Building a Virtual Reality Fire Training with Unity and HTC Vive

Bettina Schlager*

Supervised by: Anton Fuhrmann†

VRVis Research Center, Vienna / Austria

Institute of Computer Graphics and Algorithms - Vienna University of Technology, Vienna / Austria

Abstract

Simulations in virtual environments for educational purpose are well-known in many domains to achieve an interactive learning experience. This paper addresses the development of a virtual reality application for training non-professionals in using a fire extinguisher. Virtual firefighting is advantageous compared to fire training in a real environment in terms of security, practicality and the selection of training scenarios. Additionally, it is a cost-effective alternative while offering a more engaging training experience. The state-of-the-art virtual reality device HTC Vive, paired with the high-level game development framework Unity, is used to develop a 360-degree fire simulation and therefore allows trainees to interact with a fire in a safe setting. The scenario can be controlled from an external supervisor for professional learning feedback. Through visual effects, realistic lighting, suitable shading methods and a physical approximation of fire growth, we achieved to develop a realistic and an immersive learning environment which is also maintainable for further development.

Keywords: Virtual Reality, Realtime Rendering, Virtual Learning, Fire Training, Haptics

1 Introduction

Fire training for non-professionals can be an expensive and complicated setting and mostly carried out in a minimalist context to reduce trainees risk. This paper guides through the development of a virtual fire training application to offer an insight on designing a practical training alternative with an immersive learning practice in virtual reality.

A good training experience can help to recall right choices in emergency situations fast. Due to safety security requirements of a real fire training, non-professional trainees are learning within limited circumstances and are not dealing with impact of smoke development and enormous fire growth in indoor rooms. Within the practical offline training in Austria, the fire scenario is simulated by putting out a gas-burner with different types of a fire extinguisher. Figure 1 shows an attempt of a trainee to aim at a minimal fire with a dry powder fire extinguisher. The instructor is supervising the situation for further instructions or interventions. The main objective of this exercise is to teach the suitable usage of the various types of fire extinguisher and how to use a fire extinguisher correctly. Further, it is important to learn when to flee instead of attempting to fight the fire, feel the weight of using a real extinguisher and get to know the basics of a proper fire extinguisher maintenance. By working through a practical fire training people are more confident when dealing with an emergency and are able to analyze critical decisions faster. Despite the high learning output of the real training, a virtual setting would offer more freedom regarding the fire environment, growth and space. Especially indoor fire, where most of the incidents occur, can be simulated with the aid of virtual reality for an even better learning practice.

The developed application guides trainees through a safe fire training experience without the need to endanger human life or overspend on unrealistic fire training setup. Therefore, the proposed application is more advanced regarding training experience by offering virtual fire fighting indoor scenarios. Virtualization of the interaction space allows dynamic adaption of the fire environment and is not involved with an inconvenient setup process, which would be the case in a real dynamic fire environment.

Figure 1: Insight on a practical fire fighting training in Austria. Image courtesy from VRVis GmbH

Learning in a virtual environment is known as beneficial

*schlager@vrvis.at

†fuhrmann@vrvis.at

in many ways and is already used in many different domains like education, military and health. For a true learning experience, there are several characteristics of virtual education software to consider during design and implementation. This work will guide through the development, including comparison of similar applications which were built to train people within virtual simulations.

While virtual reality hardware is well-studied and evolved throughout the last decade, virtual reality frameworks and development platforms emerged recently. This application is a base for a low-cost education for non-professional fire fighters and future fire security officers. This approach does not simulate real heat, but it teaches the user about the fast development, growth of fire and smoke and handling of a fire extinguisher. The application is set up for trainees to understand a fire emergency situation through a hands-on experience.

Figure 2: User engagement with the aid of a stereoscopic display. The trainee is able to interact in a 3D space by shutter glasses and a stereoscopic display. Image courtesy of Maschek [7]

2 Related Work

Serious gaming for educational purpose is not a new concept, but adapted to today's virtual reality hardware, new possibilities arise. Immersion in training is well discussed and aims to support the learning experience by using skills in a context which is similar to the real world. Dede [6] investigated several studies concerning the degree of immersion in learning and noted that immersive virtual environments can enhance learning by allowing situated settings, transferring knowledge in the belonging context and changing perspectives on the learning background.

Encouraging right reactions in emergencies through serious games is also used for other domains which expect a fast decision-making process from responsible persons like during biohazards, terror attacks, floods or power outages. Cohen et al. [3] worked on applications that prepare hospital employees for certain emergency scenarios like a bomb blasting and its consequences. The software is developed with experienced surgeons and hospital leaders to brief staff members the emergency standards. The user study evaluation showed a strong interest on learning in virtual environments, since this could be a low-cost alternative to real-world practice, which takes place on a regular basis. Although the approach showed great results, bringing this learning output to virtual reality instead of a desktop applications, has the possibility to achieve even better results.

There are few comparative papers for virtual fire training published. Despite all of them are happening in a virtual environment with approximated fire simulations, they face different challenges and have contrasting objectives and priorities. Williams-Bell et al. [8] investigated several approaches for a good overview of current research. The majority focused on fire simulations for professional fire fighters to enhance their decision-making and team coordination skills. The review also described different utilities aiming to increase the immersive experience such as early versions of head mounted displays, stereoscopic glasses, breathing apparatus [4], cave automatic virtual environments and other hardware.

One of them is called Blaze [5], a fire training for individuals. The user faces fire emergency in the kitchen and has to react correctly depending on the type of fire. During the simulation, trainees can choose between several obstacles, like the fire extinguisher, water or baking soda for facing the fire. When the user puts the fire out, feedback is given by increasing or decreasing a user score and awarding trophies. To simulate the fire hazard over time, the user's avatar signals critical situations of smoke development by coughing. The application is based on Unity and uses virtual heating blocks, which are distributed in the environment space, to put fire spreading into practice. Although the user interaction is very advanced, the software is not available for virtual reality glasses or head mounted displays, but regular 2D displays.

Another comparable work is developed as a virtual reality fire training application to train prospective fire protection officers, business or private clients for fire emergencies. [7] The main objective is to overcome the difficulties and disadvantages of a real fire training. Mascheks setup is based on sensors on a fire extinguisher, the trainee's glasses and the sleeve of the extinguisher which are interacting with a stereoscopic large-screen projection. The learner perceives the projection as a 3D scene through circular polarized glasses and is able to put a fire out while even using a real fire extinguisher. Figure 2 shows the interaction of a user with the fire training software. Sensors on the extinguisher register user engagement and give immediate feedback on the display in form of water, carbon dioxide or powder. Through the three-dimensional experience on a two-dimensional screen, the trainee's perception is limited to a narrow field of view and has a finite interacting area.

Moohyun et al. [2] developed a training program where a head mounted display with a motion tracking system is used to guide a trainee through virtual fire safety challenges. It aims to enhance the practical training of future fire officers. An outside instructor is able to track the trainee's achievements, mission status and other parameters as fire, smoke and temperature development. The behavior of fire, smoke and other fuzzy objects were calculated by computational fluid dynamics. Compared to the mentioned application before, Moohyun et al.'s approach is more complex and has a broader decision-making context like when to escape the burning scene and how to help other people in the fire. Mascheks training software, on the other hand, is developed for trainee's which seek confidence in using a fire extinguisher.

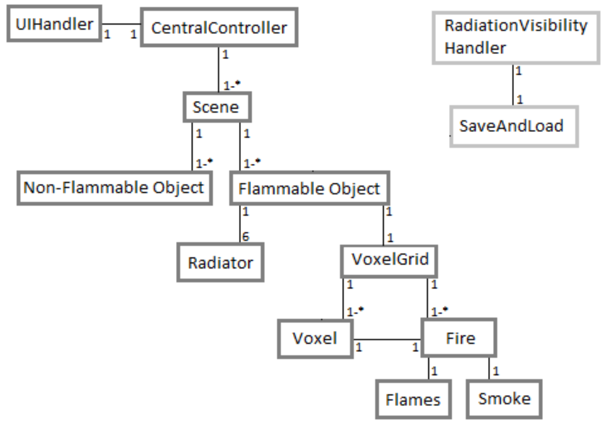

Figure 3: A system diagram of the developed application which reveals a high-level view of the internal structure.

3 Implementation

The aim of the development is to offer an immersive learning software which should also be adjustable for further enhancement. For the framework we chose Unity, which is a high-level game engine and supports a fast development process. One of the advantages with Unity is large-scale of already implemented functionalities such as particle systems, shading programs and lighting calculations. Since we could use and adjust build-in components, we were able to focus on customizing the framework for our need. Unity also supports common file formats for 3D models like Autodesk Collada (.fbx) or Wavefront (.obj). Compared to traditional game development, building virtual reality applications for state-of-the-art head mounted displays is a new concept and is facing unknown challenges. Unity recently started to offer further functionalities for virtual reality development with HTC Vive and other head mounted displays.

For an immersive experience, several requirements were considered before implementation. A good and fluent real-time performance with heavy calculations is a challenge, but for a virtual reality application to offer a good quality experience, it must achieve at last 90 FPS. with few allowed drops at start-up of each scene. Such a high frame rate is especially important when working with virtual reality head mounted displays, because low frame-rates can cause negative effects on the well-being of the user, like motion sickness. Further, to allow the user learn fire dynamics in a small environment, the approximated fire growth should be as real as possible. The next sections explain the development of the requirements in detail.

3.1 System Design

Figure 3 shows the scene graph of a fire scenario. Every scene has flammable and non-flammable models. Models which are non-flammable models are static and do not alter during run-time. A 3D grid of voxels is placed on the corresponding flammable object's position. Voxels are used to save information such as current temperature, physical properties and current state. Figure 4 displays the voxels with their associated rendered models. The grid information is loaded from a 2D raw image file by reading it as a bytestream, where black pixels indicate empty placeholders and white ones, above a certain threshold, represent voxels. This image is calculated by an internal software beforehand, the grid is shaped like the respective 3D model in the simulation and accessed at run-time. Due to heat transfer, which is explained in another paragraph, every flammable object has six radiators and are placed on each side in a cubic shape. A radiator is a non-visible object which is loaded and activated at run-time and continuously receives and sends heat from other burning objects in the scene.

Concerning the architecture, the software should be maintainable as possible for future changes, like implementation of further simulation complexity, adding customized simulation environments and more user engagement possibilities. We decided to work with an object-oriented approach for the fire simulation and component structure for single object behaviors like the fire extinguisher. Further, a combination of the singleton and lazy initialization design pattern is used where every object in the scene has responsibilities according to their behavior in the real world. Figure 3 reveals the structure of the main components, where a single central controller is handling the user input and creation or deletion of simulation scenes.

Due to performance optimization, the child objects of all flammable objects, like the voxelgrids and the radiators, are destroyed and created during run-time. The voxelgrid object is responsible for the main functionalities such as heat transfer and fire handling. It is able to start a fire in case the temperature of a voxel in the grid exceeds the autoignition temperature of the related material or to stop the fire if the trainee used the right fire extinguisher fluid.

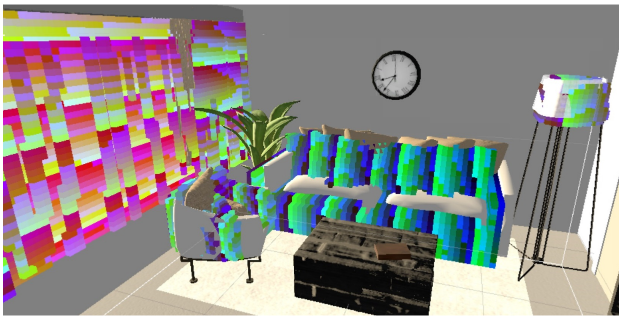

Figure 4: Every flammable object has a corresponding voxelgrid. Voxels save information about heat transfer, state changes and material properties.

Next to the scene structure, there are several static executive functions which support needed calculations such as a conversion, material dictionary for fire growth and file reading and writing.

3.2 System Interaction

Since it is going to be a supervised learning process, the instructor has to be able to control the simulation and to see what the user is doing. The user interface for the supervisor is forwarding the commands to the central controller object, to achieve a separation of concerns. While the trainee is interacting with the scene through the head mounted display with a first-person view, the camera of the user interface is positioned from an aerial perspective, to allow supervision of all burning objects and the reactions of the trainee in real-time. This allows personal feedback from the fire expert to the trainee during and after the simulation.

This version of the learning software offers two possible simulation scenes, a kitchen and a living room. Each of them contains flammable objects with different materials like burning furniture, electric devices or oil.

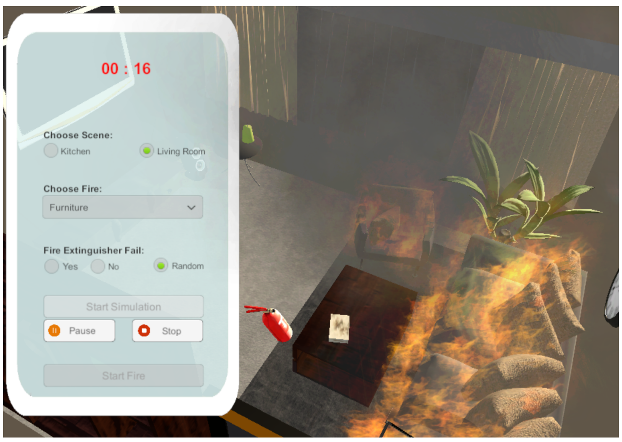

Figure 5: The instructor is observing the simulation from a bird's-eye view to analyze scene events and give immediate feedback.

In the living room, it is possible to choose between electric and furniture and in the kitchen, between burning oil and furniture fire. As Figure 5 shows, through the user interface the instructor can choose to start either the kitchen or the living room simulation, what kind of fire should be triggered and when, and to pause or stop the simulation. Further, it can be selected if the fire extinguisher should fail during the simulation process.

The interaction possibilities for the trainee in-game are important for an immersive engagement. For simplicity we chose two common types of fire extinguisher, water and carbon dioxide. For the trainee to feel the impact of the weight of a real fire extinguisher, the controllers of the head mounted display, which act as virtual hands, are going to be attached to a real fire extinguisher each. For an optimal learning effect, the real extinguishers can be placed in the simulation space of the Vive before training start. The simulation space is the area the user can move and the controllers can be tracked by the sensors of the HC Vive. The instructor can then start the desired fire in the desired room. During the simulation, the trainee has to make the right decision which fire extinguisher he should grab according to fire type. If the trigger of the controller, which represents the trigger of the extinguisher, is used, a virtual particle system is seen by the user. Figure 6 shows the difference between the carbon dioxide and water particle system. With the fire extinguisher, the learner can then put the fire out. To ensure safe distance to the fire source and fast reactions, the vision of the user blurs if he is too close to the fire or if the fire has grown too big. As mentioned before, the kitchen scene offers burning oil. If the user tries to extinguish the oil with the water, an explosion occurs and simulation is over. Should the trainee extinguish the fire with the right extinguisher on time, a winning sound is triggered and on failure, a failing sound.

Figure 6: The trainee can choose between two types of fire extinguisher, the one on top simulates water and the one on the bottom, carbon dioxide

On top of the visual feedback of the water and carbon dioxide particle system, the user is also able to hear the sound of the extinguisher if the trigger on the controller is enabled.

3.3 Fire, Smoke and Other Particle System Development

In Unity, several functionalities are already implemented and ready to use. Particle systems are available and come with great adjustment possibilities. In our application, several particle systems are used such as flames, smoke, convection, water and carbon dioxide. The latter two simulate the liquid of the fire extinguisher. The convection particle system, which is later explained in detail, is used for heat transfer and the other two represent the fire itself. Particle systems allow fast movements and texture rendering to achieve the look of a random change in the shape of a real fire or smoke.

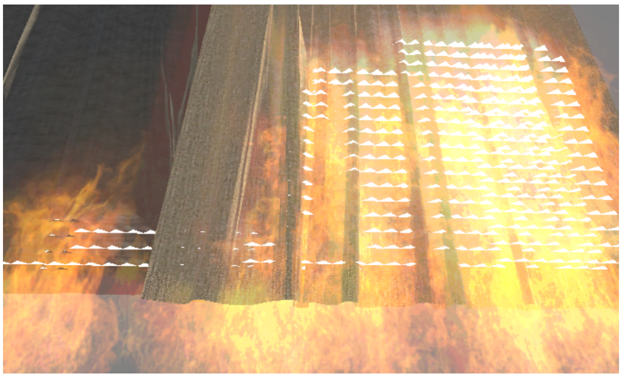

Figure 7: The flame and smoke particle system is using a dynamic changing mesh as emitter, which is built at run-time to achieve visual, local fire spread on a flammable object. Local fire spread is similar to the physical heat transfer phenomenon conduction.

Almost all of them are using the standard particle shader program and a billboard as particle shape, except the flames of the fire. To get a more realistic fire impact effect on the environment, we achieved the impression of heat haze by implementing a special distortion shader program on the material of the particles which allows a blurred effect around the fire. A closer look on Figure 7 reveals the small distortion effect around the flames. Moving the calculations of the heat haze effect to the shader program is also enhancing performance compared to the alternative of adding a third particle system to the fire object to simulate such a distortion. Concerning the amount of particles, the flames and smoke particle systems dynamically adjust to the size of the belonging fire. If a certain ratio of the voxelgrid is burning, the size of the particle system is growing and the emission rate rises. Also, if the fire is extinguished, the amount of burning voxels, the fire size and the fire emission rate decreases.

When using functionalities which are already available in the framework, there are always limitations when it comes to custom demands. For the heat transfer, we had to use an alternative procedure. For the physical calculations of the fire spread, a heat transfer framework is used which is provided by an internal project at VRVis. The calculations are based on physical conditions of the environment and the material of the burning object. [1] In general, heat is transferred by three phenomenons: conduction, radiation and convection.

Conduction: In physics, conduction is the transmission of high temperatures. In terms of heat transfer, local fire spread is achieved. To implement this process in our software we decided to use a dynamic mesh as emitter for the flame and smoke particle system. Figure 7 shows the emitting mesh of the flames of a fire. As a high consuming calculation, the mesh is built in real-time depending on the current heat of the voxel. If the temperature of a voxel in the grid is high enough, a triangle in size of this voxel is added to the emitting mesh on the respective position. If

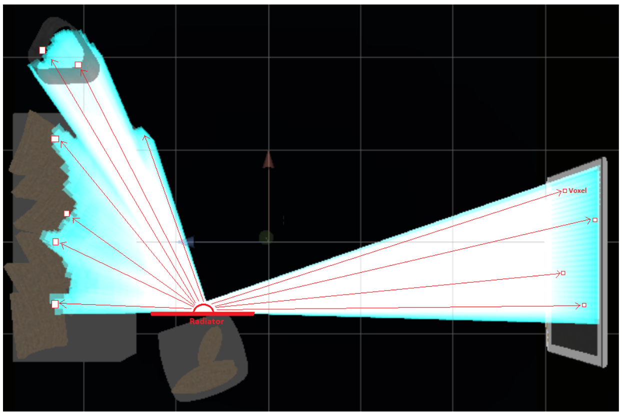

Figure 8: Visibility of a radiator of the chair. Each line, some are highlighted, represents a connection between the radiator and a voxel of another object. This information is used for heat transfer at run-time. If the chair burns, the radiators spread heat to all voxels it sees.

the voxel is hit by particles of the fire extinguisher fluid, the belonging triangle is removed from the emitting mesh. For our approach to work, the state and temperature of all voxels in the scene have to be updated during run-time. To enhance performance and balance the workload, we decided to update only a certain amount of voxels of each flammable object per frame.

Radiation: Thermal radiation is achieved through precalculation of heat transfer information. By loading the references of the voxels every radiator in the scene is seeing, which Figure 8 explains, a burning object can transfer heat to nearby flammable objects. Thanks to preprocessing the visibility, radiation heat transfer is carried out without dramatic impact on the performance. The radiation visibility text file saves the information in this format:

{ "Sofa" : { "Sofa RadXPos" : { "Chair:102019" : 7.465889E-05, "Chair:101919" : 7.723208E-05,..} } }

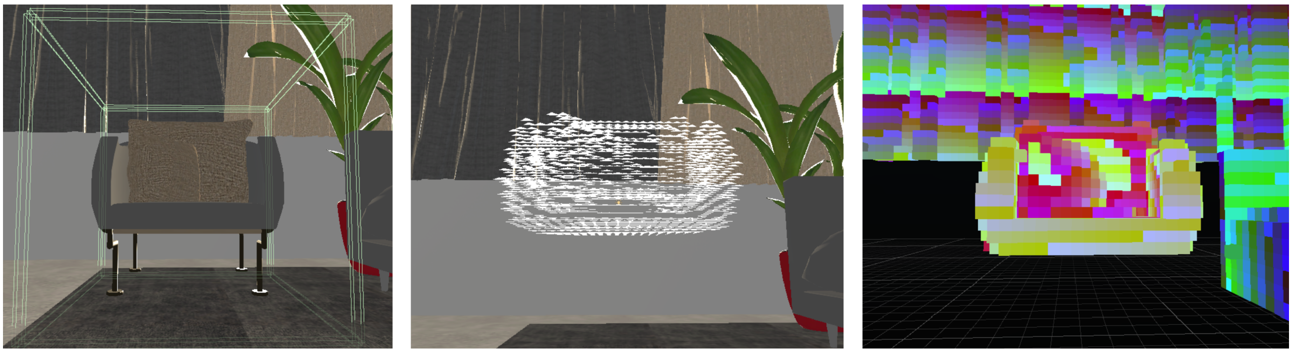

Figure 9: Information Layers of flammable objects. The first image shows the rendered mesh and the respective radiators on each side, which is seen by the trainee. The two images on the right represent the information which is usually not rendered, like the emitting mesh of the fire and voxels of the voxelgrid in shape of the flammable object

For every flammable object, and each of the six radiators, a dictionary is saved. The dictionary contains the position of the voxel in the voxelgrid of the belonging flammable object and a form-factor which represents the distance to the given radiator. From the file example above: The radiator of the sofa, which is pointing in the direction of the positive x-axis, sees a voxel of the chair object. The voxel's position in the grid is given and a calculated distance to the positive x-axis radiator. Figure 8 is an example of the visibility of a radiator. Every line represents a connection between a radiator of the chair a voxel of nearby objects. This information is accessed during run-time to transfer heat to other objects in the room.

Convection: Explains physical heat transfer through gas movements and allows the heat in the simulation to move up. This is achieved through an invisible particle system, placed in the center of every flammable object when the respective fire starts burning. More convection particles are emitted, if the temperature of the voxels in the grid is increasing. When the particles of the convection collide with other flammable objects in the scene, heat is transferred. Due to convection, flammable objects above an active fire are likely to burn faster.

3.4 Realtime Calculation Constraints and Preprocessing

Since the recommended frame-rate of a virtual reality application is 90 FPS, we have to optimize performance to a large degree. As mentioned previous, for several properties of the scene setup, we decided to preprocess the information and make them accessible during run-time through external files to achieve a performance without interference. The visibility of each radiator is computed beforehand and saved to an external file. This file is accessed at start-up and processed in real-time. Further, to add more realistic shading to the scene, burning or burned vertices of the rendered mesh of a flammable object are tinted in a darker colour. For this process we also needed to precalculate a vertex-voxel dictionary, where every vertex of the rendered mesh of a flammable model is mapped to a voxel in the voxelgrid. At run-time, it is then possible to change the colour of the vertex without complicated real-time calculations.

Due to the dynamic fire growth, not all information can be processed beforehand. Calculations which are consuming high computation time are the real-time updates of voxel parameters such as temperature, the particle systems and radiation calculation. The previous section already mentioned implemented real-time processes like heat transfer through radiation, conduction and convection, adjustment of fire size and shape, editing the dynamic mesh for the fire emission and handling collisions of fire extinguisher fluid particles. Concerning lighting and rendering, Unity is very adjustable and no further calculations have to be done manually. Due to the fact that all objects, except the fire extinguishers, are static, the lightening of the scene is baked to a light map and loaded at run-time. Baking a scene is already one of the implemented functionalities in Unity. For high realistic flames with the given transparent particle shader, we decided to light the scene in an evening atmosphere. Further, concerning lighting, which is usually a high cost real-time effect, we decided to limit the quantity of dynamic lights to a small number, such as the fire lights which simulate fire flickering.

4 Results

An approximate simulation and visualization of a complex real world phenomenon is not straight forward. Figure 9 shows the resulting information layers of a flammable object which are processed at run-time to simulate the heat transfer and fire in the scene. Figure 10 shows the final lighting of the living room scene. Through continuous improvements of the workflow and communication between dynamic objects, the resulting application achieved a fluent simulation running with with an acceptable frame-rate of 85-90 FPS, depending on the amount of burning objects in the scene. The setup was an Intel(R) Core(TM) i7-4770 CPU @3.4GHz, 32GB working memory, a NVIDIA GeForce GTX 980 for the graphical processing unit and Windows 10 64-bit as operating system. Due to the independent structure of the program and separation of concerns, the application is adjustable for further enhancement. With few adaptations in Unity, scenes can be adjusted or new scenes can be added and visibility data is calculated automatically.

Figure 10: The resulting simulation scene with pre-baked lighting. To enhance the visual effects of the fire, the room is lit in an evening atmosphere.

Compared to mentioned related work, the application is developed with a state-of-the-art head-mounted display, which offers a large space for movement in the virtual world and a 360-degree experience. The implemented features, like audible effects, physical fire growth, interaction with the virtual fire extinguisher, which is also present in reality and dynamic control of the simulation through an expert, offer trainees immediate learning feedback and a situated learning effect. With this training software, future fire executive officers or other non-professionals can learn how to use a fire extinguisher in a safe environment and are able to get a new perspective on fire dynamics. Related applications usually focused on virtual training for professional fire fighters or enhancing teamwork in case of an emergency.

Our visual senses are one of the most important data providers for perception. No computer simulation can simulate heat or smoke impact on the body without the help of external hardware, but the resulting behavior and appearance of the virtual fire and smoke particle systems and the implemented blur feedback is able to simulate the heat by fooling our perception a hot environment. Since there are no user studies executed, the degree of immersion is just tested internally.

5 Conclusions

Our objective, to create an immersive fire training simulation is reached through a well-thought system. Despite the fact that the simulation is running on the required hardware with the recommended frame-rate, Unity might not be suitable if the virtual training is enhanced by more complex features in the future. The game engine is a high-level designing tool and not perfectly suited for custom calculations and processing at run-time. Further, virtual reality support is rather new to Unity and there are custom adjustments necessary for an experience without frame-rate drops. During development, we recognized that, several implemented functionalities in Unity do not offer support for virtual reality products yet.

Concerning performance, the frame-rate highly depends on the number of burning or flammable objects in the scene. Our setting of the scenes can be played without major frame-rate drops, but adding more flammable objects may have negative effects on the performance.

In general, simulations have to deal with a trade-off between performance and realness of real-time effects. For further studies it is important to see what degree of immersion in fire training is necessary to achieve a high learning value with less performance expensive methods. An evaluation of the learning output of our application was not part of this development process, but might provide important information for further work.

6 Future Work

The success of a learning software always depends on the learning attitude of the user. That leads to the fact that building such an application should include user engagement and expertise which is missing by the developer. Therefore, future user studies with the right target group will help to evaluate the actual learning output of the fire training. For further enhancement of the product, fire fighting experts can be involved to evaluate and correct, certain aspects of the virtual training.

The application serves as a base for a practical fire training. Theoretical education is not included with this version and it is assumed that trainees already had a short introduction about fire handling beforehand. For further work, theoretical knowledge like learn videos can be added to the simulation workflow. For more training value the user might want to interact with more virtual objects like the door, if the fire is too big to handle, or a towel for fire deletion if the fire extinguisher should fail.

Working with Unity was efficient for this context, but might face difficulties with further development. Due to the fact that virtual reality hardware and software is more evolving than ever, Unity will probably increase their support for developing virtual reality products during the next years.

7 Acknowledgement

During development, I was able to use the HTC Vive and other equipment from VRVis which helped me develop the application. Especially I want to thank Anton Fuhrmann, who was always willing to answer my questions, give valuable suggestions or possible improvements anytime.

References

[1] Hal Anderson. Heat transfer and fire spread. *U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station*, page 20, 1969.

[2] Moohyun Chaa, Soonhung Han, Jaikyung Leea, and Byungil Choi. A virtual reality based fire training simulator integrated with fire dynamics data. *Fire Safety Journal*, 50:12–24, 2012.

[3] Daniel Cohen, Nick Sevdalis, David Taylor, Karen Kerr, Mick Heys, Keith Willett, Nicola Batrick, and Ara Darzi. Emergency preparedness in the 21st century: training and preparation modules in virtual environments. *Resuscitation*, 84:78–84, 2013.

[4] Carolina Cruz-Neira, Daniel Sandin, Thomas DeFanti, Robert Kenyon, and John Hart. The CAVE: audio visual experience automatic virtual environment. *Commun. ACM*, 35(6):64–72, 1992.

[5] Aaron DeChamplain, Eric Rosendale, Ian McCabe, Matthew Stephan, Veronica Cole, and Bill Kapralos. Blaze: A serious game for improving household fire safety awareness. pages 1–4, 2012.

[6] Chris Dede. Immersive interfaces for engagement and learning. *Science*, (323):66–69, 2012.

[7] Matthias Maschek. Real time simulation of fire extinguishing scenarios. Master's thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2010.

[8] Michael Williams-Bell, Bernadette Ann Murphy, Bill Kapralos, Andrew Hogue, and Elizabeth Weckman. Using serious games and virtual simulation for training in the fire service: A review. *Fire Technology*, (51):553–584, 2007.