Gravity Revision

Computergraphik SS15
Plaz Georg

Description of Implementation

A camera within ego perspective follows the player's movement. When the level gravity changes, the camera rotates in the shortest direction possible.
Custom collision detection is used. It works mostly on cubes. Some levels contain moving platforms which carry the user while standing upon.
Objects can be textured, e.g., the pink crystal level-cube which defines the end of each level.
The player can walk ('W', 'A', 'S', 'D'), jump (Space) and shoot (left-click with mouse) with the Gravity Gun (if it is already obtained).
When shooting the surrounding coloured (not the inactive gray-ish) level walls, the gravity will be changed into the targeted direction, which enables complex puzzles to be solved.
Furthermore, portals were implemented with which you can be transported to the corresponding other portal.
Moreover, the game supports Gravity Buttons. When you jump on a Gravity Button, your gravity changes according to how the button's gravity is set. You can see this feature very well in the second level.

Key	**Effect**
W,A,S,D	Walk around
Space	Jump
Left Mouse Click	Shoot Gravity Gun
Mouse	Look around

Complex Objects

Only .obj models with normals can be loaded.

Animated Objects

Any object can be attached to any other object and will move along with it. objects can be stuck to moving platforms and other objects can be stuck upon them.
The most obvious way to see the animation, are the 6 cube particles circling around the goal. if the goal moves (like it does in the very last level and the sliding-puzzle) the particles will continue circleing the moving goal.

View-Frustum-Culling

very simple frustum culling implemented, which calculates the position of each object relative to the players viewing direction. All objects behind the player are not rendered

Transparency

The small particles emitted by the goal are partially transparent. Additionally blending was used to render them.

Features

- Shoot with Gravity Gun
- Changing gravity with Gravity Gun
- Changing gravity with Gravity Buttons
- Portals
- Own collision detection
- Savepoints
- Multiple levels

Objects

- Player - can be controlled. The camera is positioned and fixed in a first person view
- Gravity Gun - the gun used by the player to alter gravity.
- Moving and stationary platforms - can carry the player and other objects. Will function as platforms or elevators, depending on the levels orientation
- Portals - Objects can move through a portal and will be placed on both sides of it.
- Buttons - jump onto them, to activate (currently only buttons in place, which will change gravity)
- The levels goal - touch it to get to the next level
- Cubes affected by gravity - Will be essential to solving some puzzles
- Level walls - the walls surrounding each level. Some are active (colourfull) and some are inactive (grey-ish coloured). The active ones can be shot by the player to alter gravity.

Illumination

All objects are illuminated by a point light, which won't cast any shadows. Additionally every level can produce a spot light which casts shadows. This was only used in the first level to show the effect, since it doesn't make a lot of sense in most levels to place a non-omnidirectional light.
The end of each level which is induced by a pink crystal is texturised.

Additional Libraries

No additional libraries are used.

Effects

- GPU-Particle System
- Shadow Maps

Implementation of Effects

GPU-Particle System

The goal in each level produces particles. Two shaders were created like descriped in several tutorials online. One for updating and one for rendering, along with a transform feedback. There is a single spawing particle, which can be moved to always match the goals position. It produces particles which are later rendered. The particles remaining life-time is also used to calculate it's alpha channel, so it will slowly fade out.
There is a texture used for each particle, which has increasing transparency as you move outwards from the center. The particles texture will be multiplied with a colour which will be passed to the particle shader.
In order to achieve a nice effect with the particles, blending is used.
The transpareny of the objects can be turned off with by pressing F9

Shadow Maps

The shadow mapping works pretty much like it is descriped on the slides in the wiki. All objects (ignoring the basic frustum culling) are rendered before the rest of the scene is rendered. Only depth values are stored and the colour is discarded. This texture will then be passed as a texture to all other shaders, where it can be used for rendering. Some objects ignore the shadows, like the goal-object, along with all of it's small and the 6 cube-particles. Also the portal and the Higgs-Beam will ignore shadows. The Beam will also not cast a shadow.

References:

- Slides to this course: Particle Effects, Shadow Mapping (https://lva.cg.tuwien.ac.at/cgue/wiki/doku.php?id=students:slides)
- Shadow Mapping Tutorials:
  - http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/
  - http://ogldev.atspace.co.uk/www/tutorial23/tutorial23.html
  - http://www.rastertek.com/dx11tut40.html
- GPU-Particle System Tutorials:
  - http://www.raywenderlich.com/37600/opengl-es-particle-system-tutorial-part-1
  - http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/

Tools

No tools are used.

Instructions

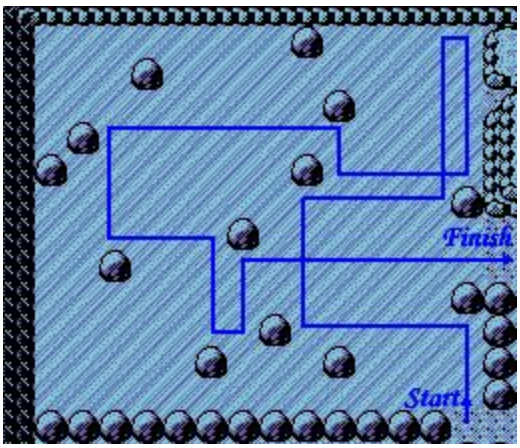
Cheats

- Leftshift + W change gravity to “North”
- Leftshift + D change gravity to “East”
- Leftshift + S change gravity to “South”
- Leftshift + A change gravity to “West”
- Leftshift + Q change gravity to “Up”
- Leftshift + E change gravity to “Down”
- **Leftshift + F finish current level (actually moves the player to the goal. If the goal is cramped into a tiny hole, it will not work. This might be a problem in the sliding puzzle)**

Graphics

- F1 - Erase all displayed text (periodically displayed text will still pop up until turned off)
- F2 - Display fps: On/Off
- F3 - Wireframe: On/Off
- F4 - Texture Sampling: Nearest Neighbor/Linear
- F5 - Mip Mapping: Off/Nearest Neighbor/Linear
- F6 - Display particles: On/Off
- F8 - View Frustum Culling: On/On+display/Off
- F9 - Using transparency: On/Off

Solution for sliding puzzle



(Quelle: https://malexos.files.wordpress.com/2010/08/ice_cave.jpg)