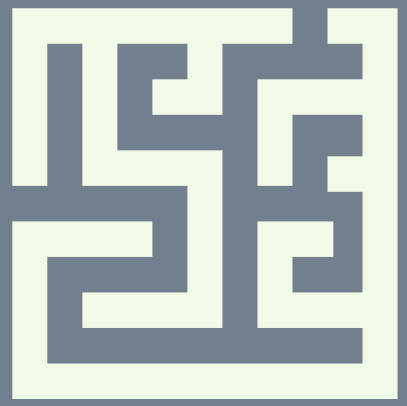
AGAINST TIME



A Tetris' Production

CONCEPT

- Welcome to Against Time, a first-person 3D maze game inspired from one of the best TV shows of the last decade, *SEVERANCE.*
- In this era of a corporate driven world, people are becoming consumed by work, gradually losing sight of what truly matters. The pursuit of productivity blinds them to the real meaning of life: spending time with loved ones and simply enjoying the moments-even the sad ones, that make life worth living.

USE CASE

Main Success Scenario

1. System presents the main menu with play options
2. Player selects "Start Game" or "New Game"
3. System loads Level 1 (easier difficulty) and starts the countdown timer
4. Player navigates the 3D maze using keyboard (WASD) and mouse for looking
5. Player locates and reaches the exit before the timer expires
6. System congratulates player and presents option to proceed to Level 2
7. Player selects to continue to Level 2 (harder difficulty)
8. System loads Level 2 and resets/adjusts the countdown timer
9. Player navigates the more complex Level 2 maze
10. Player reaches the final exit before the timer expires
11. System congratulates player on completing the game and displays final score/time

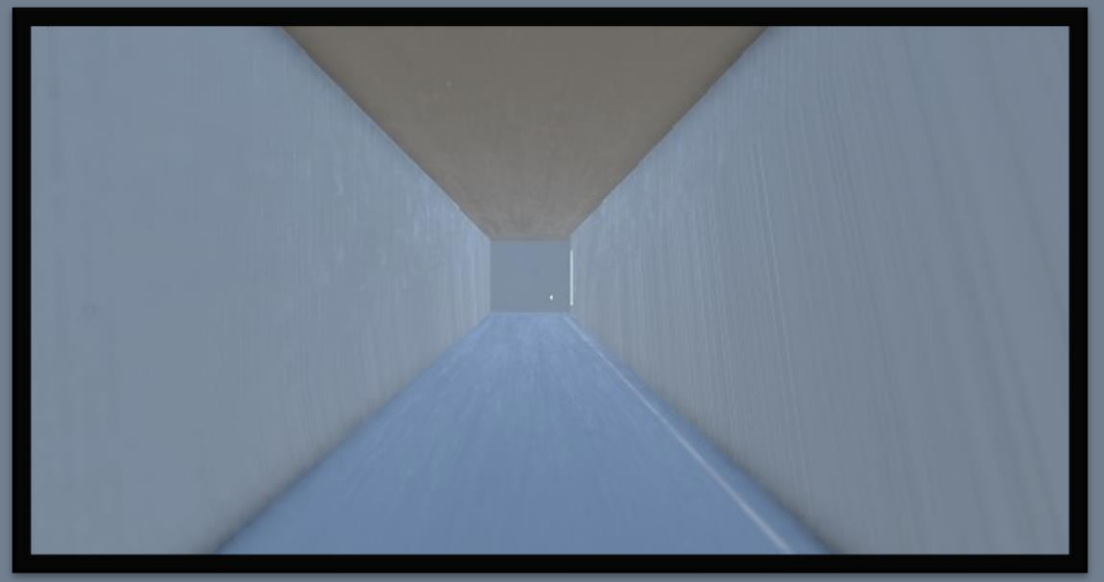
GAME OVERVIEW

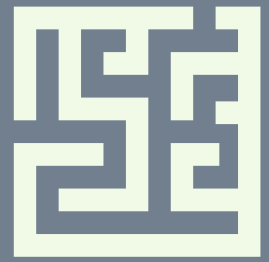
- Title: Against Time
- Genre: First-Person Puzzle/ Psychological thriller
- Platform: PC(Unity)
- Game Duration: Short session(5 min per run)
- Core Mechanics: Time-limited navigation, memory-based progression, environmental storytelling.
- Themes: Workaholism, lost-time, self reflection.

GAMEPLAY MECHANICS

CORE LOOP:

- Player wakes up in a fixed maze with a timer.
- Navigate maze through first person controls.
- If time runs out, the player respawns and find himself at the beginning of the level.
- The maze always remains the same.
- The goal: **reach exit before timer hits 00:00.**



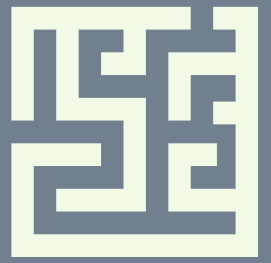


GAME MECHANICS:

- First-Person Movement: Walk, run, and explore the maze.
- Timer System: A countdown that resets the player's position upon failure.
- Static Maze Layout: No shifting walls- players must memorize paths.

GAME CONTROLS:

- **W, A, S, D:** Move forward, left, backward, right respectively.
- Mouse Movement: Look around and turn.
- Spacebar: Jump
- Shift: Sprint



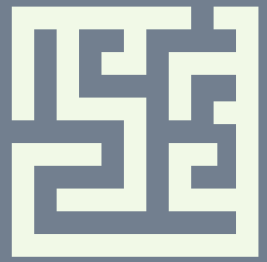
Levels and Progression

LEVELS

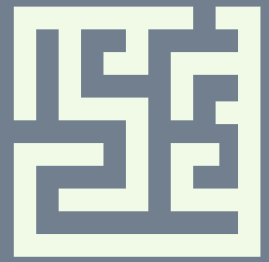
- LEVEL 1
- LEVEL 2

PROGRESSION

- Timer of 1 minute.
- Respawn to the beginning of the maze upon failure to exit.
- Timer of 3 minutes.
- More difficult maze.
- Respawn at the beginning of the maze.



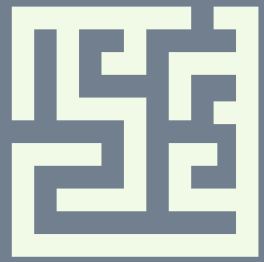
SCRIPTS



MAZE TMER:
Resets when player respawns.

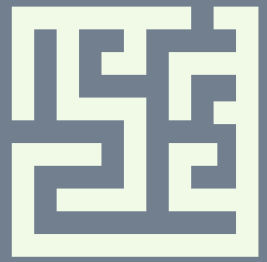
```
using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement; // Import SceneManager for scene reloading
using TMPro;  // Import TextMeshPro for UI text

public class Timer : MonoBehaviour
{
    public float timeRemaining = 180f; // 3 minutes = 180 seconds
    public TextMeshProUGUI timerText; // Reference to the UI text
    private bool timerRunning = true;

    void Start()
    {
        UpdateTimerDisplay(); // Show initial time
        StartCoroutine(StartCountdown()); // Start the coroutine
    }

    IEnumerator StartCountdown()
    {
        while (timeRemaining > 0 && timerRunning)
        {
            yield return new WaitForSeconds(1f); // Wait for 1 second
            timeRemaining--;
            UpdateTimerDisplay();
        }

        if (timeRemaining <= 0)
        {
            OnTimerEnd();
        }
    }

    void UpdateTimerDisplay()
    {
        int minutes = Mathf.FloorToInt(timeRemaining / 60);
        int seconds = Mathf.FloorToInt(timeRemaining % 60);
        timerText.text = $"{minutes:00}:{seconds:00}"; // Format as MM:SS
    }

    void OnTimerEnd()
    {
        Debug.Log("Time's Up! Restarting Scene...");
        ReloadScene();
    }
```

```
// Reloads the current scene
void ReloadScene()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}

// Optional: Call this to stop the timer
public void StopTimer()
{
    timerRunning = false;
}

// Optional: Call this to restart the timer manually
public void RestartTimer()
{
    timeRemaining = 180f;
    timerRunning = true;
    UpdateTimerDisplay();
    StartCoroutine(StartCountdown()); // Restart the coroutine
}
```



TO CHANGE LEVELS:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class nextscene : MonoBehaviour
{
    public string scenename;

 void OnTriggerEnter(Collider other){
  if(other.CompareTag("Player")){
   SceneManager.LoadScene(scenename);
  }
 }
}
```



ASSETS

UNITY ASSET STORE

https://assetstore.unity.com/?srsltid=AfmBOoq8cE6VaRykv1X-Uu83s1NPCfq52pmuvD_Gua9E4pKsNTfZPPeG

1. Maze Generator
2. Texture pack-Floor plan
3. Texture pack-Wall plan
4. First person Controller

CONTRIBUTORS

1.Abhinav K Nair

- Planning and development of Mazes
- Implementing First Person Controller

2. Alna Jaison

- Main Menu
- UI/UX of Maze

3.Annabel Marianne Victor

- Planning and Development of Mazes

4.Cebatina Treesa Joseph (Team Lead)

- Research and Storyline