

Space Tank Wars

Controls:

W and S: move the tank forwards and backwards OR up/down in Menus

A and D: turn the tank left and right

Q and E: move sideways

mouse: aiming, camera steering

left mouse button: fire primary weapon

right mouse button: fire secondary weapon (not available on every tank)

!Debug Commands are only available ingame, not in Main Menu!

F2: debug mode; shows active modes, fps and the number of drawn vertices

F3: toggle wireframe mode

F4: toggle texture sampling quality: nearest neighbour/bilinear

F5: toggle mipmapping quality: off/nearest neighbour/linear

F6: toggle anisotropic filtering: isotropic/anisotropic filtering x4/AFx8/AFx16

F8: toggle view frustum culling

F9: toggle transparency

ESC: quit program instantly

Enter: toggle steering mode: playable/unplayable OR in menu: Klick Button

A-Z: Name input on highscorelist

Description of the implementation:

- complex objects: every object in the game was modeled in 3DS MAX or Blender; as file format we used Collada. The models are imported with Assimp.
- animated objects: We implemented hierarchical animation: The tanks move, the turret on the tank rotates and the barrels on the turret move up and down.
- view frustum culling: We used the collision spheres from our collision detection to check if the objects are in the view frustum. Tanks as well as projectiles are clipped if they are not on the screen.
- transparency: The user interface is transparent, the health bars over the enemies too, and also the particles are transparent.
- collision detection is done with bounding spheres; because the tanks are quite "flat", the

spheres are cut off over the tanks, so that projectiles can fly over the tanks without hitting them.

Features:

- different goals for each level (finish the mission in a given time, defeat all enemies, survive for a given time)

Illumination:

- The terrain, the tanks and the projectiles are illuminated by a directional light (the sunlight). They all have textures and make use of normal and specular mapping.

Additional libraries:

- Assimp for importing the models <http://assimp.sourceforge.net/>
- FMOD for sound <http://www.fmod.org/>
- GLFW for OpenGL-Window and texture loading <http://www.glfw.org/>
- FlexGL for OpenGL Loading <https://github.com/ginkgo/flexGL>
- GLM for Vector Math <http://glm.g-truc.net/>

Effects:

- normal/specular mapping on terrain, tank and projectile; not implemented exactly after 1 source. Most helpful tutorials were: <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-13-normal-mapping/> and CG1 Course.

The whole Shader for objects is implemented in WorldSpace. So Normal-mapping is implemented by transforming the texture normal into the Worldspace. Specular Mapping is implemented by reading the specular color from a texture. Tangent and Bitangent are loaded with assimp model loader.

- particle effects; we extended our particle system with the ability to rotate the particles into their flying direction and scale them along this direction.