Virtual Reality Simulation

Team Members:

Manish Singh

Pravinkumar Singh

Abstract

An augmented experience framework was created to recreate crisis departures amid flames. The spreading of the fire and smoke in the virtual flame was displayed dependent on numerical flame reenactments, so the conditions are like reality. A multi-network, multi-base-state database demonstrate was utilized to defeat the disservices of customary smoke spreading recreations. Finished pictures and molecular frameworks give representation of fire and smoke. The framework submerges the client in a virtual domain with nitty gritty collaborations between the client and the virtual condition. The framework can demonstrate which clearing strategies are compelling for structure security assessments.

The main aim of the current system is to create an identical enviornment similar to real one, this includes identical dimensions, area and geographical conditions of the real enviornment. The geographical conditions can work along particle system with integrated real time physics engine to provide accurate results of simulation. The system does not focus on the hardware end of the virtual reality rather than the software end of virtual reality simulation. The project is built on unreal engine 4 and models/shaders are processed in blender.

Contents

List of Figures

Chapter 1

Introduction

This chapter deals with brief introduction on use cases of virtual reality simulation, the chapter further expands into comparing traditional system to modern simulation system. The chapter introduces new technology used during research and implemented in project, why this project is required and what is the future scope of this project. The chapter also deals with motivation behind choosing this project and what objectives it deals with.

1.1 Introduction to Virtual Reality.

Virtual reality simulation is the use of 3D objects and environments to create immersive and engaging learning experiences, the project focuses on creating an identical environment of a local subway station to deal with realistic problems in case of natural calamities or an accident. The use case used here is an example of fire accident, instead of low poly animated graphics, the motive of this project is to introduce photo realistic rendered environment, this is just an example of use case that can be built into real-time high-end simulation.

The principle of virtual reality e-learning is to impart, practice and check a user's knowledge using interactive scenarios and environments to reflect real-life situations. The project experiments with additional features like geographical positioning, optimized physically based rendering which reduces load on CPU and improves features like shadow and light mapping. The global positing helps to create identical climate structures and wind conditionings

1.2 Motivation

The traditional user training for any employee or citizen in case of an accident or natural calamities is based on theory, people are taught the traditional ways to act in the event of such incident, but this does not work when real incident hits the ceiling. Panic takes over the mind and traditional theory training doesn’t pan out to be helpful.

The idea of throwing a person in actual simulated environment with actual physics acting on him, trains his motor skills and forces him to act according to the situation, which results in minimum causalities. Plus, this environment and simulated structure can a lay a brief foundation for military training, education or even world simulation.

1.3 Problem statement

The lack of real time training during actual incident can lead to huge human causalities or even lead to fatal incident. This often led to uncoordinated movement, which can lead to panic among human employees which can be directed to an incident like stamped. Instruction manuals are useless during the action of the incident, because the panic takes over the instructions taught. This paper presents a VR framework for reproductions, penetrates, and preparing amid flame crisis departures. The framework furnishes clients with a virtual structure condition like this present reality.

1.4 Scope

The main aim of this project is to design identical architecture similar to the real-life architecture with photorealistic models and real time lightnings. The project focuses on

real life particle system with real time environment that has real time wind condition responsive with smoke and fire to give an edge to simulation[1]. Further down the road the project features can be compatible with other real time simulation like military drills, industrial training, scuba diving simulation etc. photogrammetry can be introduced to create real time semi polished models with the help of photos or videos.

1.5 Objectives

To design a virtual photo realistic environment identical to the real one. The primary focus for current project is subway system. The main aim is design photo realistic subway, implement NPCs and particle system. Further make the project compatible for both VR and non-VR user. Implement cost efficiency technologies like texture compressing, models optimization and low poly particles to make it efficient for low end rigs.

Chapter 2

Review of literature

Literature Review acts as the basis of research and study of the various concepts required for a particular domain. It describes the theories and other methodologies that can be adopted in order to implement modules of the proposed system. This chapter includes literature survey of technical papers along with the advantages and disadvantage of each system.

2.1 Car driving simulator

Driving simulation has become an important tool in the field of car industry. Researchers are providing an environment which is safer for drivers because there are no physical obstacles or potential for harm in the form of driving simulation using VR. The way that people look around while they're driving, and control of the vehicle (speed control, braking and lane changing) can manifest as driver behaviour errors. Recognition errors, decision making errors and performance errors, in particular, are 3 of the most common errors[2]. Driver error has been identified as an important factor in road traffic accidents.

For instance, if a driver accelerates instead of braking, it is considered an error. Also, it is considered an error if a driver performs an incorrect action intentionally or unintentionally such as passing a yellow light when they can stop, or braking hard on an icy or slippery road. Most errors occur when someone drives at a high speed in a situation that requires quick decision making within a small window of time. According to McLean, there are four types of collisions occurring at intersections: Right-Angel, Right-turn-Against, Rear-End and crashes involving pedestrians.

Advantage

- Controllability, reproducibility, and standardization.
- Ease of data collection.
- Possibility of encountering dangerous driving conditions without being at physical risk.

Disadvantages

- Limited physical, perceptual, and behavioural fidelity.
- Shortage of research demonstrating validity of simulation.
- Simulator sickness, especially in older people or under demanding driving conditions.

2.2 Flight Simulator

A flight simulator is a device that recreates an aircraft and its environment or any events where it flies. It is a world where someone who wants to become a pilot, or someone who just wants to learn how to drive a plane, or just to play without having to use the original plane. In this case, the flight simulator is very important for the learning of prospective pilots or airline pilots. To support this pilot prospective study, the flight simulator should be made as closely as possible to the real situation they would have when riding a plane. In this manner, this device needs to include equations of how an aircraft is flying, how flight controls react when it is triggered, effects of other aircraft systems, and reaction of aircraft to external factors such as damping, gravity, air density, turbulence, etc.

In a flight simulator, it is presented some of the best models that each plane has its own character and its specifications[3]. Generally, if a prospective new pilot learns to ride a plane, then the plane chosen is usually the best Cessna 182 because the plane is the most frequently used by someone who first learned and also has a relatively affordable price for most laymen that laypeople also sometimes have the aircraft. Some other models are often used in the world of flight simulator is the Air Force, Army, and Navy. A military air force would often practice for driving a fighter by first using a flight simulator beforehand so it can be more proficient and is used in an exercise using an actual plane.

Advantages

• Permit novice pilots to experience simulated flight without risk.
• Risk free introduction to emergencies and system failures.
• Cost effective and fuel savings.

Disadvantages

• Simulator operator cannot experience same stress level as would be expected in anticipation of real accident.
• Shorter flight cycles do not accurately reflect pilot fatigue or boredom.

2.3 Virtual classroom simulation

In the education system, virtual inclusion is a practice which allows a student to participate in classroom activities through a mobile robotic telepresence system. Using a bot, allows the student to interact with his/her teachers, peers and other school personnel as if the student is present in the classroom. By using such a system, a student sitting at one location, can still engage himself and participate in classroom activities and feel as if he is part of the class. For a homebound student, this is done using an internet connection. Once a student logs in, the individual can see, hear, talk and interact with others present in the classroom. If any student wishes to ask a query, the bot can give a subtle indication to the teacher who can give the student attention.

In this system, on one end we have a remotely controlled robot and at the other end we have the human operator[4]. Telepresence is a system by which the human operator forgets about his current location and feels as if he is in the robot's local environment. Such type of system can be very valuable in the educational field, especially to students with disabilities as it allows them to participate in classroom activities by remotely controlling a robot which is in the classroom attending lectures.

Advantage

- Provides outstanding visualizations that aren't possible in the traditional classroom.
- Increase student's engagement.
- Improves the quality of education in different field.

Disadvantage

- Deteriorates human connections.
- Functionality issues.
- Addiction to the virtual world.

Chapter 3

Requirement analysis

Virtual reality project is heavily reliable on graphical rendering processor, this makes requirement analysis a core chapter of this report. This chapter focuses on hardware requirement, software requirement, functional and non-functional requirement.

3.1 Functional requirement

Information gathering of profiles, contexts and services.

Information storage of profiles, session, files and settings or configuration.

Device, features and network discovery of authentication and authorization.

Task execution such as trigger action.

Reflect from standard review plan.

3.1.1 Normal requirement

Response time is basically the time taken to do something after it has been given enough input. Whatever is been given as an output must be measurable in the real system. Care must be taken to ensure that the performance measurement is unambiguous, concise and completely defined.

3.1.2 Expected requirement

A well written documentation on how to use and proceed in simulator.

Easy to install and use.

Simple GUI and good welcome screen.

Ease of human/machine interaction.

Flexible and Reliable.

3.1.3 Exciting requirement

Interactable environment allows user to play around.

Available for both VR and Non-VR user.

3D stereo sound gives an edge to realistic environment.

Optimized environment forces low computation power.

3.2 Non-functional requirement.

Project is portable, that is a person can enjoy the state of simulation even at home.

Since project is uploaded on steam market, user enjoys unlimited updates without interruption of any experience.

Project is available and accessible 24x7 and works even on offline mode.

Cost efficient, once project is implemented it can be used for years without any cost of maintenance.

3.3 Minimum hardware/software requirement

Consists of minimum hardware/software requirement required to run this project smoothly without any fps drop.

3.3.1 Hardware requirement

GPU: Nvidia GeForce GTX 970 or AMD Radeon R9 290.

CPU: Intel i5-4590 or AMD FX 8350 equivalent.

RAM: 4GB

Video Output: HDMI 1.4, Display Port 1.2 or newer.

USB Port: 1X USB 2.0 or better port.

3.3.2 Software requirement

Operating System: Windows 7 or Higher

Software: Blender, Unity, Unreal Engine, C# programming language, DirectX 9 or higher.

Chapter 5

Report on present investigation.

5.1 Proposed system

The current focus of the system is not on improving the mechanics of virtual reality, but rather the reality part. Current VR tech has enough to serve the mechanics require for movement accuracy, but suffers with realism provided by virtual reality system. Hitboxes are not accurate this can cause non-realistic perspective view during the simulation. Odd lightings and low moly models can cause severe issues during projection of models. The purpose of the system is to create realistic environment with modern technologies to get realistic perceptive to user, using the system. This system can lay foundation for future realistic simulation projects like military tech simulation, architectural visualization or even modern VR ecommerce store.

5.1.1 System architecture

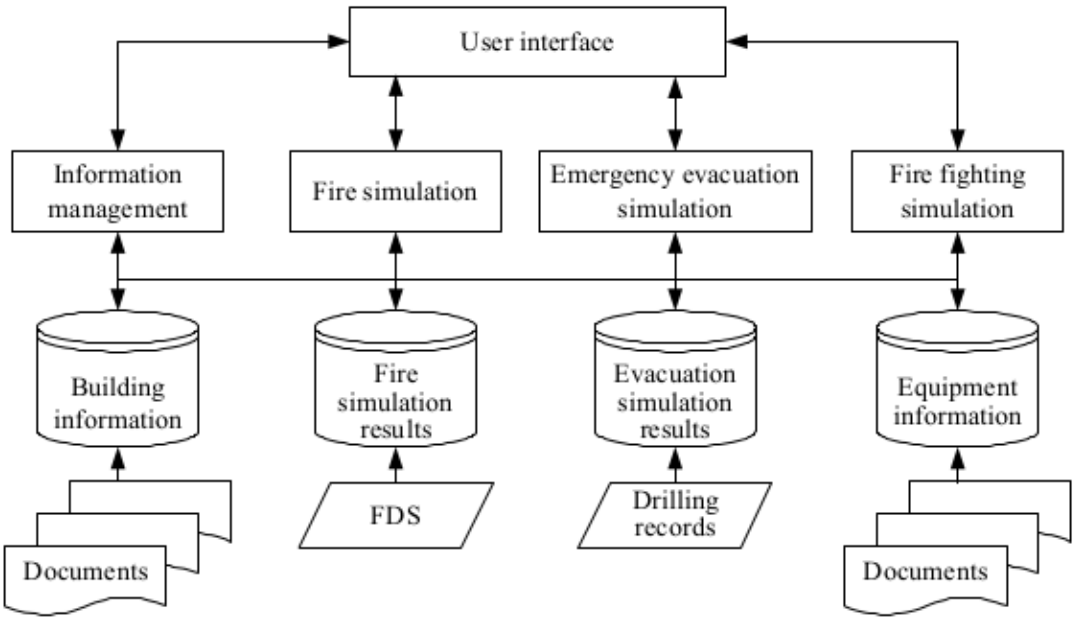

Fig. 2.1 System Architecture

Figure 2.1 depicts system architecture of the current system, all the components corresponds with UI that includes four system as stated in diagram, the building information contains valid information about building that includes area of the building as well as all the docs necessary further the very next component deals with simulation as well as evacuation results and last component deals with safety equipment, equipped in the building.

5.1.2 Unreal engine

One of the major features planned for UE4 was real-time global illumination using voxel cone tracing, eliminating pre-computed lighting [5]. However, this feature has been replaced with a similar but less computationally-expensive algorithm prior to release for all platforms including the PC due to performance concerns. UE4 also includes new developer features to reduce iteration time, and allows updating of C++ code while the engine is running. The new

"Blueprints" visual scripting system (a successor to UE3's "Kismet") allows for rapid development of game logic without using C++, and includes live debugging. The result is reduced iteration time, and less of a divide between technical artists, designers, and programmers.

5.1.3 Unreal script

Unreal Script (often abbreviated to UScript) was Unreal Engine's native scripting language used for authoring game code and gameplay events before the release of Unreal Engine 4. The language was designed for simple, high-level game programming. The Unreal Script interpreter was programmed by Sweeney, who also created an earlier game scripting language, ZZT-oop. Similar to Java, Unreal Script is object-oriented without multiple inheritance (classes all inherit from a common Object class), and classes are defined in individual files named for the class they define. Unlike Java, Unreal Script does not have object wrappers for primitive types. Interfaces are only supported in Unreal Engine generation 3 and a few Unreal Engine 2 games. Unreal Script supports operator overloading, but not method overloading, except for optional parameters.

5.2 Particle system

A particle system is a technique in game physics, motion graphics, and computer graphics that uses a large number of very small sprites, 3D models, or other graphic objects to simulate certain kinds of "fuzzy" phenomena, which are otherwise very hard to reproduce with conventional rendering techniques - usually highly chaotic systems, natural phenomena, or processes caused by chemical reactions.

Introduced in the 1982 film Star Trek II: The Wrath of Khan for the fictional "Genesis effect", other examples include replicating the phenomena of fire, explosions, smoke, moving water (such as a waterfall), sparks, falling leaves, rock falls, clouds, fog, snow, dust, meteor tails, stars and galaxies, or abstract visual effects like glowing trails, magic spells, etc. - these use particles that fade out quickly and are then re-emitted from the effect's source. Another technique can be used for things that contain many strands - such as fur, hair, and grass - involving rendering an entire particle's lifetime at once, which can then be drawn and

manipulated as a single strand of the material in question. Particle systems may be two-dimensional or three-dimensional.

5.2.1 Simulation stage

During the simulation stage, the number of new particles that must be created is calculated based on spawning rates and the interval between updates, and each of them is spawned in a specific position in 3D space based on the emitter's position and the spawning area specified. Each of the particle's parameters (i.e. velocity, color, etc.) is initialized according to the emitter's parameters. At each update, all existing particles are checked to see if they have exceeded their lifetime, in which case they are removed from the simulation.

5.2.2 Rendering stage

After the update is complete, each particle is rendered, usually in the form of a textured billboarded quad (i.e. a quadrilateral that is always facing the viewer). However, this is sometimes not necessary for games; a particle may be rendered as a single pixel in small resolution/limited processing power environments. Conversely, in motion graphics particles tend to be full but small-scale and easy-to-render 3d models, to ensure fidelity even at high resolution. Particles can be rendered as Metaballs in off-line rendering; isosurfaces computed from particle-metaballs make quite convincing liquids. Finally, 3D mesh objects can "stand in" for the particles — a snowstorm might consist of a single 3D snowflake mesh being duplicated and rotated to match the positions of thousands or millions of particles.

5.2.3 Photogrammetry

Photogrammetry is the art and science of making measurements from photographs, especially for recovering the exact positions of surface points[6]. Photogrammetry is as old as modern photography, dating to the mid-19th century and in the simplest example, the distance between two points that lie on a plane parallel to the photographic image plane, can be determined by measuring their distance on the image, if the scale (s) of the image is known.

5.3 Ragdoll physics

Early video games used manually created animations for characters' death sequences. This had the advantage of low CPU utilization, as the data needed to animate a "dying" character was chosen from a set number of pre-drawn frames. As computers increased in power, it became possible to do limited real-time physical simulations. A ragdoll is, therefore, a collection of multiple rigid bodies (each of which is ordinarily tied to a bone in the graphics engine's skeletal animation system) tied together by a system of constraints that restrict how the bones may move relative to each other. When the character dies, their body begins to collapse to the ground, honouring these restrictions on each of the joints' motion, which often looks more realistic.

The term ragdoll comes from the problem that the articulated systems, due to the limits of the solvers used, tend to have little or zero joint/skeletal muscle stiffness, leading to a character collapsing much like a toy rag doll, often into comically improbable or compromising positions. A physics engine is computer software that provides an approximate simulation of certain physical systems, such as rigid body dynamics (including collision detection), soft body dynamics, and fluid dynamics, of use in the domains of computer graphics, video games and film. Their main uses are in video games (typically as middleware), in which case the simulations are in real-time. The term is sometimes used more generally to describe any software system for simulating physical phenomena, such as high-performance scientific simulation. Ragdoll physics is a type of physics engine procedural animation which is often used as a replacement for traditional static death animations in video games and animated films

5.4 Engines

One of the first general purpose computers, ENIAC, was used as a very simple type of physics engine. It was used to design ballistics tables to help the United States military estimate where artillery shells of various mass would land when fired at varying angles and gunpowder charges, also accounting for drift caused by wind. The results were calculated a single time only, and were tabulated into printed tables handed out to the artillery commanders. Physics engines have been commonly used on supercomputers since the 1980s to perform computational fluid dynamics modeling, where particles are assigned force vectors that are combined to show circulation. Due to the requirements of speed and high precision, special computer processors known as vector processors were developed to accelerate the calculations. The techniques can be used to model weather patterns in weather

forecasting, wind tunnel data for designing air- and watercraft or motor vehicles including racecars, and thermal cooling of computer processors for improving heat sinks. As with many calculation-laden processes in computing, the accuracy of the simulation is related to the resolution of the simulation and the precision of the calculations; small fluctuations not modeled in the simulation can drastically change the predicted results.

In most computer games, speed of the processors and gameplay are more important than accuracy of simulation. This leads to designs for physics engines that produce results in real-time but that replicate real world physics only for simple cases and typically with some approximation. More often than not, the simulation is geared towards providing a "perceptually correct" approximation rather than a real simulation. However, some game engines, such as Source, use physics in puzzles or in combat situations. This requires more accurate physics so that, for example, the momentum of an object can knock over an obstacle or lift a sinking object. Physically-based character animation in the past only used rigid body dynamics because they are faster and easier to calculate, but modern games and movies are starting to use soft body physics[7]. Soft body physics are also used for particle effects, liquids and cloth. Some form of limited fluid dynamics simulation is sometimes provided to simulate water and other liquids as well as the flow of fire and explosions through the air.

Chapter 6

Result and discussion

6.1 Result

Fig. 3.1 Reflections

Figure 3.1 depicts pbr characteristics in the environment with advance lightning effects in the simulated environment.

Fig. 3.2 Subway before fire simulation.

Figure 3.2 depicts subway system before fire was simulated in environment without any particles or effects.

Fig. 3.2.3 Fire in subway

Fig.3.2.3 depicts subway system after fire was simulated in the simulated environment.

Chapter 7

Conclusion and Literature Cited

7.1 Conclusion:

An augmented experience framework created to assess crisis clearing strategies was presented. The framework can be utilized to direct modest and safe crisis fire departure penetrates in a virtual situation. The multi-framework and multi-base-state show beats the burdens of conventional models for smoke spread recreations. Surface mapping and molecule frameworks are utilized to picture the fire and smoke. The virtual structure condition is made utilizing Multigen Creator and Vega. Collaborations between the clients what's more, the virtual condition depend on the Vega API what's more, C++ programs. The framework is incredible and simple to use. Future research will build up a circulated intelligent reproduction that will enable a few clients to at the same time take part in a clearing.

7.2 Future Work:

The current system allows user to enjoy simulation in small environment with points of health and square hitboxes that are used to calculate damage, the current system can be used in different real time simulation or even practical purposes to recreate certain events. Hitboxes being instead of square can be turned into bubble.

References

[1] Gwynne S, Galea E R, Owen M, Lawrence P J, Filippidis L. A review of the methodologies used in evacuation modeling. Fire and Materials, 1999, 23: 383-388.

[2] Yang Lizhong, Fang Weifeng, Fan Weicheng. Modeling occupant evacuation using cellular automata ʊ Effect of human behavior and building characteristics on evacuation. Journal of Fire Sciences, 2003, 21(3): 227-240.

[3] Lo S M, Fang Z, Lin P, Zhi G S. An evacuation model: The SGEM package. Fire Safety Journal, 2004, 39(3): 169-190.

[4] Shen Tzu-Sheng. ESM: A building evacuation simulation model. Building and Environment, 2005, 40: 671-680.

[5] Li Huibo, Tang Wen, Simpson D. Behaviour based motion simulation for fire evacuation procedures. In: Proceedings of the Theory and Practice of Computer Graphics 2004. Bournemouth, United Kingdom, 2004: 112-118.

[6] Gwynne S, Galea E R, Lawrence P J, Filippidis L. Modelling occupant interaction with fire conditions using the building EXODUS evacuation model. Fire Safety Journal, 2001, 36(4): 327-357.

[7] Thompson P A, Marchant E W. Computer model for the evaluation of large building populations. Fire Safety Journal, 1995, 24(2): 131-148.

[8] Shih N, Lin C, Yang C. Virtual-reality-based feasibility study of evacuation time compared to the traditional calculation method. Fire Safety Journal, 2000, 34(4): 377-391.

Technologies Used

1. Unreal Engine 4:

The Unreal Engine is a game engine developed by Epic Games, first showcased in the 1998 first-person shooter game Unreal. Although initially developed for first-person shooters, it has been successfully used in a variety of other genres, including stealth, fighting games, MMORPGs, and other RPGs. With its code written in C++, the Unreal Engine features a high degree of portability and is a tool used by many game developers today, with it being source-available. The most recent version is Unreal Engine 4, which was released in 2014.

2. Blender:

Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games. Blender's features include 3D modeling, UV unwrapping, texturing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, rendering, motion graphics, video editing and compositing.

3. Unity3D:

Unity is a cross-platform real-time engine developed by Unity Technologies, first announced and released in June 2005 at Apple Inc.'s Worldwide Developers Conference as an OS X-exclusive game engine. As of 2018, the engine has been extended to support 27 platforms.[8] The engine can be used to create both three-dimensional and two-dimensional games as well as simulations for its many platforms. Several major versions of Unity have been released since its launch, with the latest stable version being Unity 2019.1.0**.**

Publication

The research paper titled “Virtual reality simulation” is yet to be published in International Journal of Scientific Research in Computer Science Applications and management Studies (IJSRCSAMS).