Technical Design Document

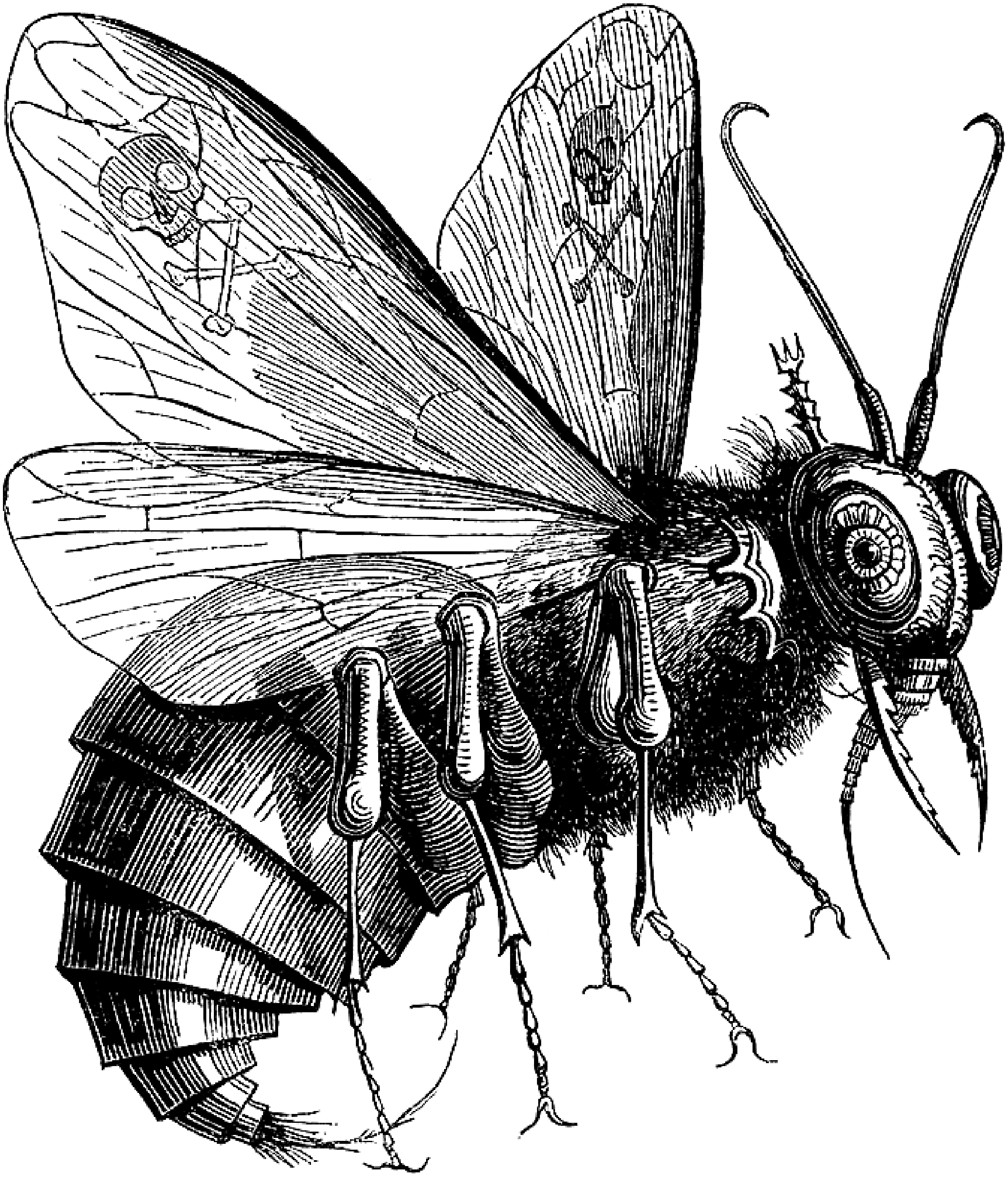
Recording 02:17

Yassir Essabbahy

Mohamed Bairouk

Driss Moussaoui

Saad El Rhandouri



1. Engine & Tools

- **Engine:** Unity 6000.0.6f21
  Language: C#
  Platform: PC (Windows)
  Perspective: 3D (2.5D side-scrolling)

2. Core Systems

1. **Player Movement & Camera**
   - 3D Game Mouvements
   - WASD To Move
   - Collision detection with walls, furniture, and room boundaries
2. **Object interaction system (PC, notes, USB…)**
3. **Puzzle system (clock, code…)**
4. **Memory loop manager (tracks loops, triggers events)**
5. **Environmental effects (lighting, shadows, visual distortion)**
6. **Scene management & transitions**

3. Systems & Components

3.1 Player System

Script: PlayerController2_5D.cs
Responsibilities:

- Move along X-axis (side-scrolling)
- Minor vertical movement (stairs, ladders)
- Detect interactable objects within range

Core Functions:

void Move(); // handles horizontal/vertical movement

void Interact(); // triggers interaction with highlighted object

void Look(); // adjust camera orientation if needed

Components:

- Rigidbody2D / Rigidbody (for physics)
- Collider (capsule or box)
- Animator (for 2.5D animations)

3.2 Interaction System

Script: ObjectInteraction.cs
Responsibilities:

- Handle all object interactions: Notes, PC, USB, Cameras, Clock
- Trigger appropriate events (open file, play audio/video, reveal story fragment)

3.3 Puzzle System

Script: ClockPuzzle.cs
Responsibilities:

- Detect player input on the kitchen clock
- Check for correct time (02:17)
- Trigger final memory loop event when correct

```
public class ClockPuzzle : MonoBehaviour {

   public int targetHour = 2;

   public int targetMinute = 17;

   public void SetTime(int hour, int minute);

   private void CheckTime();

}
```

3.4 Audio System

Script: AudioManager.cs
Responsibilities:

- Play background ambiance, environmental sounds
- Play AI-generated guidance lines
- Trigger sound effects on interaction/events

Structure:

```
public void PlaySound(AudioClip clip);

public void PlayAIInstruction(string lineID);

public void StopAll();
```

3.5 Camera & Visual Effects (Cinemachine 4)

Camera System Overview:

- Uses **Cinemachine 4** (Unity package) to control the in-game camera.
- Designed for **2.5D side-scrolling perspective**, with a fixed plane on X-axis and limited depth (Y/Z) for parallax effects.
- Handles both **player-follow camera** and **cinematic cutscenes**.

Cinemachine Components:

1. **Cinemachine Virtual Camera**
   - Follows the player smoothly along the 2.5D plane.
   - Can be triggered to focus on key objects (PC, clock, notes).
2. **Cinemachine Brain**
   - Placed on the main camera to blend between virtual cameras.
   - Handles smooth transitions during cutscenes and puzzle events.
3. **Cinemachine Confiner**
   - Keeps the camera within room bounds (Child Room, Kitchen, Basement).
   - Ensures the camera doesn’t show empty space outside environment.

3.6 In-Game PC (UI Toolkit)

Overview:

- The PC is an interactive object inside the house.
- Uses **Unity UI Toolkit** to create a **responsive, layered UI**.
- Player interacts with it by pressing **E** when in range.
- Layers (folders, notes, cameras) show/hide dynamically by setting **GameObject.active = true/false**.

3.6.1 UI Structure

Hierarchy (UI Toolkit):

```
PC_UI (UIDocument)
 ├– BackgroundPanel (VisualElement)
 ├– FolderPanel
│    ├– FamilyFolder
│    ├– CamerasFolder
│   └─ NotesFolder
 ├– USBPanel
└─ CloseButton
```

Key Points:

- Each folder/panel is a **VisualElement** or container.
- Initially, only **BackgroundPanel** is visible.

- Clicking on a folder activates the corresponding panel and deactivates others.

3.6.2 Interaction System

Script: PCInteraction.cs

Responsibilities:

- Detect player proximity and input (E)
- Toggle PC UI on/off
- Handle folder selection → show/hide corresponding layers

3.6.3 UI Structure (UXML)

Example UXML:

```
<ui:UXML xmlns:ui="UnityEngine.UIElements">
  <ui:VisualElement name="PC_UI">
    <ui:VisualElement name="BackgroundPanel">
      <ui:Button name="FamilyFolder" text="Family"/>
      <ui:Button name="CamerasFolder" text="Cameras"/>
      <ui:Button name="NotesFolder" text="Notes"/>
      <ui:Button name="USBFolder" text="USB"/>
      <ui:Button name="CloseButton" text="Close"/>
    </ui:VisualElement>
  </ui:VisualElement>
```

```
</ui:UXML>
```

- Each folder panel is a **VisualElement**.
- Initially, only the background panel is visible; folder panels are toggled on/off dynamically.

3.6.4 Styling (USS)

Example USS:

```
#PC_UI {
  width: 800px;
  height: 600px;
  background-color: #1a1a1a;
}

#BackgroundPanel {
  flex-direction: column;
  align-items: center;
}

Button {
  width: 200px;
  height: 50px;
```

```
    margin: 5px;
    color: white;
    background-color: #333333;
}

Button:hover {
    background-color: #555555;
}
```

- USS handles **layout, colors, fonts, hover effects**, and responsive scaling.
- Dynamic visibility is handled by toggling style.display = DisplayStyle.Flex/None

3.6.5 Features & Behavior

- **Responsive UI:** Scales with screen resolution automatically via UI Toolkit.
- **Active/Inactive Layers:** Folders show/hide by toggling **DisplayStyle.Flex / None**.
- **Keyboard Interaction:** Press **E** to open/close the PC.
- **Click Handling:** Clicking folder buttons calls OpenFolder("FolderName").

- **Integration with Story System:**
    - Clicking notes or camera folders triggers events