

# Developing a Virtual Reality Game

Results of a practical course at the Chair for Computer Graphics and Multimedia  
(RWTH Aachen University, Germany)

Sebastian Filla\*

Tanja Schmelter†

Nils Lukas‡

Philip Schwarzbauer§



Figure 1: *Cave Pursuit* is about surviving as long as possible.

## Abstract

Our Game "Cave Pursuit" is a horror game, putting the player into a closed cave system. The player is not able to leave the cave. The goal of the game includes escaping the monster in the cave and surviving as long as one can. With time passing the monster will duplicate and multiple monsters will go after the player, therefore increasing the difficulty of the game. The player is able to interact with his environment to archive a bigger lead on the chasing monster which is key to surviving as long as possible.

**Keywords:** game programming, virtual reality

## 1 General Information

The game aspects of "Cave Pursuit" are subdivided into 5 main topics. These are Game Mechanics, Code Structur, Graphics, Content Creation and Sound. They are explained in details in the following sections.

## 2 Game Mechanics

As with other games, this game also has certain game mechanics. They include environmental features like the lava, certain player items the player can interact with and the live management. For physic simulation and raytracing, the library "Bullet Physics" is used. In the following this is described in more detail.

\*sebastian.filla@rwth-aachen.de

†tanja.schmelter@rwth-aachen.de

‡hendrik.nils.lukas@rwth-aachen.de

§philip.schwarzbauer@rwth-aachen.de

### 2.1 Environmental Features

The game is centered around a monster patrolling the cave on pre-defined waypoints. As soon as the monster has visual contact with the player it starts hunting the player down until the player dies or the line of sight is intersected. If it loses sight of him, the monster will continue on its patrol path. Furthermore, if it happens that the monster is stuck somewhere, it will teleport to a random waypoint.

The Cave also has moving objects. The axe above the water ditch kills the player on contact. The player can disable the swinging axe by pushing a button in an nearby room while gates found in the cave can be opened and closed by buttons next to them. These gates should be used to slow down the monster, allowing the player to escape from it for a certain time.

Finally there are two life-threatening areas, the water ditch and a canyon filled with lava. Both will instantly kill the player if he falls into one of them.

### 2.2 Player Items

The player has items he can use to interact with the environment. With the torch in the player's left hand, he is able to ignite other torches scattered around the cave. This can be used to get a better overview of the surroundings and see the monster from further away. However, there is a set period of time in which the torch is ignited. After it runs out the torch's flame will extinguish.

The second object the player is able to directly use is the magic ball. The player is able to shoot this ball at his enemy, the monster, to stun it for a short period of time. This gives the player time to escape out of the range of the monster. However, the player cannot

shoot the magic ball continuously as there is a short delay until he is able to use it again, and duplicated monsters are immuneto magic balls.

### 2.3 Live Management

As the player cannot escape the cave system, he will eventually die, resulting in the game ending. There are two groups of objects able to kill the player, the first one being the monster which searches and hunts for the player throughout the game. The other is the environmental objects like the water, lava and the axe. If the player is killed by any of these, the death animation will play and the game restarts.

## 3 Code Structure

The code structure of the game includes an Entity-Component System. Every object created in the game is an entity only consisting of a unique ID. All functionality will be handled by the game systems using data of its corresponding components. Components can dynamically be assigned to entities with no constraint to diversity, amount and type of components. For example, every torch alongside the cave walls consist of a physics component, a rendering component and a gamelogic component. The gamelogic component manages the addition of particle and light components in dependence to user interaction.

## 4 Graphics

The graphics are an important aspect of every game. These include shadows, illumination, post processing effects, the particle system and deferred rendering.

### 4.1 Shadows and Illumination

The blinn-phong-lighting model is used to illuminate the world. To achieve smooth lighting the surface normals of the meshes are distorted using normalmapping. Basic shadowmaps are supported to allow shading from multiple light sources.

### 4.2 Post Processing

Postprocessing effects are used after the complete scene is drawn. Several effects are supported, like blurring the output or making objects glow. The death animation consists of a radial blur effect emitting from the screen center.

### 4.3 Deferred Rendering

Applying the abovementioned shadow and illumination techniques is done by a deferred rendering pipeline. By first drawing world position, surface normals, glow and diffuse color to textures appended to the Geometry Buffer one is able to minimize the amount of expensive lighting calculation. These four textures are used in the lighting-step, where lights are rendered as meshes containing their light volume. Light calculation is only done if an object is contained in this light volume, which is determined by using the stencil test. Since every light is drawn individual it is very easy to apply custom modifiers like shadow maps.

When all lights are drawn, the postprocessing pipeline starts with applying glow, drawing particles using alpha blending and blur if needed.

## 4.4 Particle System

The particle system is responsible for the torch's flames and the teleport effect of the monster. By adding a particle component to an entity one is able to emit particles with custom configurations ranging from lifetime, amount of particles spawned and displacement to sprite and color animations. The particle system will choose random numbers in between those predefined min/max-ranges at spawn and interpolate between texture and color steps from the color-configuration, resulting in smooth transitions between the different states each particle could have.

For rendering, every particle consists of a single vertex containing position, size and similar information. All particles will be pushed into a single object thus allowing to handle all particles with a single draw call. The geometry shader enlarges points to quads aligned with the camera to allow efficient rendering of big amounts of particles.

## 5 Content Creation

Another important aspect of the game is the content creation. In this case, the whole content was created by ourselves, using the program Blender. The most important part was the modelling of the cave system. The whole cave system consists of 9 individual parts that fit perfectly together at their connection points. The process of creating them was stretched across the whole practical course time.

Other models created with Blender where the furniture, torches and the player's body. The furniture was included as decorative items with the hint that once upon a time humans inhabited the cave system. Furthermore there was only one model made for a torch which was reused in the game. They are either used as interactive objects scattered in the cave as a lightable light source or as a torch carried around by the player the whole time.

Finally, there are the body parts of the player which are visible time to time or always. These are his arms and hands as well as his feet and legs. The only body part which is always visible is his left hand holding the torch to lighten the environment.

However modelling is just the first part of the content creation. The second part includes searching for suitable textures for the models or making them yourself. Most of the used textures were not self-made but open source material, except the magic ball, the gate, the axe and the monster. With the help of UV Mapping in Blender the models were textured. All models use a single texture (e.g. table, chair). The only exception is the cave texture which is composed of 3 different textures in one.

## 6 Sound

The sound in the game was included with the help of the library OpenAl. There are 3 categories of sounds included in the game. Environmental sound, movement sound and others. The environmental sound is largely responsible for the creepy atmosphere. But it also includes the sounds made by the water and the lava.

Movement Sounds are only made by the player himself or the monster patrolling the cave. As the player is barefoot he does not create much sound while walking. The monster however does not have any walking sounds but intensive snarling. The remaining sound includes the opening and closing of the gates.