Documentation flip-it

How to start the game

Unpack the archive and click on the flip-it.exe file. If you want to, you can change settings like the window size, the refresh rate or custom properties like the level you want to start in or the illumination multiplier.
We recommend to start with Level 1. This is the introduction level that shows the player the most relevant key controls and environmental objects.
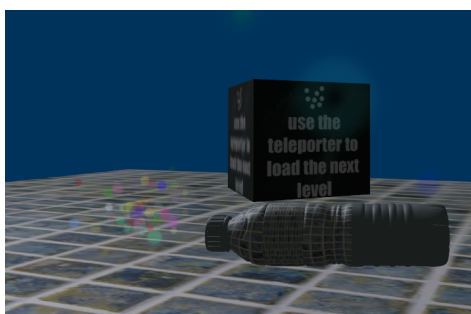
Level Overview

We have implemented 3 playable Levels.
- Level 1, the introduction level
- Level 2, the advanced level
- Level 0, the demo level in which most of the testing and implementation was done. There you can find all the effects we used in one place.

Controls

- Gameplay
  - WASD: Roll around
  - SPACE: Jump
  - X: a powerful high jump (costs water)
  - E: boosts the player forward (costs water)
  - Q: opens and closes the glider while in the air
- Debug
  - F1:   toggle Wireframe-Mode
  - F2:   toggle Backface Culling
  - F3:   toggle Debug Camera
  - F7:   toggle Lens Flares
  - F8:   toggle Culling of Objects
  - F9:   toggle HUD
  - F10:   select Level of Detail for Spheres
  - G: deprecated fly hack
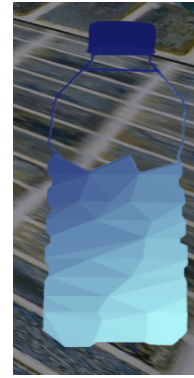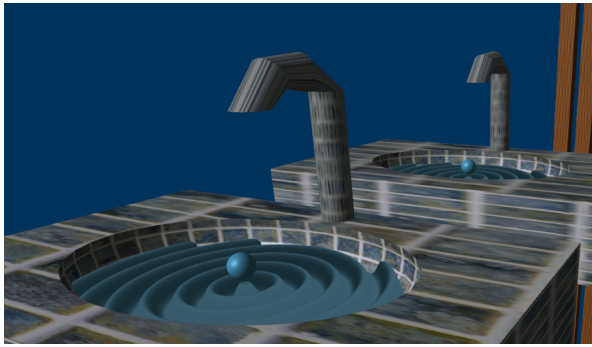
Game Explanation:

Your character is a bottle that you can maneuver around the world. If you fall into the emptiness you "die" and respawn (either at the start of the level or at certain checkpoints).
Your goal is to get to the end of the level and step in the teleporter to the next level.
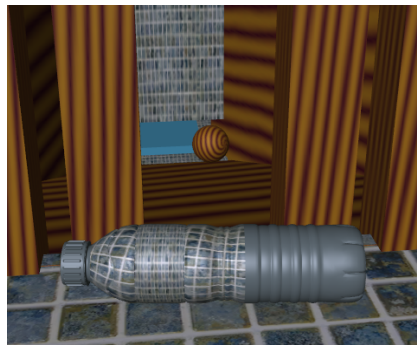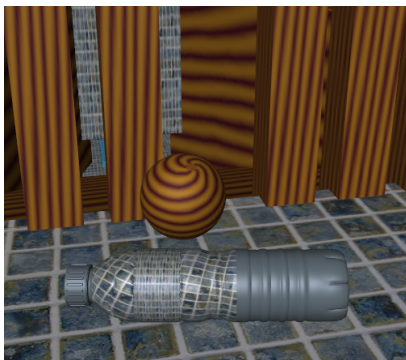Teleporters are small colorful particle systems.

If you use your special abilities (E to boost or X to high jump) you lose your water. This is indicated in the

HUD: You can refill your water at the sinks that can be found in every level.
Just pick up the small blue bubble and you are good to go.
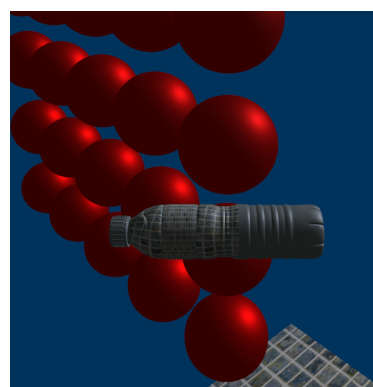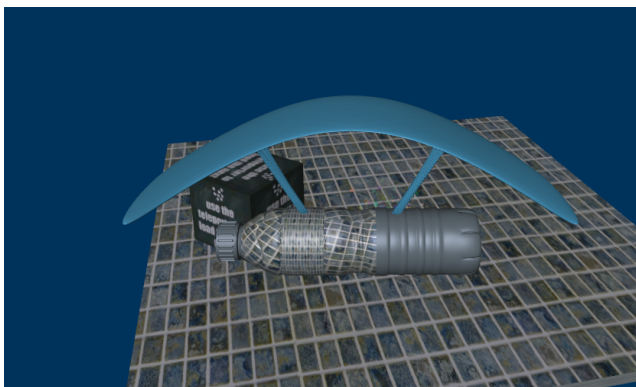

If you encounter a sphere that seems a bit out of place, you could try to move it with your character. In this case the sphere triggers a pressure plate at the bottom of it's path which will throw you in the air.

Pressure plates are small structures that catapult you in the air (doesn't matter who activates them)

If you need to get to a far platform. Jump off your platform and press Q to activate the glider. You can also use X and E while gliding to extend your flight!
In level 2 you will come to a point at which you will have to abuse the game mechanics. At our "climbing wall" you can jump up by using the space between the floating spheres to get to the top.

Features
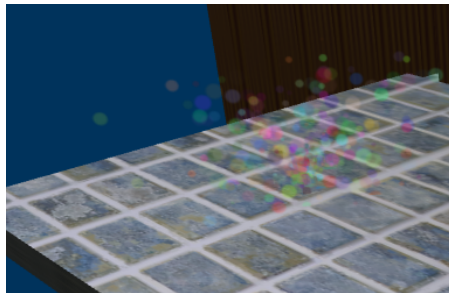- Rotating horizontal player character

- - physix does not support this and recommends to rotate the whole world instead. This leads to a whole lot of problems, but at that point it was too late to change the whole system.
  - Jump and Run and Glide
    - an insanely fun character movement
  - Power up System
    - Power Ups spawn Periodically in sinks and can be collected to refill the water meter.
  - Trigger Plates
    - Trigger Plates can be activated by stepping on them or by rolling an object on them and give a great upwards boost.
  - Teleporters
    - Teleporters are used to transition to the next Level, but can also be used to teleport inside one level (which is only shown in level0)
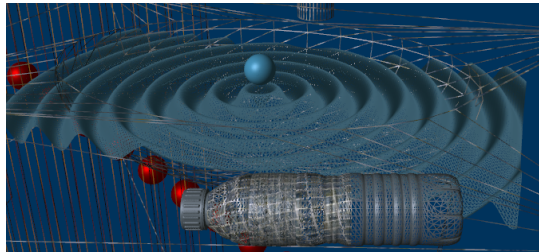
Advanced Gameplay
  - Collision detection
    - physx is used to detect collisions with other solid objects (ground/wall) and trigger objects (teleporters, savepoints)
  - Advanced Physics
    - physx is used to have a rolling sphere, which can be used to activate Pressure plates
  - View-Frustum Culling
    - Geometry that is far away will not be rendered, to increase fps
    - Particle Systems have a shorter distance, as those need much more processing power
  - HUD - F9
    - Displays the water meter, as explained in the Game Section
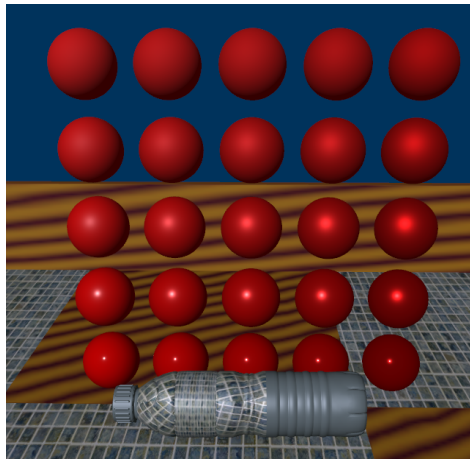
Effects
  - CPU Particle System

    -
    - can be seen at the end of every level
    - Tutorial used: http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/
  - Vertex Shader Animation

- ○
- ○ can be seen at every sink (use wireframe mode)
- ○ Theory from Animations from the Revision Course 2018
- Procedural Texture
  - ○ The wood Texture, as seen in other screenshots and everywhere in the game
  - ○ Theory from: https://lodev.org/cgtutor/randomnoise.html. Extended by change of generation and color
- Physically Based Shading

  - ○
  - ○ can be seen in the demo-level (Level 0)
  - ○ Theory from https://learnopengl.com/PBR/Theory
- Lens Flares

  - ○
  - ○ simply turn around, as the sun is behind the player spawning position
  - ○ Theory from https://www.youtube.com/watch?v=OiMRdkhvwqg

Illumination & Shading
- All Objects are illuminated by a directional light source.
- All Spheres use physically based shading.

Libraries
- Physx (https://www.nvidia.com/en-us/drivers/physx/physx-9-19-0218-driver/)
- Assimp (https://www.assimp.org/index.php/downloads)
- stb_load_image (https://github.com/nothings/stb)