International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 02 | Feb-2018 www.irjet.net p-ISSN: 2395-0072

Lost: The Horror Game

Omkar Kute[1], Amit Pathak[2], Shoaib Sayyed[3], Prathmesh Patil[4]

[1,2,3,4] *Student, Department of Information Technology, P.C. Polytechnic, Pune, India*

Abstract: *Gaming is a phenomenon that has become a fundamental element in today's digital youth culture. So far very little effort has been made to study games in particular with respect to its potentials to positively influence research developments in other areas. In the fast-growing field of software engineering and development and even more rapidly growing a sector of game development the future is hard to predict. We are working on this game as our software project lab-II.SPL-II is a 3 credit course and as part of our degree, we choose this type of work for doing better with development cycle, development period, graphics, scripting, adopting new technology, animation. In general software project is a project focusing on the creation of software. Consequently, Success can be measured by taking a look at the resulting software. In a game project, the product is a game. But and here comes the point: A game is much more than just its software. It has to provide content to become enjoyable. Just like a web server: without content the server is useless, and the quality cannot be measured. This has an important effect on the game project as a whole. The software part of the project is not the only one, and it must be considered in connection to all other parts: The environment of the game, the story, characters, game plays, the artwork, and so on. A website called Raywenderlich.com shows basic tutorials of using Unreal Engine 4 with the help of images, graphics, and videos. A survival horror storyline usually involves the investigation or confrontation of horrific forces, and thus many games transform common elements from horror fiction into gameplay challenges. LOST typically features a variety of monsters with unique behavior patterns. Enemies can appear unexpectedly or suddenly, and the level is designed with script sequences where enemies can crash through any part of the game.*

1. INTRODUCTION

Survival horror is a subgenre of video games inspired by horror fiction that focuses on survival of the character as the game tries to frighten players with either horror graphics or scary ambiance. Entertainment by far has been the most acknowledged way to spend time; gaming is one of them where an average person can almost spend a whole day playing a game. A person's choice of the game may differ based on his likings, where one likes to play a thriller or Action other may like racing or a horror game. The "Lost" is one of the games where a person who likes horror and thrill game will enjoy playing this at the same, in which the player cycles through various areas ranging from dense forest to an undead area, in which the protagonist must survive in all the areas to proceed further in the story the player is equipped with a torch which helps to manure through dark areas plus spooky audio effect which enhances the gaming experience. This game also contains cutscenes to give a dramatic effect as well as the storyline stiff. Overall this game contains good graphics, to begin with, where you won't get bored. In spite of negative effects of video games, certain studies indicate that they may have value in terms of academic performance, perhaps because of the skills that are developed in the process. LOST will help to raise self-esteem and build confidence. It gives people an opportunity to do and experience horror and thrill that they cannot do in real world, and to discover new things about themselves.

2. GAME ENGINE

The game engine is a software structure, designed for creation and development of the game for consoles, mobiles devices, and personal computers. The core functionality typically provided by a game engine is it includes a rendering engine for 2d or 3d graphics, a physics engine, sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and may include video support for cinematic. The process of game development is often economized in large part, by use again/familiarizing the same game engine to create different games, or to make it easier to port games to multiple platforms.

3. Unreal Engine

Unreal Engine 4 is a complete collection of development gears made for anyone working with real-time technology. From enterprise applications and cinematic experiences to high-quality games across PC, console, mobile, VR and AR, Unreal Engine 4 gives you everything you need to start, ship, grow and stand out from the crowd. A top-notch toolset and accessible workflows allow developers to quickly restate on ideas and see immediate results without stirring a line of code, while full source code access gives everyone in the Unreal Engine 4 community the liberty to transform and extend engine features.

4. BLUEPRINTS VISUAL SCRIPTING

The Graphical Codding system in game Engine is a complete gameplay codding system acknowledged on the concept of using a node based mapping interface to create gameplay elements from within Unreal Editor. As with many common scripting languages, it is used to define object-oriented classes or objects in the engine. As you use a game engine, you'll frequently find that objects defined using blueprints. This system is enormously malleable and powerful as it provides the capability for designers to use virtually the full

range of concepts and gears usually only available to programmers. In addition, Blueprint specific markup language available in Unreal Engine. C++ execution enables programmers to create standard systems that can be extended by designers.

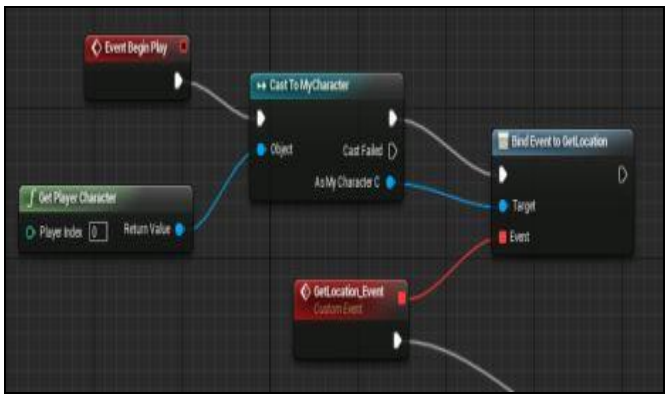

Figure 1: sample blueprint of unreal engine

5. ASSETS

Game resources contain everything that can be included into a game, including 3-dimensional models, sprites, audio effects, music and components, Blueprints, and even whole projects that can be used by a game engine. Here's a list of:

2D/3D Design:

- Characters
- Environments
- Vehicles

Scripting:

- AI
- Special effects
- Networking
- Physics

Figure 2: example of assets

6. Textures

Textures are somewhat like images that are used on Materials. If you're at all familiar with the movie and video game 3D pipelines, you know they're similar. They both utilize modeling, texturing, rigging, animation, etc. However, they're actually quite different in the number of limitations between the two, texturing in particular. Let's take a closer look at texturing, how it is utilized within the video game pipeline, and what are some very important things to consider when texturing for game assets. So how does texturing vary from movies to video games? Well, you need to keep in mind that video games have a very strict polygon budget to be considered throughout the entire modeling process. Composite Textures

- Support Textures
- Material Textures

A single material may make use of numerous textures that are all experimented and applied for different purposes. For example, a simple material may have a base texture, a specular texture, and a normal map. In addition, there may be a map for the Emissive and Irregularity stored in the alpha channels of one or more of these textures.

Figure 3: example of textures

7. CODING

Unreal Engine 4 provides two toolsets for programmers which can also be used to speed up development workflows. New gameplay classes, Slate and Canvas user interface elements, and editor functionality can be written with C++, and all changes will be reflected in Unreal Editor after compiling with either Visual Studio. The Blueprints visual scripting system is a strong tool which allows classes to be created in-editor through wiring together function blocks and property references. C++ classes can be used as a base for Blueprint classes, and in this way, programmers can set up fundamental gameplay classes that are then sub-classed and iterated on by level designers.

```
Actor.cpp
AActor.PostLoad                    void AActor::PostLoad()

void AActor::BeginDestroy()
{
    UnregisterAllComponents();
    Super::BeginDestroy();
}

bool AActor::IsReadyForFinishDestroy()
{
    return Super::IsReadyForFinishDestroy() && DetachFence.IsFenceComplete();
}

void AActor::PostLoad()
{
    Super::PostLoad();

    // add ourselves to our Owner's Children array
    if (Owner != NULL)
    {
        checkSlow(!Owner->Children.Contains(this));
```

Figure 4: example of code

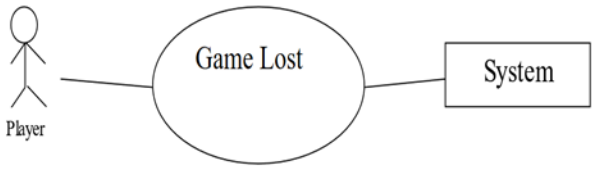

Fig 1: Level 0 for Game

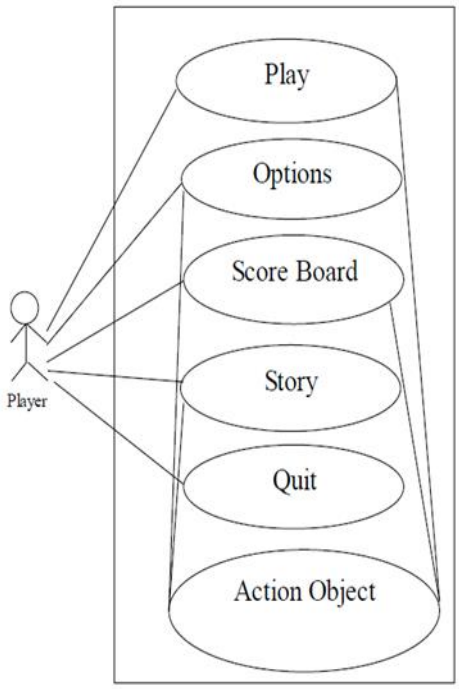

Fig 2: Level 1 for Game UX

8. Architectures

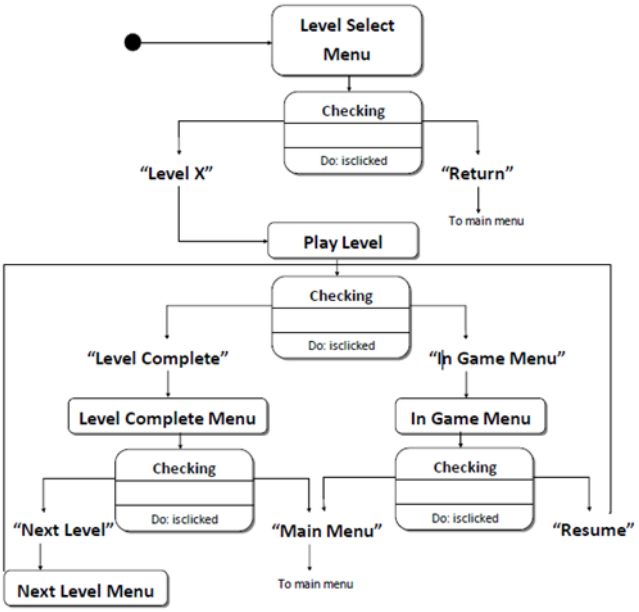

Fig 29: play level state diagram

9. System Requirements

Hardware Requirements	
Processor	Intel i5 760 2.8Ghz
	AMD Athlon II X4 620
GPU	2Gb Dedicated
Ram	8Gb
HDD Space	10Gb

Software Requirements	
OS	Windows-10 64-bits
DirectX	Fully DirectX 11.0 Compatible

10. Conclusion

Game development tools are finding their way into younger and younger hands. As a result, the next frontier of educational gaming will be created by the very generation who were nurtured by games throughout childhood. "LOST" TYPICALLY FEATURES A VARIETY OF MONSTERS WITH UNIQUE BEHAVIOUR PATTERNS.

ACKNOWLEDGMENT

It is being rightly said that we are built on the shoulder of others. For everything I achieved, the credit goes to all those who had really helped me to complete this work successfully. I am extremely thankful for my Project Guide Miss Nisha Karolia for guidance and review of this paperwork

REFERENCES

1. "Alexa Ranking". Alexa Internet. Archived from the original on December 29, 2017. Retrieved November 7, 2017.

2. Steiner, Brian (June 24, 2013). "How the Unreal Engine Became a Real Gaming Powerhouse". Popular Mechanics. Hearst Corporation. Archived from the original on May 30, 2015. Retrieved October 6, 2014.

3.: ab "Most successful video game engine". Guinness World Records. Archived from the original on December 22, 2015.

4."First video game created with the Unreal Engine". Guinness World Records. Archived from the original on September 24, 2015. Retrieved November 22, 2015.

5. Sweeney, Tim (2005). "GPU Gems 2 – Foreword". NVidias Developer. Retrieved October 3, 2017.

6. Nutt, Christian (March 21, 2014). "Epic's Tim Sweeney lays out the case for Unreal Engine 4". Gama sutra. Retrieved October 3, 2017.