Obsession

Rebecca Brunner

Revision: 0.0.0

GDD Template Written by: Benjamin “HeadClot” Stanley

Special thanks to Alec Markarian
Otherwise this would not have happened

License

If you use this in any of your games. Give credit in the GDD (this document) to Alec Markarian and Benjamin Stanley. We did work so you don't have to.

Feel free to Modify, redistribute but **not sell** this document.

TL;DR - Keep the credits section of this document intact and we are good and do not sell it.

Overview

Theme / Setting / Genre

- OCD, series of repeating rooms (house/dungeon), horror/suspense

Core Gameplay Mechanics Brief

- unlock doors to progress, go down stairs to next level
- doors unlocked by completing rituals/compulsions
- random chance of encountering “breaking point”
- temporary relief items

Targeted platforms

- PC

Influences (Brief)

- <Influence #1>

- *Amnesia: The Dark Descent*
- Environment/Atmosphere design

- <Influence #2>

- *Slender: the Arrival*
- Sound design

The elevator Pitch

You travel through a series of repeating rooms, and as you progress the atmosphere thickens and inte

Project Description (Brief):

Obsessive Compulsive Disorder, or more commonly known as OCD, is defined as a repetitive thought or behavior that is out of the sufferer's control and causes significant negative impact to their overall quality of life.

In the interactive experience *Obsessive*, the goal is to invoke empathy from the viewer for those who suffer from OCD, and this is done by re-creating the sensation and feeling of going through a compulsion.

One of the primary missions of *Obsession* is to steer away from stereotypical misconceptions about what OCD is and who suffers from it. This is why *Obsession* takes a very atmospheric approach in order to better describe the overall feeling of suffering from the illness as opposed to focusing solely on the highly external and potentially misleading physical actions of OCD sufferers

Project Description (Detailed)

--expanding off of previous statements

-The game consists of an infinitely repeating set of levels in which the player is constantly going downwards. As the player continues downwards, the atmosphere thickens and intensifies, and the chances of them running into their breaking-point continue to grow. The descent and intensifying atmosphere is meant to show the degeneration of an OCD sufferer's mental state as they continue to repeat/obsess. The breaking point serves as an antagonist despite having no physical form, and serves as an instant game over upon contact. The player's primary form of avoiding the breaking-point is to hide and cower and wait for the feeling to pass.

What sets this project apart?

- Unlike many college level “horror” games, this game will have an actual threat.
- This game will not rely on jump-scares/scripted events
- Highly metaphorical with no connection to supernatural
- Potentially informative and educational

Core Gameplay Mechanics (Detailed)

- Each level gets more intense

- <Details>

As the player descends the lighting and textures change to show a slow degradation of the player’s mental state.

- <How it works>

Each level simply reloads itself, however a script is preserved that counts how many levels you have completed. As the number increases, the script will alter assets in the level.

- Encountering the breaking point

- <Details>

As the player progresses they have an increased chance of encountering the breaking-point

- <How it works>

Using the variable described above, a random number will be generated using this variable. If the variable falls into a certain category, the breaking point will spawn.

- Breaking-Point AI

- <Details>

Breaking-point AI will hunt the player.

- <How it works>

Recycling code from my old project, the AI consists of three states, idle, investigation, and chase.

- Other changing elements on each level

- <Details>

Certain objects/collectibles have a certain chance to appear on each level

- <How it works>

Same as breaking-point spawn

- Inventory system

- <Details>

Used to collect items to help relieve the player temporarily of the breaking-point

- <How it works>

Further research into the programming is necessary but I imagine the basics have something to do with data structures and/or arrays.

Assets Needed

- 2D

- Textures
  - Environment Textures (upstairs/dungeon)
  - Inventory UI
  - Inventory sprites
  - Title
  - Menu UI

- 3D

- Characters
  - Particle effects for breaking-point
- Environmental Art Lists
  - Wall sets
  - Interactable items (pickups)
    - teddy bear, family photo, medicine, etc.
  - Furniture
    - nightstand, light source, etc.

- Sound

- Sound List (Ambient)
  - Outside

- Inside
    - creaking for feet walking (provided default by Unity)
    - wind noises
    - drum beats
    - scratchy noises
    - high pitched squealy noises

- Sound List (Player)
    - Character Movement Sound List
        - Default provided by Unity
    - Character on Injured / Death sound list
        - Dramatic chord change

- Code
- Character Scripts (Playercontroller)
    - Inventory script
    - Health script
- Ambient Scripts (Runs in the background)
    - Level control script - determines how level loads
- NPC Scripts
    - AI control script

- Animation
- Environment Animations
    - Door opening
    - Lights flickering
- Character Animations
    - NPC
        - Particles for breaking-point

Schedule

- **<2D & 3D Assets finished>**
    - Time Scale - End of October
        - Milestone 1 - 3D environment modeling done
        - Milestone 2 - 3D object modeling done
        - Milestone 3 - 2D Textures (upstairs & dungeon) done
        - Milestone 4 - UI elements done
        - Milestone 5 - Enemy particle effect R&D done
        - Milestone 7 - Enemy particle effect done

- **<Code>**
    - Time Scale - Early November to mid November
        - Milestone 1 - Inventory R&D done
        - Milestone 2 - Inventory done
        - Milestone 3 - Level progression done
        - Milestone 4 - Enemy AI and player avoidance done

- **<Sound>**
    - Time Scale - Mid November to December
        - Milestone 1 - Sound collection done
        - Milestone 2 - Sound implementation done

- **<Other things>**
    - Time Scale - December to due date
        - Milestone 1 - Title screen and game over screen
        - Milestone 2 - Bug testing