Documentation

**Group/Game Name:** FairyEscape  
  
**Brief description of implementation:** You play a fairy that is stuck at the bottom of the well and is trying to reach the top of it to win.  
  
**Additional libraries:** tinyobjloader, stb\_easy\_font

**Gameplay:**

**Mandatory:**

* 3D Geometry: The wings, the finish-flag and the cloud the flag stands on are complex externally loaded objects.
* Playable: Yes.
* Advanced Gameplay: Yes.
* Min 60 FPS and Framerate Independence: Yes.
* Win/Lose Condition: You win by reaching the finish-flag. You lose if your flying power hits 0.
* Intuitive controls: WASD and Mouse for movement.
* Intuitive Camera: Third-person camera which follows the fairy. Key “C” opens the debug camera, which is movable with WASD and the mouse. The debug camera pauses the flow of the game.
* Illumination model: One directional light shines from the top of the well into it. One point light illuminates the well itself, while another point light follows the fairy to illuminate objects around it more.
* Textures: Yes.
* Moving Objects: The fairy (sphere and wings) is movable.
* Documentation: This.
* Adjustable Parameters: The Fullscreen mode and the screen resolution are adjustable using a config file. Additionally, the key “F3” toggles the Fullscreen mode at runtime.

**Optional:**

* Collision Detection (Basic Physics): Basic collision detection is implemented. The player cannot move though the walls of the well. The Camera also cannot clip out of the wall
* Advanced Physics: /
* Scripting Language Integration: /
* View-Frustum Culling: /
* Heads-up Display: Key “1” toggles the visibility of the HUD. It shows the remaining flying power with a bar and a heart on the bottom left corner.

**Effects:**

**Lighting:**

* Lightmap using Separate Textures: /
* Lightmap using In-Game Calculation: The well wall is precomputed on startup and thus illuminated.
* Shadow Map with PCF: /
* Shadow Volumes: /

**Advanced Modelling:**

* CPU Particle System: Key “2” toggles this effect. Particles (white, small quads) are sprayed from under the fairy, representing glitter particles for aesthetic purposes.
* GPU Particle System using Transform Feedback:/
* GPU Particle System using Compute Shader:/
* Blobby Object Using Marching Cubes:/
* Subdivision Surface:/

**Terrain:**

* Tessellation from Height Map:/
* Voxel Terrain using an Octree:/

**Animation:**

* Hierarchical Animation:/
* Vertex Shader Animation: Key “3” toggles this effect. At the bottom of the well is a blue plane, that is animated as if there are water waves.
* GPU Vertex Skinning:/

**Texturing:**

* Procedural Texture:/
* Video Texture:/
* Specular Map:/
* Environment Map:/

**Shading:**

* Simple Normal Mapping:/
* Cel Shading:/
* Style Transfer:/
* Brush Strokes:/
* Physically Based Shading:/

**Advanced Data Structures:**

* BSP Tree:/
* kd-Tree:/
* LOD using an octree:/

**Post Processing:**

* Bloom/Glow: Any pixel brighter than a set threshold is copied to a “bright” texture (all others go black), that bright texture is Gaussian‐blurred, and finally the blurred result is added back onto the original image to create the bloom/glow. It is seen best on the fairy and the rings.
* Lens Flares:/
* Contours via Backfaces:/
* Contours via Edge Detection:/

**Other special features:** /  
  
**Walk-through:** You start at the bottom of the well. You can see your starting flying power at the bottom of the HUD. You can move around freely in the well. The flying power is starting to deplete instantly, so you need to fly through a magic ring to gain a certain amount of flying power back. There is a maximum you can have. If you look up, you can see a cloud with a flag on it at the top of the well, that is the finishing point you need to get to in order to win the game. As long as you have flying power you can continue moving. If you do not move the fairy character, it falls downwards because of gravity. If you don’t manage to get to the finishing point, and have 0 flying power left, the fairy starts falling and hits the bottom of the well, resulting in a loss.

**Controls:**

WASD and Mouse: Player Movement

ESC: Quit

F1: Wireframe toggle

F2: Culling toggle

F3: Fullscreen toggle

N: Draw Normals

T: Draw Texture Coordinates

C: Debug Camera

1: HUD toggle

2: Particle System toggle

3: Vertex Shader Animation toggle  
  
Third-Party Asset Credits:

For full details and attributions of all third-party 3D models and textures used in this project, please see the README.md in the root of the project repository.