

Konstruksi Pemrograman C# 2

Pemrograman Game

Content:

- Fungsi dan Kelas
- Basic Movement

Fungsi dan Kelas

Fungsi

- ✓ Fungsi Sederhana
- ✓ Fungsi berparameter
- ✓ Fungsi dengan kembalian (return value)
- ✓ Fungsi Overloading
- ✓ Fungsi dengan Optional Parameter
- ✓ Fungsi dengan 'Param' keyword
- ✓ Fungsi memanggil dirinya sendiri (recursive)

Fungsi Sederhana

```
public class Fungsi_Sederhana : MonoBehaviour {  
    public int jumlahPanggilan;
```

```
    // Update is called once per frame
```

```
    void Update () {  
        HitungPembaharuan();  
    }
```

```
    void HitungPembaharuan()  
    {  
        jumlahPanggilan++;  
  
        print("Pembaharuan "+jumlahPanggilan);  
    }  
}
```

Fungsi Berparameter

```
public class Fungsi_Berparameter : MonoBehaviour {  
  
    // Use this for initialization  
    void Start () {  
        CetakPesan("Game Programming!");  
    }  
  
    void CetakPesan(string pesan)  
    {  
        print(pesan);  
    }  
}
```

Fungsi dengan Return value

```
public class Fungsi_Return_Val : MonoBehaviour {  
    // Use this for initialization  
    void Start()  
    {  
        print("Jumlah: "+Penambahan(3, 2));  
    }  
  
    int Penambahan(int angka1, int angka2)  
    {  
        int jumlah = angka1 + angka2;  
        return jumlah;  
    }  
}
```

Fungsi Overloading

```
public class Fungsi_Overloading : MonoBehaviour {  
  
    void Start()  
    {  
        print("Jumlah: " + Penambahan(3, 2));  
  
        print("Jumlah: " + Penambahan(3.2f, 2.3f));  
    }  
  
    int Penambahan(int angka1, int angka2)  
    {  
        int jumlah = angka1 + angka2;  
        return jumlah;  
    }  
  
    float Penambahan(float angka1, float angka2)  
    {  
        float jumlah = angka1 + angka2;  
        return jumlah;  
    }  
}
```


Fungsi dengan Optional Parameter

```
public class Fungsi_Berparameter : MonoBehaviour {  
  
    // Use this for initialization  
    void Start () {  
        CetakPesan("Game Programming!");  
    }  
  
    void CetakPesan(string pesan)  
    {  
        print(pesan);  
    }  
}
```

Fungsi dengan Param Keyword

```
public class Fungsi_Params : MonoBehaviour {  
  
    // Use this for initialization  
    void Start () {  
  
        print(Penambahan(2,3));  
        print(Penambahan(4,5));  
    }  
  
    int Penambahan(params int[] angka)  
    {  
        int sum = 0;  
  
        foreach (int i in angka)  
        {  
            sum += i;  
        }  
  
        return sum;  
    }  
}
```

Fungsi Rekursive

```
public class Fungsi_Rekursive : MonoBehaviour {
```

```
    // Use this for initialization  
    void Start () {  
        print(Faktorial(5));  
    }
```

```
    int Faktorial(int angka)
```

```
    {  
        if (angka == 0)  
        {  
            return 1;  
        }
```

```
        if (angka < 0)  
        {  
            return 0;  
        }
```

```
        int result = angka * Faktorial(angka - 1);
```

```
        return result;
```

```
    }  
}
```

Kelas

```
BasicMove.cs  + X
BasicMove
+ using ...
- public class BasicMove : MonoBehaviour {
    public float kecepatan = 0.1f;

    // Update is called once per frame
    - void Update () {
        Gerak();
    }

    + public virtual void Gerak() ...

    //property
    + public Vector2 posisi ...
}
```

Kelas: Inheritance

BasicMove2.cs

BasicMove2

```
using UnityEngine;
using System.Collections;

public class BasicMove2 : BasicMove {
    public float kecepatan = 0.1f;

    // Update is called once per frame
    void Update () { ... }

    public override void Gerak() { ... }
}
```

Basic Movement

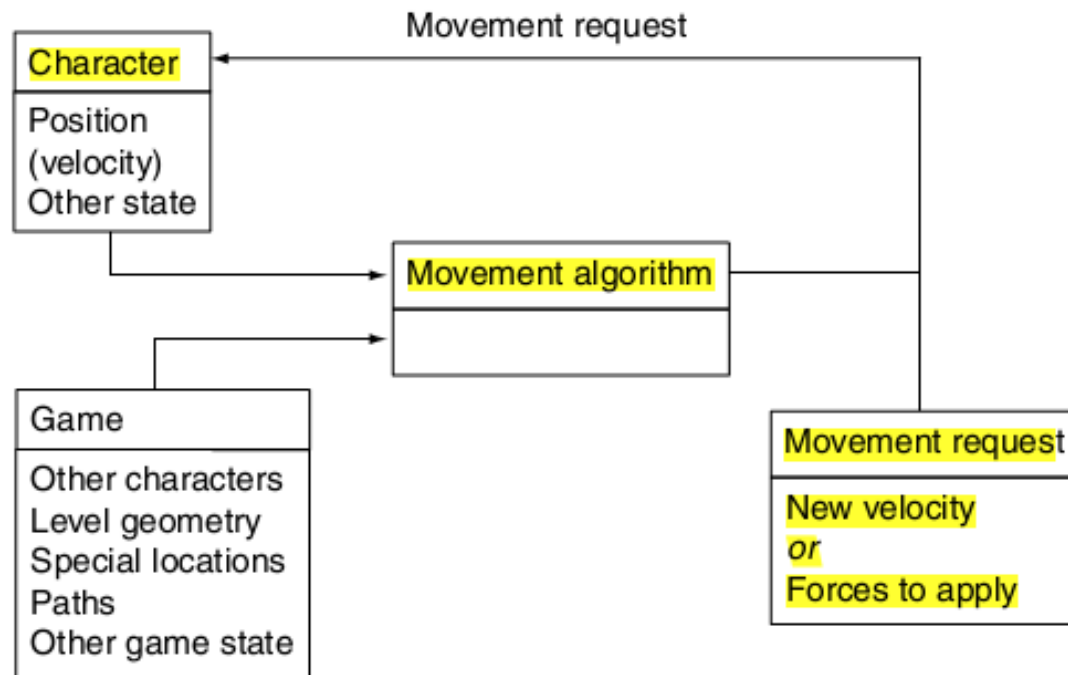
Algoritma Dasar Pergerakan

- Setiap karakter pada Game memiliki *current position* dan properti fisik yang mengatur pergerakan karakter tersebut.
- Algoritma pergerakan mengatur properti karakter untuk menentukan dimana posisi karakter selanjutnya.
- Bentuk dasar algoritma pergerakan:
Mengambil data Geometri *state* karakter dan *state* lingkungan dan menghasilkan data geometri yang menunjukkan perubahan posisi karakter.

Algoritma Dasar Pergerakan

- Beberapa algoritma pergerakan bisa jadi hanya memiliki *input* yang sedikit, misalnya: posisi karakter dan posisi *enemy* yang akan dikejar.
- Pada game yang lawas hanya memiliki dua kecepatan: Bergerak dan diam.
- Contoh algoritma pergerakan: *Steering Behaviour* yang dibuat oleh Craig Reynolds

Algoritma Dasar Pergerakan



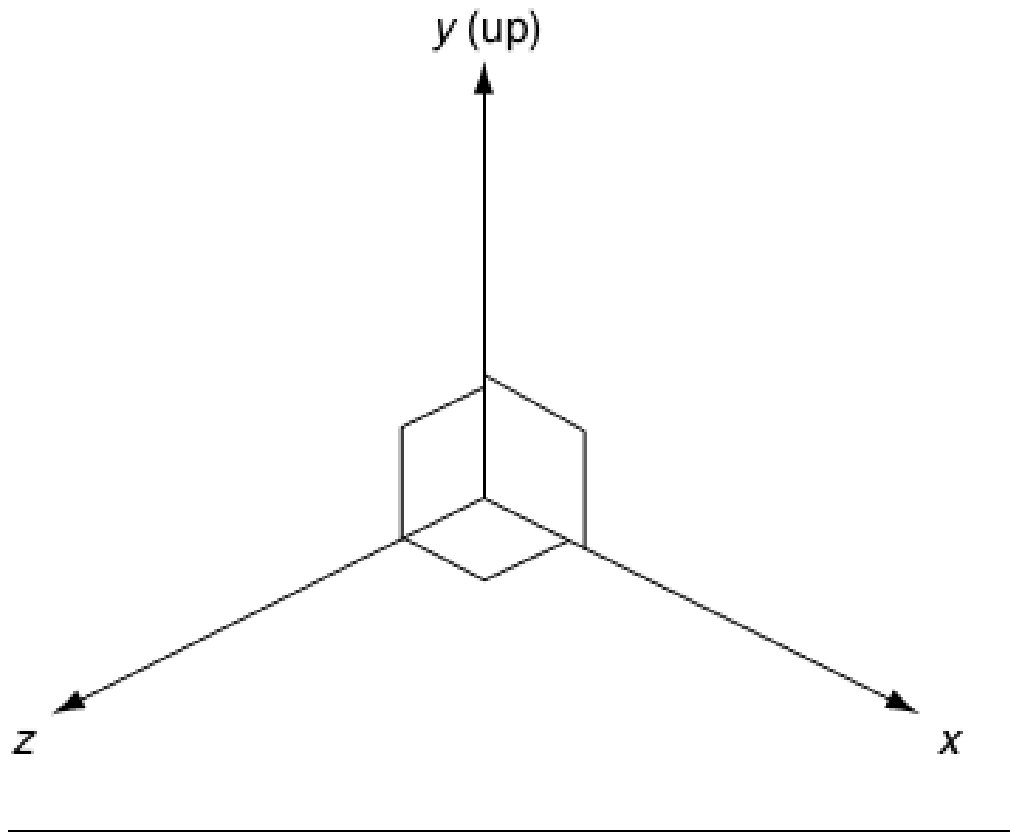
The movement algorithm structure

Algoritma Dasar Pergerakan

Dua tinjauan gerak:

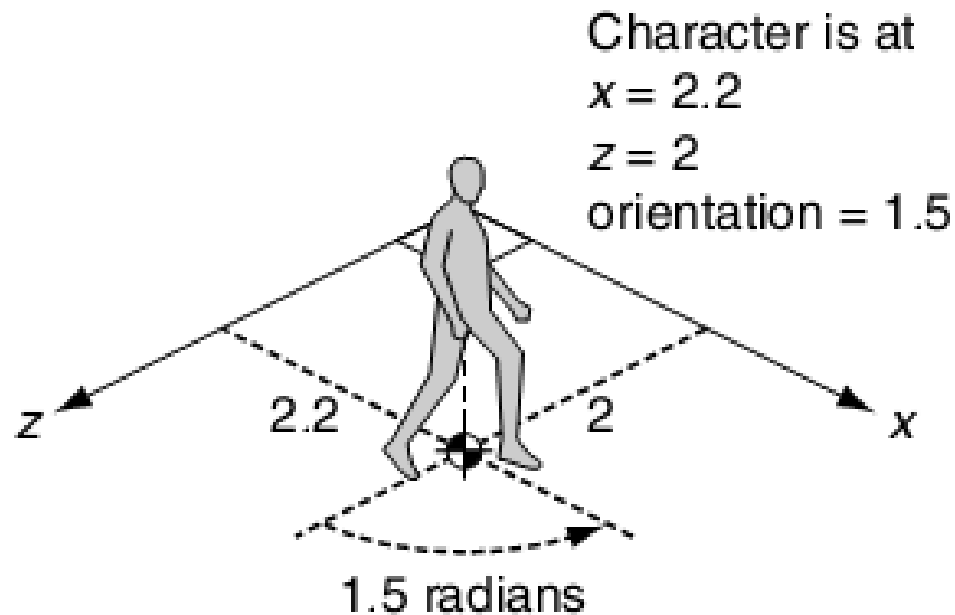
- ❑ Dinamika: Penyebab gerak diperhitungkan
- ❑ Kinematika: Hanya memperhitungkan posisi, karakter dianggap sebagai titik

Kinematika



The 2D movement axes and the 3D basis

Kinematika



The positions of characters in the level

Kinematika

Implementasi kinematika sederhana di UNITY

```
public virtual void Gerak(){  
    Vector2 posisiSementara = posisi;  
    posisiSementara.x += kecepatan*Time.deltaTime;  
    posisi = posisiSementara;  
}
```

```
public override void Gerak(){  
    Vector2 posisiSementara = posisi;  
    posisiSementara.y = Mathf.Sin(Time.time*Mathf.PI*2)*2;  
    posisi = posisiSementara;  
    base.Gerak();  
}
```

Daftar Pustaka

Ian Millington, Artificial Intelligence For Games
, Elseiver 2006