

| Administrative        |   |
|-----------------------|---|
| Game Name             | Twisted Paths   |
| GitHub Link           | <a href="https://github.com/e11906247/ptvc25-20">https://github.com/e11906247/ptvc25-20</a>   |
| Students              | Maximilian Kraus 11816865<br>Florentina Gara 11906247   |
| Genre                 | Action / Puzzle / Adventure   |
| Goal                  | The player must navigate through a 3D maze, avoid traps and collect a key in order to reach the goal.   |
| Game Idea and Content |   |
| Story                 | The player finds themselves trapped in a mysterious labyrinth. To escape, they must traverse a series of deadly traps and discover hidden paths. Only the key, hidden in the labyrinth, opens the door to freedom.  |
| Gameplay              | The player moves through the maze from a first-person perspective. Various traps such as lava pools increase the difficulty. The goal is to reach the exit as quickly as possible, before the time runs out.  |
| User interaction      | <ul style="list-style-type: none"> <li>● <b>WASD</b>: Movement to traverse through the maze</li> <li>● <b>Space</b>: Jump</li> <li>● <b>LMB</b>: Click to Interact with objects (picking up the key, push a wall, click the final door)</li> <li>● <b>F1</b>: Press to switch to Polygon-Mode</li> <li>● <b>F11</b>: Change between Fullscreen and Windowed Mode</li> <li>● <b>N</b>: Press to see the normals of the lava pools and objects</li> </ul> |
| 3D Objects            | <p>Static:</p> <ul style="list-style-type: none"> <li>● Walls, floor and the roof for separating rooms of the maze</li> <li>● The key</li> <li>● The end door that opens, when the key is collected</li> </ul> <p>Dynamic:</p> <ul style="list-style-type: none"> <li>● Vertex Shader Animation of the lava pool</li> <li>● A Box that needs to be pushed over</li> </ul>   |
| Scene lighting        | Multiple point lights (lava), ambient/emissive lighting, lightmap   |

| Compulsory Gameplay                          |  |
|--|--|
| <b>3D Geometry</b>                           | We have the maze as a whole complex 3D object and a key. Which we both modeled in Blender.   |
| <b>Playable</b>                              | Our game is playable with an in-game character in first person view and various game mechanics, like opening a door and falling into lava.   |
| <b>Min 60 FPS and Framerate Independence</b> | The framerate is tested with a capped framerate of 144fps, and it never falls under 143fps.  |
| <b>Win/Lose Condition</b>                    | <p>The player spawns into the labyrinth, when the key is found the exit door can be opened. If walked through the door, the CMD window outputs a "Winning"- print.</p> <p>A timer is counting down, if it hits 0, the game is lost. There are also various lava traps, where your position is reset (and the key, if found, is lost), if the player falls into them.</p>   |
| <b>Intuitive Controls</b>                    | WASD, Space, LMB, F1, F11, N are the keys, which are used.   |
| <b>Intuitive Camera</b>                      | The first person camera can be moved around with the mouse. It is bound to the character, which can move freely inside the maze.   |
| <b>Illumination Model</b>                    | There are PointLights defined over the lava surfaces, all objects in our game have textures and each object provides Normal Vectors.   |
| <b>Textures</b>                              | Every object in the scene has a texture attached to it.  |
| <b>Moving Objects</b>                        | The exit door is opening when clicked while having the key and there is a crate in the room, which can be pushed over.   |
| <b>Documentation</b>                         | <p>This is the Documentation.</p> <p>We used Blender for the objects, Assimp to load in the models and Physx for the collision detection and basic physics in the game.</p> <p>We integrated Assimp with the help of <a href="https://learnopengl.com/Model-loading/Assimp">https://learnopengl.com/Model-loading/Assimp</a>. We included the assimp.dll and assimp.lib</p> <p>For PhysX we used the Version 5.3.1</p> |
| <b>Adjustable Parameters</b>                 | We have a settings file to adjust the parameters such as screen resolution and with the F11 key, it can be switched between Fullscreen - and Windowed-mode.  |

| Optional Gameplay and Features |   |  |
|--------------------------------|---|--|
| <i>Category</i>                | <i>Feature</i>  | <i>Description (Usage)</i>   |
| <b>Optional Gameplay</b>       | Collision Detection:<br>(6 Points)                              | The player is colliding with the walls, doors and objects inside the maze with Physx.  |
|                                | Advanced Physics<br>(4 Points)                                  | An object (crate) that has to be pushed in order to fall over is added. The push is implemented with raycasting.   |
|                                | Heads-Up Display<br>(4 Points)                                  | Displays in-game text - A key icon, when the key is collected is displayed in the upper right corner, a crosshair in the middle of the screen and a “Winning” or “Losing” Overlay. |
| <b>Effects</b>                 | <u>Lighting:</u><br>Lightmap using Separate Textures (8 Points) | A lightmap is pre-baked in Blender and added to the textures in-game to simulate lighting.   |
|                                | <u>Animation:</u><br>Vertex Shader Animation (8 Points)         | Simulating waves on the lava surface. Normals are calculated dynamically.  |
|                                | <u>Advanced Modelling:</u><br>CPU Particle System (8 Points)    | Yellow Line Particles that are shown at the area where the player has to go through in order to win the game.  |
|                                | <u>Texturing:</u><br>Video Texture (8 Points)                   | A tutorial video is added at the start of the maze, for the to get information on what to do.  |