

Code

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>


// Function to clear the console screen (works on Windows & Linux)

void clearScreen() {

#ifdef _WIN32

    system("cls");

#else

    system("clear");

#endif

}


int main() {

    int board[6][6], shown[6][6];

    int size, totalScore = 0;

    int i, j;


    // Title

    clearScreen();

    printf("=====\n");

    printf("    MEMORY MATCH MADNESS\n");

    printf("=====\n");

    printf("Match all pairs to win!\n");

    printf("Scoring: +10 for a match, -2 for a wrong guess\n\n");


    // Choose difficulty
```

```
printf("Choose difficulty:\n");

printf("1) Easy (2x2)\n2) Medium (4x4)\n3) Hard (6x6)\n> ");

int level;

scanf("%d", &level);

while (getchar() != '\n');


if (level == 1)
    size = 2;
else if (level == 2)
    size = 4;
else size = 6;


// Start from chosen level to Hard
for (int lvl = level; lvl <= 3; lvl++) {
    if (lvl == 1)
        size = 2;
    else if (lvl == 2)
        size = 4;
    else
        size = 6;


// Initialize shown array
for (i = 0; i < size; i++)
    for (j = 0; j < size; j++)
        shown[i][j] = 0;


int total = size * size;

int pairs = total / 2;
```

```

int nums[36];

int k = 0;


// Fill pairs
for (i = 1; i <= pairs; i++) {
    nums[k++] = i;
    nums[k++] = i;
}


// Shuffle numbers
srand(time(NULL));
for (i = total - 1; i > 0; i--) {
    int r = rand() % (i + 1);
    int temp = nums[i];
    nums[i] = nums[r];
    nums[r] = temp;
}


// Fill the board
k = 0;
for (i = 0; i < size; i++)
    for (j = 0; j < size; j++)
        board[i][j] = nums[k++];


// Game loop
int found = 0, score = 0, tries = 0;
while (found < pairs) {
    clearScreen();

```

```
    printf("Level %dx%d | Total Score: %d | Attempts: %d\n",size, size, totalScore +
score, tries);
```

```
    // Print board
```

```
    printf("\n  ");
```

```
    for (j = 0; j < size; j++)
```

```
        printf("%3d", j + 1);
```

```
    printf("\n  +");
```

```
    for (j = 0; j < size; j++)
```

```
        printf("---");
```

```
    printf("\n");
```

```
    for (i = 0; i < size; i++) {
```

```
        printf("%2d |", i + 1);
```

```
        for (j = 0; j < size; j++) {
```

```
            if (shown[i][j])
```

```
                printf("%3d", board[i][j]);
```

```
            else
```

```
                printf("%3s", "*");
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    int r1, c1, r2, c2;
```

```
    // First card
```

```
    while (1) {
```

```
        printf("\nSelect first card:\n Row (1-%d): ", size);
```

```
        scanf("%d", &r1);
```

```

printf(" Col (1-%d): ", size);
scanf("%d", &c1);
while (getchar() != '\n');
r1--; c1--;
if (r1 < 0 || r1 >= size || c1 < 0 || c1 >= size) {
    printf("Invalid position!\n");
    continue;
}
if (shown[r1][c1]) {
    printf("Already revealed!\n");
    continue;
}
break;
}

shown[r1][c1] = 1;
clearScreen();

// Show after first pick
printf("Level %dx%d | Total Score: %d | Attempts: %d\n",
    size, size, totalScore + score, tries);
printf("\n  ");
for (j = 0; j < size; j++)
    printf("%3d", j + 1);
printf("\n  +");
for (j = 0; j < size; j++)
    printf("---");
printf("\n");

```

```

for (i = 0; i < size; i++) {
    printf("%2d |", i + 1);
    for (j = 0; j < size; j++) {
        if (shown[i][j])

            printf("%3d", board[i][j]);

        else

            printf("%3s", "*");

    }
    printf("\n");
}

```

// Second card

```

while (1) {

    printf("\nSelect second card:\n Row (1-%d): ", size);
    scanf("%d", &r2);
    printf(" Col (1-%d): ", size);
    scanf("%d", &c2);
    while (getchar() != '\n');
    r2--; c2--;

    if (r2 < 0 || r2 >= size || c2 < 0 || c2 >= size) {
        printf("Invalid position!\n");
        continue;
    }

    if (r1 == r2 && c1 == c2) {
        printf("Same card! Try again.\n");
        continue;
    }

    if (shown[r2][c2]) {

```

```

        printf("Already revealed!\n");
        continue;
    }
    break;
}

shown[r2][c2] = 1;
tries++;
clearScreen();

// Display board again
printf("Level %dx%d | Total Score: %d | Attempts: %d\n",
        size, size, totalScore + score, tries);
printf("\n  ");
for (j = 0; j < size; j++)
    printf("%3d", j + 1);
printf("\n  ");
for (j = 0; j < size; j++)
    printf("---");
printf("\n");
for (i = 0; i < size; i++) {
    printf("%2d |", i + 1);
    for (j = 0; j < size; j++) {
        if (shown[i][j])
            printf("%3d", board[i][j]);
        else
            printf("%3s", "*");
    }
}

```

```

        printf("\n");
    }

    // Check match
    if (board[r1][c1] == board[r2][c2]) {
        printf("\nMATCH! +10 points.\n");
        score += 10;
        found++;
    } else {
        printf("\nNot a match! -2 points.\n");
        score -= 2;
        if (score < 0)
            score = 0;

        shown[r1][c1] = 0;
        shown[r2][c2] = 0;
    }

    printf("\nPress Enter to continue...");
    getchar();
    clearScreen(); // <-- clears after every try
}

printf("\nLEVEL COMPLETE! Score gained: %d\n", score);
totalScore += score;
printf("Press Enter to continue...");
getchar();
}

```



```
clearScreen();

printf("=====\n");

printf("  GAME COMPLETE! FINAL SCORE\n");

printf("=====\n");

printf("Total Score: %d\n", totalScore);

printf("Thanks for playing!\n");


return 0;

}
```