Xplore Project Report

CS174A Spring 2020

Name	Student ID	Email	GitHub Handle
Nathan March	404827938	nathanzmarch@gmail.com	nathanzmarch
Akram Sabbah	504751933	akramsabbah9@gmail.com	akramsabbah9
Alec Serrano Deneken	804754567	alecdeneken@gmail.com	alecdeneken
Matthew Valley	704747855	matthew.valley1@gmail.com	matthewvalley

Idea

The idea behind Xplore was to create a puzzle-exploration type game, taking inspiration from titles like Monument Valley, Fire Watch, and The Witness. We approached the design of this game with the goal of making it as vibrant as possible, with multiple different levels and landscapes to traverse.

How To Play

Xplore is a minimalist adventure game where the player traverses through multiple landscapes to reach the end goal. WASD keys move the player around and mouse-picking adjusts the view perspective. Each level has a trigger that the player needs to stand on in order to transition into the next landscape. There are 6 levels total.

Advanced Features

Collision Detection: Implemented to keep the player inside the game domain, and to keep it out of certain structures, such as trees, cacti and the pyramid. Collision detection was also used in the last level: if the player touched the lava floor, they would respawn at the beginning of the lava, and if they touched a purple obelisk, they would spawn a new platform to walk on.

Shadowing: Projects the shadows of the Forest level's trees onto the floor. (Shader by Jonathan Mitchell)

Particle Simulation: Designed player-tracked snow particles to give the illusion of snowfall. Rather than drawing snow objects across the entire level, localizing the snowfall to the player location notably optimized runtime performance. Snow particles are first drawn at a set height and accelerated towards the ground. When a particle hits the floor, it is redrawn at its initial height again. This is then enhanced by adding several more particles to the operation. This loop of actions creates a seamless illusion of global snow.

Mouse Controls: To properly implement Xplore's controls, the default Movement_Controls (in dependencies.js) were removed. The commands were re-written from the ground up to allow for camera rotation without any z-axis skew. This allowed the orientation of the camera matrix to stay aligned with the level. Javascript event listeners were also added to allow for click-and-drag mouse rotation. Our implementation only rotates the camera when the mouse is dragged in a direction, unlike the default Movement_Controls which rotates the camera as long as the mouse is slightly off-center.

Implementation

All Levels

- Collision detection to bound the domain
- Ground, landscape and sky textures
- Mouse and key controls

Forest - Everyone

- Location randomizer to generate a forest of 100 trees
- Time-dependent sinusoidal motion of flames to achieve desired fire motion
- The trees cast shadows on the ground

Desert - Alec

- Time-dependent location for the flying vultures
- Collision detection such that the pyramid can only be entered through a specific passage
- Multiple objects with different texture coordinates to achieve desired pattern in pyramid
- Heavy use of transformation matrices to generate desired vulture and cacti geometry

Snow - Matthew

- Camera-tracked snow particles increase level immersion without sacrificing performance
- Tree and snowman randomization to create a more natural layout of the environment
- Creative model design: snowmen, mountain, trees
- Large focus on detailed textures: snow, ice, wood, leaves

Ocean - Nathan

- Imports .obj files for Turtle, Whale, Snail, and Shell. Only can extract triangles from one object.
- Whale, Turtle, and Snail Movement
- Generated multiple seemingly random sizes and colors using modulus.
- Texture Scrolling on the roof to appear that the water is moving

Lava - Akram

- Creates a skydome, rather than a skybox, to implement the background.
- Moves the lava texture constantly to the edge, to simulate a sea of flowing lava.
- Adds glass “Platform” objects for the player to stand on, tracked using an array.
    - The player starts off with only one platform at the beginning of the level, but can spawn a path to the goal by touching obelisks scattered around the level.
- Extra collision detection to determine if the player fell off the platform, or touched an obelisk. Uses Math.abs to reduce the number of computations by half.

Contributions

Nathan

Imported snail, shell, turtle, and whale objects, and implemented Level 4 (Ocean), texture scrolling, and creature movement.

- https://github.com/intro-graphics/team-project-team-gfx/commit/effb50551b6c114c7641032a2a7cf9a43cac9121

- https://github.com/intro-graphics/team-project-team-gfx/commit/57bc3709507a7ce4b82740c5729d55ef50dccf7d

- https://github.com/intro-graphics/team-project-team-gfx/commit/06299f115aab5d6f2e24ba3bce236f5830320024

Akram

Implemented camera controls, movement controls, Level 5 (Lava), texture scrolling, shadows

- https://github.com/intro-graphics/team-project-team-gfx/commit/0795b90664c31978d5965d6913c1b50904240dbe

- https://github.com/intro-graphics/team-project-team-gfx/commit/0b8165cbb791f7443bec774ea4fddebb8556bf2b

- https://github.com/intro-graphics/team-project-team-gfx/commit/ffb78966aa8c6a72681cc4615ad49bd6b8ea8648

Alec

Implemented collision detection, cacti and tree forests, Level 2 (Pyramid), flying birds

- https://github.com/intro-graphics/team-project-team-gfx/commit/fd814e002dcca40e076e7a101c790ec4839ab62c

- https://github.com/intro-graphics/team-project-team-gfx/commit/90451aec544ef105f736a32d6f3b09bd024007c6

- https://github.com/intro-graphics/team-project-team-gfx/commit/ec5987fd15190b9a015c2a1a7782fcae6172e99d

Matthew

Implemented snow level and associated features. Snow particles, textures, code organization

- https://github.com/intro-graphics/team-project-team-gfx/commit/caf5806f7c97c18393028362bbb376ead3d5d48a

- https://github.com/intro-graphics/team-project-team-gfx/commit/0da453d260e9230477d4e37d68cbdc63827b4c4d

- https://github.com/intro-graphics/team-project-team-gfx/commit/2928438e3010dead24bdc2aa73ca90c660fbebda

Citations

Shadow shader:
https://github.com/JonathanCMitchell/CS174A_DrivingSimulator/blob/master/shadow_shader.js
Model Imports:

1. Fan Shell - https://www.turbosquid.com/FullPreview/Index.cfm/ID/1562122
2. Snail - https://www.turbosquid.com/FullPreview/Index.cfm/ID/1551575
3. Whale - https://www.turbosquid.com/3d-models/whale-swimming-animation-model-1560985
4. Turtle - http://www.cadnav.com/3d-models/model-47651.html