Brief Game Design Document

Pixelated Horror

Contents

Short Description / Key concept

This game is a classic survival horror with tank controls, fixed camera angles and physics based puzzles. The game splits into two sections, day and nightmare. The day sections allows the player to discover clues that help the player to survive the nightmare sections. The player transitions between worlds by going to sleep or passing out. The games target audience is ages 18 – 40 from western countries who play psychological horror games.

The game's story revolves around the main character, a former psychologist, who suffers from lack of sleep caused by his past experiences with her patients. This happened because the main character had all the information she had to prevent the deaths of her patients, but she failed to see the connections in the past. The sudden trauma has made her realize that she could have been able to prevent the deaths. In shock, she denies the fact that the patients are dead, and starts to believe that the patients are still alive. Now, she has to go through each patient's case, and come to terms with the reality.

Game Mechanics

Core Loop

Day sections: 1. Explore 2. Collect items/clues 3. Enter the nightmare section
Nightmare sections: 1. Explore 2. Avoid nemesis 3. Use items/clues to escape the nightmare

Objects, Attributes and States

Object: Player

Atributes:

- Health: int   (0..*)
- Sprint speed: float   (0..*)
- Walk speed: float   (0..*)
- Is moving: bool   (Is moving or not)
- Is sprinting: bool   (Is sprinting or not)
- Is reversing: bool   (Is walking backwards or not)
- Is aiming: bool   (Is aiming the weapon or just holding it in a hand)
- Is reloading: bool   (Is reloading or not)
- Is alive: bool   (Alive or dead)
- In puzzle: bool   (Interacting with puzzle or moving normally)
- Has weapon: bool   (Is armed or unarmed)

Object: Enemy (nemesis)

Atributes:

- Health: int   (0..*)
- Speed: float   (0..*)
- Player visible: bool   (Sees the player or not)
- Attacking: bool   (Is attacking or not)
- Roaming: bool   (Is roaming or not)
- Is alive: bool   (Alive or dead)

Object: Puzzle door

Atributes:

- Puzzle index: int   (0..*)
- Requires key: bool   (Needs a key to interact with or not)
- Key: string   (Name of the required key item)

Object: Key

Atributes:

- Name: string   (Name of the item)
- Stackable: bool   (Can be stacked or not inside the inventory)
- Quantity: int   (0..*)
- Thumbnail: texture   (Texture file)
- Mesh: static mesh   (3D-mesh)

Skill

To master the game, the player needs to think of combat or avoiding the enemies as a puzzle. This skill can be improved upon by learning more about the game's story and the environment, and then applying the information to solve the problem. For example, instead of simply shooting the enemy, the player could lure the enemy somewhere else on the map, or use an item or the environment to defeat the enemy.

Unique Selling Proposition

What makes this game unique is that it uses physics based puzzles that are each built around a narrative. For example, a door with a physics puzzle is like an enemy character with a backstory. The physics puzzle could have many different forms, one of which being a maze that needs to be completed by moving an object through it.

Look and feel

The art style is a mix of German expressionism, steampunk and PS1 aesthetic. The visuals can be divided into two different types, which are day and nightmare. Day sections have slightly distorted geometry and somewhat possible architecture. These sections will have usual colors that are expected for the environment. The nightmare sections have distorted geometry and weird architecture. The world will have a dominant color and other complementary colors that guide the player. The soundscape contains warm and chill sounds in the normal world. Trippy and atmospheric (Creaks, drones) sounds in the nightmare world.

Demographics

The games target audience is ages 18 – 40 from western countries who play psychological horror games, and who seek the thrill of horror, identification, emotional bonding and intriguing narrative.

Resources

This game requires at least a programmer, a narrative designer, a character artist, an environment artist, a level designer and a sound designer. These task can be divided among 5-7 people, but will require the team members to have generalist skills. The approximate development time would be around 1,5 years.

The minimum required software: Blender, Unreal 5, Gimp, Any version control (For example Diversion).

The minimum required hardware: a computer setup that can reliably run Unreal 5, Blender, and Gimp.