illumination. The raccoons, as well as the trashcans and all other static objects (fences, houses, fountain) are illuminated with the directional light source and are cel shaded..

In addition the houses and the floor are textured with a brick texture and the raccoons as well as fences and the fountain are textured in one color, in order to see the cel shading effect nicely.

In addition there is a skybox implemented to simulate a wide landscape.

HUD

We have a HUD to display the crosshairs, the current hit points of our trash cans and the messages at the end of the game. We are using the FreeType library to load fonts and generate for each one a bitmap. We render the bitmaps on small quads and we are using an orthographic projection matrix because we do not need perspective when we render the text.

- <https://learnopengl.com/In-Practice/Text-Rendering>

Tools

Blender

Blender is used to create the scene model (houses, terrain, trashcans,..) and to animate the raccoon models. The raccoons are downloaded models, moved according to the leg animation and then exported.

Controls

The character is moved through the map using the keys W, A, S, D and the weapon can be fired with the left mouse button. Without a gun model and with no crosshairs you have to no scope it.

Adjustable parameters

All parameters listed below can be adjusted in the config.ini file in the folder *TrashWars/assets/config.ini*.

- Screen resolution: the width and height can be set separately
- Brightness: if adjusted, a uniform in the shaders is set
- Refresh rate: default is 60 fps
- Fullscreen mode: default false

Libraries

We use the library "Assimp" for the model loading (raccoons, trashcans, scene). All objects are modelled in Blender and the models are imported with its meshes, materials and textures. GLFW is used for windows, GLM for math functions, GLAD for OpenGL extensions, FreeType for text rendering and stb_image for image loading.

- Assimp
- GLFW
- GLM
- GLAD
- stb_image
- FreeType

Planned features

Due to time constraints a lot of ideas were sadly not implemented. We hope we can finish these till the presentation:

- Prettier crosshairs
- Gun model and particles from it on each shot
- More complex paths for the raccoons
- Multiple raccoon waves
- Prettier HUD
- Collision for buildings