

Cat to the Past

Documentation

Group/Game Name: Cat to the Past

- Munir Yousif Elagabani, 12022518
 - Studienkennzahl 033 534
 - GPU NVIDIA GeForce RTX 2060 SUPER
- Stefan Brandmair, 12024754
 - Studienkennzahl 033 534
 - GPU NVIDIA GeForce GTX 1060

Brief description of implementation:

A puzzle game, with the main mechanic being that you can rewind time.

Implemented using Rust and Vulkan, without the ECG Framework.

For a list of used sources, see <https://github.com/stefnotch/cgue23-cat-to-the-past/tree/v0.0.2#used-sources>

Additional libraries:

- vulkano - a Vulkan API wrapper
- winit - cross platform window creation
- nalgebra - linear algebra library
- rapier3d - physics engine written in Rust
- bevy_ecs - Entity Component System library
- tracing-tracy - integrating tracy
- gltf - gltf loader
- For a complete list, we recommend consulting https://github.com/search?q=repo%3Astefnotch%2Fcgue23-cat-to-the-past+path%3A**%2FCargo.toml+dependencies&type=code, which lists all direct dependencies.

Gameplay:

Mandatory:

- 3D Geometry: 3D cube with cat logo in 2nd level. We also have a gltf loader.
- Playable: Yes, see walk-through.
- Advanced Gameplay: Time rewinding. We also have objects that can be picked up and thrown.
- Min 60 FPS and Framerate Independence: Yes, on our GPUs. We're using deltatime for all relevant calculations.
- Win/Lose Condition: Win by reaching the final level. Lose a level by running out of "rewind power", which is displayed in the top right. When losing, a "game over" text is displayed, followed by the level being time-rewinded to the starting position. The player then is respawned.
- Intuitive controls: WASD+Mouse+Space for movement and basic actions. Esc to exit. Quake-style character controller.
- Intuitive Camera: First person camera, freecam can be toggled with T. See controls below.
- Illumination model: PBR, as detailed below. Last level has a secondary light that does not cast shadows.
- Textures: Yes, the cubes have mipmapped textures that are linearly sampled.
- Moving Objects: Yes, the cubes can be picked up. There is also a moving platform in level 3.
- Documentation: Yes, see also README

- Adjustable Parameters: Yes, see “assets/config.json”. Depending on the system, the mouse sensitivity can be configured to a lower value such as “0.1”.

Optional:

- Collision Detection (Basic Physics): Yes, you can’t walk through walls.
- Advanced Physics: Yes, the laser in the 1st level is a physics trigger. So are the pressure plates in the other levels. The 3rd level also has a moving platform that pushes the player. We are raycasting to find objects that can be picked up.
- View-Frustum Culling: Yes, we’re doing cheap sphere-sphere intersection tests. The number of culled models is printed to the console every once in a while.
- Heads-up Display: Yes, including a rotating rewind arrow when rewinding time. When running out of rewind power, as displayed in the top left, one will see a “Game Over” text appear.

Effects:

Lighting:

- Shadow Map with PCF: Yes, we implemented it for point lights using cube maps and are using hardware PCF. For optimization, only certain models are marked as shadow casters. Verifying that the artifacts have been countered can be done by playing around with the cat cube in the 2nd level. In certain setups, a bit of shadow acne can be noticed. Eliminating shadow acne is significantly harder with point lights and cube maps, as a constant bias does not work. We also have a shadow_demo.exe.

Animation:

- Vertex Shader Animation: Partial, the time rewinding effect animates vertices in the scene renderer and shadow renderer. It doesn’t recalculate normals.

Shading:

- Physically Based Shading: Yes, most objects have a plain PBR material. The cubes are a textured material. Level 3 has a moving platform with a higher roughness. Level 4 has metallic objects. To properly verify the implementation, we also have a pbr_demo.exe.

Post Processing:

- Bloom/Glow: Yes, see laser. Implemented using compute shaders, with multiple downsampling and upsampling passes. We optimized it since the last submission using shared memory. First, the threads are batched into groups. Then, each thread samples 1 to 2 pixels of the texture, and saves it into shared memory. After that, the downsampling operations are done with the pixels in the shared memory. Reading from shared memory is significantly cheaper than sampling a texture.
We are also reusing the command buffer on the CPU side.
We have a bloom_demo.exe.

Other special features:

- Open source
- Memory safety
- Cross platform, tested on Linux

Walk-through:

The controls needed for the walk-through are

- WASD+Mouse for moving
- Space to jump
- Right mouse button for time rewinding
- Left mouse button for interacting
- Esc to quit

Other controls, for advanced purposes, are

- Shift and right mouse button for fast time rewinding
- F8 enables/disables view frustum culling
- T for swiTching to freecam
 - WASD + Shift + Space to move around in the freecam mod

First Level:

1. There's an open door.
2. Walk through the laser, and the door closes.
3. Rewind time, by pressing the right mouse button. The door reopens.
4. Walk through the door, and give yourself a pat on the shoulder for solving the puzzle.

Second Level:

1. There are tall stairs in front of you, and a cat cube to the right.
2. Pick up the cat cube.
3. Place it on the blue pressure plate.
4. Leave it on the blue pressure plate for a bit.
5. Pick up the cube again, and use it to climb up the stairs.
6. Now, stand on the blue pressure plate that is at the top of the stairs.
7. Rewind time, such that the blue cube lands on the pressure plate. (basically rewind to step 3.)
8. Door should open.
9. Walk to the door, and rewind time a bit if you don't make it to the door in time.

Third Level:

1. There is a platform and a cat cube on top of it. There is a pressure plate to your right.
2. Step on the pressure plate.
3. Run to the platform and pick up the cube. Rewind time if you don't make it to the platform in time.
4. Put the cube on the pressure plate.
5. The platform stays at the bottom.
6. Jump on the platform
7. Rewind time, until the platform fully moved up to get to the second pressure plate
8. While the cube is still on the other pressure plate, step on the second pressure plate
9. Door should open like in the level before

Fourth Level:

1. There are two pillars in front of you. To the left of you is a step that's low enough to climb on, with the help of a cube.
2. Pick up a cube, destroying one of the pillars.
3. Use the cube to climb up.
4. Rewind time, until the pillars are back.
5. Jump over the pillars.
6. Yay