A MAJOR PROJECT REPORT ON

"Erica-The Warrior"

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

To

Guru Gobind Singh Indraprastha University, Delhi

Under the Guidance of:

Dr. Barkha Bahl

Director

Submitted By:

Ankur Chaurasia (35120602019)

BCA 6th Semester

TRINITY INSTITUTE OF PROFESSIONAL STUDIES

(Affiliated to Guru Gobind Singh Indraprastha University, Delhi)

Ranked "A+" Institution by SFRC, Govt. of NCT of India

Recognized under section 2(f) of the UGC Act, 1956

NAAC Accredited "B++" Grade Institution

CONTENTS

S. NO.	Topic	Page No.
1	Certificate (s)	3
2	Acknowledgements	4
3	List of Abbreviations	5
4	Chapter-1: Introduction	6
5	Chapter-2: Motivation	10
6	Chapter-3: Systems Requirement Analysis	11
7	Chapter-4: Methodology	12
8	Chapter-5: Systems Development (Coding/Blueprints)	24
9	Chapter-6: Systems Implementation (Screen shots)	81
10	Chapter-7: Systems Testing	194
11	Chapter-8: Future Scope	195
12	Conclusion	196
13	References	197

TRINITY INSTITUTE OF PROFESSIONAL STUDIES

(Affiliated to Guru Gobind Singh Indraprastha University, Delhi)

Ranked "A+" Institution by SFRC, Govt. of NCT of India

Recognized under section 2(f) of the UGC Act, 1956

NAAC Accredited "B++" Grade Institution

To Whom It May Concern

I **Ankur Chaurasia**, Enrolment No. **35120602019** from BCA-VI Sem of the Trinity Institute of Professional Studies, Delhi hereby declare that the Major Project Report entitled

"Erica The Warrior" is an original work and the same has not been submitted to any other Institute for the award of any other degree.

Date: Signature of the Student

Certified that the Project Report submitted in partial fulfilment of Bachelor of Computer Applications (BCA) to be awarded by G.G.S.I.P. University, Delhi by **Ankur Chaurasia**, Enrolment No. **35120602019** has been completed under my guidance and is Satisfactory.

Date: Signature of the Guide

Name of the Guide: Dr. Barkha Bahl

Designation: Director

ACKNOWLEDGEMENT

We would like to express special thanks of gratitude to our Guide Dr. Barkha Bahl and the Institute for giving us the golden opportunity to do this amazing project on the development of a piece of technology for the **Erica The Warrior**; which also helped us in doing a lot of research and we came to know about so many new things and gain a knowledgeable experience.

Secondly, we would also like to thank our parents who helped us and supported us in the time of helplessness, stage of giving up. We would also like to thank and show our gratitude to a friend who not only supported us but also helped by letting his system to work on. We are heartiest thankful to our friends who helped us in finishing this project within the limited time.

The purpose of making this project was not merely for marks, but to gain new experience, learn new things, increase our knowledge and to know more about open world RPG to explore gaming world.

List of Abbreviations

S. NO.	Actual Word	Abbreviated Word
1	Unreal Engine	UE
2	Role Playing Game	RPG
3	Blend Space	BS
4	Blueprints	BP

Ankur Chaurasia
(35120602019)

Chapter-1

INTRODUCTION

Nowadays, it is a truth that video games are the world's most beloved merriment industry. Even though video games are frequently played by kids and youth, there are many older people who also play those games. Moreover, video games are progressively becoming even more of a part of our culture. Today's kids and human beings are wasting their time on technological devices by playing video games. This examination is planned as an entrance to present thought as regards turn-based RPG strategy video games' role. Role-Playing Games (RPGs) form one of the primary genres of games and are possibly one of the most widely varied game forms around. Furthermore, RPGs have been designed for or ported to every existing hardware platform produced for games, in addition to tabletop and live-action variations. RPGs formed one of the primary sources of inspiration for the early evolution of computer games and retain a strong influence to this date.

The game engine ensures game originators withal the obligatory characteristics to construct games rapidly, influentially, and yieldingly. Game engines are too momentous for developing video games and muster numerous primaries structs. It is kind of a spine for game development.

For a game developer, this part is the most crucial decision. Game engine choice is able to repulse the standard of the game advanced or back in whichever facet. The most important familiar game engines own their own ways and characteristics. There are any inquiries and responses your necessity to do your selection any of them the size of your squad, your warrant on the game engine, the structure of your project, your license the estimates, your choice of many details, such as 2D, 3D, mobile, console, and PC. For our game we have chosen Unreal Engine.

Unreal Engine 5

Unreal Engine 5 (UE5) is now available as an Early Access build. We are excited to share this new version of Unreal with the world, giving everybody a chance to get hands-on with the future of Unreal Engine.

WORLD BUILDING AND COLLABORATION

One of our ongoing goals is to make the creation of dynamic, open worlds faster, easier, and more collaborative for teams of all sizes. With Unreal Engine 5, a new World Partition system changes how levels are managed and streamed, automatically dividing the world into a grid and streaming the necessary cells. Additionally, teams can simultaneously work in the same World without treading on each other's toes, thanks to the new One File Per Actor system, which breaks each level into a series of files. This means that when you make changes to the level, you are only saving and submitting changes to the individual Actors you've changed or added. There have also been significant improvements to how Unreal works with Source Control, specifically with Perforce, allowing developers to do almost all of their SCM operations directly in the editor.

NEW USER INTERFACE

In Unreal Engine 5, the Unreal Editor gets a makeover, with an updated visual style, streamlined workflows, and optimized use of screen real estate, making it easier, faster, and more pleasing to use. To free up more space for viewport interactions, we've added the ability to easily summon and stow the Content Browser and to dock any editor tab to the collapsible sidebar. You can now quickly access frequently used properties in the Details panel with a new favouriting system, while the new Create button on the main toolbar lets you easily place Actors into your world. There's also a streamlined, easier workflow for creating new projects.

CHAOS PHYSICS

Chaos Physics is a lightweight physics simulation solution available in Unreal Engine 5, built from the ground up to meet the needs of next-generation games. The system includes the following major features:

- Rigid Body Dynamics
- Rigid Body Animation Nodes and Cloth Physics
- Destruction
- Ragdoll Physics
- Vehicles
- Physics Fields
- Fluid Simulation
- Hair Simulation

MIXAMO

Mixamo is a 3D computer graphics technology company. Based in San Francisco, the company develops and sells web-based services for 3D character animation. Mixamo's technologies use machine learning methods to automate the steps of the character animation process, including 3D modeling to rigging and 3D animation.

Edit the animation

1. Use the editor panel to control the animation settings such as speed and arm spacing after you've chosen an animation and applied it to your character.
2. You can remove an animation by clicking the **X** in the top of the animation editing section to return to your static character.
3. Select the **Download** button when you're finished editing the animation.

Ankur Chaurasia
(35120602019)

Chapter-2

Motivation

Our aim for this project is to make a turn-based RPG game. We intend to make it run on PCs and laptops. We used the Unreal Engine 5 to develop our game. We wanted to develop a role-playing video game within the electronic game genre in which players advance through a story quest and often many side quests, for which their character or party of characters gain experience that improves various attributes and abilities.

This is not something we would develop for the sake of a minor project, but something that is truly of our interest, will help us better understand gaming technology, and is also very different and unique.We did a little research about which engine should be the best platform for developing our game, and the Unreal Engine came out as the best of all.

We did face a lot of issues and practical problems, but we overcame them all with the guidance of our teachers and research. The most important thing about our project is that the future scope is endless and every new idea is welcomed.

Chapter-3

TOOLS: SOFTWARE & HARDWARE REQUIREMENTS

Hardware Requirements: -

- CPU: 2.9GHz Intel Core i7-7500U (dual-core, 4MB cache, up to 3.5GHz)
- RAM: 8 GB DDR4
- Graphics: Intel HD Graphics 620
- Storage: 1 TB Hard Drive
- Keyboard & Mouse
- Active Internet Connection

Software Requirements: -

- Unreal Engine 5
- Mixamo

Chapter-4

METHODOLOGY

Unreal Engine enables game developers and creators across industries to realize next-generation real-time 3D content and experiences with greater freedom, fidelity, and flexibility than ever before.

Unreal Engine 5 gets a UI makeover!

With Quixel Bridge now fully integrated, you have drag-and-drop access to the entire Megascans library. It’s part of the new Create menu for acquiring content and creating and placing Actors.

To free up space in the viewport, you can now easily summon and stow the Content Browser and dock any editor tab to a sidebar. Plus, it’s faster to find the properties you’re looking for in the Details panel.

The **Blueprint Visual Scripting** system in Unreal Engine is a complete gameplay scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor. As with many common scripting languages, it is used to define object-oriented (OO) classes or objects in the engine. As you use UE4, you'll often find that objects defined using Blueprint are colloquially referred to as just "Blueprints."

This system is extremely flexible and powerful as it provides the ability for designers to use virtually the full range of concepts and tools generally only available to programmers. In addition, Blueprint-specific markup available in Unreal Engine's C++ implementation enables programmers to create baseline systems that can be extended by designers.

How Do Blueprints Work?

In their basic form, Blueprints are visually scripted additions to your game. By connecting Nodes, Events, Functions, and Variables with Wires, it is possible to create complex gameplay elements.

Blueprints work by using graphs of Nodes for various purposes - object construction, individual functions, and general gameplay events - that are specific to each instance of the Blueprint in order to implement behavior and other functionality.

Commonly Used Blueprint Types

The most common Blueprint types you will be working with are Level Blueprints and Blueprint Classes.

Level Blueprint

The Level Blueprint fills the same role that Kismet did in Unreal Engine 3, and has the same capabilities. Each level has its own Level Blueprint, and this can reference and manipulate Actors within the level, control cinematics using Matinee Actors, and manage things like level streaming, checkpoints, and other level-related systems. The Level Blueprint can also interact with Blueprint Classes (see the next section for examples of these) placed in the level, such as reading/setting any variables or triggering custom events they might contain.

Blueprint Class

Ankur Chaurasia
(35120602019)

Blueprint Classes are ideal for making interactive assets such as doors, switches, collectible items, and destructible scenery. In the image above, the button and the set of doors are each separate Blueprints that contain the necessary script to respond to player overlap events, make them animate, play sound effects, and change their materials (the button lights up when pressed, for example).

In this case, pressing the button activates an event inside the door Blueprint, causing it to open - but the doors could just as easily be activated by another type of Blueprint, or by a Level Blueprint sequence. Because of the self-contained nature of Blueprints, they can be constructed in such a way that you can drop them into a level and they will simply work, with minimal setup required. This also means that editing a Blueprint that is in use throughout a project will update every instance of it.

What Else Can Blueprints Do?

You have read about Level Blueprints and Blueprint Classes, listed below are a handful of examples that can be accomplished with the Blueprint system.

Create Customizable Prefabs with Construction Scripts

The Construction Script is a type of graph within Blueprint Classes that executes when that Actor is placed or updated in the editor, but not during gameplay. It is useful for creating easily customizable props that allow environment artists to work faster, such as a light fixture that automatically updates its material to match the color and brightness of its point light component, or a Blueprint that randomly scatters foliage meshes over an area.

In the Content Examples maps, the long rooms that contain each example (pictured above) are actually a single Blueprint made up of many components. The Blueprint's Construction Script creates and arranges the various Static Meshes and lights according to parameters exposed in the Blueprint's **Details** panel. With each Content Example map we created, we were able to drop in the demo room Blueprint, set values for the length, height, and number of rooms that would be generated (and a few other options), and have a complete set of rooms ready in moments.

A Blueprint like this can be time-consuming to create initially, but if you know you will use it often, the time saved when building a level and the ease of making changes can make it very worthwhile.

Create A Playable Game Character

Pawns are also a type of Blueprint Class, and it is possible to put together every element you need for a playable character in the Blueprint graph. You can manipulate camera

behavior, set up input events for mouse, controller, and touch screens, and create an Animation Blueprint asset for handling skeletal mesh animations.

When you create a new Character Blueprint, it comes with a character component that has much of the behavior needed for moving around, jumping, swimming, and falling built-in, and all that is required is to add some input events in accordance with how you want your character to be controlled.

Create A HUD

Blueprint script can be used to create a game's HUD as well, which is similar to Blueprint Classes in that it can contain event sequences and variables, but is assigned to your project's GameMode asset instead of being added directly to a level.

You can set up a HUD to read variables from other Blueprints and use them to display a health bar, update a score value, display objective markers, and so on. It is also possible to use the HUD to add hit-boxes for elements like buttons that can be clicked on or, in the case of mobile games, can respond to touch input.

Ankur Chaurasia
(35120602019)

Modules:

1. Engine download and project setup

In the first place, we need to setup the Unreal Engine for game initialization. To download and install Unreal Engine, you need to:

A. Download and install the **Epic Games Launcher**.
B. Sign up for an Epic Games account, if you don't have one.
C. Sign in to the Epic Games Launcher.
D. Install Unreal Engine.

When you create a new project, Unreal Engine gives you a list of templates you can choose from. These templates contain some ready-to-use Assets, such as level geometry, a character you can control, and simple character animations. Many tutorials use one of these templates as a starting point.

In a third-person game, the player sees the game world from a Camera that's located at a fixed distance behind and slightly above their character. In Unreal Engine, you can control camera distance and position, and adjust it as needed.

The Third Person template in Unreal Engine 5 contains the following elements:

- A playable third-person character that can move and jump.
- Additional meshes for the character.
- A Level with basic geometry (ramps, platforms).
- Physics-enabled cubes that react when the player collides with them.

The template also comes with redesigned Mannequins.

1.1 Creating a Third Person Project

Launching Unreal Engine 5 opens the **Project Browser** window, where you can choose to open an existing Unreal project or create something new. To create a third-person project, select the **Games** category on the left, and then select the **Third Person** template.

1.2 Template Contents

The Third Person template contains all of the basic elements of a simple third-person experience. You can use it as a starting point for a traditional role-playing game (RPG), third-person shooter, or any other kind of application. The following section details the main template elements and where to find them in the **Content Browser**.

1.3 Third-Person Character

Assets for the player character are located in the Content/Characters folder. By default, the Third Person template starts with the feminine Unreal Engine 5 Mannequin. This folder contains additional Skeletal Meshes for the player character, both in the style of Unreal Engine 5 and the legacy Unreal Engine Mannequin. Unreal Engine 5 Mannequins also come with configurable **Level of Detail** (LOD) settings. LODs help optimize your application for different platforms.

1.4 Animations

The Animation Blueprints make full use of the new IK Rig system in Unreal Engine 5. Unlike legacy animations, IK Rig can be used to dynamically modify pose-based solver parameters. An example of this can be seen in the screenshots below: the position of Quinn's feet adjusts dynamically to match the kind of terrain she is standing on.

Also in the ***Rigs*** folder, the **Mannequin Body** Control Rig Asset can be used to easily pose and animate the Mannequin directly in the editor.

1.5 Character Setup

You can change your character's appearance by changing its **Static Mesh**. As an example, let's change the default Mannequin mesh. To do this, follow these steps:

I. In the **Content Browser**, go to Content/ThirdPerson/Blueprints, then double-click the **BP_ThirdPersonCharacter** Blueprint to open it in the Blueprint Editor.

II. In the Blueprint Editor, in the **Components** panel, click the **Mesh (CharacterMesh)** Component to select it.

III. With the **Mesh (CharacterMesh)** Component selected, find the **Details** panel on the right side of the Blueprint Editor. Then, in the **Mesh** section, click the drop-down next to the **Skeletal Mesh** parameter and select SKM_Manny from the list.

IV. Still in the **Details** panel, in the **Animation** section, set the following options:
   - **Animation Mode**: Use Animation Blueprint
   - **Anim Class**: ABP_Manny

V. **Compile** and **Save** the Blueprint.

VI. Click the **Viewport** tab to confirm that the mesh has updated.

2. Character Initialization

To create **Character Movement**, several parts are needed, each working together to produce the end result of a character moving around in your game. You will need a character, inputs for that character, how the character responds to inputs from a player, animations for the character moving, transitions to and from animations, and the logic behind the transitioning between animations.

We are using the **Animation Starter Pack**, which is available for free on the **MIXAMO**, as it contains several animations that can be used to create character movement. Once you have downloaded the Animation Pack, you can add it to an existing project (preferably a New Blank Project) via the **Add to project** button in the Launcher then specifying which project you want to add the assets to.

3. Equipping Weapons

By default, Action RPG (ARPG) provides the player with a single weapon that they can use for battle. As the game progresses, and the player gathers more and more souls from the enemies that they vanquish, the player can unlock four additional weapons to use. We will now take a look at adding a new weapon for the player to purchase and use in ARPG.

First, locate the **Weapons** folder and then you will need to locate a **Skeletal Mesh** to use for a weapon. For this example, we will be using the Skeletal Mesh from the Infinity Blade Weapons pack.

Then, In the Content Browser, go to the **Blueprints > Weapons** folder and then create a new Blueprint that has **WeaponActor** as the Parent Class and a name of **WoodenSword_BP.**

Open up the **WoodenSword_BP** Blueprint and in the Components tab, select the **Skeletal Mesh**(inherited) option. Then in the Details panel, under the Mesh section, change the Skeletal Mesh to point to the **SK_Blade_HeroSword11** Skeletal Mesh.

Next, in the Component tab, select the **CapsuleCollision** (Inherited) component and then in the Details panel under the Shape section, set the Capsule Half Height to 49 and the **Capsule Radius** to 10.

Now we need to create a **Blend Space** that will be used to connect all of the needed assets for the Blade Talon sword to function together. To do this, first, **right-click** in

the Content Browser and from the displayed list, go to the **Miscellaneous** section and then select the **Blend Space** option from the displayed list.

Inside of the **Armed_BS Blend Space**, under the Weapon section, set the **Weapon Actor** to the **WoodenSword_BP** Blueprint. This will make sure that when this weapon is selected, the correct **Skeletal Mesh** will be used.

With the **Armed_BS Blend Space** now set up, make sure to save all of the assets and then launch Action RPG. When the game loads, open up the store and you should now see the **WoodenSword_BP** as a weapon you have in your hand by pressing button Z.

4. Setting up attack and Damage Variables

In this module, we have done some camera tweaking and going over things like timelines and lerps, and getting the beginnings of our attack function in place.

Also, we have been expanding on our attack functionality and setting up some of the players' attributes, including health, mana, experience points, level, etc.

Variables are properties that hold a value or reference an Object or Actor in the world. These properties can be accessible internally to the **Blueprint** containing them, or they can be made accessible externally so that their values can be modified by designers working with instances of the Blueprint placed in a level.

Variables are displayed as rounded boxes that contain the name of the variable:

Variables can be created in a variety of different types, including data types - such as Boolean, integer, and float - as well as reference types for holding things like Objects, Actors, and particular Classes. Arrays of each variable type can also be created. Each type is color coded for easy identification.

Ankur Chaurasia
(35120602019)

5. Developing an AI based enemy

In this module, we have started working on our enemy's AI and getting him to walk around. We are first working on the basics of our enemy's AI; we have made him walk, run, and also chase him down. In addition, we will make our adversary AI chase us down and start fighting back.

Now, we'll be tweaking the AI a bit, making him face the player when attacking and so on, and implementing the start of our player HUD. We'll be killing off the AI and giving the player experience points so they can level up.

6. Adding Magic Spells to the Character

Examining the first of our magical spells, a **FIREBALL**. We'll create the spell to make it fly, have our character perform it, and have it cause damage to our opponents.

Adding onto our characters magic abilities, we'll be making an **ICE SPELL** this time that bursts up from the ground to impale our enemies.

In this section, we'll build up a **HEALING SPELL** and begin our **MANA** regeneration ability. We can recuperate the harm done by the adversaries with the assistance of a healing spell. Our mana is depleted whenever we use spells, but it gradually recovers.

We will update our magic selection by adding in a shortcut menu that'll slow down time, giving us a little more time to pick what magic we need for the situation.

7. Creating Landscape

Unreal Engine 4 (UE4) is capable of creating massive terrain-based worlds using its suite of powerful terrain editing tools. The **Landscape** tool allows you to create immersive outdoor terrain pieces that optimized and can still maintain playable frame rates across a multitude of different devices. You can create your Landscape using any of the following methods:

- You can create a new Landscape heightmap from scratch using the in-engine Landscape tools.
- You can import a Landscape heightmap created previously in Unreal Editor or created through external tools
- You can even create your own format for importing Landscapes.

7.1 Enabling the Water System Plugin and Content

The Water system enables you to create rivers, lakes, and oceans that all interact and work together with your Landscape terrains using a spline-based workflow. The Water system unifies the shading and mesh rendering pipeline, with surfaces that support physics interactions and fluid simulation with gameplay, such as ripples caused by footsteps or the wake behind a boat moving through the water.

7.2 Foliage Tool

Foliage Tool enables you to quickly paint or erase sets of Static Meshes or Actor Foliage on filter-enabled Actors and geometry.

7.3 Types of Lights

Unreal Engine 4 has four light types: **Directional**, **Point**, **Spot**, and **Sky**. Directional lights are primarily used as your primary outdoor light or any light that needs to appear as if it is casting light from extreme or near infinite distances. Point lights are your classic "light bulb" like light, emitting light in all directions from a single point. Spot lights emit light from a single point, but have their light limited by a set of cones. Sky lights capture the background of your scene and apply it as lighting to your level's meshes.

7.4 Sun and Sky Actor

The **Sun Position Calculator** plugin includes a Geographically Accurate Sun Positioner that gives you fine control over the Sun's position based on a geographic location and date in time. The **SunSky** Actor is part of this same plugin. It uses the same mathematical equations to govern the position of the Sun in the sky and includes a bundle of components—a Directional Light, Sky Light, and SkyAtmosphere—to produce true-to-life renderings that show realistic patterns of sunlight and shadow.

The SunSky Actor makes it simple and quick to set up your scenes no matter your aesthetic choice, with settings for Daylight Saving Time (DST), the date, and the time of day. It's designed to work well within games and other industries, like our Architectural, Engineering and Construction (AEC), or Automotive, Product Design and Manufacturing.

7.5 Volumetric Clouds

The Volumetric Cloud component is a physically-based cloud rendering system that uses a material-driven approach to give artists and designers the freedom to create any type of clouds they need for their projects. The cloud system handles dynamic time-of-day setups that is complemented by the Sky Atmosphere and Sky Light using the real

time capture mode. The system provides scalable, artist-defined clouds that can adapt to projects using ground views, flying, and ground to outer space transitions.

8. Inventory

In this module, we'll go through how to create an inventory menu, how to design it and get all of our buttons in place, and how to have it manage our inventory properly.

We're also working on getting our crafting menu up and running. Some design elements are tiresome, yet they may significantly improve the gamers' overall experience.

9. Skill Tree and Travel System

In this module, we are firstly setting up a fast travel system, we'll make it to where you can open a menu, select which waypoint you want to travel to, and fade the camera out as you travel to it. In the next one, we'll set up restrictions so that you can only travel to them if you've found it before hand and there are no enemies nearby, but lets get the base of it established.

Now that we have our fast travel system in place, then we will add some restrictions. namely, making sure the player has found the lodestone first, and that there are no enemies nearby.

Finally, we move to our skill tree where we'll start off with making it pretty and setting up all our buttons for use.

Now, we are setting up the functionality behind our skill tree, making it check how many skills points our player has, using one up and applying the updates to our characters stats. We'll also set in limiters so that certain stats can't be boosted too much, and give the player too much of an edge.

10.Adding SFX

Sound is vital to creating believable and immersive environments. From ambient sounds to interactive sounds of vehicles or weapons, from music cues to spoken dialog, audio in a game can make or break the user experience.

The Audio Engine system in Unreal Engine provides tools and features that offer ways for audio designers and engineers to mold audio to match their visions. This means that a clean version of the sound can be produced once in an external application, imported, then crafted within the engine to create the desired result.

The **Audio Mixer** is Unreal Engine's fully-featured, multi-platform audio renderer. This powerful audio engine was first launched in Fortnite and was later added to Unreal Engine 4.24.

Since its initial release, the Audio Mixer has continued to evolve its legacy feature set, and has added next-generation audio rendering features that provide unprecedented control and flexibility to audio designers working on real-time experiences.

This overview covers the current legacy feature set, as well as the next-generation features added to Unreal Engine 5.

The Next Generation of Sound Rendering

Many of the features developed for Audio Mixer expand on the legacy features found in Unreal Engine and address the challenges mentioned above. Other features were newly developed to give Unreal Engine capabilities similar to other professional audio authoring tools.

MetaSounds address the issues present in Sound Cues and serve as a replacement for Unreal Engine's default audio object. Unlike Sound Cues, MetaSounds are fundamentally a Digital Signal Processing (DSP) rendering graph. They provide audio designers the ability to construct arbitrarily-complex procedural audio systems that offer sample-accurate timing and control at the audio-buffer level.

MetaSounds

Unreal Engine 5 introduces **MetaSounds**, a new high-performance audio system that provides audio designers with complete control over Digital Signal Processing (DSP) graph generation for sound sources.

Audio Modulation

Audio Modulation is the evolution of these legacy classes and it liberates the concept of parameter control and modulation to a generic "parameter bus". Anything can be a modulation source and anything can be a modulation destination.

Chapter-4

SYSTEM DEVELOPMENT

- **Equipping Weapons**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Guard Mode**

Ankur Chaurasia
(35120602019)

- **Camera Tweaking and Attack**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Attacking and player variables**

Ankur Chaurasia
(35120602019)

- **Enemy AI Beginnings**

Ankur Chaurasia
(35120602019)

- **Advancing The AI**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **AI Tweaking and Player HUD**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **AI Death And Levelling The Player**

Ankur Chaurasia
(35120602019)

Player_BP > Gain XP
BLUEPRINT

PlayerHud > Get Text 2
Get Text 2
Return Node
Return Value
ToText (Float)
Value
Return Value
Player Ref
Target
Next Lvl
WIDGET BLUEPRINT

- **Fire Magic**

Ankur Chaurasia
(35120602019)

- **Ice Magic**

Ankur Chaurasia
(35120602019)

- **Healing Spell And Mana Regen**

Ankur Chaurasia
(35120602019)

- **Interacting And Chests Beginning**

Ankur Chaurasia
(35120602019)

- **Status Menu Beginnings**

Ankur Chaurasia
(35120602019)

- **Expanding Leveling And Harvesting Plants**

Ankur Chaurasia
(35120602019)

- **Tidying PlayerBP**

Ankur Chaurasia
(35120602019)

- **Expanding On Harvesting**

Ankur Chaurasia
(35120602019)

- **Magic Shortcut and Regen Tweaking**

Ankur Chaurasia
(35120602019)

- **Landscape Beginnings**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Player_BP > Event Graph
BLUEPRINT

PlayerHud > Get Visibility 1 (pure)
WIDGET BLUEPRINT

- **Getting our Widgets Ready**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Inventory Menu**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

GreenPlant_BP > Event Graph
BLUEPRINT

YellowPlant_BP > Event Graph
BLUEPRINT

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Potion Functionality**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Equipment Menu Design**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

SkillTree > Event Graph
WIDGET BLUEPRINT

SkillTree > Event Graph
WIDGET BLUEPRINT

- **Fast Travel System & Fast Travel Restrictions**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Chapter-6

SYSTEM IMPLEMENTATION

- **Engine Setup & Character Initialization**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Equipping Weapons**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Guard Mode**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Camera Tweaking and Attack**

Ankur Chaurasia
(35120602019)

PlayerAnim_BP > AnimGraph > New State Machine
Attacking
MeleeReady
SheatheWeapon
EquipWeapon
Entry
Initialize
ANIMATION

- **Attacking and player variables**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Enemy AI Beginnings**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Advancing The AI**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **AI Tweaking and Player HUD**

Ankur Chaurasia
(35120602019)

No Animation Selected

- **AI Death And Levelling The Player**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Fire Magic!**

Ankur Chaurasia
(35120602019)

- **ICE Magic!**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Healing Spell and Mana Regen**

Ankur Chaurasia
(35120602019)

- **Interacting and Chests.**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Status Menu Beginnings**

Ankur Chaurasia
(35120602019)

Name Name
Level Name
HP 00
MP 00
Strength 00
Defense 00
Experience 00
Next Level 00
No Animation Selected

Fire
Name Kyrina
Level 1
HP 50
MP 30
Strength 2
Defense 2
Experience 0
Next Level 20
Gold 0

Ankur Chaurasia
(35120602019)

- **Expanding Levelling and Harvesting Plants**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Expanding on Harvesting**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Magic Shortcut and Regen Tweaking**

Ankur Chaurasia
(35120602019)

Text Block
Text Block
No Animation Selected

Fire
Magic
Fireball
Ice Spike
Heal

- **Landscape Beginnings and More**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Compiling Shaders (95)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Sprint and Stamina Drain, Recovery**

Ankur Chaurasia
(35120602019)

- **More Landscape, Plus Foliage Tools and More!**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Fire
0

Fire
0

Ankur Chaurasia
(35120602019)

- **Designing the level and collision fixes**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

\

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Getting our Widgets Ready**

Name
Name
Level
Name
HP
00
MP
00
Strength
00
Defense
00
Experience
00
Next Level
00
No Animation Selected

Name
Name
Level
Name
HP
00
MP
00
Strength
00
Defense
00
Experience
00
Next Level
00
No Animation Selected

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Inventory Menu**

Designer
Graph
0
No Animation Selected

Designer
Graph
Text Block
No Animation Selected

Ankur Chaurasia
(35120602019)

-Inventory-
0

-Inventory-
0

- **Crafting Menu and Base Functionality**

WIDGET BLUEPRINT

-Crafting-
Potions
Food
Upgrades
Crafting Level
Novice

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

InventoryScreen > Get Text 20
Get Text 20
Return Node
Return Value
ToText (string)
In String
Return Value
Append
A
Return Value
B
Add pin
Player Ref
Target
Blue Berries
WIDGET BLUEPRINT

-Inventory-
X 0
X 0
X 0
X 2
X 0
X 0
X 2
X 5
X 2
X 10
X 0
X 0
0

Ankur Chaurasia
(35120602019)

- **Hunger and Feeding**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Adding More Harvestables**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Potion Functionality**

-Inventory-
x0
x0
x0
x0
x0
x1
x0
x0
x0
x0
x0
x0
0

- **Equipment Menu Design & Changing Equipment**

Ankur Chaurasia
(35120602019)

Name Kyrina
Level 1
HP 50
MP 30
Strength 2
Defense 2
Experience 0
Next Level 20

-Equipment-
Wood Sword
Iron Sword
Steel Sword
Dragon Blade
Wood Shield
Iron Shield
Weapon Level 1
Armor Level 1

Ankur Chaurasia
(35120602019)

- **Fast Travel System & Fast Travel Restrictions**

Town
Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)
DPI Scale 0.67
No Animation Selected

Town
Observatory
Hut
Ocean
Device Content Scale 1.0
No Device Safe Zone Set
1280 x 720 (16:9)
DPI Scale 0.67
No Animation Selected

Ankur Chaurasia
(35120602019)

Town
Observatory
Hut
Ocean

Stamina

Stamina

Town
Stamina
Observatory
Hut
Ocean

- **Skill Tree Functionality and Limiters**

Ankur Chaurasia
(35120602019)

Ankur Chaurasia
(35120602019)

- **Character SFX**

Ankur Chaurasia
(35120602019)

Chapter-7

SYSTEM TESTING

- Testing: Testing is a major part of creating our website as we want our customer use a problem free sign recognition website. For this, we test the site over multiple browsers and devices to check whether it is working like supposed to. As at this point, the URL is private and should launch after confirming all the things.

- Unit Testing: It focuses verification effort on the smallest unit of software design, the module.

- Conditional Testing: In this part of the testing, each of the conditions were tested to both true and false aspects. And all the resulting paths were tested so that each path that may generate on particular condition is traced to uncover any possible errors.

- Data Flow Testing: This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variables were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

- Loop Testing: In this type of testing, all the loops are tested to all the possible limits. All the loops were tested at their limits, just above them and just below them. For nested loops test, the inner most loop first and then work outwards. For concatenated loops, the values of dependent loops were set with the help of connected loop. Unstructured loops were resolved into nested loops or concatenated loops are tested as above.

Chapter-8

FUTURE SCOPE

- This game is made for PC in future we can make an IOS version & Android version.
- We will add a database to our game which will store the process of the player.
- We can add multiple characters to our game.
- We can also create dynamic environments and levels to make the experience more interactive for the users.
- We can also create a setting for customizing graphics for end users.
- We can also add cut-scenes to add a story line to the game.
- We can also add the feature of multiplayer.

CONCLUSION

Erica-The Warrior can be the most fascinating role-playing game for the end users. Role-playing games sit at the intersection of four phenomena – play, roles, games, and media culture. Role-playing games are quite interactive.

Erica-The Warrior is free to move anywhere in the environment and can perform magic spells, hit, kill, and heal herself. This open world game also has many animals to make it look more lively, as well as many other graphical animations that can entice users to play our game.

Unreal Engine fastens to layouts, colour, sound, physics, and animation. Today's contemporary game engines supply realistic. graphics. Game engine choice could repulse the standard of the game advanced or back in any facet.

REFERENCES

- Unreal Engine 5: https://www.unrealengine.com/en-US/?sessionInvalidated=true
- Mixamo: https://www.mixamo.com/#/
- https://www.slideshare.net/NadiaIIT/final-project-report-of-a-game
- https://www.researchgate.net/publication/330942093_Video_Game_Report_-_Riko_The_Adventurer
- http://norma.ncirl.ie/2390/1/kellymcguinness.pdf
- https://www.theseus.fi/bitstream/handle/10024/71525/Xia_Peng.pdf?sequence=1&isAllowed=y
- https://www.scribd.com/document/433077579/Project-Report-Game
- Cabbit. (2014). 24x32 Characters Big Pack. Available from: https://opengameart.org/content/24x32-characters-with-faces-big-pack [Accessed on 15 December 2018]
- Beast. (2011). Sewer Tileset. Available from: https://opengameart.org/content/sewer-tileset [Accessed on 15 December 2018]
- Angelee. (2018). Little Servant Devil Animation. Available from: https://opengameart.org/content/little-servant-devil-animation [Accessed on 15 December 2018]
- C Tatz. (2014). Free UI Asset Pack 1. Available from: https://opengameart.org/content/free-ui-asset-pack-1 [Accessed on 30 December 2018]
- Kenney. (2014). Onscreen Controls (8 styles). Available from: https://opengameart.org/content/onscreen-controls-8-styles [Accessed on 30 December 2018]