

Összetett transzformáció

Példa: A képernyő közepéből az origóból induló szakaszt forgassunk egy adott sebességgel az origó körül, a szakasz végéből egy másik szakasz indul, és a csatlakozási pont körül forgassuk egy adott sebességgel.

```
#include <raylib.h>
#include <raymath.h>

int main() {
    const int screenWidth = 800;
    const int screenHeight = 450;
    InitWindow(screenWidth, screenHeight, "Rotating line segments");

    Vector2 origo = { screenWidth / 2.0f, screenHeight / 2.0f };
    float hossz1 = 100.0f;
    float hossz2 = 75.0f;
    float szog1 = 0.0f;
    float szog2 = 0.0f;
    float sebesseg1 = 90.0f; // Fok/perc
    float sebesseg2 = 180.0f; // Fok/perc

    SetTargetFPS(60);

    while (!WindowShouldClose()) {
        szog1 += sebesseg1 * GetFrameTime();
        szog2 += sebesseg2 * GetFrameTime();

        // Első szakasz transzformációja
        Matrix transzformacio1 = MatrixRotateZ(DEG2RAD * szog1);
        Vector2 vegpont1 = Vector2Transform((Vector2){ hossz1, 0.0f },
transzformacio1);
        vegpont1 = Vector2Add(vegpont1, origo);

        // Második szakasz transzformációja
        Matrix transzformacio2 = MatrixMultiply(MatrixRotateZ(DEG2RAD *
szog2), MatrixTranslate(hossz1, 0.0f, 0.0f));
        transzformacio2 = MatrixMultiply(transzformacio2, transzformacio1);
        Vector2 vegpont2 = Vector2Transform((Vector2){ hossz2, 0.0f },
transzformacio2);
        vegpont2 = Vector2Add(vegpont2, origo);

        BeginDrawing();
        ClearBackground(RAYWHITE);

        DrawLineV(origo, vegpont1, RED);
        DrawLineV(vegpont1, vegpont2, BLUE);

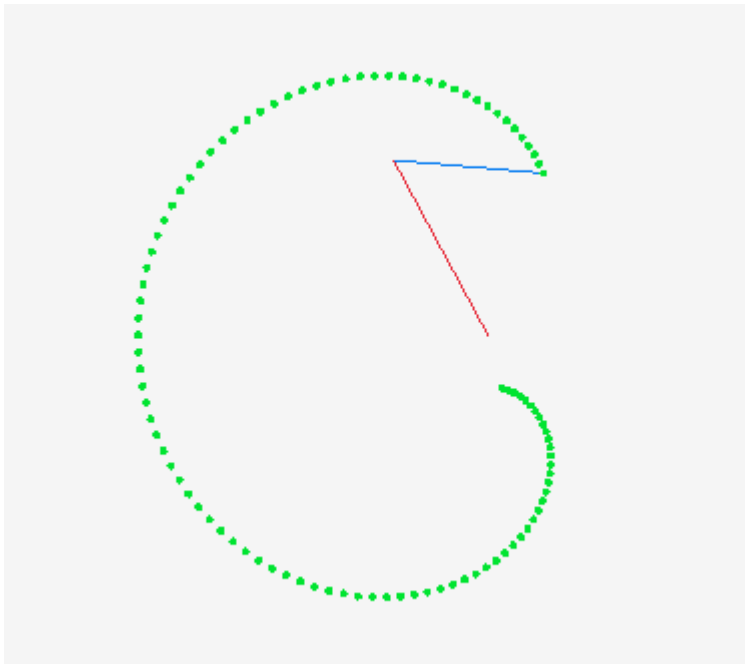
        EndDrawing();
    }
}
```

```
}

CloseWindow();

return 0;
}
```

Feladat: egészítsük ki a fenti példát, hogy előre definiált végpontot is elmentsen egy tömbben. pl. `Vector2 pontok[MAX_POINTS];` Mentsük le a második szakasz végpontjait és `DrawCircleV()` függvénnnyel for ciklusban rajzoljuk ki a trajektóriát.



```
#include <raylib.h>
#include <raymath.h>

#define MAX_POINTS 100 // Max számú pont a trajektóriához

int main() {
    const int screenWidth = 800;
    const int screenHeight = 450;
    InitWindow(screenWidth, screenHeight, "Rotating Segments with Trajectory");

    Vector2 origo = { screenWidth / 2.0f, screenHeight / 2.0f };
    float hossz1 = 100.0f;
    float hossz2 = 75.0f;
    float szog1 = 0.0f;
    float szog2 = 0.0f;
    float sebesseg1 = 90.0f; // Degrees per second
```

```
float sebesseg2 = 180.0f; // Degrees per second

Vector2 pontok[MAX_POINTS];
int pontIndex = 0;

SetTargetFPS(60);

while (!WindowShouldClose()) {
    szog1 += sebesseg1 * GetFrameTime();
    szog2 += sebesseg2 * GetFrameTime();

    Matrix transzformacio1 = MatrixRotateZ(DEG2RAD * szog1);
    Vector2 vegpont1 = Vector2Transform((Vector2){ hossz1, 0.0f },
transzformacio1);
    vegpont1 = Vector2Add(vegpont1, origo);

    Matrix transzformacio2 = MatrixMultiply(MatrixRotateZ(DEG2RAD *
szog2), MatrixTranslate(hossz1, 0.0f, 0.0f));
    transzformacio2 = MatrixMultiply(transzformacio2, transzformacio1);
    Vector2 vegpont2 = Vector2Transform((Vector2){ hossz2, 0.0f },
transzformacio2);
    vegpont2 = Vector2Add(vegpont2, origo);

    // Pontok hozzáadása a tömbhöz
    pontok[pontIndex++] = vegpont2;
    if (pontIndex >= MAX_POINTS) pontIndex = 0;

    BeginDrawing();
    ClearBackground(RAYWHITE);

    DrawLineV(origo, vegpont1, RED);
    DrawLineV(vegpont1, vegpont2, BLUE);

    // Trajektória rajzolása
    for (int i = 0; i < MAX_POINTS; i++) {
        DrawCircleV(pontok[i], 2, GREEN);
    }

    EndDrawing();
}

CloseWindow();

return 0;
}
```

From:
<https://edu.iit.uni-miskolc.hu/> - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:muaszaki_informatika:oesszetett_transzformacio

Last update: **2024/02/28 14:00**

