**Unreal S2**

**Actor**

An Actor is any object that can be placed into a level, such as a Camera, static mesh, or player.

**Actor Component**

Base/Root of an Actor.

Actor component are the building blocks of actors. Different actor components control different pieces of an actor, such as how the actor moves, how it is rendered, where it exists in the world, how it is viewed by the player and so on.

**Pawn (inherits from Actor and gives base abilities to Character)**

Pawn is the base class for any actor that can be controlled by a player or AI.

**Character**

Special type of pawn designed for a vertically-oriented player representation that can walk, run, jump, fly, and swim through the world.

**World** [🡪Made a Bowling Alley]

World is the top-level object representing a map in which Actors and Components will exist and be rendered. Contains the persistent level and many other objects like gamestate, gamemode, and lists like Pawns and Controllers currently on the map.

**Game Instance**

The game instance class persists throughout the lifetime of the game. Traveling between maps and menus maintain the same instance of this class. This class is used to manage information and systems that need to exist throughout the lifetime of the game.

**User Interface, HUD, Camera**

UI consist of game menus, HUD, and other elements drawn over the game screen.

The HUD is the base object for displaying elements overlaid on the screen.

Camera represents the player’s POV of the world.

**Real-time lights and baked lights**

**Subsurface Scattering**

**Engine-Garbage Collection**

**IMP:**

End key – Stick object to below surface

Alt key (while moving object) – Duplicates object

Why camera is still there in world settings after deleting it in UE5

Delete Player, Add CameraActor in Scene

AutoActivate SceneComponent and CameraComponent and AutoPlayerActivation for Player 0

Right Click on CameraActor, move object to camera & vice versa

**Level Design**

**How to make a level:**

Flow

Challenge

Reward

Clarity

Immersion

**Pre-Production**

About planning the basic shape and scope of the project. What is the project about? What are the design goals and constraints?

**Combat Design**

**Layout**

**Blockout**

A blockout is a playable rough draft of the level, built with simple blocky 3D shapes in low detail.

**Scripting**

**Lighting**

**Environment Art**

**Game Design – Defines rules and mechanics**

**Level Design – Builds spaces where those rules come alive**

**Environment Art – Adds detail, lighting, and emotion to those spaces**

**Flow, Space, and Player Guidance**

Flow is the emotional rhythm of experience – moments of tension and release, challenge and relief. In level design, we shape this with space, light, and timing.

**Spatial Composition:**

Start Macro – Large zones of movement

Then Micro – Detailed loops and encounters

Then Detail – Lighting, Props, Story cues

**Player Path and Guidance**

Use landmarks

Use lighting contrasts

Use colour direction

Use sound or motion cues

**Choke Points & Landmarks**

Players remember spaces by landmarks – unique visual anchors that make navigation natural.

**Environmental Storytelling**

A scene should speak before a character does. Every prop, every crack, every colour is a line of dialogue in the world.

**Testing, Iteration, and Polish –**

Every level evolves. The magic word is iteration – test, observe, fix, test again.

**Playtesting & Metrics**

Observe player routes

Measure completion time

Track engagement

Refine clarity

**Balancing Challenge & Flow**

Challenge keeps players engaged; clarity keeps them moving; reward keeps them coming back.

**Visual Polish**

Final emotional tuning

**Optimisation**

Balance art and performance

**Presenting Your Work**

Process documentation – Concept, Layout, Flowcharts, Iterations & Final Renders