

NAME:

Midterm Madness

BRIEF DESCRIPTION:

2-D game in which the main character is a student who's dreaming that he's late for his exam and lost in a big school building (like Humanities), looking for the room in which the exam is being held.

He has to find keys to doors, navigate through other students, and complete certain tasks, all without getting too stressed. A stress meter measures the stress level and a timer shows how much time is left till the exam begins. Power-ups may be placed all over that give him more time, or more speed or the ability to walk through walls.

He can walk, run, jump and climb staircases.

DETAILED DESCRIPTION:

The game starts off with maybe a simple room, where the immediate task of the game looks simple (ie. go down the hall and make a left, the exam is around the corner). Once the player reaches the room they'll realize they've made a mistake, and begin a "quest" of increasingly difficult tasks to obtain the location of the actual exam room. Since the game takes place in a dream, the player may encounter obstacles that seem at times to be ridiculous.

Some of the obstacles may involve finding "keys" to doors that the player thinks lead to the real exam room, but only provide certain clues (think of toad in the original Mario Bros). The player will have to play detective work, doing favors for people (bring item A to location B). They may also encounter doors and walls that seem like dead ends, needing certain power-ups or upgrades in order to pass through them.

All of this turns out to be no easy task, as the stress meter begins to increase and an overall game timer begins to get lower. The player can find "clocks" throughout the building along with "recovery" items (such as cups of coffee for speed) that help them out, though these are found in limited quantities. The stress meter is affected by bumping into other people in the hallways (and getting yelled at), or perhaps becoming jarred by music that surprises the player in the hallway. Making a mistake and going in a wrong direction may also cause the player's stress meter to increase.

The game will have very simple graphics with barely any character animation modeled.

The game would most likely use sprites in order to portray the game environment.

Designs of the game levels would be static images, while dynamic objects on the screen would be produced using sequences of images. Dialogue boxes would also be used in the game, to show what is being said to the player.

SCALABILITY PLAN:

The first plan would be to get the basic game mechanics down first, with simple obstacles, power-ups, and a smaller level design. As complexity increases, larger floors, rooms, and hallways can be added, along with more detailed puzzles that may not have very obvious solutions. Perhaps better dialogue could also be something for later, if the rest of the game is complete.

GAME PRINCIPLES DISCUSSION:

Challenge release cycle - After a hard task, player finds power-ups that decrease stress.

Feedback - Stress meter gives feedback on how you're doing . An inventory shows what items you have. The timer also gives visual feedback. Other cues like changing colors could indicate power-ups.

Punishment - Increase stress if puzzle/task isn't being solved fast enough.

Rewards - Power-Ups, prolonged play(game gets over if you run out of time or if the stress meter becomes full)

DESIGN CHALLENGES:

- 1.Connecting all the small tasks into one giant puzzle.
- 2.Organizing time (a timer for each task or one for the entire game).
- 3.Designing puzzles.
- 4.Placing obstacles and power-ups so that the challenge release cycle is maintained.

TECHNICAL OVERVIEW:

You could start by getting the player movement done, a basic level/room and transitioning between hallways/rooms or different parts of the building. Another thing to work on would be keeping track of the multiple objects on the screen without too much overhead. Getting the puzzles to work would be one of the main challenges and can be scaled in complexity.

TECHNICAL CHALLENGES:

It might be difficult storing a game, and, depending how elaborate it is, it might also be difficult to organize some kind of "save" system. The game will be more fun the more complex it is, and a simple version might not be very compelling. This game will be challenging, and to do so it will require creating complex areas and floors for our academic building. If we load everything we need into memory upon first loading the game, there might be some issues.