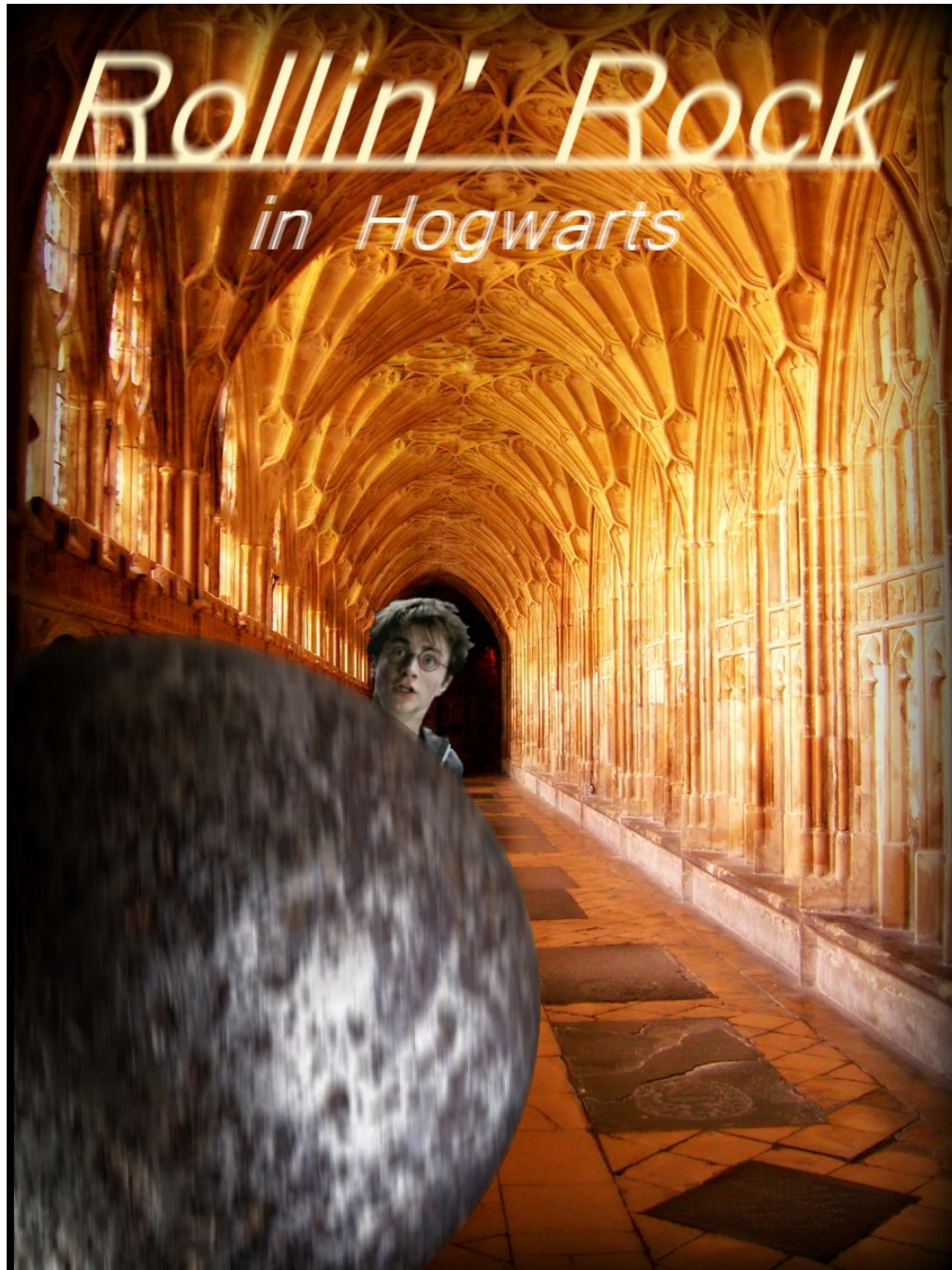


Rock'n Roll with Rollin' Rocks

- Welcome to Hogwarts -

User Manual



Controls:

Keyboard:

I,J,K,L,U,O - tilt the playground (more presses tilts it faster and faster)
Space - start/pause/resume (there is only a symbolic menu, space will start levels)
W,A,S,D - rotates the camera which looks always to the center of the playground
Arrow up,down - zoom in, out

Mouse:

Mouse Button 1 - tilt the playground
Mouse Button 2 - turn the camera
Mouse Button 3 - zoom in/out
Alt + Mouse 1 - tilt the playground on the y axis

The camera is usual a orbiting camera. To change it to a free moveable one alter the movement/cam_orbiting line in the config.ini to false.

Free moveable debug camera:

W,A,S,D - changes the viewing direction
Arrows - moves and strafes the camera
Page up,down - moves it up and down

Alt + Return - *switch to fullscreen*

To change the rendering resolution edit the config.ini "default_resolution_index" to an other one.

F2 - toggle debug output
F3 - Wireframe (set DoF also to off there will be nothing visible)
F4 - switch texture filtering
F5 - toggle Mip Mapping
F6 - reset to Anisotropic Filtering
F7 - toggle MSAA on/off
F8 - toggle View Frustum Culling
F9 - toggle Transparency
F10 - toggle DoF
F11 - toggle Glow

F12 - finished level successfully

Development:

The engine is using Qt for the window handling, and its event system for some timings. The Qt version used is currently 5.0.2 with the desktop OpenGL switch. For the physics NVidia's PhysX is used, which FBX delivers Maya modelling import and a scene graph, although the transformations of the scene elements are partly taken from the physics scene graph, and pushed there from the animation stack of the FBX animation system.

Nearly nothing was hardcoded in the game part, except that the two spot lights above the playground are always shining in the direction of the ball. Some naming conditions are made for modelling new levels. If these naming conditions are fulfilled it is able to create arbitrary levels, with physically correct animated models with help of the Maya animation system., not only flat playgrounds. Actually it is a quite opened system and one would be able to develop various games with it without touching the code in just fulfilling the naming conditions of the connecting parts of the physics, the FBX and the game part.

Effects implemented are:

- Depth of Field
- Glow
- VSM Shadow Mapping

Problems:

- PhysX runs partly buggy, especially with some background apps running (i.e. Maya)
 - also if objects are moved too fast tunnelling appears (i.e. the ball falls through the field, or gets stuck)
 - in pause the simulation sometimes gets stuck
- fbx unpacking takes some time

Problems appeared in the implementation:

- FBX not threadsafe, i would have needed to implement a threadsafe interface to it.
- PhysX - Maya Plugin crashes Maya very often, Multithreading was instable and lead to unstable results.

Have Fun