

UNIwersytet Warmiński-Mazurski

Wydział Matematyki i Informatyki

Kierunek: Informatyka ogólna

**Piotr Bońkowski**

# **Wizualizacja map na potrzeby projektu RoboGuide**

Visualization of building maps for the RoboGuide project

**PRACA MAGISTERSKA**

Wykonana w Katedrze

Metod Matematycznych Informatyki

pod kierunkiem

dra Piotra Artiemjew

Olsztyn 2017

## **Streszczenie**

Celem mojej pracy była wizualizacja wybranego obszaru budynku Wydziału Matematyki i Informatyki, UWM w celu zastosowania w projekcie RoboGuide – dedykowanym do oprowadzania gości po budynku wydziału. Projekt został wykonany przy użyciu silnika graficznego Unity i połączony z implementacją RoboGuide napisaną w języku C++ w bibliotece NXT++.

## **Summary**

The aim of the work was the visualization of the selected area of the building of the Department of Mathematics and Computer Science, University of Warmia and Mazury in Olsztyn. The visualization is a part of the project ROBOGUIDE - dedicated to guide visitors of the building. The project was completed using Unity graphics engine and connected with the implementation of ROBOGUIDE written in C ++ based on library NXT ++.

# Spis treści

1. Wstęp.....	3
2. Opis projektu.....	5
3. Opis implementacji .....	6
4. Funkcjonalności projektu.....	40
5. Finalny wygląd projektu.....	43
6. Testy i analiza.....	76
7. Podsumowanie.....	77
8. Literatura.....	78

# 1. Wstęp

## 1.1 Cel i zakres pracy

Celem pracy było wizualizacja poruszania się robota po wygenerowanej mapie za pomocą danych uzyskanych z projektu RoboGuide.

W skład projektu wchodzi następujące funkcjonalności. Po uruchomieniu programu wybiera się lokalizację plików danych z mapą, plikiem zawierającym aktualną pozycję robota i danymi z radaru osadzonego na robocie. Po naciśnięciu przycisku start program generuje wizualizację mapy. Robot przemieszczając się po mapie aktualizuje swoją pozycję co jest uwzględniane w wizualizacji w sposób ciągły - realtime. Podczas wizualizacji robot może pobrać odległości od przeszkód za pomocą radaru, obraz jest generowany za pomocą figury płaskiej o liczbie wierzchołków zgodnej z liczbą badanych kątów.

Podczas działania można zmienić widok kamery pomiędzy widokiem całej mapy a widokiem dotyczącym obszaru, w którym jest robot.

Program jest w dwóch wariantach. W pierwszym użytkownik sam wybiera lokalizację plików z mapą i celem, w drugiej używa plików na dysku twardym w domyślnie zdefiniowanej lokalizacji.

## 2. Opis projektu

### 2.1 Opis programów współpracujących

**2.1.1 Unity3D** - jest to silnik graficzny do tworzenia gier na różne platformy między innymi PC, PS4, Xbox, Android. Nadaje się też do różnego typu programów graficznych interaktywnych. Program wykorzystuje dwa języki programowania C# oraz Java. Zawiera różne zaawansowane systemy, takie jak: UI (User Interface), dzięki któremu można tworzyć różnego rodzaju panele, w tym też na obiektach, czy też Animator, w którym można łatwo i wygodnie tworzyć powiązania pomiędzy animacjami.

**2.1.2 Projekt Robo-Guide** jest realizowany przez Naukowe Koło Robotyki Wydziału Matematyki i Informatyki, UWM. Celem tego projektu było stworzenie oprogramowania dla robota do oprowadzania po wydziale. W moim projekcie nie zajmowałem się szczegółami działania projektu Robo-Guide a jedynie generowanie wizualizacji mapy z plików danych, które zapisuje robot podczas poruszania się po danym obszarze.

### 3. Opis implementacji

W tej części pracy zawarłem informacje o stworzonych przeze mnie skryptach potrzebnych do działania programu.

#### 1. Opis skryptów (klas),

**Nazwa:** aktywacja\_znaczaika\_celu.cs

**Krótki opis:** Kod wskazuje gdzie aktualnie udaje się robot

**Kod:**

```
using UnityEngine;
using System.Collections;

public class aktywacja_znaczaika_celu : MonoBehaviour {
    public GameObject obrazek;
    public GameObject punkt;
    string cel=null;

    // Update is called once per frame
    void Update () {
        cel = GameObject.Find ("Sterownik").GetComponent<wyznaczanie_celu> ().nazwa_celu;
        obrazek.SetActive (punkt.name == cel);
    }
}
```

**Nazwa:** automatyczne\_przypisanie.cs

**Krótki opis:** Przypisuje domyślne pliki do programu o ile istnieją

**Kod:**

```
using UnityEngine;
using System.Collections;
using System.IO;

public class automatyczne_przypisanie : MonoBehaviour {
    public GameObject pliki;
    public GameObject przetwarzane_dane;
    public GameObject start;
    // Use this for initialization
    void Start () {
        if (File.Exists(@"C:\mapa.txt")&&File.Exists(@"C:\cel.txt"))
        {
            pliki.GetComponent<przypisz_domyslne_pliki> ().przypisz ();
            przetwarzane_dane.GetComponent<pobieranie_z_txt_do_tablic> ().wczytywa
nie_mapy();
            start.GetComponent<uruchamianie> ().do_sceny ();
        }
    }
}
```

```
}  
}
```

**Nazwa:** Czy\_istnieje\_pliki.cs

**Krótki opis:** Sprawdza czy istnieją pliki domyślne

**Kod:**

```
using UnityEngine;  
using UnityEngine.UI;  
using System.Collections;  
using System.IO;  
  
public class Czy_istnieje_pliki : MonoBehaviour {  
    public Text tekst;  
    // Use this for initialization  
    void Start () {  
        if (!(File.Exists(@"C:\mapa.txt") && File.Exists(@"C:\cel.txt")))  
        {  
            gameObject.GetComponent<Button> ().interactable = false;  
            tekst.text = "Brak plików domyślnych";  
        }  
    }  
}
```

**Nazwa:** długość\_x.cs

**Krótki opis:** Wyznacza długość ścian zachodniej i wschodniej oraz umieszcza je wzdłuż terenu.

**Kod:**

```
using UnityEngine;  
  
using System.Collections;  
  
public class długość_x : MonoBehaviour {  
  
    public float z; //zmienna liczbowa  
  
    void Update () {  
  
        GameObject teren = GameObject.Find ("Terrain"); //odnośnik do wyszukiwanego obiektu  
  
        float x = teren.GetComponent<Terrain> ().terrainData.size.x;  
  
        //przypisanie wartości do zmiennej  
  
        gameObject.transform.position= new Vector3(x/2,3f,z); //zmiana wartości parametru  
  
        gameObject.transform.localScale = new Vector3 (x, 6f, 0f); //zmiana wartości parametru
```

```
    }  
}
```

**Nazwa:**długość\_z.cs

**Krótki opis:** Wyznacza długość ścian północnej i południowej oraz umieszcza je wzdłuż terenu.

**Kod:**

```
using UnityEngine;  
  
using System.Collections;  
  
public class długość_z : MonoBehaviour {  
  
    public float x;//zmiennaliczbowa  
  
    void Update () {  
  
        GameObjectteren = GameObject.Find ("Terrain");  
  
        //odnośnik do wyszukiwanego obiektu  
  
        float z = teren.GetComponent<Terrain> ().terrainData.size.z;  
  
        //przypisanie wartości do zmiennej  
  
        gameObject.transform.position= new Vector3(x,3f,z/2);  
  
        //zmiana wartości parametru  
  
        gameObject.transform.localScale = new Vector3 (0f, 6f, z);  
  
        //zmiana wartości parametru  
  
    }  
}
```

**Nazwa:** pobierania\_daych\_do\_radaru .cs

**Krótki opis:** Pobiera z wskazanego pliku txt do tablic danych dotyczących radaru .

**Kod:**

```
using UnityEngine;  
using System.Collections;  
using UnityEngine.UI;
```



```

using System.IO;
using System.Collections.Generic;
public class pobierania_daych_do_radaru : MonoBehaviour {

    public string ścieżka;//zmienna tekstowa
    string ostatni_zapis;//zmienna tekstowa
    public List<string>dane=new List<string>();//odnośnik do obiektu
    public GameObject przycisk;// odnośnik do obiektu UI button
    // Update is called once per frame
    void Update () {
        if (Application.loadedLevel==0) {
            ścieżka=przycisk.GetComponent<wybieranie_pliku>().path;//przypisywanie
wartości
        }

        if (ścieżka.Length!=0) {
            if (ostatni_zapis!=""+File.GetLastWriteTime(ścieżka)) {
                read();//wykonywanie funkcji
                ostatni_zapis=""+File.GetLastWriteTime(ścieżka);//przypisywanie wa
rtości

                GameObject robot=GameObject.FindWithTag("robot");
                if (robot!=null&&dane.Count>0) {
                    robot.GetComponent<radar>().sondowanie(dane);
                }
            }
        }
    }
    void read()
    {
        dane.Clear ();
        string curline; //zmienna tekstowa
        System.IO.StreamReader file = new System.IO.StreamReader(ścieżka);//szczyt
ywanie zawartości wskazanego pliku
        while((curline = file.ReadLine()) != null)
        {
            dane.Add(curline);//przypisywanie wartości
        }
    }
}

```

**Nazwa:**pobieranie\_z\_txt\_do\_tablic.cs

**Krótki opis:** Pobiera z wskazanego pliku txt do tablic danych dotyczących mapy .

**Kod:**

```

using UnityEngine;

using System.Collections;

using System.Collections.Generic;

using System.IO;

using System.Xml;

```

```

using System.Xml.Serialization;

using System.Text;

using UnityEngine.UI;

public class pobieranie_z_txt_do_tablic : MonoBehaviour {

    public string plik; //zmienna tekstowa

    private string _dane; //zmienna tekstowa

    public wzór_mapy WM; //odnośnik do klasy

    public Button przycisk; // odnośnik do obiektu UI button

    public void wczytywanie_mapy(){

        plik = przycisk.GetComponent<wybieranie_pliku> ().path;

//przypisywanie wartości do zmiennej

        odczyt_z_XML (); //wykonanie funkcji

        WM = (wzór_mapy)deserializacja (_dane);

//przypisanie wartości do danych klasy

    }

    void odczyt_z_XML()

    {

        StreamReader reader = File.OpenText(plik); //odczytuje zawartość pliku

        string zawartość_pliku = reader.ReadToEnd(); //przypisuje ciąg znaków z pliku

        reader.Close (); //wyłączenie

        _dane = zawartość_pliku; //przypisuje ciąg znaków

    }

    object deserializacja(string dane)

    {

        XmlSerializer serializerxml = new XmlSerializer (typeof( wzór_mapy ));

//przypisuje wzór zapisu

        MemoryStream pamięć = new MemoryStream (do_tablicy(dane)); //przypisuje dane

        XmlTextWriter writer = new XmlTextWriter (pamięć, Encoding.UTF8); //przetwarzane

        return serializerxml.Deserialize (pamięć); //zwraca przetworzone dane
    }
}

```

```

    }

    byte[] do_tablicy(string dane)
    {
        UTF8Encoding kodowanie = new UTF8Encoding (); //kodowane znaków
        byte[] tablica = kodowanie.GetBytes (dane); //przypisuje dane do tablicy
        return tablica; //zwraca tablicę
    }
}

```

### **Nazwa: poruszanie\_robotem.cs**

**Krótki opis:** Przemieszcza obiekt robot po mapie jeśli istnieje cel.

#### **Kod:**

```

using UnityEngine;

using System.Collections;

public class poruszanie_robotem : MonoBehaviour {

    public GameObject cel; //odnośnik do obiektu

    NavMeshAgent agent; //odnośnik do komponentu

    void Start () {

        agent = gameObject.GetComponent<NavMeshAgent> (); //przypisanie komponentu
    }

    void Update () {

        if (cel != null) {

            agent.SetDestination(cel.transform.position); //uruchomienie funkcji
        }
    }
}

```

**Nazwa:pozycja\_x.cs**

**Krótki opis:** Przesuwa ścianę na południowy skraj mapy.

**Kod:**

```
using UnityEngine;

using System.Collections;

public class pozycja_x : MonoBehaviour {

    void Update () {

        GameObject teren = GameObject.Find ("Terrain");//odnośnik do wyszukiwanego obiektu

        gameObject.GetComponent<długość_z> ().x = teren.GetComponent<Terrain>
().terrainData.size.x;//zmiana wartości parametru

    }

}
```

**Nazwa:pozycja\_z.cs**

**Krótki opis:** Przesuwa ścianę na wschodu skraj mapy.

**Kod:**

```
using UnityEngine;

using System.Collections;

public class pozycja_z : MonoBehaviour {

    void Update () {

        GameObject teren = GameObject.Find ("Terrain");//odnośnik do wyszukiwanego obiektu

        gameObject.GetComponent<długość_x> ().z = teren.GetComponent<Terrain>
().terrainData.size.z; //zmiana wartości parametru

    }

}
```

**Nazwa:przetwarzanie\_pliku\_txt.cs**

**Krótki opis:** Odczytuje cel dla robota z pliku txt a potem jeśli nastąpiła zmiana ostatniej edycji pliku txt ponownie odczytuje cel.

**Kod:**

```
using UnityEngine;

using System.Collections;

using UnityEngine.UI;

using System.IO;

public classprzetwarzanie_pliku_txt : MonoBehaviour {

    public string ścieżka;//zmienna tekstowa

    string ostatni_zapis;//zmienna tekstowa

    public string cel;//odnośnik do obiektu

    public GameObject przycisk;// odnośnik do obiektu UI button

    void Update () {

        if (Application.loadedLevel==0) {

            ścieżka=przycisk.GetComponent<wybieranie_pliku>().path;//przypisywanie wartości

        }

        if (ścieżka.Length!=0) {

            if (ostatni_zapis!=""+File.GetLastWriteTime(ścieżka)) {

                read();//wykonywanie funkcji

                ostatni_zapis=""+File.GetLastWriteTime(ścieżka);//przypisywanie wartości

            }

        }

    }

    voidread()

    {

        string curline; //zmienna tekstowa
```

```

        System.IO.StreamReader file = new System.IO.StreamReader(ścieżka);//wczytywanie
zawartości wskazanego pliku

        while((curline = file.ReadLine()) != null)

        {

            cel = curline;//przypisywanie wartości

        }

    }
}

```

### **Nazwa:przełącznik.cs**

**Krótki opis:** Zmienia wartość przełącznik (switch) do skryptu sterowanie kamerą.

#### **Kod:**

```

using UnityEngine;

using System.Collections;

public class przełącznik : MonoBehaviour {

    public void przypisanie(int wartość){

        GameObject.Find ("MainCamera").GetComponent<sterowanie_kamerą> ().przełącznik =
wartość; //przypisywanie wartości

    }

}

```

### **Nazwa: przypisz\_domyslny\_plik .cs**

**Krótki opis:** przypisuje pliki domyślne

#### **Kod:**

```

using UnityEngine;
using System.Collections;

public class przypisz_domyslny_plik : MonoBehaviour {

    public GameObject mapa;
    public GameObject cel;

    public void przypisz()
    {

```

```

        mapa.GetComponent<wybieranie_pliku>().path = @"C:\mapa.txt";
        cel.GetComponent<wybieranie_pliku>().path = @"C:\cel.txt";
    }
}

```

**Nazwa:** radar.cs

**Krótki opis:** Wizualizuje radar

**Kod:**

```
using UnityEngine;
```

```
using System.Collections;
```

```
using System.Collections.Generic;
```

```
public class radar : MonoBehaviour {
```

```
    // Use this for initialization
```

```
    public List<GameObject>obiekty=new List<GameObject>();
```

```
    public GameObject prefabrykant;
```

```
    private GameObject robot;
```

```
    public void sondowanie(List<string> dane){
```

```
        float kont = 0;
```

```
        GameObject obiekt=gameObject;
```

```
        float a1=0;
```

```
        float b1 = 0;
```

```
        foreach (var item in obiekty)
```

```
        {
```

```
            Destroy(item);
```

```
        }
```

```
        foreach (var item in dane) {
```

```
Quaternion obrot=new  
Quaternion(gameObject.transform.rotation.x,gameObject.transform.rotation.y,gameObject.  
transform.rotation.z,gameObject.transform.rotation.w);
```

```
string[] podzial=item.Split(':');  
  
int dystans=int.Parse(podzial[1]);  
  
kont=int.Parse(podzial[0]);  
  
float a=Mathf.Sin(kont)*dystans;  
  
float b=Mathf.Cos(kont)*dystans;  
  
if (kont!=0) {  
    obiekt=  
Instantiate(prefabrykant,gameObject.transform.position,obrot) as GameObject;  
  
    objekty.Add(obiekt);  
  
    obiekt.AddComponent<MeshFilter> ();  
  
    obiekt.AddComponent<MeshRenderer> ();  
  
    Mesh mesh =obiekt.GetComponent<MeshFilter> ().mesh;  
  
    mesh.Clear ();  
  
    mesh.vertices = new Vector3[] {new Vector3 (0, 1, 0), new  
Vector3 (a1, 1, b1), new Vector3 (a, 1, b)};  
  
    mesh.uv = new Vector2[] {new Vector2 (0, 0), new Vector2 (0,  
1), new Vector2 (1, 1)};  
  
    mesh.triangles = new int[] {0, 1, 2};  
  
}  
  
a1=a;  
  
b1=b;  
  
}
```



```

    }

    public Vector3 pozycja_robota=new Vector3(0,0,0);

    public Vector3 w1;

    void Update (){

        robot = GameObject.FindGameObjectWithTag ("robot");

        if (robot!=null) {

            if(pozycja_robota.x!=robot.transform.position.x
|| pozycja_robota.z!=robot.transform.position.z){

                foreach (var item in objekty) {

                    Destroy(item);

                }

            }

            pozycja_robota=robot.transform.position;

        }

    }

}

```

**Nazwa: sterowanie\_kamerą.cs**

**Krótki opis:** Odpowiada za sterowanie kamerą w zależności od wybranego trybu:

1. (domyślny) widok na całą mapę
2. widok wycelowany na robota z możliwością zmiany wysokości.

Kod:

```
using UnityEngine;
using System.Collections;

public class sterowanie_kamera : MonoBehaviour {

    public GameObject teren;//odnośnik do obiektu

    private RectTransform ramka;//odnośnik do obiektu

    private float maxY;//zmienna liczbowa

    private float maxX;//zmienna liczbowa

    public int przełącznik;//zmienna liczbowa

    public float prętkość_oddalania;//zmienna liczbowa

    private float oddalenie;//zmienna liczbowa

    void Update () {

        float x=0f;//przypisywanie wartości

        float y=0f;//przypisywanie wartości

        float z=0f;//przypisywanie wartości

        switch (przełącznik) {

            case 1:

                x = teren.GetComponent<Terrain> ().terrainData.size.x/2;//przypisywanie wartości

                z = teren.GetComponent<Terrain> ().terrainData.size.z/2;//przypisywanie wartości

                y = teren.GetComponent<Terrain> ().terrainData.size.x;//przypisywanie wartości

                oddalenie=0f;//przypisywanie wartości

                break;

            case 2:

                if (GameObject.Find("robot")!=null) {

                    GameObject robot=GameObject.Find("robot");

                    //przypisywanie wyszukiwanego obiektu

                    x=robot.transform.position.x;//przypisywanie wartości

                    z=robot.transform.position.z;//przypisywanie wartości

                    if (Input.GetAxis("Mouse ScrollWheel")>0 &&oddalenie>-34f)
```

```

        {
            oddalenie-=prętkość_oddalania;//przypisywanie wartości
        }
        if (Input.GetAxis("Mouse ScrollWheel")<0)
        {
            oddalenie+=prętkość_oddalania;//przypisywanie wartości
        }
        if (oddalenie<-34f) {
            oddalenie=0f;//przypisywanie wartości
        }
        y=40f+oddalenie;//przypisywanie wartości
    }
    else {
        przełącznik=1;//przypisywanie wartości
    }
    break;
}

gameObject.transform.position = new Vector3 (x, y, z);

//przypisywanie wartości
    }
}

```

### **Nazwa: uruchamianie**

**Krótki opis:** Sprawcza czy są wybrane ścieżki do plików. Gdy wynik jest true to odblokowuje przycisk. Po kliknięciu przechodzi do sceny z mapą.

### **Kod:**

```

using UnityEngine;

using UnityEngine.UI;

```

```

using System.Collections;

public class uruchamianie : MonoBehaviour {

    public GameObject obiekt;//odnośnik do obiektu

    public Button przycisk_ścieżki;// odnośnik do obiektu UI button

    public Button przycisk_mapy;// odnośnik do obiektu UI button

    public void do_sceny()

    {

        DontDestroyOnLoad (obiekt);

        //aktywowanie funkcji ktura sprawia że nie usuwa się wskazany obiekt przy przejściu do nowej sceny

        Application.LoadLevel ("mapa");//przejście do wybranej sceny

    }

    void Update () {

        if (przycisk_ścieżki.GetComponent<wybieranie_pliku>().path.Length!=0
        &&przycisk_mapy.GetComponent<wybieranie_pliku>().path.Length!=0) {

            gameObject.GetComponent<Button>().interactable=true; //przypisywanie wartości

        }

        else {

            gameObject.GetComponent<Button>().interactable=false;//przypisywanie wartości

        }

    }

}

```

**Nazwa:**ustawianie\_mapy.cs

**Krótki opis:** Przypisuje pobrane wcześniej dane z pliku txt do obiektu terrain.

**Kod:**

```

using UnityEngine;

```

```

using System.Collections;

using System.Collections.Generic;

public class ustawianie_mapy : MonoBehaviour

{

    boolczy_wygenerować = true;//zmienna typu bool i przypisanie wartości

    GameObject przetwornik;//oddnośnik do obiektu

    public GameObjectprefab_pola;//oddnośnik do obiektu


    void Update ()

    {

        if (czy_wygenerować) {

            przetwornik = GameObject.Find ("przetwarzanie_pliku_do_listy");//przypisanie
znalezionego obiektu

            if (przetwornik != null) {

                wzór_mapy WM =
przetwornik.GetComponent<pobieranie_z_txt_do_tablic> ().WM;//zmienna wzór_mapy i przypisanie wartości

                double[,] north = newdouble[WM.ilość_punktów_x,
WM.ilość_punktów_y];//tablica liczbowa

                double[,] south = newdouble[WM.ilość_punktów_x,
WM.ilość_punktów_y];//tablica liczbowa

                double[,] east = newdouble[WM.ilość_punktów_x,
WM.ilość_punktów_y];//tablica liczbowa

                double[,] west = newdouble[WM.ilość_punktów_x,
WM.ilość_punktów_y];//tablica liczbowa

                double[,] obstacles = newdouble[WM.ilość_punktów_x,
WM.ilość_punktów_y];//tablica liczbowa

                double x = 0;//przypisywanie wartości

                double y = 0;//przypisywanie wartości

                north = zwróć_tablice (WM.north);// przypisanie wartości za pomocą funkcji

                south = zwróć_tablice (WM.south);//przypisanie wartości za pomocą funkcji

                east = zwróć_tablice (WM.east);//przypisanie wartości za pomocą funkcji

                west = zwróć_tablice (WM.west);//przypisanie wartości za pomocą funkcji

```

funkcji

```
obstacles = zwróć_tablice (WM.obstacles);//przypisanie wartości za pomocą

for (int i = 0; i <WM.ilość_punktów_x; i++) {

    x += north [i, 0];//przypisywanie wartości

    x += south [i, 0];//przypisywanie wartości

}

for (int i = 0; i <WM.ilość_punktów_y; i++) {

    y += east [0, i];//przypisywanie wartości

    y += west [0, i];//przypisywanie wartości

}

gameObject.GetComponent<Terrain> ().terrainData.size = new Vector3

((float)x, 10f, (float)y);

x = 0;//przypisywanie wartości

y = 0;//przypisywanie wartości

GameObject punkt;//odnośnik do obiektu

for (int i = 0; i <WM.ilość_punktów_x; i++) {

    for (int j = 0; j <WM.ilość_punktów_y; j++) {

        if (i==0) {

            x+=north[i,j];//przypisywanie wartości

        }

        else {

            for (int k = 0; k <i; k++) {

                x+=north[k,j];//przypisywanie wartości

                x+=south[k,j];//przypisywanie wartości

            }

            x+=north[i,j];//przypisywanie wartości

        }

        y+=west[i,j];

        punkt=Instantiate(prefab_pola,new

Vector3((float)x,0f,(float)y),prefab_pola.transform.rotation)as GameObject;
```

```

//dodanie nowego obiektu i przypisywanie obiektu

        punkt.name=(i+1)+","+(j+1);//przypisywanie wartości

        punkt.transform.GetChild(0).gameObject.transform.localScale=new
Vector3((float)(north[i,j]+south[i,j]),10f,(float)(east[i,j]+west[i,j]));//przypisywanie wartości
        punkt.transform.GetChild(0).gameObject.SetActive(obstacles[i,j]==1);
//przypisywanie wartości

        y+=east[i,j];//przypisywanie wartości

        x=0;//przypisywanie wartości

    }

    y=0;//przypisywanie wartości

}

czy_wygenerować = false;//przypisywanie wartości

}

}

private double[,] zwróć_tablice (List<string>lista)
{

    wzór_mapy WM = przetwornik.GetComponent<pobieranie_z_txt_do_tablic>
().WM;//przypisywanie wartości

    double[,] tablica = new double[WM.ilość_punktów_x, WM.ilość_punktów_y];

//tablicaliczbowa

    foreach (var item in lista) {

        string[] podział = item.Split ('=');

//tablica tekstowa i przybisanie podzielonej zmiennej

        string[] podział1 = podział [0].Split (',');

//tablica tekstowa i przybisanie podzielonej zmiennej

        int x = int.Parse (podział1 [0]);//zmienna liczbowa i przydzielenie wartości

        int y = int.Parse (podział1 [1]);//zmienna liczbowa i przydzielenie wartości

        if (podział[1].Contains(",")) {

            podział1 = podział [1].Split (',');// przybisanie podzielonej zmiennej

            podział[1]=podział1[0]+"."+podział1[1];//przypisanie wartości

```

```

        }

        tablica [x - 1, y - 1] = double.Parse (podział [1]); //przypisanie wartości

    }

    return tablica; //zwrót wartości

}

}

```

### **Nazwa: wybieranie\_pliku**

**Krótki opis:** Wyświetla wybraną ścieżkę do pliku.

Kod:

```
using UnityEngine;
```

```
using System.Collections;
```

```
using UnityEngine.UI;
```

```

public class wybieranie_pliku : MonoBehaviour {

    public Text text; //odnośnik do obiektu UI text

    public string path; //zmienna tekstowa

    public GameObject zapis; //odnośnik do obiektu

    public Button przycisk_zatwierdzenia; // odnośnik do obiektu UI button

    public void przypisanie_obiektu(){

        przycisk_zatwierdzenia.GetComponent<wybierz_plik>().obiekt_potrzebujący_pliku=gameObject; //prz
upisanie obiektu

    }

    void Update () {

        text.text = path; //przypisanie wartości

    }

}

```



**Nazwa:**wyjście\_z\_programu.cs

**Krótki opis:** Wyłącza program.

**Kod:**

```
using UnityEngine;

using System.Collections;

public class wyjście_z_programu : MonoBehaviour {

    public void wyjście(){

        Application.Quit ();//wyjście z aplikacji

    }

}
```

**Nazwa:** wyznaczanie\_celu.cs

**Krótki opis:** Oczekuje na cel dla robota i w przypadku pierwszego celu tworzy w wskazanym miejscu obiekt robota a później przekazuje mu nowy cel do którego ma się udać.

**Kod:**

```
using UnityEngine;

using System.Collections;

public classwyznaczanie_celu : MonoBehaviour {

    GameObjectprzetearzanie_pliku_do_listy;//odnośnik do obiektu

    string nazwa_celu;//zmienna tekstowa

    GameObject robot;//odnośnik do obiektu

    public GameObjectprefab;//odnośnik do obiektu

    void Update () {

        if (GameObject.Find("przetearzanie_pliku_do_listy")) {

            przetearzanie_pliku_do_listy=GameObject.Find("przetearzanie_pliku_do_listy");//przypisanie obiektu

        }

    }

}
```

```

        nazwa_celu=przetwarzanie_pliku_do_listy.GetComponent<przetwarzanie_pliku_txt>().cel;//przypisanie
        e wartości

        if (nazwa_celu.Length!=0) {

            if (GameObject.FindWithTag("robot")==null) {

                if (GameObject.Find(nazwa_celu)!=null) {

                    robot=Instantiate(prefab,GameObject.Find(nazwa_celu).transform.position,
                    GameObject.Find(nazwa_celu).transform.rotation) as GameObject;

                    //dodanie nowego obiektu i przypisanie obiektu

                }

            }

            else {

                if (GameObject.Find(nazwa_celu)!=null) {

                    GameObject
cel=GameObject.Find(nazwa_celu);//odnośnik do obiektu i przypisanie wartości

                    robot.GetComponent<poruszanie_robotem>().cel=cel;//przypisanie
obektu

                }

            }

        }

    }

}

```

**Nazwa: wzór\_mapy.cs**

**Krótki opis:** Klasa z danymi do mapy.

**Kod:**

```
using UnityEngine;
```

```
using System.Collections;
```

```
using System.Collections.Generic;
```

```
public class wzór_mapy {
```

```

public int ilość_punktów_x;//zmienna liczbowa

public int ilość_punktów_y;//zmienna liczbowa

public List<string>north = new List<string> (); //listatekstowa

public List<string>south = new List<string> ();//listatekstowa

public List<string>east = new List<string> ();//listatekstowa

public List<string>west = new List<string> ();//listatekstowa

public List<string>obstacles = new List<string> ();//listatekstowa

}

```

## Skrypty do przeglądania plików

**Nazwa:** do\_poprzedniego

**Krótki opis:** Skraca ścieżkę do pliku o jeden poziom.

**Kod:**

```
using UnityEngine;
```

```
using System.Collections;
```

```

public class do_poprzedniego : MonoBehaviour {

    public GameObject okno;//odnośnik do obiektu

    public string[] w;//tablica tekstu

    public string s;//zmienna tekstowa

    public string poprzednia;//zmienna tekstowa

    public void poprzedni(){

        string[] podzielona_ścieżka= okno.GetComponent<przeglądanie_plików>
().obecne_położenie.Split ("\\");//tablica tekstowa i przypisanie podzielonego tekstu

        w = podzielona_ścieżka;//przypisanie wartości

        if(podzielona_ścieżka.Length>1) {

```

```

string nowa_ścieżka=null;//zmienna tekstowa
for (int i = 0; i < podzielona_ścieżka.Length-1; i++) {
    if (i==podzielona_ścieżka.Length-3) {
        if (podzielona_ścieżka[i+1].Contains(".txt")) {
            nowa_ścieżka+=podzielona_ścieżka[i];

//przypisanie wartości

            if (podzielona_ścieżka.Length==2) {
                nowa_ścieżka+="\\";

//przypisanie wartości

            }
            break;
        }
    }

    if (i==podzielona_ścieżka.Length-2) {
        nowa_ścieżka+=podzielona_ścieżka[i];

//przypisanie wartości

        if (podzielona_ścieżka.Length==2) {
            nowa_ścieżka+="\\";//przypisanie wartości

        }
    }
    else {
        nowa_ścieżka+=podzielona_ścieżka[i]+"\\";

//przypisanie wartości

    }
}

s=nowa_ścieżka;//przypisanie wartości

```

```

        okno.GetComponent<przeglądanie_plików> ().obecne_położenie
=nowa_ścieżka;//przypisanie wartości

        if (poprzednia!=nowa_ścieżka) {

            okno.GetComponent<przeglądanie_plików>().lista_plików();//wykonanie funkcji

        }

        poprzednia=nowa_ścieżka;//przypisanie wartości

    }

}

```

### **Nazwa: otwieranie\_wyszukiwania.cs**

**Krótki opis:** Skrypt otwiera okno przeglądania plików oraz wskazuje startową ścieżkę wyszukiwania.

### **Kod:**

```

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

public class otwieranie_wyszukiwania : MonoBehaviour {

    public GameObjectokno;//odnośnik do obiektu

    public void otwieranie()

    {

        okno.SetActive (true);//zmiana wartości parametru typu bool

        okno.GetComponent<przeglądanie_plików> ().obecne_położenie = @"C:\Users";//przypisanie
wartości

        okno.GetComponent<przeglądanie_plików> ().lista_plików();//wykonanie funkcji

    }

}

```

**Nazwa:**przeglądanie\_plików.cs

**Krótki opis:** Skrypt wyświetla listę folderów oraz plików txt w danym folderze.

**Kod:**

```
using UnityEngine;

using System.Collections;

using UnityEngine.UI;

using System.IO;

using System.Collections.Generic;

public class przeglądanie_plików : MonoBehaviour {

    public Button przycisk_startowy;//odnośnik do obiektu UI typu button

    public string obecne_położenie;//zmienna tekstowa

    public List<string>pliki=new List<string>();//listatypu string

    public List<string>wyświetlane_pliki = new List<string> ();//listatypu string

    public Button prefab_folderu;//odnośnik do obiektu UI typu button

    public GameObject panel;//odnośnik do obiektu

    public voidlista_plików()

    {

        if (obecne_położenie!="") {

            pliki.Clear();//czyszczenielisty

            foreach (string file in Directory.GetDirectories(obecne_położenie))

            {

                pliki.Add(file);//dodanie wartości do listy

            }

            foreach (string file in Directory.GetFiles(obecne_położenie)) {

                if (file.Contains(".txt")) {

                    pliki.Add(file);//dodanie wartości do listy

                }

            }

        }

    }

}
```

```

    }

    float i=0;//zmienna liczbowa i przypisanie wartości

    int j=0;//zmienna liczbowa i przypisanie wartości

    foreach (string item in pliki) {

        Button przycisk= Instantiate(prefab_folderu) as Button;

//dodanie obiektu i przypisanie wartości

        przycisk.GetComponent<RectTransform>().parent=panel.GetComponent<RectTransform>();
//przypisanie komponentu

        i=i+30f;//przypisanie wartości

        przycisk.GetComponent<RectTransform>().anchoredPosition=new
Vector2(0f,przycisk_startowy.GetComponent<RectTransform>().anchoredPosition.y-i);//przypisanie wartości

        przycisk.GetComponent<RectTransform>().sizeDelta=new
Vector2(0f,przycisk_startowy.GetComponent<RectTransform>().sizeDelta.y);//przypisanie wartości

        przycisk.GetComponent<przejście_do_folderu>().ścieżka_do_folderu=item;//przypisanie wartości

        string[]podzielona_ścieżka=item.Split("\\");

//tablica tekstu i przypisanie podzielonego tekstu

        przycisk.GetComponent<przejście_do_folderu>().nazwa_folderu=podzielona_ścieżka[podzielona_ścież
ka.Length-1];//przypisanie wartości
        przycisk.GetComponent<przejście_do_folderu>().przycisk_do_poprzedniego=przycisk_startowy;//przy
pisanie wartości

        j+=1;//przypisanie wartości

        if (j>10) {

            panel.GetComponent<RectTransform>().offsetMin=new
Vector2(0f,-(j-10)*30f);//przypisaniewartości

        }

        else {

            panel.GetComponent<RectTransform>().offsetMin=new
Vector2(0f,0f);//przypisaniewartości

        }

    }

}

```

```
}  
  
}
```

**Nazwa: przejście\_do\_folderu.cs**

**Krótki opis:** Skrypt w przypadku folderu przechodzi o jeden poziom do głębiej dodając do ścieżki a w przypadku pliku txt zostaje wybrany plik.

**Kod:**

```
using UnityEngine;  
  
using System.Collections;  
  
using UnityEngine.UI;  
  
public class przejście_do_folderu : MonoBehaviour {  
  
    public string ścieżka_do_folderu;//zmienna tekstowa  
  
    public string nazwa_folderu;//zmienna tekstowa  
  
    public Button przycisk_do_poprzedniego;//odnośnik do obiektu UI typu button  
  
    public Text tekst;//odnośnik do obiektu UI typu tekst  
  
    public void od_folderu(){  
  
        GameObjectokno = GameObject.Find ("Panel_wyszukiwania");  
  
        //odnośnik do obiektu i przypisanie do wartości  
  
        okno.GetComponent<przeglądanie_plików> ().obecne_położenie =  
        ścieżka_do_folderu;//przypisanie wartości  
  
        if (!ścieżka_do_folderu.Contains(".txt")) {  
  
            okno.GetComponent<przeglądanie_plików> ().lista_plików ();//wykonanie funk  
  
            przycisk_do_poprzedniego.GetComponent<do_poprzedniego>().poprzednia=ścieżka_do_folderu;//prz  
            ypisanie wartości  
  
        }  
  
    }  
  
    void Update () {  
  
        tekst.text = nazwa_folderu;//przypisanie wartości  
  
        GameObjectokno = GameObject.Find ("Panel_wyszukiwania");
```



```
//odnośnik do obiektu i przypisanie wyszukanego obiektu

        if (!okno.GetComponent<przeglądanie_plików>().pliki.Contains(ścieżka_do_folderu)) {

                Destroy(gameObject); //usuwanie wskazanego obiektu

        }

    }

}
```

### **Nazwa:wybierz\_plik**

**Krótki opis:** Skrypt przydziela wybrany plik.

#### **Kod:**

```
using UnityEngine;

using System.Collections;

public class wybierz_plik : MonoBehaviour {

    public GameObject obiekt_potrzebujący_pliku; //odnośnik do obiektu

    public GameObject panel; //odnośnik do obiektu

    public void przypisanie_pliku(){

        obiekt_potrzebujący_pliku.GetComponent<wybieranie_pliku>().path =
panel.GetComponent<przeglądanie_plików>().obecne_położenie; //przypisanie wartości

        panel.SetActive (false); //zmiana wartości parametru typu bool

    }

}
```

### **Nazwa:Zamykanie\_okna**

**Krótki opis:** Skrypt zamyka okno wyszukiwania.

#### **Kod:**

```
using UnityEngine;

using System.Collections;

public class Zamykanie_okna : MonoBehaviour {

    public GameObject okno; //odnośnik do obiektu
```

```

        public void zamykanie()
        {
            okno.SetActive (false); //zmiana wartości parametru typu bool
        }
    }
}

```

## Programy zewnętrzne

**Nazwa:** generator\_celu

**Krótki opis:** Program na podstawie mapy generuje wybraną ilość celów, jeden co określony czas.

**Kod:**

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Xml;
using System.Xml.Serialization;
using System.Threading;

namespace generator_celu
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public string plik; //zmienna tekstowa
        private string _dane; //zmienna tekstowa
        public wzór_mapy WM; //odnośnik do klasy

        private void button1_Click(object sender, EventArgs e)
        {
            plik = textBox1.Text;
            int ilość_przesunięć = Int32.Parse( textBox2.Text);
            bool czy_powtorzyc=true;
            int x = 0;
            int y = 0;
            textBox3.Text = "";
            if (File.Exists(plik)&&File.Exists("cel.txt"))
            {
                odczyt_z_XML();//wykonanie funkcji
                WM = (wzór_mapy)deserializacja(_dane); //przypisanie wartości do
danych klasy
                int max_x = WM.ilość_punktów_x;
                int max_y = WM.ilość_punktów_y;
                Random rndx = new Random();
                Random rndy = new Random();
            }
        }
    }
}

```

```

List<string> pola =new List<string> ( WM.obstacles);
int aktualny_x = Int32.Parse(textBox4.Text);
int aktualny_y = Int32.Parse(textBox5.Text);
textBox3.Text += "" + x + "," + y + " \n";
File.WriteAllText("cel.txt", "" + x + "," + y);
for (int i = 0; i < ilość_przesunięć; i++)
{
    while (czy_powtorzyc)
    {
        int rx = rndx.Next(0, 4);
        int ry = rndx.Next(0, 4);
        x = aktualny_x+rx- 2;
        y = aktualny_y +ry - 2;
        if (x > 0 && x <= max_x && y > 0 && y <= max_y)
        {
            string pole = "" + x + "," + y + "=0";
            if
(pla.Contains(pole)&&!(aktualny_x==x&&aktualny_y==y))
            {
                czy_powtorzyc = false;
                aktualny_x = x;
                aktualny_y = y;
                textBox3.Text += "" + x + "," + y+" \n";
                File.WriteAllText("cel.txt", "" + x + "," + y);
            }
        }
        Thread.Sleep(2000);
        czy_powtorzyc = true;
    }
}
else
{
    textBox3.Text = "Brak plików";
}
}

void odczyt_z_XML()
{
    StreamReader reader = File.OpenText(plik);//odczytuje zawartość pliku
    string zawartość_pliku = reader.ReadToEnd();//przypisuje ciąg znaków z
    pliku
    reader.Close();//wyłączenie
    _dane = zawartość_pliku;//przypisuje ciąg znaków
}
object deserializacja(string dane)
{
    XmlSerializer serializerxml = new
    XmlSerializer(typeof(wzór_mapy));//przypisuje wzór zapisu
    MemoryStream pamięć = new MemoryStream(do_tablicy(dane));//przypisuje
    dane
    XmlTextWriter writer = new XmlTextWriter(pamięć,
    Encoding.UTF8);//przetwarza dane
    return serializerxml.Deserialize(pamięć);//zwraca przetworzone dane
}
byte[] do_tablicy(string dane)

```

```

        {
            UTF8Encoding kodowanie = new UTF8Encoding();//kodowane znaków
            byte[] tablica = kodowanie.GetBytes(dane);//przypisuje dane do tablicy
            return tablica;//zwraca tablicę
        }
    }
}

```

**Nazwa:** generowanie\_mapy

**Krótki opis:** Program przerabia plik bmf z danymi mapy na txt

**Kod:**

```

using System;
using System.IO;
using System.Xml;
using System.Xml.Serialization;
using System.Text;

namespace generowanie_mapy
{
    class Program
    {

        public string lokalizacja_zapisu;
        private string dane_do_string;
        void stwórz()
        {
            lokalizacja_zapisu = @"C:/mapy/D-West.txt";
            wzór_mapy _obiekt = new wzór_mapy();
            string lokalizacja_odczytu = @"C:/mapy/D1-West.bmf";
            string zawartosc = File.ReadAllText(lokalizacja_odczytu);
            string[] dzielniki = { "=", "(", ")=", "\r\n", " " };
            string[] podzial =
            zawartosc.Split(dzielniki,StringSplitOptions.RemoveEmptyEntries);

            _obiekt.ilosc_punktów_x =Int32.Parse(podzial[1]);
            _obiekt.ilosc_punktów_y =Int32.Parse(podzial[3]);
            bool poczatek = false;
            bool poczatek_N = false;
            bool poczatek_S = false;
            bool poczatek_W = false;
            bool poczatek_E= false;
            bool poczatek_O = false;
            for (int i = 4; i <podzial.Length-1; i++)
            {
                if (podzial[i].Contains("FIELDS_SIZE"))
                {
                    poczatek = true;
                }
                if (poczatek)
                {
                    if (podzial[i].Contains("NORTH") &&
!podzial[i].Contains("NORTH_DIRECTION"))

```

```

{
    i += 1;
    poczatek_N = true;
    poczatek_S = false;
    poczatek_W = false;
    poczatek_E = false;
    poczatek_O = false;
}
if (podzial[i].Contains("EAST"))
{
    i += 1;
    poczatek_N = false;
    poczatek_S = false;
    poczatek_W = false;
    poczatek_E = true;
    poczatek_O = false;
}
if (podzial[i].Contains("SOUTH"))
{
    i += 1;
    poczatek_N = false;
    poczatek_S = true;
    poczatek_W = false;
    poczatek_E = false;
    poczatek_O = false;
}
if (podzial[i].Contains("WEST"))
{
    i += 1;
    poczatek_N = false;
    poczatek_S = false;
    poczatek_W = true;
    poczatek_E = false;
    poczatek_O = false;
}
if (podzial[i].Contains("OBSTACLES"))
{
    i += 1;
    poczatek_N = false;
    poczatek_S = false;
    poczatek_W = false;
    poczatek_E = false;
    poczatek_O = true;
}
if (podzial[i].Contains("NORTH_DIRECTION"))
{
    poczatek_N = false;
    poczatek_S = false;
    poczatek_W = false;
    poczatek_E = false;
    poczatek_O = false;
}
}
if (poczatek_N)
{
    _obiekt.north.Add(podzial[i]+"="+podzial[i+1]);
    i += 1;
}
if (poczatek_S)

```

```

        {
            _obiekt.south.Add(podzial[i] + "=" + podzial[i + 1]);
            i += 1;
        }
        if (poczatek_E)
        {
            _obiekt.east.Add(podzial[i] + "=" + podzial[i + 1]);
            i += 1;
        }
        if (poczatek_W)
        {
            _obiekt.west.Add(podzial[i] + "=" + podzial[i + 1]);
            i += 1;
        }
        if (poczatek_O)
        {
            if (podzial[i+1]=="0")
            {
                _obiekt.obstacles.Add(podzial[i] + "=" + 0);
            }
            else
            {
                _obiekt.obstacles.Add(podzial[i] + "=" + 1);
            }

            i += 1;
        }
    }

    dane_do_string = serializacjaobiektów(_obiekt);
    zapis_do_txt();

}

string serializacjaobiektów(object dane_obiektów)
{
    string otrzymany_string = null;//przechowuje ciąg znaków
    MemoryStream pamięć = new MemoryStream();//przechowuje pamięć
    XmlSerializer serializerxml = new
XmlSerializer(typeof(wzór_mapy));//przypisuje wzór zapisu
    XmlTextWriter writer_xml = new XmlTextWriter(pamięć,
Encoding.UTF8);//przetwarza dane
    serializerxml.Serialize(writer_xml, dane_obiektów);//koduje ciąg
znaków
    pamięć = (MemoryStream)writer_xml.BaseStream;//przypisuje przetworzone
dane
    otrzymany_string = do_stringa(pamięć.ToArray());//przypisuje ciąg
znaków
    return otrzymany_string;//zwraca string
}

string do_stringa(byte[] tablica)
{
    UTF8Encoding kodowanie = new UTF8Encoding();//kodowane znaków
    string otrzymany_string = kodowanie.GetString(tablica);//przypisuje
dane do tablicy
    return otrzymany_string;//zwraca tablicę
}

void zapis_do_txt()
{

```

```

        StreamWriter writer;//odnośnik do pliku tekstowego
        FileInfo info = new FileInfo(lokalizacja_zapisu);//przechowuje
informacje o lokacji
        if (!info.Exists)
        {
            writer = info.CreateText();//tworzy plik tekstowy
        }
        else
        {
            info.Delete();//usuwa plik
            writer = info.CreateText();//tworzy plik tekstowy
        }
        writer.Write(dane_do_string);//wpisuje ciąg znaków do pliku
        writer.Close();//wyłącza StreamWriter
    }

    static void Main(string[] args)
    {
        Program mapa=new Program();
        mapa.stwórz();
    }
}

```

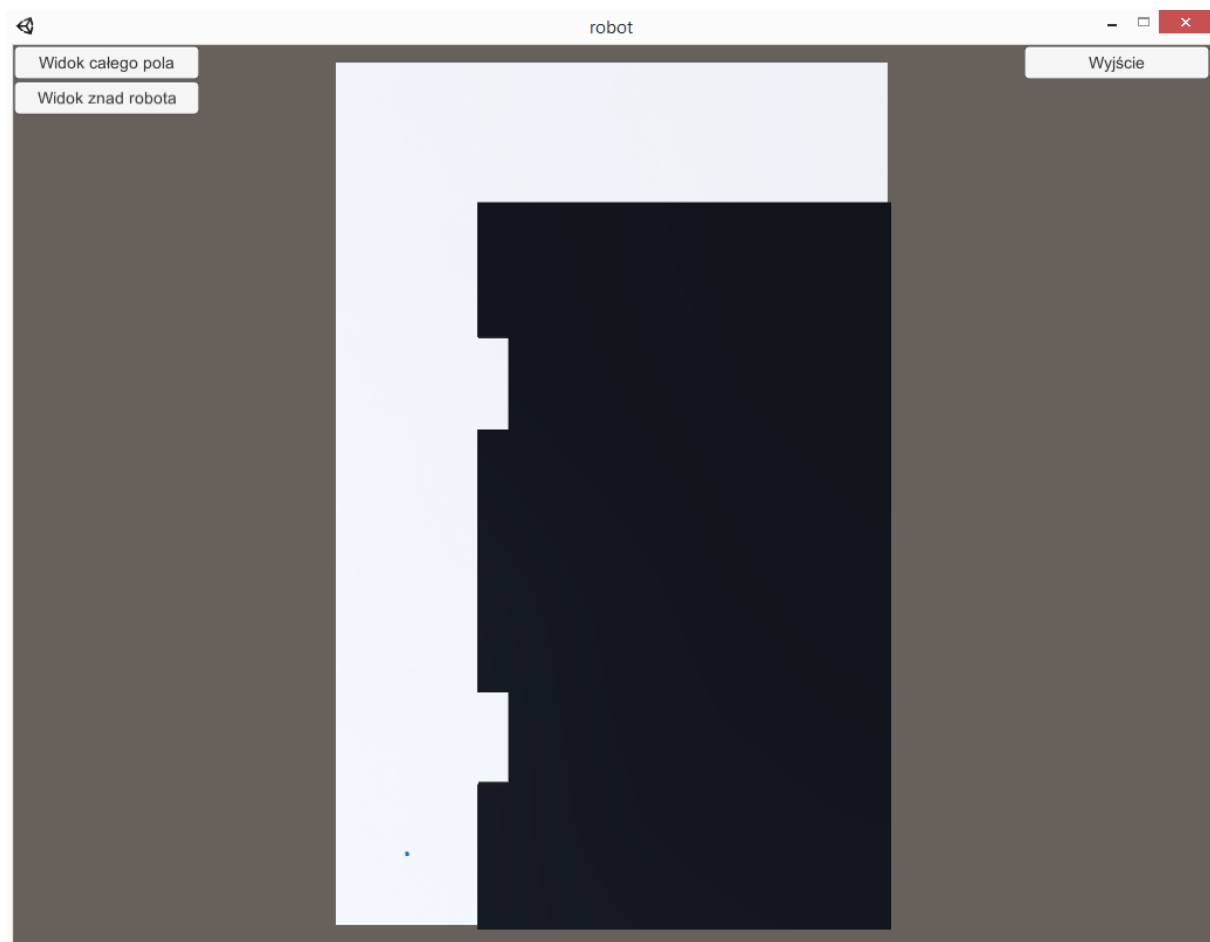
## 4.Funkcjonalności projektu

W tym rozdziale opisane jest działanie poszczególnych funkcji działania programu.

### Proces generowania mapy,

**Opis:** Program przelicza wielkości pól na długość i szerokość całej mapy następnie tworzy pola o wyznaczonych wymiarach obok siebie w zależności od ich nazw na koniec wypełnia je poprzez aktywacje pod obiektu lub zostawia puste

### Screenshot:



### Proces przemieszczania robota,

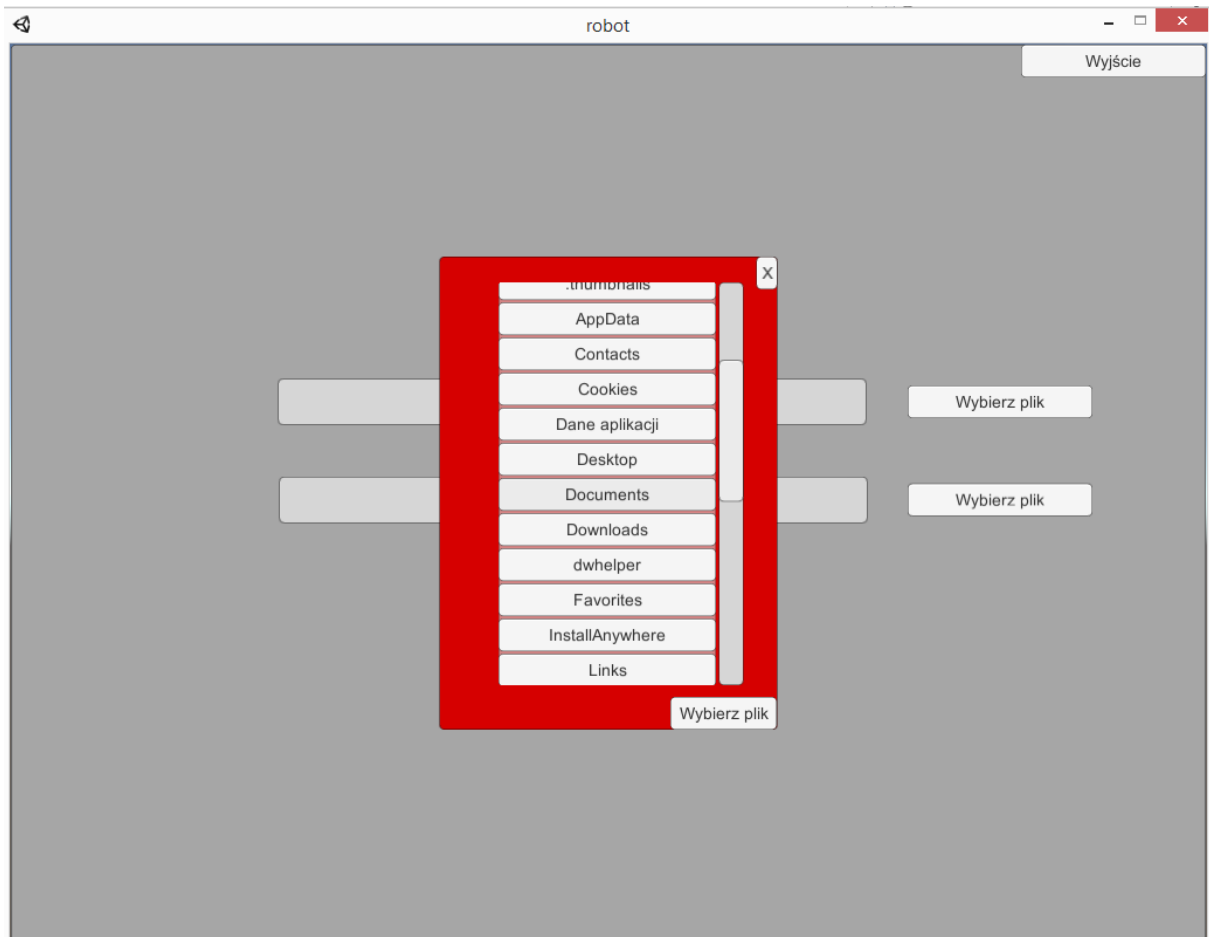
**Opis:** Robot porusza się do wskazanego pola najkrótszą dostępną ścieżką za pomocą funkcji navmashagent.



## Uproszczony eksplorator,

**Opis:** Fragment programu odpowiedzialny za wybranie plików z mapą i celem dla robota.

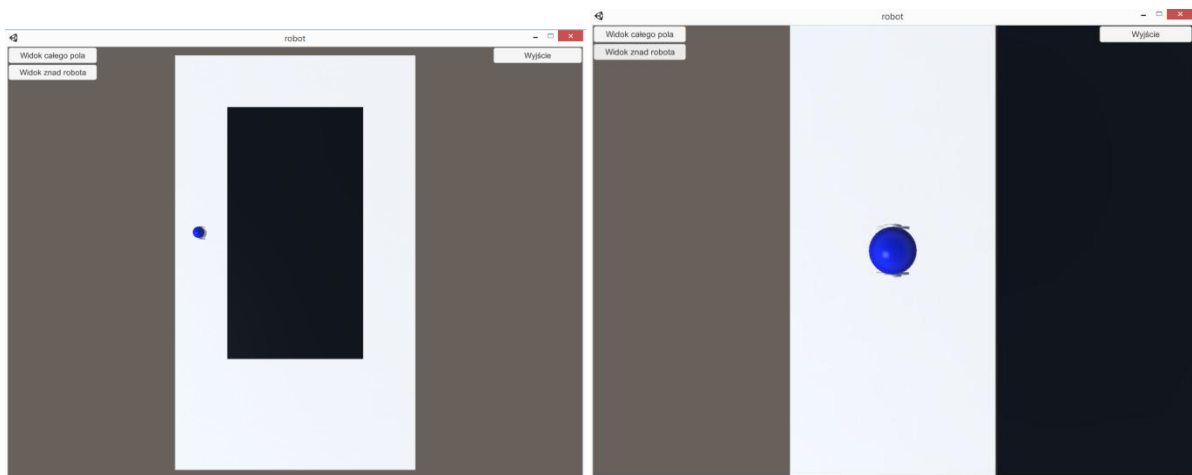
### Screenshot:



## Zmiana typu kamery:

**Opis:** Pozwala zmienić widok pomiędzy widokiem całej mapy lub widokiem znad robota który można zoomować.

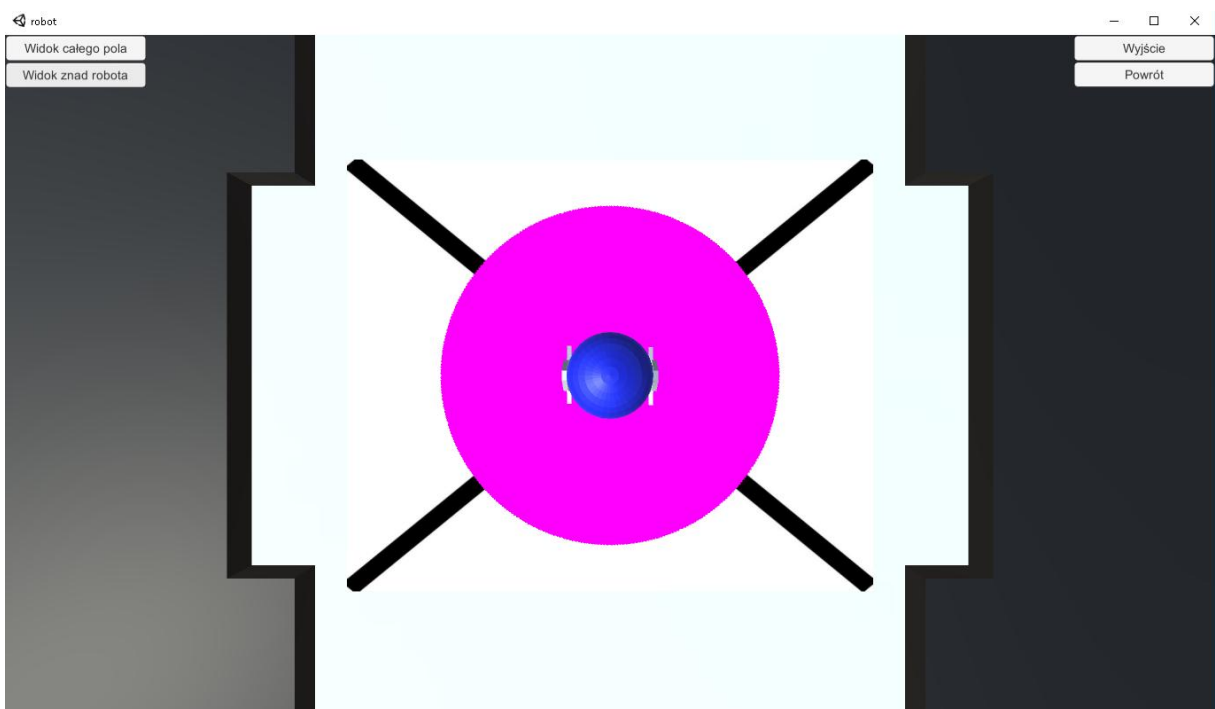
## Screenshot:



## Radar:

**Opis:** Wyświetla obszar od ścian zbadany przez robota po badanych kontaktach

## Screenshot



## 5. Finalny wygląd projektu:

W tym rozdziale zawarte informacje o mapach Wydziału, które dostałem do wizualizacji. Mapy przedstawione s w trzech częściach: dane z prodramy RoboGuide, przetworzone przez zemnie dane do ułatwienia wizualizacji( zawiera dane o ilości pól ,odległości granic pól od ich środka oraz które pola są przeszkodami) oraz zwizualizowaną mapę.

### Mapy – numeryczny opis i wizualizacje,

#### E0 – Hall

#### Dane numeryczne:

#### Mapa w formacie dla projektu roboguide (.bmf),

MAP\_ROWS=7

MAP\_COLS=5

x\_goal=1 // ilośćpółpoziomo

y\_goal=5// ilośćpółpionowo

DISTANCE\_TO\_OBSTACLES

NORTH

(1,1)=120 (1,2)=120 (1,3)=255 (1,4)=120 (1,5)=120

(2,1)=255 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=255

(3,1)=255 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=255

(4,1)=255 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=255

(5,1)=255 (5,2)=60 (5,3)=60 (5,4)=60 (5,5)=255

(6,1)=255 (6,2)=170 (6,3)=170 (6,4)=170 (6,5)=25570 // górna odległość sierotka pola od najbliższej ściany

(7,1)=255 (7,2)=255 (7,3)=255 (7,4)=255 (7,5)=255

EAST

(1,1)=255 (1,2)=255 (1,3)=255 (1,4)=255 (1,5)=120

(2,1)=120 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=120

(3,1)=100 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=120

(4,1)=120 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=120

(5,1)=255 (5,2)=255 (5,3)=255 (5,4)=255 (5,5)=120

(6,1)=255 (6,2)=255 (6,3)=255 (6,4)=255 (6,5)=120

(7,1)=255 (7,2)=255 (7,3)=255 (7,4)=255 (7,5)=120// prawa odległość sierotka pola od najbliższej ściany

#### SOUTH

(1,1)=255 (1,2)=120 (1,3)=120 (1,4)=120 (1,5)=255

(2,1)=255 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=255

(3,1)=255 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=255

(4,1)=255 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=255

(5,1)=255 (5,2)=255 (5,3)=255 (5,4)=255 (5,5)=255

(6,1)=255 (6,2)=255 (6,3)=255 (6,4)=255 (6,5)=255

(7,1)=145 (7,2)=145 (7,3)=165 (7,4)=145 (7,5)=145//dolna odległość sierotka pola od najbliższej ściany

#### WEST

(1,1)=120 (1,2)=255 (1,3)=255 (1,4)=255 (1,5)=255

(2,1)=120 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=120

(3,1)=120 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=120

(4,1)=120 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=120

(5,1)=120 (5,2)=255 (5,3)=255 (5,4)=255 (5,5)=255

(6,1)=140 (6,2)=255 (6,3)=255 (6,4)=255 (6,5)=255

(7,1)=120 (7,2)=255 (7,3)=255 (7,4)=255 (7,5)=255// lewa odległość sierotka pola od najbliższej ściany

#### FIELDS\_SIZE

#### NORTH

(1,1)=120 (1,2)=120 (1,3)=120 (1,4)=120 (1,5)=120

(2,1)=255 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=255

(3,1)=63 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=63

(4,1)=255 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=255

(5,1)=60 (5,2)=60 (5,3)=60 (5,4)=60 (5,5)=60

(6,1)=50 (6,2)=50 (6,3)=50 (6,4)=50 (6,5)=50

(7,1)=145 (7,2)=145 (7,3)=145 (7,4)=145 (7,5)=145// górna odległość sierotka pola od końca pola

#### EAST

(1,1)=120 (1,2)=90 (1,3)=130 (1,4)=90 (1,5)=120

(2,1)=120 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=120

(3,1)=100 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=120

(4,1)=120 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=120

(5,1)=120 (5,2)=90 (5,3)=130 (5,4)=90 (5,5)=120

(6,1)=120 (6,2)=90 (6,3)=130 (6,4)=90 (6,5)=120

(7,1)=120 (7,2)=90 (7,3)=130 (7,4)=90 (7,5)=120// prawa odległość sierotka pola od końca pola

#### SOUTH

(1,1)=120 (1,2)=120 (1,3)=120 (1,4)=120 (1,5)=120

(2,1)=255 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=255

(3,1)=63 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=63

(4,1)=255 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=255

(5,1)=60 (5,2)=60 (5,3)=60 (5,4)=60 (5,5)=60

(6,1)=50 (6,2)=50 (6,3)=50 (6,4)=50 (6,5)=50

(7,1)=145 (7,2)=145 (7,3)=165 (7,4)=145 (7,5)=145// dolna odległość sierotka pola od końca pola

#### WEST

(1,1)=120 (1,2)=90 (1,3)=130 (1,4)=90 (1,5)=120

(2,1)=120 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=120

(3,1)=120 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=120

(4,1)=120 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=120

(5,1)=120 (5,2)=90 (5,3)=130 (5,4)=90 (5,5)=120

(6,1)=140 (6,2)=90 (6,3)=130 (6,4)=90 (6,5)=120

(7,1)=120 (7,2)=90 (7,3)=130 (7,4)=90 (7,5)=120// lewa odległość sierotka pola od końca pola

## NORTH\_DIRECTION

(1,1)=188 (1,2)=160 (1,3)=188 (1,4)=172 (1,5)=224

(2,1)=182 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=140

(3,1)=144 (3,2)=0 (3,3)=0 (3,4)=0 (3,5)=210

(4,1)=182 (4,2)=0 (4,3)=0 (4,4)=0 (4,5)=194

(5,1)=204 (5,2)=180 (5,3)=186 (5,4)=212 (5,5)=180

(6,1)=180 (6,2)=186 (6,3)=130 (6,4)=180 (6,5)=180

(7,1)=154 (7,2)=200 (7,3)=186 (7,4)=186 (7,5)=180

## OBSTACLES

(1,1)=0 (1,2)=0 (1,3)=0 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=5 (2,3)=5 (2,4)=5 (2,5)=0

(3,1)=0 (3,2)=5 (3,3)=5 (3,4)=5 (3,5)=0

(4,1)=0 (4,2)=5 (4,3)=5 (4,4)=5 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=0 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=0 (6,4)=0 (6,5)=0

(7,1)=0 (7,2)=0 (7,3)=0 (7,4)=0 (7,5)=0

## Mapa w formacie potrzebnym do wizualizacji

### Konwersja z pliku .bmf:

```
<?xml version="1.0" encoding="utf-8"?><wzór_mapyxmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<ilość_punktów_x>7</ilość_punktów_x><ilość_punktów_y>5</ilość_punktów_y>
```

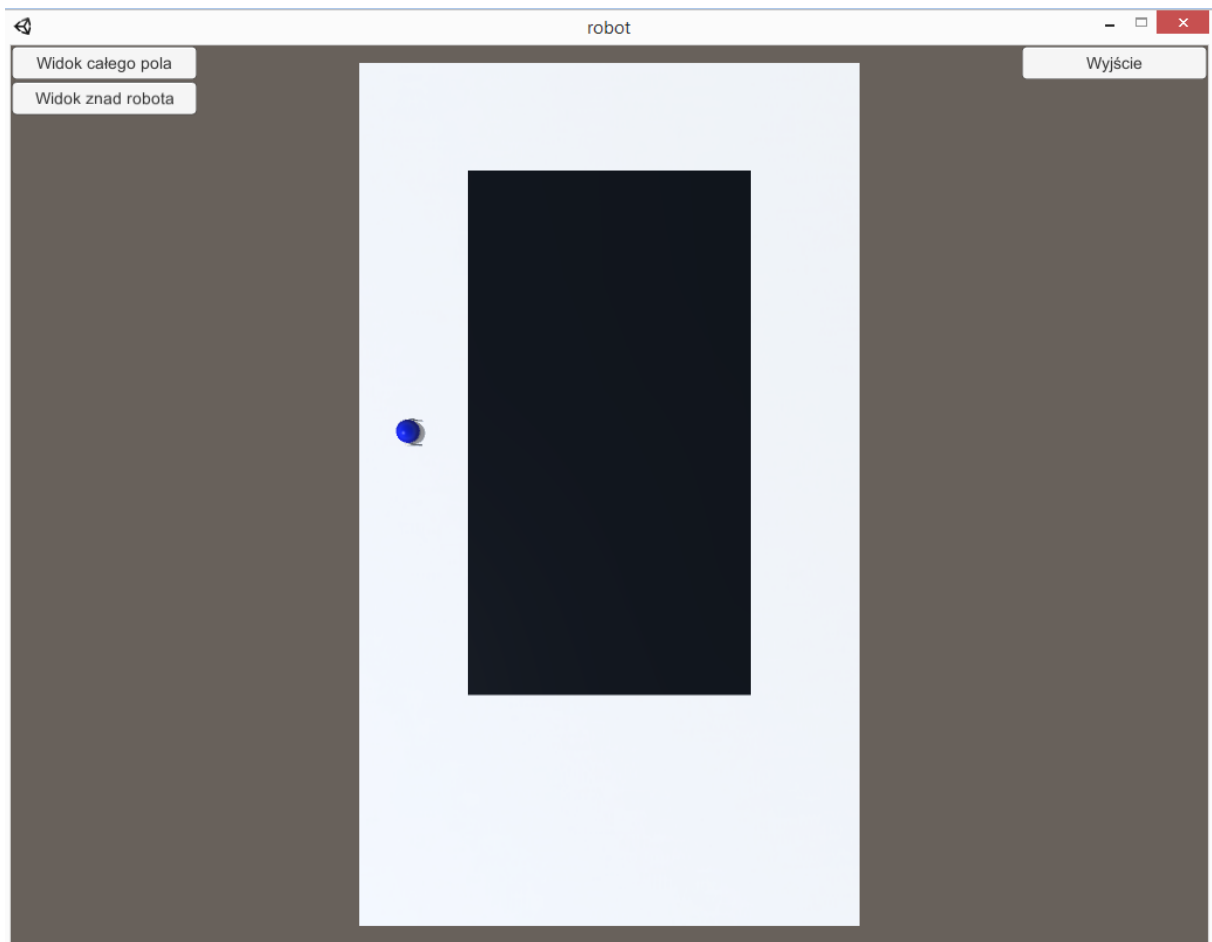
```
//ilości pól poziomopionowo
```

```
<north><string>1,1=120</string><string>1,2=120</string><string>1,3=120</string><string>1,4=120</string>
<string>1,5=120</string><string>2,1=255</string><string>2,2=255</string><string>2,3=255</string><string>
2,4=255</string><string>2,5=255</string><string>3,1=63</string><string>3,2=63</string><string>3,3=63</st
ring><string>3,4=63</string><string>3,5=63</string><string>4,1=255</string><string>4,2=255</string><stri
ng>4,3=255</string><string>4,4=255</string><string>4,5=255</string><string>5,1=60</string><string>5,2=60
</string><string>5,3=60</string><string>5,4=60</string><string>5,5=60</string><string>6,1=50</string><stri
ng>6,2=50</string><string>6,3=50</string><string>6,4=50</string><string>6,5=50</string><string>7,1=145<
/string><string>7,2=145</string><string>7,3=145</string><string>7,4=145</string><string>7,5=145</string>
</north>
```

```
// górna odległość sierotka pola od końca pola
```

```
<obstacles><string>1,1=0</string><string>1,2=0</string><string>1,3=0</string><string>1,4=0</string><string>1,5=0</string><string>2,1=0</string><string>2,2=1</string><string>2,3=1</string><string>2,4=1</string><string>2,5=0</string><string>3,1=0</string><string>3,2=1</string><string>3,3=1</string><string>3,4=1</string><string>3,5=0</string><string>4,1=0</string><string>4,2=1</string><string>4,3=1</string><string>4,4=1</string><string>4,5=0</string><string>5,1=0</string><string>5,2=0</string><string>5,3=0</string><string>5,4=0</string><string>5,5=0</string><string>6,1=0</string><string>6,2=0</string><string>6,3=0</string><string>6,4=0</string><string>6,5=0</string><string>7,1=0</string><string>7,2=0</string><string>7,3=0</string><string>7,4=0</string><string>7,5=0</string></obstacles></wzór_mapy>//opis ktore polasapełne
```

**Zwizualizowana mapa (Screenshot z projektu):**



## E0-West-corridor

**Dane numeryczne:**

**Mapa w formacie dla projektu roboguide (.bmf),**

**MAP\_ROWS=6**

**MAP\_COLS=3**

**x\_goal=1**

**y\_goal=1**

**DISTANCE\_TO\_OBSTACLES**

**NORTH**

**(1,1)=70 (1,2)=70 (1,3)=70**

**(2,1)=205 (2,2)=0 (2,3)=0**

**(3,1)=255 (3,2)=0 (3,3)=0**



(4,1)=255 (4,2)=0 (4,3)=0

(5,1)=255 (5,2)=0 (5,3)=0

(6,1)=255 (6,2)=0 (6,3)=0

#### EAST

(1,1)=255 (1,2)=255 (1,3)=100

(2,1)=70 (2,2)=0 (2,3)=0

(3,1)=85 (3,2)=0 (3,3)=0

(4,1)=70 (4,2)=0 (4,3)=0

(5,1)=85 (5,2)=0 (5,3)=0

(6,1)=70 (6,2)=0 (6,3)=0

#### SOUTH

(1,1)=255 (1,2)=70 (1,3)=70

(2,1)=255 (2,2)=0 (2,3)=0

(3,1)=255 (3,2)=0 (3,3)=0

(4,1)=255 (4,2)=0 (4,3)=0

(5,1)=110 (5,2)=0 (5,3)=0

(6,1)=32 (6,2)=0 (6,3)=0

#### WEST

(1,1)=70 (1,2)=240 (1,3)=255

(2,1)=70 (2,2)=0 (2,3)=0

(3,1)=85 (3,2)=0 (3,3)=0

(4,1)=70 (4,2)=0 (4,3)=0

(5,1)=85 (5,2)=0 (5,3)=0

(6,1)=70 (6,2)=0 (6,3)=0

#### FIELDS\_SIZE

##### NORTH

(1,1)=70 (1,2)=70 (1,3)=70

(2,1)=65 (2,2)=0 (2,3)=0

$(3,1)=45$   $(3,2)=0$   $(3,3)=0$

$(4,1)=127$   $(4,2)=0$   $(4,3)=0$

$(5,1)=45$   $(5,2)=0$   $(5,3)=0$

$(6,1)=32$   $(6,2)=0$   $(6,3)=0$

#### EAST

$(1,1)=70$   $(1,2)=100$   $(1,3)=100$

$(2,1)=70$   $(2,2)=0$   $(2,3)=0$

$(3,1)=85$   $(3,2)=0$   $(3,3)=0$

$(4,1)=70$   $(4,2)=0$   $(4,3)=0$

$(5,1)=85$   $(5,2)=0$   $(5,3)=0$

$(6,1)=70$   $(6,2)=0$   $(6,3)=0$

#### SOUTH

$(1,1)=70$   $(1,2)=70$   $(1,3)=70$

$(2,1)=65$   $(2,2)=0$   $(2,3)=0$

$(3,1)=45$   $(3,2)=0$   $(3,3)=0$

$(4,1)=127$   $(4,2)=0$   $(4,3)=0$

$(5,1)=45$   $(5,2)=0$   $(5,3)=0$

$(6,1)=32$   $(6,2)=0$   $(6,3)=0$

#### WEST

$(1,1)=70$   $(1,2)=100$   $(1,3)=100$

$(2,1)=70$   $(2,2)=0$   $(2,3)=0$

$(3,1)=85$   $(3,2)=0$   $(3,3)=0$

$(4,1)=70$   $(4,2)=0$   $(4,3)=0$

$(5,1)=85$   $(5,2)=0$   $(5,3)=0$

$(6,1)=70$   $(6,2)=0$   $(6,3)=0$

#### NORTH\_DIRECTION

$(1,1)=190$   $(1,2)=188$   $(1,3)=224$

$(2,1)=188$   $(2,2)=0$   $(2,3)=0$

(3,1)=178 (3,2)=0 (3,3)=0

(4,1)=168 (4,2)=0 (4,3)=0

(5,1)=140 (5,2)=0 (5,3)=0

(6,1)=152 (6,2)=0 (6,3)=0

#### OBSTACLES

(1,1)=0 (1,2)=0 (1,3)=0

(2,1)=0 (2,2)=5 (2,3)=5

(3,1)=0 (3,2)=5 (3,3)=5

(4,1)=0 (4,2)=5 (4,3)=5

(5,1)=0 (5,2)=5 (5,3)=5

(6,1)=0 (6,2)=5 (6,3)=5

**Mapa w formacie potrzebnym do wizualizacji,**

#### **Konwersja z pliku .bmf:**

```
<?xml version="1.0" encoding="utf-8"?><wzór_mapyxmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<ilość_punktów_x>6</ilość_punktów_x><ilość_punktów_y>4</ilość_punktów_y>
```

```
<north><string>1,1=70</string><string>1,2=70</string><string>1,3=70</string><string>1,4=70</string><stri  
ng>2,1=65</string><string>2,2=65</string><string>2,3=65</string><string>2,4=65</string><string>3,1=45</s  
tring><string>3,2=45</string><string>3,3=45</string><string>3,4=45</string><string>4,1=127</string><strin  
g>4,2=127</string><string>4,3=127</string><string>4,4=127</string><string>5,1=45</string><string>5,2=45  
</string><string>5,3=45</string><string>5,4=45</string><string>6,1=70</string><string>6,2=70</string><stri  
ng>6,3=70</string><string>6,4=70</string></north>
```

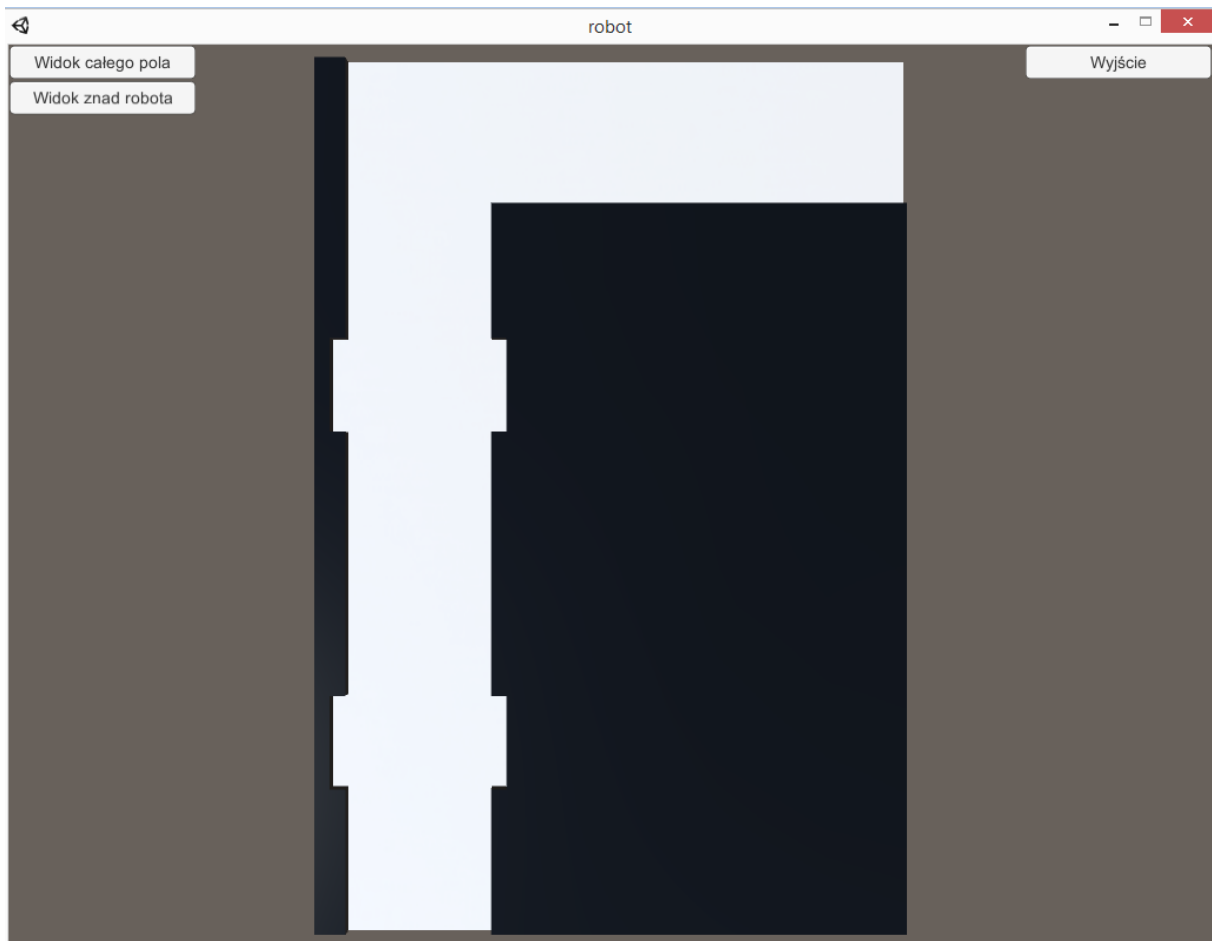
```
<south><string>1,1=70</string><string>1,2=70</string><string>1,3=70</string><string>1,4=70</string><stri  
ng>2,1=65</string><string>2,2=65</string><string>2,3=65</string><string>2,4=65</string><string>3,1=45</s  
tring><string>3,2=45</string><string>3,3=45</string><string>3,4=45</string><string>4,1=127</string><strin  
g>4,2=127</string><string>4,3=127</string><string>4,4=127</string><string>5,1=45</string><string>5,2=45  
</string><string>5,3=45</string><string>5,4=45</string><string>6,1=70</string><string>6,2=70</string><stri  
ng>6,3=70</string><string>6,4=70</string></south>
```

```
<east><string>1,1=15</string><string>1,2=70</string><string>1,3=100</string><string>1,4=100</string><stri  
ng>2,1=15</string><string>2,2=70</string><string>2,3=100</string><string>2,4=100</string><string>3,1=7,5  
</string><string>3,2=85</string><string>3,3=92,5</string><string>3,4=100</string><string>4,1=15</string><  
string>4,2=70</string><string>4,3=100</string><string>4,4=100</string><string>5,1=7,5</string><string>5,2  
=85</string><string>5,3=92,5</string><string>5,4=100</string><string>6,1=15</string><string>6,2=70</strin  
g><string>6,3=100</string><string>6,4=100</string></east>
```

```
<west><string>1,1=15</string><string>1,2=70</string><string>1,3=100</string><string>1,4=100</string><string>2,1=15</string><string>2,2=70</string><string>2,3=100</string><string>2,4=100</string><string>3,1=7,5</string><string>3,2=85</string><string>3,3=92,5</string><string>3,4=100</string><string>4,1=15</string><string>4,2=70</string><string>4,3=100</string><string>4,4=100</string><string>5,1=7,5</string><string>5,2=85</string><string>5,3=92,5</string><string>5,4=100</string><string>6,1=15</string><string>6,2=70</string><string>6,3=100</string><string>6,4=100</string></west>
```

```
<obstacles><string>1,1=1</string><string>1,2=0</string><string>1,3=0</string><string>1,4=0</string><string>2,1=1</string><string>2,2=0</string><string>2,3=1</string><string>2,4=1</string><string>3,1=1</string><string>3,2=0</string><string>3,3=1</string><string>3,4=1</string><string>4,1=1</string><string>4,2=0</string><string>4,3=1</string><string>4,4=1</string><string>5,1=1</string><string>5,2=0</string><string>5,3=1</string><string>5,4=1</string><string>6,1=1</string><string>6,2=0</string><string>6,3=1</string><string>6,4=1</string></obstacles></wzór_mapy>//opis które polasapełne
```

## Zwizualizowana mapa (Screenshot z projektu):



E0-East-corridor

Dane numeryczne:

Mapa w formacie dla projektu roboguide (.bmf),

MAP\_ROWS=7

MAP\_COLS=5

x\_goal=4

y\_goal=3

DISTANCE\_TO\_OBSTACLES

NORTH

(1,1)=70 (1,2)=70 (1,3)=70 (1,4)=70 (1,5)=85

(2,1)=0 (2,2)=0 (2,3)=208 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=255 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=255 (6,4)=0 (6,5)=0

EAST

(1,1)=255 (1,2)=255 (1,3)=255 (1,4)=122 (1,5)=45

(2,1)=0 (2,2)=0 (2,3)=70 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=85 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=70 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=85 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=70 (6,4)=0 (6,5)=0

SOUTH

(1,1)=70 (1,2)=70 (1,3)=255 (1,4)=70 (1,5)=70

(2,1)=0 (2,2)=0 (2,3)=255 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=110 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=33 (6,4)=0 (6,5)=0

WEST

(1,1)=255 (1,2)=255 (1,3)=255 (1,4)=255 (1,5)=255

(2,1)=0 (2,2)=0 (2,3)=70 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=85 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=70 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=85 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=70 (6,4)=0 (6,5)=0

#### **FIELDS\_SIZE**

##### **NORTH**

(1,1)=15 (1,2)=15 (1,3)=15 (1,4)=15 (1,5)=0

(2,1)=70 (2,2)=70 (2,3)=70 (2,4)=70 (2,5)=85

(3,1)=67 (3,2)=67 (3,3)=67 (3,4)=67 (3,5)=67

(4,1)=45 (4,2)=45 (4,3)=45 (4,4)=45 (4,5)=45

(5,1)=125 (5,2)=125 (5,3)=125 (5,4)=125 (5,5)=125

(6,1)=45 (6,2)=45 (6,3)=45 (6,4)=45 (6,5)=45

(7,1)=33 (7,2)=33 (7,3)=33 (7,4)=33 (7,5)=33

##### **EAST**

(1,1)=55 (1,2)=150 (1,3)=70 (1,4)=30 (1,5)=48

(2,1)=55 (2,2)=150 (2,3)=70 (2,4)=30 (2,5)=48

(3,1)=55 (3,2)=150 (3,3)=70 (3,4)=30 (3,5)=48

(4,1)=55 (4,2)=142.5 (4,3)=85 (4,4)=22.5 (4,5)=48

(5,1)=55 (5,2)=150 (5,3)=70 (5,4)=30 (5,5)=48

(6,1)=55 (6,2)=142.5 (6,3)=85 (6,4)=22.5 (6,5)=48

(7,1)=55 (7,2)=150 (7,3)=70 (7,4)=30 (7,5)=48

##### **SOUTH**

(1,1)=15 (1,2)=15 (1,3)=15 (1,4)=15 (1,5)=0

(2,1)=70 (2,2)=70 (2,3)=70 (2,4)=70 (2,5)=85

(3,1)=67 (3,2)=67 (3,3)=67 (3,4)=67 (3,5)=67

(4,1)=45 (4,2)=45 (4,3)=45 (4,4)=45 (4,5)=45

(5,1)=125 (5,2)=125 (5,3)=125 (5,4)=125 (5,5)=125

(6,1)=45 (6,2)=45 (6,3)=45 (6,4)=45 (6,5)=45

(7,1)=33 (7,2)=33 (7,3)=33 (7,4)=33 (7,5)=33

#### WEST

(1,1)=55 (1,2)=150 (1,3)=70 (1,4)=30 (1,5)=48

(2,1)=55 (2,2)=150 (2,3)=70 (2,4)=30 (2,5)=48

(3,1)=55 (3,2)=150 (3,3)=70 (3,4)=30 (3,5)=48

(4,1)=55 (4,2)=142.5 (4,3)=85 (4,4)=22.5 (4,5)=48

(5,1)=55 (5,2)=150 (5,3)=70 (5,4)=30 (5,5)=48

(6,1)=55 (6,2)=142.5 (6,3)=85 (6,4)=22.5 (6,5)=48

(7,1)=55 (7,2)=150 (7,3)=70 (7,4)=30 (7,5)=48

#### NORTH\_DIRECTION

(1,1)=164 (1,2)=186 (1,3)=192 (1,4)=194 (1,5)=194

(2,1)=0 (2,2)=0 (2,3)=192 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=196 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=196 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=186 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=150 (6,4)=0 (6,5)=0

#### OBSTACLES

(1,1)=5 (1,2)=5 (1,3)=5 (1,4)=5 (1,5)=5

(2,1)=0 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=0

(3,1)=5 (3,2)=5 (3,3)=0 (3,4)=5 (3,5)=5

(4,1)=5 (4,2)=5 (4,3)=0 (4,4)=5 (4,5)=5

(5,1)=5 (5,2)=5 (5,3)=0 (5,4)=5 (5,5)=5

(6,1)=5 (6,2)=5 (6,3)=0 (6,4)=5 (6,5)=5

(7,1)=5 (7,2)=5 (7,3)=0 (7,4)=5 (7,5)=5

**Mapa w formacie potrzebnym do wizualizacji,**

**Konwersja z pliku .bmf:**

```
<?xml version="1.0" encoding="utf-8"?><wzór_mapyxmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

### Wizualizacja mapy (Screen shot z projektu):

**Wizualizacja mapy (Screen shot z projektu):**





## E1-West-corridor

**Dane numeryczne:**

**Mapa w formacie dla projektu roboguide (.bmf),**

**MAP\_ROWS=7**

**MAP\_COLS=5**

**x\_goal=1**

**y\_goal=1**

**DISTANCE\_TO\_OBSTACLES**

**NORTH**

**(1,1)=86 (1,2)=70 (1,3)=70 (1,4)=70 (1,5)=70**

**(2,1)=0 (2,2)=0 (2,3)=208 (2,4)=0 (2,5)=0**

**(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0**

**(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0**

**(5,1)=0 (5,2)=0 (5,3)=255 (5,4)=0 (5,5)=0**

**(6,1)=0 (6,2)=0 (6,3)=255 (6,4)=0 (6,5)=0**

**EAST**

**(1,1)=255 (1,2)=255 (1,3)=255 (1,4)=255 (1,5)=104**

**(2,1)=0 (2,2)=0 (2,3)=68 (2,4)=0 (2,5)=0**

**(3,1)=0 (3,2)=0 (3,3)=84 (3,4)=0 (3,5)=0**

(4,1)=0 (4,2)=0 (4,3)=68 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=84 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=68 (6,4)=0 (6,5)=0

#### SOUTH

(1,1)=70 (1,2)=70 (1,3)=255 (1,4)=70 (1,5)=70

(2,1)=0 (2,2)=0 (2,3)=255 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=136 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=46 (6,4)=0 (6,5)=0

#### WEST

(1,1)=45 (1,2)=121 (1,3)=220 (1,4)=255 (1,5)=255

(2,1)=0 (2,2)=0 (2,3)=68 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=84 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=68 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=84 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=68 (6,4)=0 (6,5)=0

#### FIELDS\_SIZE

##### NORTH

(1,1)=0 (1,2)=15 (1,3)=15 (1,4)=15 (1,5)=15

(2,1)=85 (2,2)=70 (2,3)=70 (2,4)=70 (2,5)=70

(3,1)=68 (3,2)=68 (3,3)=68 (3,4)=68 (3,5)=68

(4,1)=45 (4,2)=45 (4,3)=45 (4,4)=45 (4,5)=45

(5,1)=125 (5,2)=125 (5,3)=125 (5,4)=125 (5,5)=125

(6,1)=45 (6,2)=45 (6,3)=45 (6,4)=45 (6,5)=45

(7,1)=46 (7,2)=46 (7,3)=46 (7,4)=46 (7,5)=46

##### EAST

(1,1)=45 (1,2)=31 (1,3)=68 (1,4)=104 (1,5)=104

(2,1)=45 (2,2)=31 (2,3)=68 (2,4)=104 (2,5)=104

(3,1)=45 (3,2)=31 (3,3)=68 (3,4)=104 (3,5)=104

(4,1)=45 (4,2)=23 (4,3)=84 (4,4)=96 (4,5)=104

(5,1)=45 (5,2)=31 (5,3)=68 (5,4)=104 (5,5)=104

(6,1)=45 (6,2)=23 (6,3)=84 (6,4)=96 (6,5)=104

(7,1)=45 (7,2)=31 (7,3)=68 (7,4)=104 (7,5)=104

#### SOUTH

(1,1)=0 (1,2)=15 (1,3)=15 (1,4)=15 (1,5)=15

(2,1)=85 (2,2)=70 (2,3)=70 (2,4)=70 (2,5)=70

(3,1)=68 (3,2)=68 (3,3)=68 (3,4)=68 (3,5)=68

(4,1)=45 (4,2)=45 (4,3)=45 (4,4)=45 (4,5)=45

(5,1)=125 (5,2)=125 (5,3)=125 (5,4)=125 (5,5)=125

(6,1)=45 (6,2)=45 (6,3)=45 (6,4)=45 (6,5)=45

(7,1)=46 (7,2)=46 (7,3)=46 (7,4)=46 (7,5)=46

#### WEST

(1,1)=45 (1,2)=31 (1,3)=68 (1,4)=104 (1,5)=104

(2,1)=45 (2,2)=31 (2,3)=68 (2,4)=104 (2,5)=104

(3,1)=45 (3,2)=31 (3,3)=68 (3,4)=104 (3,5)=104

(4,1)=45 (4,2)=23 (4,3)=84 (4,4)=96 (4,5)=104

(5,1)=45 (5,2)=31 (5,3)=68 (5,4)=104 (5,5)=104

(6,1)=45 (6,2)=23 (6,3)=84 (6,4)=96 (6,5)=104

(7,1)=45 (7,2)=31 (7,3)=68 (7,4)=104 (7,5)=104

#### NORTH\_DIRECTION

(1,1)=170 (1,2)=180 (1,3)=192 (1,4)=206 (1,5)=148

(2,1)=0 (2,2)=0 (2,3)=184 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=198 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=220 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=180 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=180 (6,4)=0 (6,5)=0

## OBSTACLES

(1,1)=5 (1,2)=5 (1,3)=5 (1,4)=5 (1,5)=5

(2,1)=0 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=0

(3,1)=5 (3,2)=5 (3,3)=0 (3,4)=5 (3,5)=5

(4,1)=5 (4,2)=5 (4,3)=0 (4,4)=5 (4,5)=5

(5,1)=5 (5,2)=5 (5,3)=0 (5,4)=5 (5,5)=5

(6,1)=5 (6,2)=5 (6,3)=0 (6,4)=5 (6,5)=5

(7,1)=5 (7,2)=5 (7,3)=0 (7,4)=5 (7,5)=5

## Mapa w formacie potrzebnym do wizualizacji,

### Konwersja z pliku .bmf:

```
<?xml version="1.0" encoding="utf-8"?><wzór_mapyxmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><ilość_punktów_x>7</ilość_punktów_x><ilość_punktó
w_y>5</ilość_punktów_y><north><string>1,1=0</string><string>1,2=15</string><string>1,3=15</string><str
ing>1,4=15</string><string>1,5=15</string><string>2,1=85</string><string>2,2=70</string><string>2,3=70</
string><string>2,4=70</string><string>2,5=70</string><string>3,1=68</string><string>3,2=68</string><strin
g>3,3=68</string><string>3,4=68</string><string>3,5=68</string><string>4,1=45</string><string>4,2=45</st
ring><string>4,3=45</string><string>4,4=45</string><string>4,5=45</string><string>5,1=125</string><string
>5,2=125</string><string>5,3=125</string><string>5,4=125</string><string>5,5=125</string><string>6,1=45
</string><string>6,2=45</string><string>6,3=45</string><string>6,4=45</string><string>6,5=45</string><stri
ng>7,1=46</string><string>7,2=46</string><string>7,3=46</string><string>7,4=46</string><string>7,5=46</s
tring></north><south><string>1,1=0</string><string>1,2=15</string><string>1,3=15</string><string>1,4=15
</string><string>1,5=15</string><string>2,1=85</string><string>2,2=70</string><string>2,3=70</string><stri
ng>2,4=70</string><string>2,5=70</string><string>3,1=68</string><string>3,2=68</string><string>3,3=68</s
tring><string>3,4=68</string><string>3,5=68</string><string>4,1=45</string><string>4,2=45</string><string
>4,3=45</string><string>4,4=45</string><string>4,5=45</string><string>5,1=125</string><string>5,2=125</s
tring><string>5,3=125</string><string>5,4=125</string><string>5,5=125</string><string>6,1=45</string><str
ing>6,2=45</string><string>6,3=45</string><string>6,4=45</string><string>6,5=45</string><string>7,1=46</
string><string>7,2=46</string><string>7,3=46</string><string>7,4=46</string><string>7,5=46</string></sout
h><east><string>1,1=45</string><string>1,2=31</string><string>1,3=68</string><string>1,4=104</string><st
ring>1,5=104</string><string>2,1=45</string><string>2,2=31</string><string>2,3=68</string><string>2,4=10
4</string><string>2,5=104</string><string>3,1=45</string><string>3,2=31</string><string>3,3=68</string><
string>3,4=104</string><string>3,5=104</string><string>4,1=45</string><string>4,2=23</string><string>4,3=
84</string><string>4,4=96</string><string>4,5=104</string><string>5,1=45</string><string>5,2=31</string>
<string>5,3=68</string><string>5,4=104</string><string>5,5=104</string><string>6,1=45</string><string>6,2
=23</string><string>6,3=84</string><string>6,4=96</string><string>6,5=104</string><string>7,1=45</string
><string>7,2=31</string><string>7,3=68</string><string>7,4=104</string><string>7,5=104</string></east><
west><string>1,1=45</string><string>1,2=31</string><string>1,3=68</string><string>1,4=104</string><strin
g>1,5=104</string><string>2,1=45</string><string>2,2=31</string><string>2,3=68</string><string>2,4=104</
```

```

/string><string>2,5=104</string><string>3,1=45</string><string>3,2=31</string><string>3,3=68</string><string>3,4=104</string><string>3,5=104</string><string>4,1=45</string><string>4,2=23</string><string>4,3=84</string><string>4,4=96</string><string>4,5=104</string><string>5,1=45</string><string>5,2=31</string><string>5,3=68</string><string>5,4=104</string><string>5,5=104</string><string>6,1=45</string><string>6,2=23</string><string>6,3=84</string><string>6,4=96</string><string>6,5=104</string><string>7,1=45</string><string>7,2=31</string><string>7,3=68</string><string>7,4=104</string><string>7,5=104</string></west><obstacles><string>1,1=1</string><string>1,2=1</string><string>1,3=1</string><string>1,4=1</string><string>1,5=1</string><string>2,1=0</string><string>2,2=0</string><string>2,3=0</string><string>2,4=0</string><string>2,5=0</string><string>3,1=1</string><string>3,2=1</string><string>3,3=0</string><string>3,4=1</string><string>3,5=1</string><string>4,1=1</string><string>4,2=1</string><string>4,3=0</string><string>4,4=1</string><string>4,5=1</string><string>5,1=1</string><string>5,2=1</string><string>5,3=0</string><string>5,4=1</string><string>5,5=1</string><string>6,1=1</string><string>6,2=1</string><string>6,3=0</string><string>6,4=1</string><string>6,5=1</string><string>7,1=1</string><string>7,2=1</string><string>7,3=0</string><string>7,4=1</string><string>7,5=1</string></obstacles></wzór_mapy>

```

## Zwizualizowanamapa (Screen shot z projektu):



### E1-East-corridor

### Dane numeryczne:

### Mapa w formacie dla projektu roboguide (.bmf),

MAP\_ROWS=7

MAP\_COLS=5

x\_goal=1

y\_goal=3

## DISTANCE\_TO\_OBSTACLES

### NORTH

(1,1)=70 (1,2)=70 (1,3)=70 (1,4)=70 (1,5)=86

(2,1)=0 (2,2)=0 (2,3)=205 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=255 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=255 (6,4)=0 (6,5)=0

### EAST

(1,1)=255 (1,2)=255 (1,3)=255 (1,4)=125 (1,5)=47

(2,1)=0 (2,2)=0 (2,3)=70 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=70 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=70 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=70 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=70 (6,4)=0 (6,5)=0

### SOUTH

(1,1)=70 (1,2)=70 (1,3)=255 (1,4)=70 (1,5)=70

(2,1)=0 (2,2)=0 (2,3)=255 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=141 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=47 (6,4)=0 (6,5)=0

### WEST

(1,1)=103 (1,2)=255 (1,3)=255 (1,4)=255 (1,5)=255

(2,1)=0 (2,2)=0 (2,3)=70 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=70 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=70 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=70 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=70 (6,4)=0 (6,5)=0

#### FIELDS\_SIZE

##### NORTH

(1,1)=15 (1,2)=15 (1,3)=15 (1,4)=15 (1,5)=0

(2,1)=70 (2,2)=70 (2,3)=70 (2,4)=70 (2,5)=85

(3,1)=65 (3,2)=65 (3,3)=65 (3,4)=65 (3,5)=65

(4,1)=47 (4,2)=47 (4,3)=47 (4,4)=47 (4,5)=47

(5,1)=126 (5,2)=126 (5,3)=126 (5,4)=126 (5,5)=126

(6,1)=47 (6,2)=47 (6,3)=47 (6,4)=47 (6,5)=47

(7,1)=46 (7,2)=46 (7,3)=46 (7,4)=46 (7,5)=46

##### EAST

(1,1)=103 (1,2)=103 (1,3)=70 (1,4)=30 (1,5)=77

(2,1)=103 (2,2)=103 (2,3)=70 (2,4)=30 (2,5)=77

(3,1)=103 (3,2)=103 (3,3)=70 (3,4)=30 (3,5)=77

(4,1)=103 (4,2)=95.5 (4,3)=85 (4,4)=22.5 (4,5)=77

(5,1)=103 (5,2)=103 (5,3)=70 (5,4)=30 (5,5)=77

(6,1)=103 (6,2)=95.5 (6,3)=85 (6,4)=22.5 (6,5)=77

(7,1)=103 (7,2)=103 (7,3)=70 (7,4)=30 (7,5)=77

##### SOUTH

(1,1)=15 (1,2)=15 (1,3)=15 (1,4)=15 (1,5)=0

(2,1)=70 (2,2)=70 (2,3)=70 (2,4)=70 (2,5)=85

(3,1)=65 (3,2)=65 (3,3)=65 (3,4)=65 (3,5)=65

(4,1)=47 (4,2)=47 (4,3)=47 (4,4)=47 (4,5)=47

(5,1)=126 (5,2)=126 (5,3)=126 (5,4)=126 (5,5)=126

(6,1)=47 (6,2)=47 (6,3)=47 (6,4)=47 (6,5)=47

(7,1)=46 (7,2)=46 (7,3)=46 (7,4)=46 (7,5)=46

##### WEST

(1,1)=103 (1,2)=103 (1,3)=70 (1,4)=30 (1,5)=77

(2,1)=103 (2,2)=103 (2,3)=70 (2,4)=30 (2,5)=77

(3,1)=103 (3,2)=103 (3,3)=70 (3,4)=30 (3,5)=77

(4,1)=103 (4,2)=95.5 (4,3)=85 (4,4)=22.5 (4,5)=77

(5,1)=103 (5,2)=103 (5,3)=70 (5,4)=30 (5,5)=77

(6,1)=103 (6,2)=95.5 (6,3)=85 (6,4)=22.5 (6,5)=77

(7,1)=103 (7,2)=103 (7,3)=70 (7,4)=30 (7,5)=77

#### NORTH\_DIRECTION

(1,1)=166 (1,2)=216 (1,3)=168 (1,4)=176 (1,5)=194

(2,1)=0 (2,2)=0 (2,3)=214 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=254 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=236 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=160 (5,4)=0 (5,5)=0

(6,1)=0 (6,2)=0 (6,3)=102 (6,4)=0 (6,5)=0

#### OBSTACLES

(1,1)=5 (1,2)=5 (1,3)=5 (1,4)=5 (1,5)=5

(2,1)=0 (2,2)=0 (2,3)=0 (2,4)=0 (2,5)=0

(3,1)=5 (3,2)=5 (3,3)=0 (3,4)=5 (3,5)=5

(4,1)=5 (4,2)=5 (4,3)=0 (4,4)=5 (4,5)=5

(5,1)=5 (5,2)=5 (5,3)=0 (5,4)=5 (5,5)=5

(6,1)=5 (6,2)=5 (6,3)=0 (6,4)=5 (6,5)=5

(7,1)=5 (7,2)=5 (7,3)=0 (7,4)=5 (7,5)=5

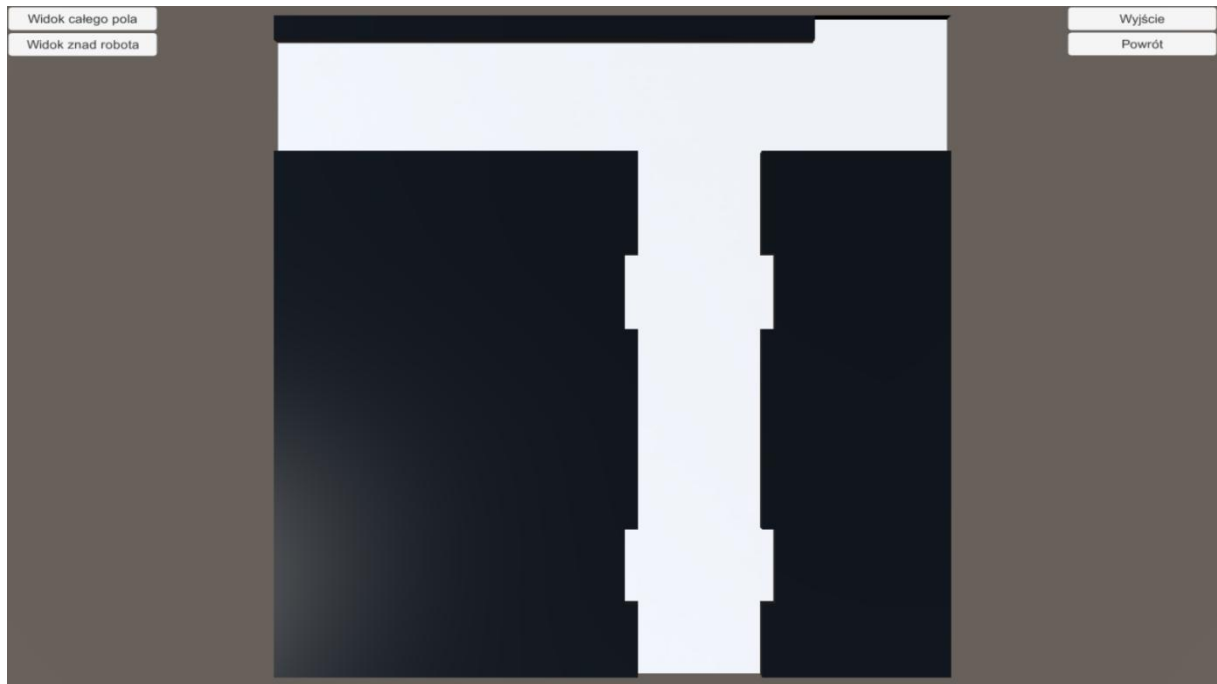
**Mapa w formacie potrzebnym do wizualizacji,**

**Konwersja z pliku .bmf:**

```
<?xml version="1.0" encoding="utf-8"?><wzór_mapyxmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><ilość_punktów_x>7</ilość_punktów_x><ilość_punktó
w_y>5</ilość_punktów_y><north><string>1,1=15</string><string>1,2=15</string><string>1,3=15</string><s
tring>1,4=15</string><string>1,5=0</string><string>2,1=70</string><string>2,2=70</string><string>2,3=70</
string><string>2,4=70</string><string>2,5=85</string><string>3,1=65</string><string>3,2=65</string><strin
```



**Zwizualizowana mapa (Screen shot z projektu):**



## D1-West-corridor

### Dane numeryczne:

### Mapa w formacie dla projektu roboguide (.bmf),

MAP\_ROWS=7

MAP\_COLS=5

x\_goal=2

y\_goal=3

DISTANCE\_TO\_OBSTACLES

NORTH

(1,1)=0 (1,2)=0 (1,3)=51 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=147 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=255 (5,4)=0 (5,5)=0

(6,1)=79 (6,2)=71 (6,3)=255 (6,4)=71 (6,5)=71

EAST

(1,1)=0 (1,2)=0 (1,3)=70 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=85 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=70 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=85 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=70 (5,4)=0 (5,5)=0

(6,1)=255 (6,2)=255 (6,3)=255 (6,4)=255 (6,5)=106

#### SOUTH

(1,1)=0 (1,2)=0 (1,3)=255 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=255 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=255 (5,4)=0 (5,5)=0

(6,1)=79 (6,2)=71 (6,3)=71 (6,4)=71 (6,5)=71

#### WEST

(1,1)=0 (1,2)=0 (1,3)=70 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=85 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=70 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=85 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=70 (5,4)=0 (5,5)=0

(6,1)=45 (6,2)=124 (6,3)=227 (6,4)=255 (6,5)=255

#### FIELDS\_SIZE

##### NORTH

(1,1)=51 (1,2)=51 (1,3)=51 (1,4)=51 (1,5)=51

(2,1)=45 (2,2)=45 (2,3)=45 (2,4)=45 (2,5)=45

(3,1)=128 (3,2)=128 (3,3)=128 (3,4)=128 (3,5)=128

(4,1)=45 (4,2)=45 (4,3)=45 (4,4)=45 (4,5)=45

(5,1)=66 (5,2)=66 (5,3)=66 (5,4)=66 (5,5)=66

(6,1)=78 (6,2)=71 (6,3)=71 (6,4)=71 (6,5)=71

$(7,1)=0$   $(7,2)=7$   $(7,3)=7$   $(7,4)=7$   $(7,5)=7$

#### EAST

$(1,1)=45$   $(1,2)=34$   $(1,3)=70$   $(1,4)=98$   $(1,5)=106$

$(2,1)=45$   $(2,2)=26.5$   $(2,3)=85$   $(2,4)=90.5$   $(2,5)=106$

$(3,1)=45$   $(3,2)=34$   $(3,3)=70$   $(3,4)=98$   $(3,5)=106$

$(4,1)=45$   $(4,2)=26.5$   $(4,3)=85$   $(4,4)=90.5$   $(4,5)=106$

$(5,1)=45$   $(5,2)=34$   $(5,3)=70$   $(5,4)=98$   $(5,5)=106$

$(6,1)=45$   $(6,2)=34$   $(6,3)=70$   $(6,4)=98$   $(6,5)=106$

$(7,1)=45$   $(7,2)=34$   $(7,3)=70$   $(7,4)=98$   $(7,5)=106$

#### SOUTH

$(1,1)=51$   $(1,2)=51$   $(1,3)=51$   $(1,4)=51$   $(1,5)=51$

$(2,1)=45$   $(2,2)=45$   $(2,3)=45$   $(2,4)=45$   $(2,5)=45$

$(3,1)=128$   $(3,2)=128$   $(3,3)=128$   $(3,4)=128$   $(3,5)=128$

$(4,1)=45$   $(4,2)=45$   $(4,3)=45$   $(4,4)=45$   $(4,5)=45$

$(5,1)=66$   $(5,2)=66$   $(5,3)=66$   $(5,4)=66$   $(5,5)=66$

$(6,1)=78$   $(6,2)=71$   $(6,3)=71$   $(6,4)=71$   $(6,5)=71$

$(7,1)=0$   $(7,2)=7$   $(7,3)=7$   $(7,4)=7$   $(7,5)=7$

#### WEST

$(1,1)=45$   $(1,2)=34$   $(1,3)=70$   $(1,4)=98$   $(1,5)=106$

$(2,1)=45$   $(2,2)=26.5$   $(2,3)=85$   $(2,4)=90.5$   $(2,5)=106$

$(3,1)=45$   $(3,2)=34$   $(3,3)=70$   $(3,4)=98$   $(3,5)=106$

$(4,1)=45$   $(4,2)=26.5$   $(4,3)=85$   $(4,4)=90.5$   $(4,5)=106$

$(5,1)=45$   $(5,2)=34$   $(5,3)=70$   $(5,4)=98$   $(5,5)=106$

$(6,1)=45$   $(6,2)=34$   $(6,3)=70$   $(6,4)=98$   $(6,5)=106$

$(7,1)=45$   $(7,2)=34$   $(7,3)=70$   $(7,4)=98$   $(7,5)=106$

#### NORTH\_DIRECTION

$(1,1)=0$   $(1,2)=0$   $(1,3)=176$   $(1,4)=0$   $(1,5)=0$

$(2,1)=0$   $(2,2)=0$   $(2,3)=194$   $(2,4)=0$   $(2,5)=0$

(3,1)=0 (3,2)=0 (3,3)=196 (3,4)=0 (3,5)=0  
 (4,1)=0 (4,2)=0 (4,3)=236 (4,4)=0 (4,5)=0  
 (5,1)=0 (5,2)=0 (5,3)=224 (5,4)=0 (5,5)=0  
 (6,1)=226 (6,2)=244 (6,3)=214 (6,4)=168 (6,5)=178

## OBSTACLES

(1,1)=5 (1,2)=5 (1,3)=0 (1,4)=5 (1,5)=5  
 (2,1)=5 (2,2)=5 (2,3)=0 (2,4)=5 (2,5)=5  
 (3,1)=5 (3,2)=5 (3,3)=0 (3,4)=5 (3,5)=5  
 (4,1)=5 (4,2)=5 (4,3)=0 (4,4)=5 (4,5)=5  
 (5,1)=5 (5,2)=5 (5,3)=0 (5,4)=5 (5,5)=5  
 (6,1)=0 (6,2)=0 (6,3)=0 (6,4)=0 (6,5)=0  
 (7,1)=0 (7,2)=5 (7,3)=5 (7,4)=5 (7,5)=5

**Mapa w formacie potrzebnym do wizualizacji,**

**Konwersja z pliku .bmf:**

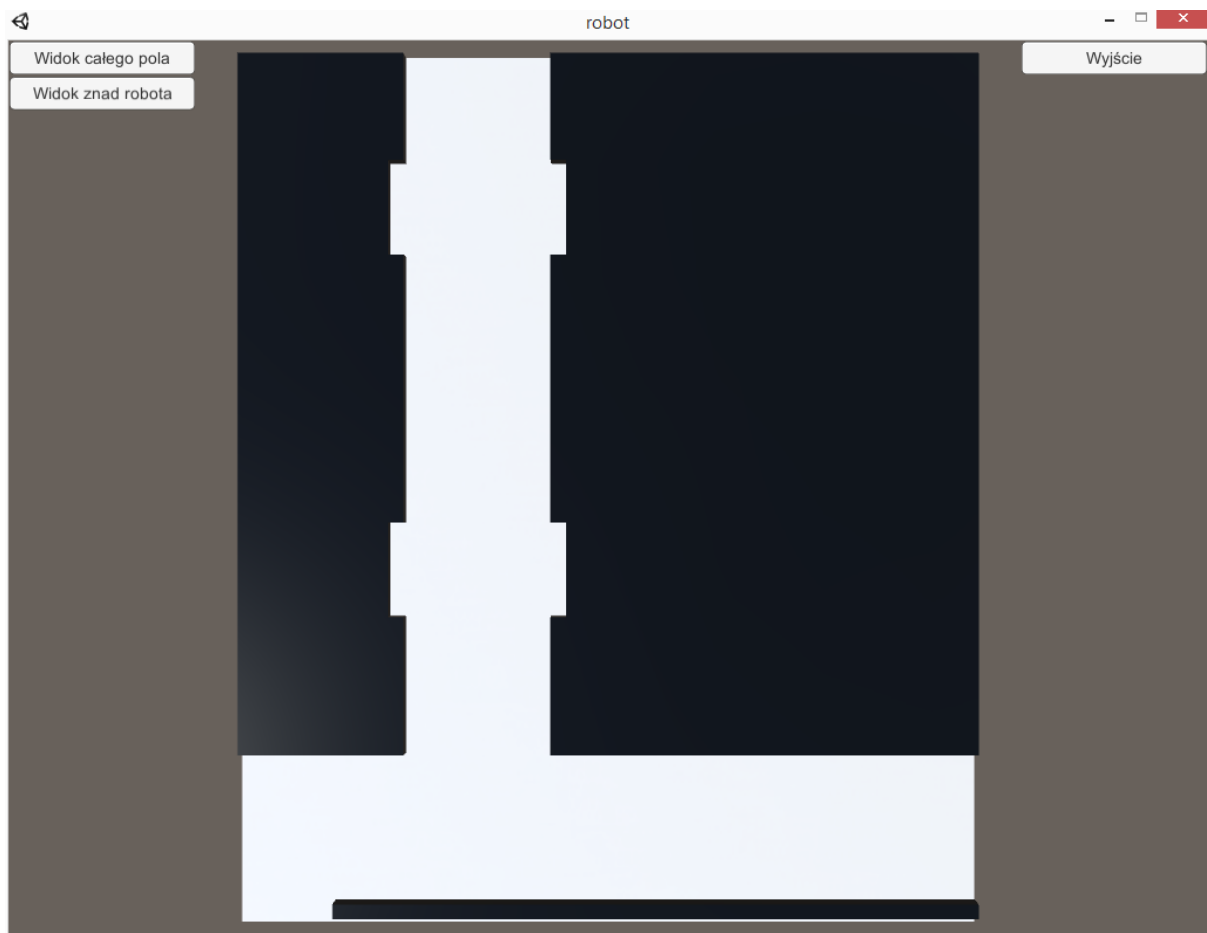
```
<?xml version="1.0" encoding="utf-8"?><wzór_mapyxmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><ilość_punktów_x>7</ilość_punktów_x><ilość_punktó
w_y>5</ilość_punktów_y><north><string>1,1=51</string><string>1,2=51</string><string>1,3=51</string><s
tring>1,4=51</string><string>1,5=51</string><string>2,1=45</string><string>2,2=45</string><string>2,3=45
</string><string>2,4=45</string><string>2,5=45</string><string>3,1=128</string><string>3,2=128</string><
string>3,3=128</string><string>3,4=128</string><string>3,5=128</string><string>4,1=45</string><string>4,2
=45</string><string>4,3=45</string><string>4,4=45</string><string>4,5=45</string><string>5,1=66</string>
<string>5,2=66</string><string>5,3=66</string><string>5,4=66</string><string>5,5=66</string><string>6,1=7
8</string><string>6,2=71</string><string>6,3=71</string><string>6,4=71</string><string>6,5=71</string><st
ring>7,1=0</string><string>7,2=7</string><string>7,3=7</string><string>7,4=7</string><string>7,5=7</string
></north><south><string>1,1=51</string><string>1,2=51</string><string>1,3=51</string><string>1,4=51</st
ring><string>1,5=51</string><string>2,1=45</string><string>2,2=45</string><string>2,3=45</string><string>
2,4=45</string><string>2,5=45</string><string>3,1=128</string><string>3,2=128</string><string>3,3=128</s
tring><string>3,4=128</string><string>3,5=128</string><string>4,1=45</string><string>4,2=45</string><stri
ng>4,3=45</string><string>4,4=45</string><string>4,5=45</string><string>5,1=66</string><string>5,2=66</s
tring><string>5,3=66</string><string>5,4=66</string><string>5,5=66</string><string>6,1=78</string><string
>6,2=71</string><string>6,3=71</string><string>6,4=71</string><string>6,5=71</string><string>7,1=0</strin
g><string>7,2=7</string><string>7,3=7</string><string>7,4=7</string><string>7,5=7</string></south><east>
<string>1,1=45</string><string>1,2=34</string><string>1,3=70</string><string>1,4=98</string><string>1,5=1
06</string><string>2,1=45</string><string>2,2=26.5</string><string>2,3=85</string><string>2,4=90.5</strin
g><string>2,5=106</string><string>3,1=45</string><string>3,2=34</string><string>3,3=70</string><string>3,
4=98</string><string>3,5=106</string><string>4,1=45</string><string>4,2=26.5</string><string>4,3=85</stri
```

```

ng><string>4,4=90.5</string><string>4,5=106</string><string>5,1=45</string><string>5,2=34</string><string>
>5,3=70</string><string>5,4=98</string><string>5,5=106</string><string>6,1=45</string><string>6,2=34</st
ring><string>6,3=70</string><string>6,4=98</string><string>6,5=106</string><string>7,1=45</string><string>
>7,2=34</string><string>7,3=70</string><string>7,4=98</string><string>7,5=106</string></east><west><stri
ng>1,1=45</string><string>1,2=34</string><string>1,3=70</string><string>1,4=98</string><string>1,5=106<
/string><string>2,1=45</string><string>2,2=26.5</string><string>2,3=85</string><string>2,4=90.5</string><
string>2,5=106</string><string>3,1=45</string><string>3,2=34</string><string>3,3=70</string><string>3,4=9
8</string><string>3,5=106</string><string>4,1=45</string><string>4,2=26.5</string><string>4,3=85</string>
<string>4,4=90.5</string><string>4,5=106</string><string>5,1=45</string><string>5,2=34</string><string>5,
3=70</string><string>5,4=98</string><string>5,5=106</string><string>6,1=45</string><string>6,2=34</strin
g><string>6,3=70</string><string>6,4=98</string><string>6,5=106</string><string>7,1=45</string><string>7,
2=34</string><string>7,3=70</string><string>7,4=98</string><string>7,5=106</string></west><obstacles><s
tring>1,1=1</string><string>1,2=1</string><string>1,3=0</string><string>1,4=1</string><string>1,5=1</strin
g><string>2,1=1</string><string>2,2=1</string><string>2,3=0</string><string>2,4=1</string><string>2,5=1</
string><string>3,1=1</string><string>3,2=1</string><string>3,3=0</string><string>3,4=1</string><string>3,5
=1</string><string>4,1=1</string><string>4,2=1</string><string>4,3=0</string><string>4,4=1</string><string>
>4,5=1</string><string>5,1=1</string><string>5,2=1</string><string>5,3=0</string><string>5,4=1</string><s
tring>5,5=1</string><string>6,1=0</string><string>6,2=0</string><string>6,3=0</string><string>6,4=0</strin
g><string>6,5=0</string><string>7,1=0</string><string>7,2=1</string><string>7,3=1</string><string>7,4=1</
string><string>7,5=1</string></obstacles></wzór_mapy>

```

## Zwizualizowana mapa (Screen shot z projektu):



## D1-East-corridor

### Dane numeryczne:

### Mapa w formacie dla projektu roboguide (.bmf),

MAP\_ROWS=6// ilość pól poziomo

MAP\_COLS=5// ilość pól pionowo

x\_goal=4

y\_goal=3

#### DISTANCE\_TO\_OBSTACLES

##### NORTH

(1,1)=0 (1,2)=0 (1,3)=51.5 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=191 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=255 (5,4)=0 (5,5)=0

(6,1)=70 (6,2)=70 (6,3)=255 (6,4)=70 (6,5)=70 // górna odległość sierotka pola od najbliższej ściany

##### EAST

(1,1)=0 (1,2)=0 (1,3)=70 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=85 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=70 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=85 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=70 (5,4)=0 (5,5)=0

(6,1)=255 (6,2)=255 (6,3)=255 (6,4)=126.5 (6,5)=48// prawa odległość sierotka pola od najbliższej ściany

##### SOUTH

(1,1)=0 (1,2)=0 (1,3)=255 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=255 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=255 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=255 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=208.5 (5,4)=0 (5,5)=0

(6,1)=70 (6,2)=70 (6,3)=70 (6,4)=70 (6,5)=86//dolna odległość sierotka pola od najbliższej ściany

#### WEST

(1,1)=0 (1,2)=0 (1,3)=70 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=85 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=70 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=85 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=70 (5,4)=0 (5,5)=0

(6,1)=106.25 (6,2)=255 (6,3)=255 (6,4)=255 (6,5)=255// lewa odległość sierotka pola od najbliższej ściany

#### FIELDS\_SIZE

##### NORTH

(1,1)=0 (1,2)=0 (1,3)=51.5 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=45 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=125 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=45 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=67.5 (5,4)=0 (5,5)=0

(6,1)=70 (6,2)=70 (6,3)=70 (6,4)=70 (6,5)=70// górna odległość sierotka pola od końca pola

##### EAST

(1,1)=0 (1,2)=0 (1,3)=70 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=85 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=70 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=85 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=70 (5,4)=0 (5,5)=0

(6,1)=106.25 (6,2)=106.25 (6,3)=70 (6,4)=30.5 (6,5)=48// prawa odległość sierodka pola od końca pola

##### SOUTH

(1,1)=0 (1,2)=0 (1,3)=51.5 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=45 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=125 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=45 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=67.5 (5,4)=0 (5,5)=0

(6,1)=70 (6,2)=70 (6,3)=70 (6,4)=70 (6,5)=86 //dolna odległość sierodka pola od końca pola

#### WEST



(1,1)=0 (1,2)=0 (1,3)=70 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=85 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=70 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=85 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=70 (5,4)=0 (5,5)=0

(6,1)=106.25 (6,2)=106.25 (6,3)=70 (6,4)=30.5 (6,5)=48// lewa odległość sierodka pola od końca pola

#### NORTH\_DIRECTION

(1,1)=0 (1,2)=0 (1,3)=162 (1,4)=0 (1,5)=0

(2,1)=0 (2,2)=0 (2,3)=204 (2,4)=0 (2,5)=0

(3,1)=0 (3,2)=0 (3,3)=188 (3,4)=0 (3,5)=0

(4,1)=0 (4,2)=0 (4,3)=234 (4,4)=0 (4,5)=0

(5,1)=0 (5,2)=0 (5,3)=216 (5,4)=0 (5,5)=0

(6,1)=234 (6,2)=210 (6,3)=236 (6,4)=212 (6,5)=180//odczyty kierunku N, zmienne,

#### OBSTACLES

(1,1)=5 (1,2)=5 (1,3)=0 (1,4)=5 (1,5)=5

(2,1)=5 (2,2)=5 (2,3)=0 (2,4)=5 (2,5)=5

(3,1)=5 (3,2)=5 (3,3)=0 (3,4)=5 (3,5)=5

(4,1)=5 (4,2)=5 (4,3)=0 (4,4)=5 (4,5)=5

(5,1)=5 (5,2)=5 (5,3)=0 (5,4)=5 (5,5)=5

(6,1)=0 (6,2)=0 (6,3)=0 (6,4)=0 (6,5)=0 //opis które pola są pełne

**Mapa w formacie potrzebnym do wizualizacji,**

**Konwersja z pliku .bmf:**

```
<?xml version="1.0" encoding="utf-8"?><wzór_mapyxmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> // opis formy zapisu pliku
```

```
<ilość_punktów_x>6</ilość_punktów_x><ilość_punktów_y>5</ilość_punktów_y>
```

```
//ilości pól poziomopionowo
```

```
<north><string>1,1=51,5</string><string>1,2=51,5</string><string>1,3=51,5</string><string>1,4=51,5</string><string>1,5=51,5</string><string>2,1=45</string><string>2,2=45</string><string>2,3=45</string><string>2,4=45</string><string>2,5=45</string><string>3,1=70</string><string>3,2=70</string><string>3,3=70</string><string>3,4=70</string><string>3,5=70</string><string>4,1=85</string><string>4,2=85</string><string>4,3=85</string><string>4,4=85</string><string>4,5=85</string><string>5,1=70</string><string>5,2=70</string><string>5,3=70</string><string>5,4=70</string><string>5,5=70</string><string>6,1=106.25</string><string>6,2=106.25</string><string>6,3=70</string><string>6,4=30.5</string><string>6,5=48</string></north>
```

**// górna odległość sierodka pola od końca pola**

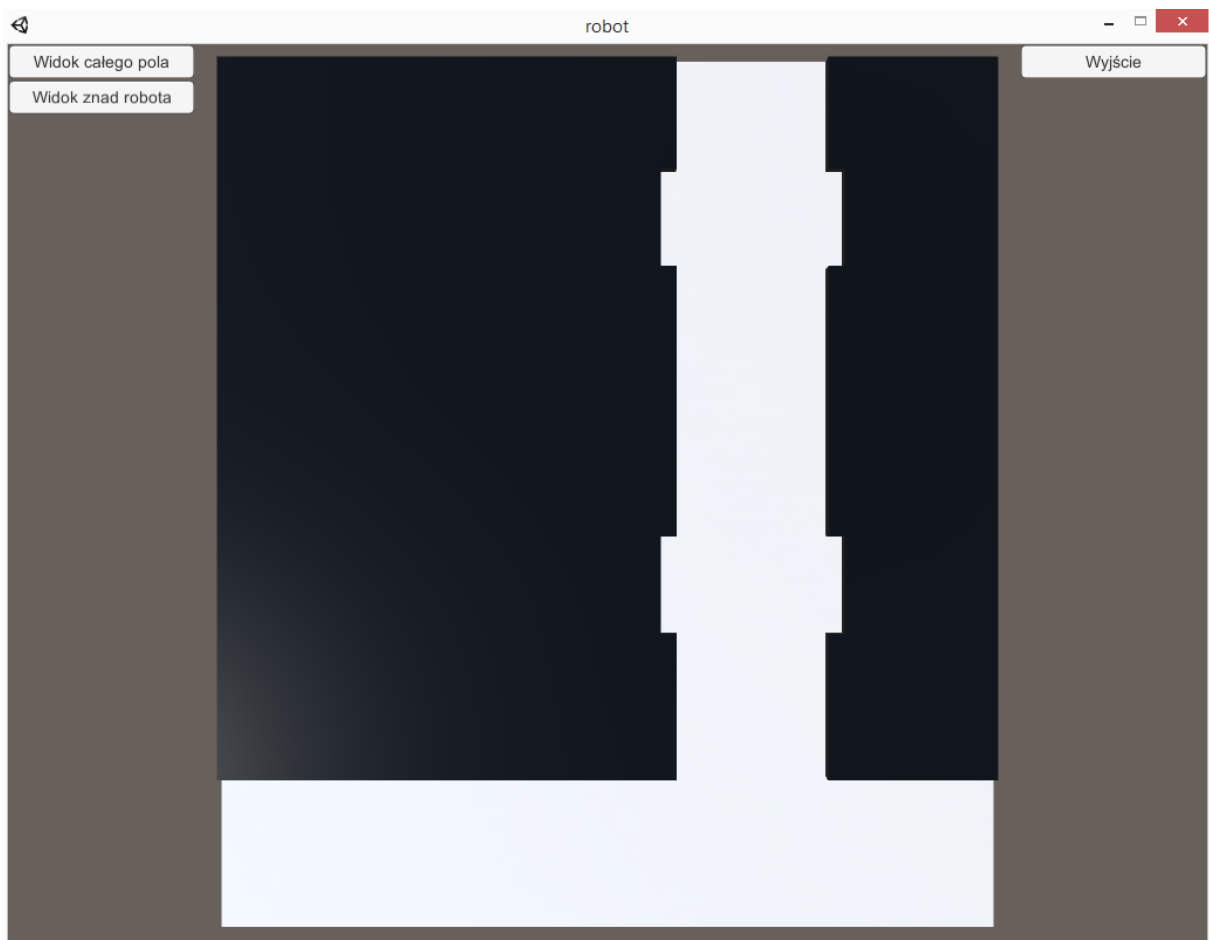
// dolna odległość sierodka pola od końca pola

// prawa odległość sierodka pola od końca pola

// lewa odległość sierodka pola od końca pola

//opis ktore pola sa peŕne

## Zwizualizowana mapa (Screenshot z projektu):



## 6. Testy i analiza

Podczas tworzenia pracy natrafiałem na kilka problemów takich jak zatrzymywanie się programu gdy jego okno nie było aktywne w systemie ten problem został rozwiązany przez stosowne konfiguracje silnika unity przed kompilacją. Innym problemem było nakładanie się warstw elementów mapy co zostało rozwiązane przez zbalansowanie ograniczeń sposobu ich wyświetlania. W systemie negatywne problemy zostały rozwiązane. Generator map jest w pełni kompatybilny z projektem ROBOGUIDE.

## **7.PODSUMOWANIE**

Projekt wizualizacji map budynku zrodził się z mojej aktywności w kole naukowym robotyki. Jednym z interesujących mnie projektów był projekt ROBOGUIDE, polegający na budowie i oprogramowaniu robota mobilnego dedykowanego do oprowadzania po budynku wydziału. Jednym z problemów do rozwiązania było wizualizacja map budynku co stało się tematem mojej pracy magisterskiej. W pracy mamy prezentację gotowego, w pełni funkcjonalnego systemu wizualizacji poruszającego się po wybranym obszarze budynku robota. Filmik demonstrujący działający projekt jest w [5].

# Literatura

[1]Will Goldstone: Projektowanie gier w środowisku Unity 3.x

[2]Projekt koła naukowego robotyki UWM- ROBOGUID link <https://github.com/boxero/robo-guide>

[3] Żmudziński, Ł., Augustyniak, A., Artiemjew, P.: Control of Mindstorms NXT robot using Xtion Pro camera skeletal tracking, In: Technical Sciences, vol. 19(1), pp. 71-81, Olsztyn, UWM Publisher (2016)

[4] P. Artiemjew: The localization of Mindstorms NXT in the magnetic unstable environment based on histogram filtering, In: Proceedings of 7th International Conference on Agents and Artificial Intelligence, ICAART'15, pp. 341-348, Lisbon, Portugal, (2015)

[5] Demonstracja projektu wizualizacji map <https://youtu.be/kp26UCKAtY4>