Dynamic VR Horror Game Final Release Summary

Group 13: Asvin Ragunathan, Isaac Sanchez, Jacob Mielczarek, Sean Comben

<u>Release One</u>:

The first release of the game included the basic level design for the 4 exploration phase environment scenes (Figure 1), as well as unique gameplay mechanics for the different scenes, such as the maze generator of the Forest scene, the map creator of the Backrooms scene, and the enemy spawner of the Hospital scene.

The first release also contained our implementation of VR controls, as well as a starting scene for the start of the game. During this release, we also began connecting the scenes by randomizing placements for keys and doors, which the player would be required to find to progress to other scenes in the game.

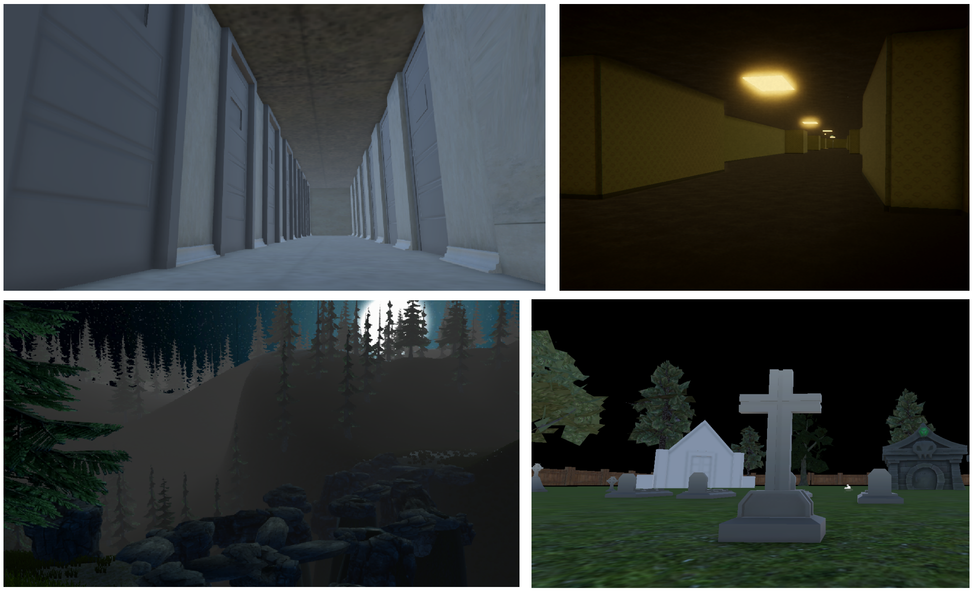
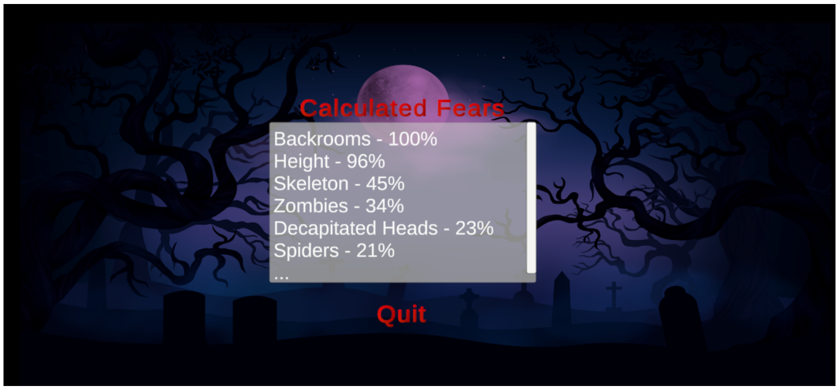


Figure 1 - The four different environments of the Exploration Phase (From top left to bottom right: Hospital, Backrooms, Forest, Graveyard)

<u>Release Two</u>:

For the second release, Group 13 decided to continue expanding upon the work of the first release by implementing the fear factor measurement system (Figure 2) and connecting the various scenes together through the use of transitions, menus, and an opening cutscene explaining the basic controls and premise for the story. The fear factor is based on certain inputs from the player such as the heart rate (monitored by the user's smart watch using the Pulsoid API) and the input volume of the user's microphone.

Group 13 also spent time cleaning up their code and environments by removing unused assets, trimming bloated scripts, and adding dynamic audio to improve performance and increase the player's immersion and help them forget they are playing a game.



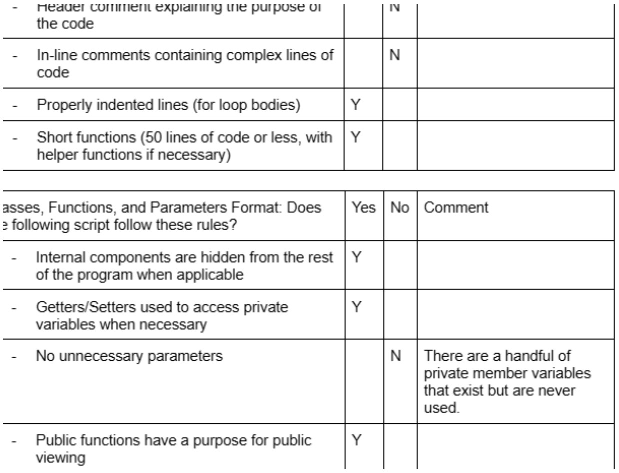
Backrooms - 100%
Height - 96%
Skeleton - 45%
Zombies - 34%
Decapitated Heads - 23%
Spiders - 21%
...

Figure 2 - Fear Factor results after a playthrough

Testing & Inspection:

Group 13 tested all the scenes' major components and menu scripts within the project, which includes the Backrooms Scene's Binary Space Partitioning algorithm, the Forest Scene's maze generation, the Hospital Scene's Entity Collision, and the menu interactions. The tests check components of these scripts, and then check the overall functionality.

Group 13 met a total of 5 times during the inspection phase. The inspection consisted of each member looking at the other group member's code and filling out the checklist for inspection (Figure 3). This consisted of checking if the code followed appropriate format, had any security flaws, and if the code worked as expected. Upon inspection, some minor bugs/flaws with certain code were found, but were corrected immediately after.



- Header comment explaining the purpose of the code		N	
- In-line comments containing complex lines of code		N	
- Properly indented lines (for loop bodies)	Y		
- Short functions (50 lines of code or less, with helper functions if necessary)	Y		

asses, Functions, and Parameters Format: Does e following script follow these rules?	Yes	No	Comment
- Internal components are hidden from the rest of the program when applicable	Y		
- Getters/Setters used to access private variables when necessary	Y		
- No unnecessary parameters		N	There are a handful of private member variables that exist but are never used.
- Public functions have a purpose for public viewing	Y		

Figure 3 - A sample of a completed Inspection Sheet

Future Improvements:

For future releases, Group 13 would begin work on the Survival Phase of the game by beginning on the generation of the custom environment designed to terrify the player and the development of a basic enemy AI that chases the player through the scene. The two phases, Exploration and Survival, would then be combined to create the final, fully playable prototype of the game.