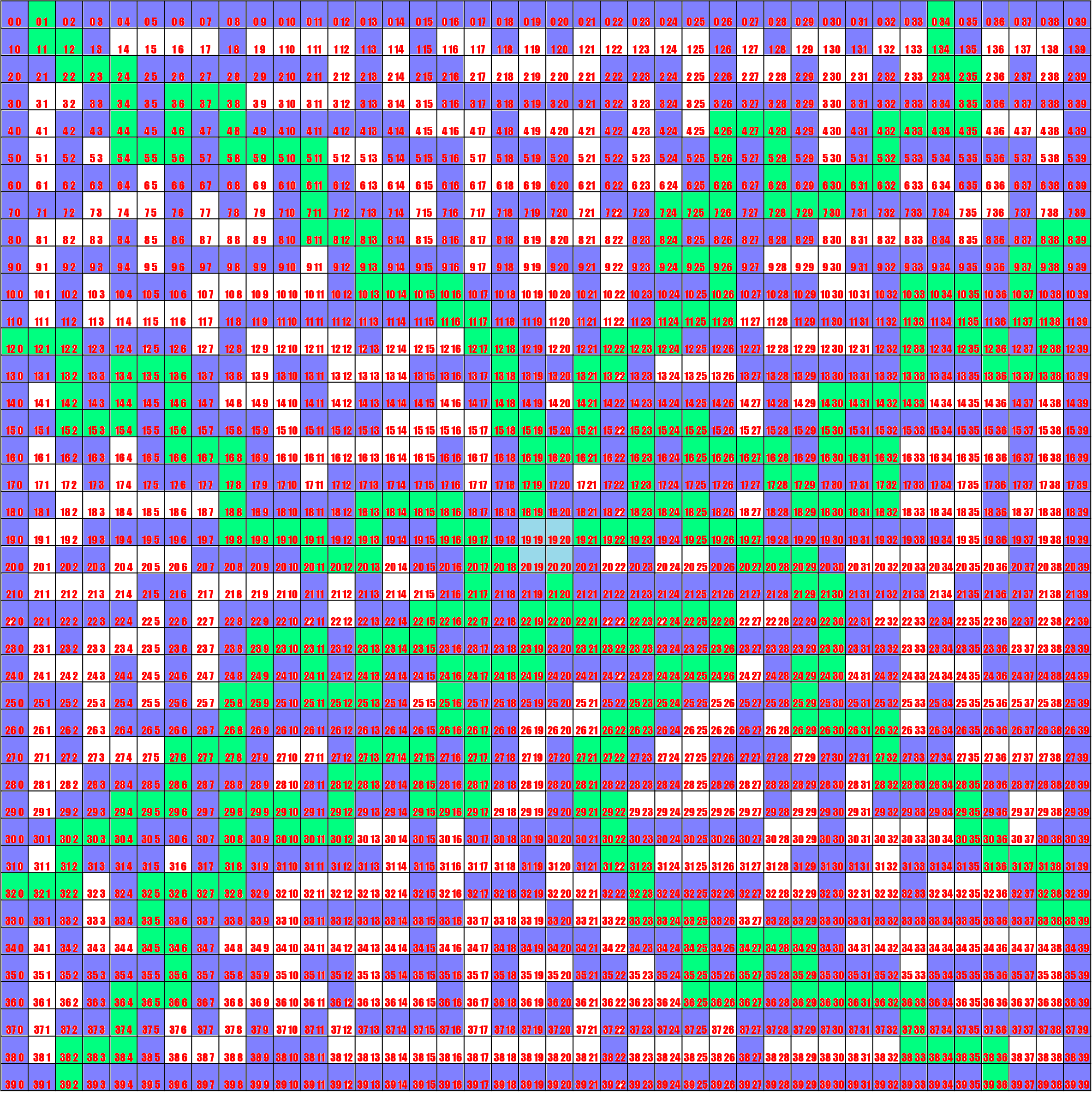
**IDEIA: JOGO DO LABIRINTO COM DOIS JOGADORES FEITO EM SISTEMAS EMBARCADOS.**

**IMAGEM DA SOLUÇÃO PLANEJADA:**

****

**MAIN.CPP**

**#define PLAYER\_1\_FLAG (1UL << 0)**

**#define PLAYER\_2\_FLAG (1UL << 1)**

Foi criado 2 macros(sinais indicativos) e dado esses dois nomes, que representam os dois jogadores, elas são usadas em conjunto com a classe EventFlags para controlar o fluxo de execução das threads associadas aos jogadores, indica quando cada jogador pode executar suas ações.

**EventFlags player\_flags;**

Cria um objeto EventFlags chamado player\_flags, que é uma classe mbed para manipular flags de eventos

**Tabuleiro tab;**

Cria um objeto da classe Tabuleiro chamado tab

**Jogador player01(PA\_0, PA\_1, "\033[91m■ \033[97m");**

**Jogador player02(PC\_1, PC\_0, "\033[32m■ \033[97m");**

Cria dois objetos da classe Jogador chamados player01 e player02, cada um com posições (pinos) e uma representação visual específica.

**Tabuleiro \*tabe = &tab;**

Cria um ponteiro para o objeto tab da classe Tabuleiro,

**void thread\_mover\_jogador1() {**

Função que controla o movimento do jogador 1

**while (true) {**

**player\_flags.wait\_any(PLAYER\_1\_FLAG);**

Execução desta thread aguarde até que seja a vez do jogador 1 se mover. **player01.mover\_jogador(tabe);**

Chama o método mover jogador, tabe é passado permitindo interação com o tabuleiro **ThisThread::sleep\_for(500ms);**

**player\_flags.set(PLAYER\_2\_FLAG);**

Levanta a flag 2, indicando ser a vez do 2 jogador

**}**

**}**

**void iniciarObjetos()** { Inicializa e monta o tabuleiro

**tab.montarTabuleiro();**

**player01.criarCasasAleatorias(tabe);** Posiciona os jogadores aleatoriamente

**player02.criarCasasAleatorias(tabe);**

**}**

**int main() {**

**BufferedSerial pc(USBTX, USBRX, 460800);**

**iniciarObjetos();** Inicializar o estado inicial do jogo

**Thread th\_jogador1, th\_jogador2;** Declara 2 objetos de thread dos 2 jogadores

**th\_jogador1.start(callback(thread\_mover\_jogador1));**

**th\_jogador2.start(callback(thread\_mover\_jogador2));**

Inicia as threads anteriores, associando-as às funções mover jogador

**while (true) {**

**if(tab.status == false){**Verifica o status do tabuleiro

**iniciarObjetos();**

**tab.status = true;** Se verdadeiro ele imprime

**}**

**tab.imprimirTabuleiro();**

**printf("\033[2J\033[H");** Limpar/atualiza o terminal

**ThisThread::sleep\_for(500ms);**

**player\_flags.set(PLAYER\_1\_FLAG);**

**player\_flags.set(PLAYER\_2\_FLAG);**

Sinaliza que é a vez de ambos os jogadores agirem, levantando suas respectivas flags.

**}**

**}**