USER MANUAL

Hanover College Video Game Development

Fire Safety Game

Made by: Max Bowe, Samuel Thielmeier, Jaxon Taylor

Table of Contents

Introduction

In this top-down game, you'll be tasked to discover a set of fire hazards scattered throughout a dorm. You can find each of these items hidden throughout the level layout. Once you find all the items that could threaten to start a fire you may leave the building safely, BUT! If you choose to leave the building and there is a single one of the fire hazard items left behind, you LOSE! It would not be safe for our college if our students and RAs can't discover these hazards before it's too late.

Github Link:

https://github.com/thielmeiers/Fire_Safety_Game.git

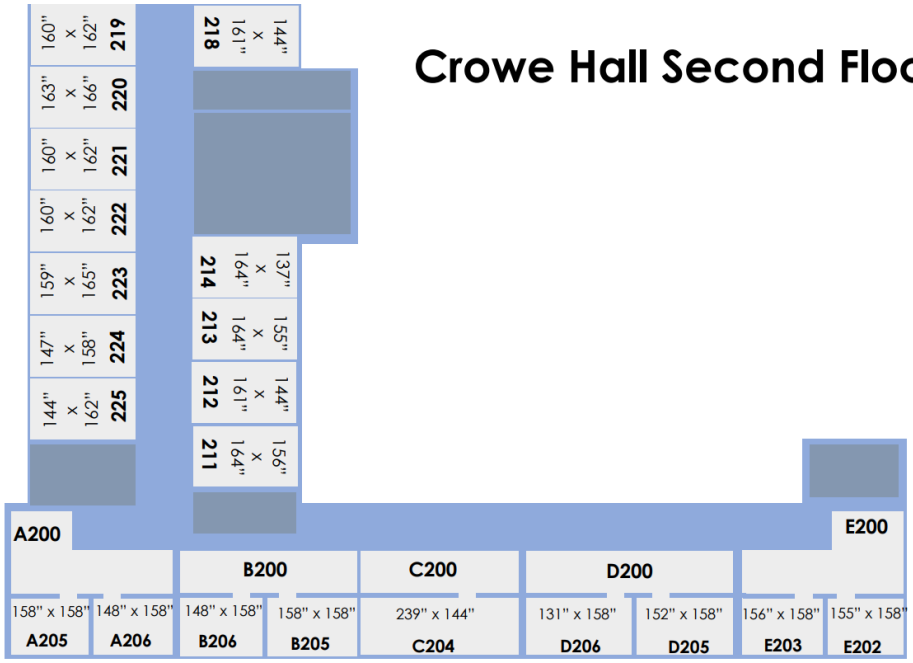
Mr. P

Mr. P aka “The Professor” is our perfect subject for this very difficult task that we have to handle. Mr. P follows all the Hanover College guidelines and never lets a single hazard slip past his eyes. He cleans up every situation to be as safe as possible for the students that are living in the dorms. Will you and Mr. P be able to keep all the student’s dorms safe from fire safety hazards?

We had a lot of different ideas for our level designs which were all centered around Hanover College dorms.

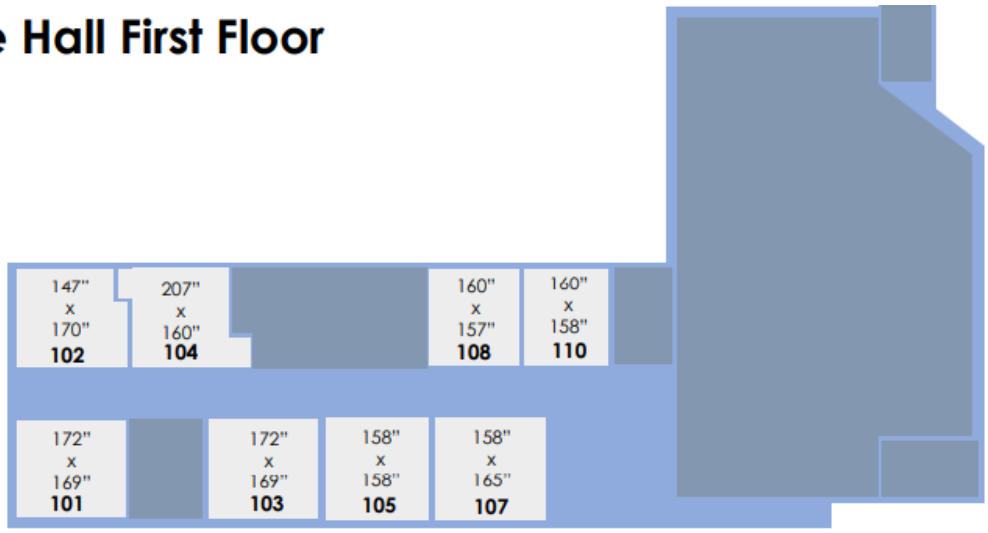
Katherine Parker Hall Second Floor

Our first choice was KP second, a pretty simple layout scheme, but most importantly out of all the dorms on campus, this one probably needs this game the most.

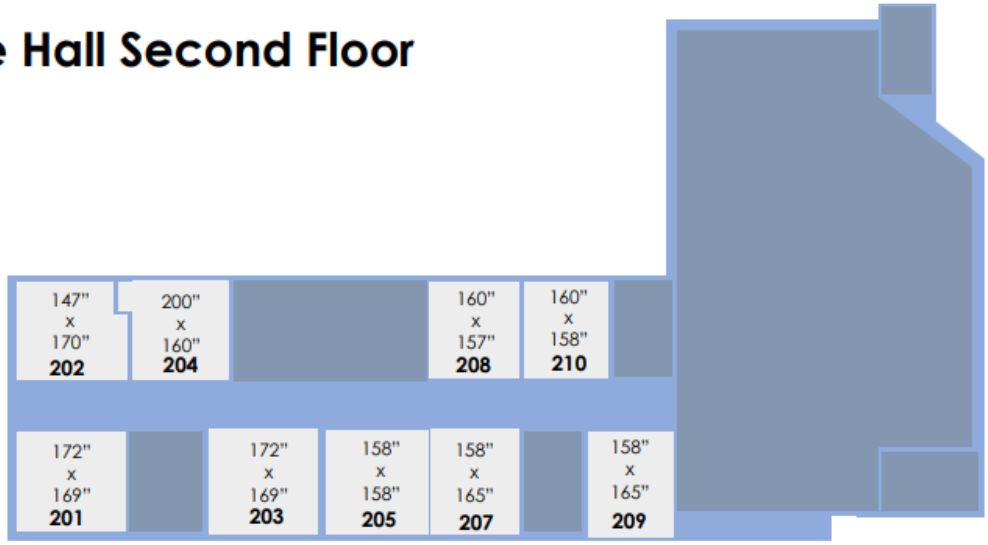


Next, we got this sort of “L” shape format that we would’ve gone for our second level if we had some more time.

Ide Hall First Floor



Ide Hall Second Floor



For our Grand Finale, It would've been a double floor layout where you could go between both floors in search of different items. This was a coding nightmare and with our time left was just not an idea worth expanding on.

Contributions made by each group member

Jaxon Taylor

I helped with fundamentally all the design assets of the game excluding the sprite himself (Mr.P) I was coding a lot of the written content for our Html pages and was a large part of the overall map design and the layout scheme. I was helping with the coding of the Tileset map onto the phaser code. Contribute to the work for the coding of certain audios and finding audios online.

Samuel Thielmeier

When coding our game, I first worked out the basic movement controls for Mr. P and map layout. First our game was coded entirely into the html code, I then moved the game code into a JavaScript file that the index html could access. Then another “upgrade” to our game I got working on was how to use scenes in phaser, which was beneficial to the organization of the game’s layout. We had multiple scenes that a JavaScript index file would access the scenes from. The scenes I had worked mainly on are the boot scene (booted the game), preloader scene (preloaded all assets into game), menu scene, ui-scene and the starting frame for the game scene. Finally, I worked on adding all beds and desks to each room after Max had made the walls for the level.

Max Bowe

From a free-to-use sprite online, I created the Mr. P sprite pixel by pixel. After Samuel figured out the player movement, I created an invisible hitbox in front of the player in order to implement interaction with objects. Next, we started creating more scenes, which act as different menus. I created the LevelEnd scene, which is activated when the player leaves the level. Then, I was tasked with creating the walls of our level, meaning the barriers that the player collides with in order to keep them in the play area. Along

with that, I placed various fire hazards around the map for the player to collect, as well as the exit area.

Development Blocks

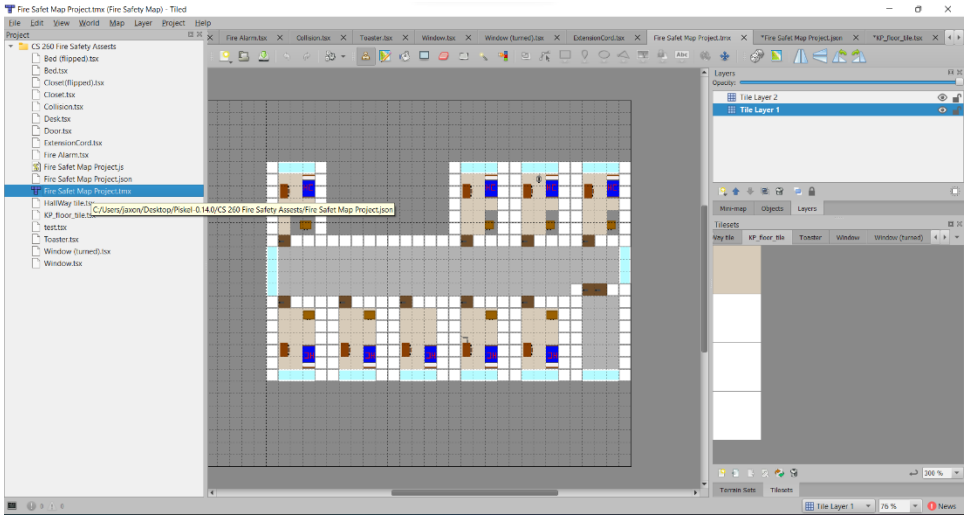
The game went through many blocks along the development process, but in my opinion there were definitely four turning points for the game.

Block 1 (Phaser Setup)

Our group felt that the Phaser software was a very good tool for our project we just needed to spend some time learning the code and setting up the “spine” of our project. But, it just wasn’t quite that easy for us. None of us really had any experience with Phaser and along with learning a new game engine entirely, we were also learning how to code as a team which was a first for most of us. We don’t regret going down the Phaser route but it was definitely a bump in the road.

Block 2 (Map Layout)

We had a decent struggle with finding out how we were going to structure our map. We originally were going to do a PNG image. But we tried to download a game map creating software called “Tiled”. Tiled was a great software for map designing of all sorts it had layers, resizing, collison commands, and many other command function to your benefits.



Block 3 (Tilesetter won’t work in Phaser code)

So the creating of the map was very simple, but putting our new json file into code on Phaser was just not working at all and were running low on time. We tried to setup the code for one extra night but after no success we finally just turned it back into a PNG and manually setup each collision block one at a time.. it was a process, and ultimately made our code less nice-looking, but it got the job done.

Block 4 (Audio)

Finally the audio for.. basically anything. So here’s the problem, Phaser for who knows why?! Likes to run all of you’re audio like its playing over a garbage disposal(potentially because it plays the audio on top of itself multiple times), making very few audio files useable and even the one that did work was usually required to be at a very specific volume or it would start messing up again. After trying to fix this for several hours, we decided that our time was better spent finishing more important parts of the game(such as actually making a fully playable game), so it was scrapped.

Apart from the audio issues, we have pushed through all of these problems and did everything that we could to remedy them.

“Work hard in silence. Let your success be your noise.” -Frank Ocean