**Defuse the Bomb**

A CSC 102 Project

Team: Maze Runners Group 2

Github: [cskaliotis/CSC-102-Final-Project](https://github.com/cskaliotis/CSC-102-Final-Project)

# Team individualization

* Maze runner element
  + Unlike the standard bomb-defusal setup, we integrated a maze exploration challenge that requires players to navigate through different obstacles before defusing the bomb. This addition enhances the gameplay by introducing an extra layer of strategy and problem-solving, making it feel more like an adventure rather than just a technical puzzle.
* Riddle based game
  + Instead of relying solely on mechanical inputs like buttons and switches, we incorporated riddle-solving as part of the defusal process. Certain phases require players to analyze clues and decipher coded messages to progress, adding a mental challenge beyond traditional bomb-defusal gameplay. This twist ensures that players must engage in critical thinking and teamwork, making each session more dynamic and unpredictable.

# Future development plans

* To add more animations
  + By introducing dynamic animations for interactions (such as wire cutting, button pressing, and countdown effects), we could create a more immersive experience. Smooth transitions, flashing alerts, and subtle motion effects would make the game feel more polished and intuitive.
* Create a more appealing game aesthetic
  + Refining the visual design would elevate the player experience. This could include an improved GUI with sleek, themed graphics, better lighting effects, and a cohesive color scheme that makes the gameplay more visually striking. Custom artwork and sound enhancements could also add personality to the bomb interface.
* Add more complex riddles
  + To make the game even more mentally stimulating, we could design deeper, multi-layered puzzles that require logical deduction, pattern recognition, and cross-referencing clues within the maze. This would challenge players beyond simple defusal mechanics and encourage problem-solving and teamwork.
* Integrate adaptive difficulty levels
  + Integrating adaptive difficulty levels would make the game more inclusive, engaging, and challenging for a diverse range of players. The goal is to ensure that both beginners and experienced players can enjoy the game while keeping the intensity balanced. Here’s how it could be implemented:

# Lessons learned

* Technical and Computational Growth
  + Working with Raspberry Pi and Python provided hands-on experience in connecting software with physical components, reinforcing key concepts in computational logic. Developing puzzle mechanics and maze navigation required optimizing algorithms, an essential skill for advanced computing problems. Encountering and resolving hardware/software compatibility issues strengthened problem-solving abilities and debugging strategies.
* Relation to The Science of Computing Curriculum
  + The project aligned with concepts in computational thinking, system design, and interactive programming. Understanding logical decision-making through riddle-based mechanics connects to broader computing principles like algorithms, conditionals, and recursion. The use of sensors and real-world interactions ties into embedded systems and hardware communication, expanding knowledge beyond traditional coding exercises.
* Problem-Solving and Critical Thinking Benefits
  + Managing different aspects of the project required effective communication and delegation which are skills necessary for both software development and broader computational teamwork. Designing adaptable difficulty levels and puzzle mechanics involved anticipating player decisions, mirroring real-world problem-solving scenarios. Constant testing and revising the game structure encouraged an iterative approach to problem-solving, a vital process in software development.
* Future Benefits in Computer Science Courses
  + Strengthened algorithmic thinking for more complex programming courses. Improved UI/UX design principles, which will be valuable for projects involving human-computer interaction. Experience with integrated hardware programming, beneficial for courses in embedded systems, robotics, or game development.