### **Project Plan & Game Design Document: *Echoes of the Past***

**Project Title:** *Echoes of the Past* **Project Lead:** *[Your Name]* **Target Platform:** PC **Genre:** Dark Fantasy Roguelite **Elevator Pitch:** A time-bending roguelite where you are an "echo" of your past self, reliving your memories to correct mistakes and defeat the villain you are destined to become.

### **1. Lore & Narrative**

You are not the hero of this story—not yet. You are an **Echo**, a fragmented consciousness of a person destined to become a powerful, malevolent force. The mind of your future self has become a prison, a labyrinth of shattered memories and corrupted actions.

Your mission is to traverse this mental landscape, reliving key moments and correcting the mistakes that led to your downfall. Every level is a twisted memory, a past event where a crucial moral choice was made. The enemies are manifestations of your guilt, regrets, and the sinister forces that influenced you.

Your ultimate goal is to reach your **future self**—the villain you are destined to become—and defeat them in a final confrontation to erase that timeline and create a new, hopeful one.

### **2. Core Gameplay Mechanics**

* **Death & Echoes:** When you die, your current run ends, but you leave behind a **Temporal Anchor**—a new kind of Echo. This anchor is not just power; it is a **memory fragment** of a past mistake.
* **The Temporal Anchor System:** Upon finding and "purifying" this anchor in a subsequent run, you gain the power and knowledge from your previous attempt. The longer the previous run lasted, the more significant the "memory fragment" and the more powerful the boon it grants you. This encourages deep runs, as the power you gain is directly tied to the effort you put in.
* **The Ascendancy & Alternate Timelines Loop:** After your first victory, the game introduces a new layer of complexity and replayability:
  + **Ascension Levels:** Each time you defeat your future villain self, you unlock a new Ascendancy level. This makes enemies stronger, adds new attack patterns to bosses, and introduces more complex challenges. The goal is to see how high you can ascend.
  + **Alternate Timelines:** Within each Ascendancy level, there will be secret objectives you can complete during a run. These objectives, when fulfilled, will transport you to a unique "Alternate Timeline" with its own narrative, a new final boss, and a unique ending. Upon completing that timeline, you will return to the main Ascendancy path.
* **Permanent Progression:** You will collect a resource from purified memories to unlock new permanent abilities, allowing you to venture deeper into the corrupted mindscape.

### **3. Art & Sound Direction**

* **Art:** For the prototype, all art will be placeholders. The game's theme suggests a focus on dark, ominous tones and a palette of grays, deep blues, and subtle, corrupted reds.
* **Sound:** We will use simple sound effects generated by an AI tool for placeholder sound, such as a "hit" sound, a "death" sound, and a "collect" sound.

### **4. Project Plan & Prototype Specification**

This is the most critical section for your coding assistant. Use this as a direct prompt to guide the initial development.

**Prototype Goal:** Create a single, playable build that tests the player controller, the Temporal Anchor system, and a basic dungeon.

**Specific Implementation Tasks:**

1. **Project Setup:**
   * Create a new 2D project in the chosen engine (Unity or Godot).
   * Set up a camera that follows the player.
2. **Player Controller:**
   * Create a player object (placeholder: a green square).
   * Implement **2D movement** using keyboard input (e.g., WASD or arrow keys). The player should have a speed variable to control movement velocity.
   * Implement a **melee attack** function. This attack should be a simple trigger that checks for enemy collisions in a small area around the player. It should have a cooldown timer to prevent spamming.
3. **Core Temporal Anchor System:**
   * Create a global memory\_manager script. This script should contain a global timer that runs continuously during a run.
   * Create a new class or script for the **Temporal Anchor object**. This object should have a single variable, run\_duration.
   * In the player's script, create a Die() function. When this function is called, it should:
     + Record the current value of the global timer.
     + Instantiate a new Temporal Anchor object at the player's current position.
     + Pass the timer value to the Temporal Anchor's run\_duration variable.
     + Reset the game to the beginning of a new run.
   * Implement logic for the Temporal Anchor: when a player collides with it, display a message (e.g., in the console) with the run\_duration value, and then destroy the object.
4. **Enemy & Dungeon:**
   * Create a single enemy object (placeholder: a red square).
   * Implement a simple **enemy AI**: when the player enters a certain radius, the enemy will chase the player.
   * Create a basic **procedural dungeon generator** that creates a few simple rooms and connects them with corridors.
5. **Initial Playtest:**
   * Once the code is working, play the prototype multiple times. Test for the following:
     + Can you move and attack correctly?
     + Does a Temporal Anchor with the correct timer value spawn when you die?
     + Does collecting the Temporal Anchor work as intended?
     + Is the enemy AI working correctly?