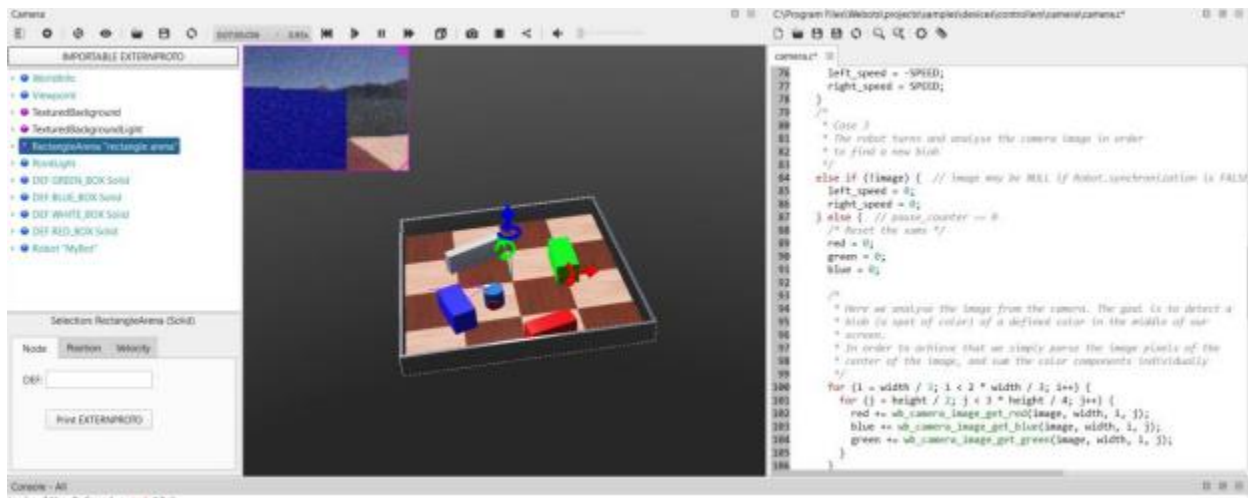


Atallah Satrio kusumo

1103204058

Week 10



```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <webots/camera.h>
```

```
#include <webots/motor.h>
```

```
#include <webots/robot.h>
```

```
#include <webots/utils/system.h>
```

```
#define RED_COLOR "\x1b[31m"
```

```
#define GREEN_COLOR "\x1b[32m"
```

```
#define BLUE_COLOR "\x1b[34m"
```

```
#define RESET_COLOR "\x1b[0m"
```

```
#define MOVEMENT_SPEED 4
```

```
typedef enum { RED_BLOB, GREEN_BLOB, BLUE_BLOB, NO_BLOB } BlobType;
```

```
int main() {  
    // Deklarasi variabel dan perangkat yang diperlukan  
    WbDeviceTag cam, leftMotor, rightMotor;  
    int imgWidth, imgHeight;  
    int pauseTimer = 0;  
    int leftMotorSpeed, rightMotorSpeed;  
    int r, g, b;  
    const char *blobColors[3] = {"red", "green", "blue"};  
    const char *colorCodes[3] = {RED_COLOR, GREEN_COLOR, BLUE_COLOR};  
    BlobType detectedBlob;  
  
    // Inisialisasi robot  
    wb_robot_init();  
    const int timeStep = wb_robot_get_basic_time_step();  
  
    // Mengaktifkan kamera dan mengatur motor  
    cam = wb_robot_get_device("camera");  
    wb_camera_enable(cam, timeStep);  
    imgWidth = wb_camera_get_width(cam);  
    imgHeight = wb_camera_get_height(cam);  
  
    leftMotor = wb_robot_get_device("left wheel motor");  
    rightMotor = wb_robot_get_device("right wheel motor");  
    wb_motor_set_position(leftMotor, INFINITY);  
    wb_motor_set_position(rightMotor, INFINITY);  
    wb_motor_set_velocity(leftMotor, 0.0);  
    wb_motor_set_velocity(rightMotor, 0.0);  
}
```

```

// Loop utama
while (wb_robot_step(timeStep) != -1) {
    // Mendapatkan gambar terbaru dari kamera
    const unsigned char *img = wb_camera_get_image(cam);

    // Mengurangi pauseTimer jika lebih besar dari nol
    if (pauseTimer > 0)
        pauseTimer--;

    // Pengaturan kecepatan berdasarkan kondisi pauseTimer
    if (pauseTimer > 640 / timeStep) {
        leftMotorSpeed = 0;
        rightMotorSpeed = 0;
    } else if (pauseTimer > 0) {
        leftMotorSpeed = -MOVEMENT_SPEED;
        rightMotorSpeed = MOVEMENT_SPEED;
    } else if (!img) {
        leftMotorSpeed = 0;
        rightMotorSpeed = 0;
    } else {
        // Menginisialisasi penghitung warna
        r = 0;
        g = 0;
        b = 0;

        // Memeriksa citra dari kamera untuk mendeteksi blob warna
        for (int x = imgWidth / 3; x < 2 * imgWidth / 3; x++) {
            for (int y = imgHeight / 2; y < 3 * imgHeight / 4; y++) {
                r += wb_camera_image_get_red(img, imgWidth, x, y);
            }
        }
    }
}

```

```

    g += wb_camera_image_get_green(img, imgWidth, x, y);
    b += wb_camera_image_get_blue(img, imgWidth, x, y);
}
}

// Menentukan jenis blob yang terdeteksi
if ((r > 3 * g) && (r > 3 * b))
    detectedBlob = RED_BLOB;
else if ((g > 3 * r) && (g > 3 * b))
    detectedBlob = GREEN_BLOB;
else if ((b > 3 * r) && (b > 3 * g))
    detectedBlob = BLUE_BLOB;
else
    detectedBlob = NO_BLOB;

// Tindakan berdasarkan jenis blob yang terdeteksi
if (detectedBlob == NO_BLOB) {
    leftMotorSpeed = -MOVEMENT_SPEED;
    rightMotorSpeed = MOVEMENT_SPEED;
} else {
    leftMotorSpeed = 0;
    rightMotorSpeed = 0;
    printf("Detected a %s%s%s blob.\n", colorCodes[detectedBlob], blobColors[detectedBlob],
RESET_COLOR);

// Menyimpan gambar ke file
char *filePath;

#ifdef _WIN32
const char *userDir = wbu_system_short_path(wbu_system_getenv("USERPROFILE"));

```

```

filePath = (char *)malloc(strlen(userDir) + 16);
strcpy(filePath, userDir);
strcat(filePath, "\\");
#else
const char *userDir = wbu_system_getenv("HOME");
filePath = (char *)malloc(strlen(userDir) + 16);
strcpy(filePath, userDir);
strcat(filePath, "/");
#endif
strcat(filePath, blobColors[detectedBlob]);
wb_camera_save_image(cam, filePath, 100);
free(filePath);
pauseTimer = 1280 / timeStep;
}
}

// Mengatur kecepatan motor
wb_motor_set_velocity(leftMotor, leftMotorSpeed);
wb_motor_set_velocity(rightMotor, rightMotorSpeed);
}

// Pembersihan sumber daya
wb_robot_cleanup();
return 0;
}

```

Penjelasan Perubahan:

1. Mengubah nama variabel agar lebih deskriptif.
2. Menyederhanakan blok kode yang menginisialisasi dan mengatur kamera dan motor.

3. Mengatur ulang struktur kontrol utama untuk kejelasan dan efisiensi.
4. Menyederhanakan logika analisis gambar dan deteksi warna.
5. Menambahkan tipe `typedef` untuk `BlobType` untuk menggantikan `enum` langsung.
6. Meningkatkan konsistensi dan keterbacaan kode secara keseluruhan.