Introduction to Robotics: Homework #7

Mini 8x8 LED Matrix game

Andrei Dumitriu andrei.dumitriu@fmi.unibuc.ro

V1.0 Last Updated: November 21th, 2023, 17:08

Due Date

Submit your completed assignment to your respective lab session during the week of November 27th - December 3rd, 2023. Ensure your Git repository is ready and the assignment is submitted before the lab begins.

1 Objective

Develop a small game on the 8x8 matrix. The game must have at least 3 types of elements: player (blinks slowly), bombs/bullets (blinks fast), wall (doesn't blink). The purpose of this to get you started with your matrix project. The basic idea is that you generate walls on the map (50% - 75% of the map) and then you move around with the player and destroy them. You can do it Bomberman style or terminator-tanks style (or another way, be creative).

2 Components Required

- Arduino Uno Board
- Joystick
- 8x8 LED Matrix
- MAX7219
- Resistors and capacitors as needed
- Breadboard and connecting wires
- (Optional) Additional sensors / components for extended functionality

2.1 Games used for inspiration:

- Bomberman video: `https://www.youtube.com/watch?v=2xErEj86Yd8`
- Bomberman game (you can play it on a simulated Nintendo environment): `https://www.retrogames.cz/play_085-NES.php`
- Tanks on terminator platform: `https://www.youtube.com/watch?v=4ZpXWn4qzzw`

3 Be Careful:

1. **LED Differentiating:** The player and the bombs/bullets need to blink at different rates. The player should blink slowly and the bomb/bullet should blink fast. The wall should not blink at all.
2. **Control:** Ensure that control is smooth. You can implement any type of control you want on the joystick, but make sure it is "pleasant" to play with.
3. **Control:** Ensure that you do not generate walls on top of the player when the game starts.

4 Submission Guidelines

Upload your code to GitHub and update the README with at least:

1. Task requirements. Include the menu structure in the description.
2. A photo of your setup
3. A link to a video demonstrating the functionality (preferred: YouTube)
4. Ensure the video is correctly oriented.

Submit your homework through MS Teams once your Git repository reflects the latest changes.

5 Coding Standards

Clean and readable code is essential for full credit. Just make sure you follow the (or a) coding standard. **Do not use delay()! Here it starts to get messy with it.**

6 Bonus Opportunities

1. **Animations:** Add animations at the start and the end of the game.
2. **Bigger map:** Add a map that is bigger than the 8x8 matrix. You can either add new rooms that you pass through or can treat the current 8x8 as a the current field of view.
3. **Advanced game:** add enemies, "food" that you can take, power-ups, lives etc. Basically continue adding game-like features.
4. **Creative LED Feedback:** Add another LED that can inform you when you are near an enemy bomb, a food or treasure etc.
5. **Extra button:** While you can use the joystick button to "fire", you will quickly notice that it is not ideal. You can add an extra button for that.
6. **Extra sensors or components:** You can add any extra sensors or components.
7. **Menu structure:** Take advantage of the menu structure built in the previous homework so that you can start / end the game and display various elements.
8. **Score:** Implement a score system that can be displayed in Serial (or with other methods). Save it in eeprom etc.
9. **Endless possibilities:** It is a game, the possibilities are endless.