**FLYING BEBEK BERBASIS UNITY-GAME ENGINE**

**Laporan ini disusun untuk mata kuliah: Game Programming II**

**Dosen pengajar: Kurnia Jaya Eliazar, S.T., M.T.**

******

Disusun Oleh:

Nizar Sapta Wiguna (061303012)

Enjang Winandar (061303004)

**SK.Ketua Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT)**

**Nomor : 041/BAN-PT/AK-XIV/S1/XII/2011**

**TEKNIK INFORMATIKA – FAKULTAS TEKNIK**

**UNIVERSITAS WIDYATAMA**

**BANDUNG**

**2017**

# KATA PENGANTAR

Puji syukur kehadirat ALLAH SWT, karena atas perkenan-Nya lah laporan tugas besar dengan judul “Flying Bebek Berbasis Unity-Game Engine” ini dapat selesai dengan baik sesuai pada waktu yang telah ditentukan.

Tugas besar ini memiliki bobot penilaian yang besar pada matakuliah Game Programming II di Universitas Widyatama Bandung. Laporan tugas besar ini disusun sebagai pelengkap tugas besar yang telah mulai dikerjakan dari awal perkuliahan semester 8.

Pada kesempatan ini tidak lupa penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah membantu penulis dalam menyelesaikan laporan tugas besar ini :

* Seluruh keluarga yang telah memberikan do’a dan dorongan baik materil maupun non materil hingga terselesaikannya penulisan laporan kerja praktek ini.
* Bapak Kurnia Jaya Eliazar, S.T., M.T. selaku dosen pembimbing yang sabar membantu penulis hingga laporan kerja praktek ini selesai.
* Rekan-rekan *Teknik Informatika 2013* yang selalu ada saat penulis dalam susah maupun senang serta senantiasa memberi saran, kritik, inspirasi, dan dorongan.
* Semua Pihak yang tidak dapat disebutkan oleh penulis yang telah memberikan bantuan kepada penulis dalam menyelesaikan laporan kerja praktek ini.

Penulis menyadari bahwa masih banyak kekurangan pada laporan ini, baik dari materi maupun teknik penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan.

Akhir kata semoga laporan tugas besar ini dapat memberikan banyak manfaat bagi kita semua.

Bandung, Juni 2017

# DAFTAR ISI

KATA PENGANTAR I

DAFTAR ISI II

DAFTAR GAMBAR III

BAB I PENDAHULUAN 1

1.1. Latar Belakang Masalah 1

1.2. Tujuan 2

1.3. Batasan Masalah 2

1.4. Sistematika Penulisan 2

BAB II ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM 3

2.1. Analisis Kebutuhan 3

2.1.1. Kebutuhan Perangkat Keras 3

2.1.2. Kebutuhan Perangkat Lunak 3

2.2. Perancangan Sistem *4*

2.2.1. Download Asset dari Asset Store 4

2.2.2. Membuat Game Object dan Scripting 5

2.2.3. Membuat Game Object Particle 4

BAB III IMPLEMENTASI DAN TAMPILAN AKHIR 15

3.1. Implementasi 15

3.2. Tampilan Akhir 15

BAB IV PENUTUP 16

4.1. Kesimpulan 16

# DAFTAR GAMBAR

[Gambar 221.1. Animated Coins 4](#_Toc484570540)

[Gambar 221.2. speed tree 5](#_Toc484570541)

[Gambar 221.3. surface texture 5](#_Toc484570542)

[Gambar 222.1. Object dan Script in Game 6](#_Toc484570543)

[Gambar 222.2. Directional Light 7](#_Toc484570544)

[Gambar 222.3. Plane 7](#_Toc484570545)

[Gambar 222.4. Main Camera 8](#_Toc484570546)

[Gambar 222.5. Inspector Player 9](#_Toc484570547)

[Gambar 223.1. Particle System 14](#_Toc484570552)

# BAB I PENDAHULUAN

## 1.1. Latar Belakang Masalah

Teori Permainan adalah suatu pendekatan matematis untuk merumuskan situasi persaingan dan konflik antara berbagai persaingan. Teori ini dikembangkan untuk menganalisa proses pengambilan keputusan dari situasi persaingan yang berbeda dan melibatkan dua atau lebih kepentingan. Teori permainan mula-mula dikemukakan oleh seorang ahli matematika Prancis yang bernama Emile Borel pada tahun 1921. kemudian, John Von Neemann dan Oskar Morgenstern mengembangkan lebih lanjut sebagai alat untuk merumuskan perilaku ekonomi yang bersaing.

Pada perkuliahan Game Programming II ini, mahasiswa dituntut untuk membuat sebuah game sederhana, namun menarik dengan menggunakan teknik-teknik yang telah dipelajari selama perkuliahan. Tools yang digunakan adalah Unity-Game Engine. Game untuk tugas besar ini berbasis Unity.

Game ini akan menggambarkan seekor bebek terbang yang mengumpulkan koin dengan menghindari beberapa halang rintang, yaitu sebuah pohon. Tujuan dari game ini adalah bebek tersebut mengumpulkan koin sebanyak-banyaknya dengan menghindari sebuah pohon yang menghadangnya, jika tidak dapat menghindar, maka bebek tersebut akan mati dan permainanpun gameover.

## 1.2. Tujuan

Tujuan yang ingin dicapai dari Rancang Bangun Mini Game Flying Bebek Berbasis Unity-Game Engine ini adalah sebagai berikut:

* Merancang bangun sebuah game sederhana dan menarik berbasis Unity.
* Menerapkan ilmu-ilmu yang telah dipelajari pada matakuliah Game Programming II di dalam game.

## 1.3. Batasan Masalah

Dalam penyusunan laporan tugas besar ini, kami mencoba membuat batasan-batasan agar tidak terjadi penyimpangan dari tujuan yang diharapkan. Batasan-batasan tersebut adalah sebagai berikut:

* Game sebatas akan dimainkan melalui Unity-Game Engine.
* Audio pada game tidak akan kami prioritaskan.

## 1.4. Sistematika Penulisan

*Bab I Pendahuluan*. Bagian ini berisi tentang penjelasan latar belakang masalah yang ditemukan saat kerja praktek, tujuan masalah, batasan masalah, dan sistematika penulisan laporan.

*Bab II Analisis Kebutuhan dan Perancangan Sistem*. Bagian ini berisi tentang langkah-langkah prosedural yang ditempuh dalam membangun mini game “Run Forrest Run”.

*Bab III Implementasi dan Tampilan Akhir.* Bagian ini berisi tentang manual book dan tampilan akhir dari mini game “Run Forrest Run”.

*Bab IV Penutup*. Bagian ini berisi kesimpulan dari hasil tugas besar.

# BAB II ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

## 2.1. Analisis Kebutuhan

Dalam pembuatan game ini digunakan perangkat yang dibagi menjadi dua perangkat, yaitu perangkat keras (hardware) dan perangkat lunak (software).

### 2.1.1. Kebutuhan Perangkat Keras

Perangkat keras *(hardware)* yang digunakan pada saat perancangan dan pengujian system yaitu sebagai berikut :

1. *Processor* : AMD A8-6410 APU with Radeon R5 Graphics (4Cpus), 2.70 GHz
2. *Memory* : 4096MB RAM
3. *Video Card* : AMD Radeon(TM) R5 Graphics
4. *Monitor*
5. *Mouse*
6. *Keyboard*

### 2.1.2. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan oleh kami untuk merancang bangun game ini adalah sebagai berikut:

|  |  |
| --- | --- |
| **Nama Perangkat Lunak** | **Fungsi** |
| Windows 7 | Sebagai *platform* sistem operasi |
| Unity-Game Engine v5.6.0f3 Personal (64bit | Aplikasi yang digunakan untuk membuat game |

## 2.2. Perancangan Sistem

Perancangan sistem pada tugas besar ini akan mencakup beberapa aspek, yaitu:

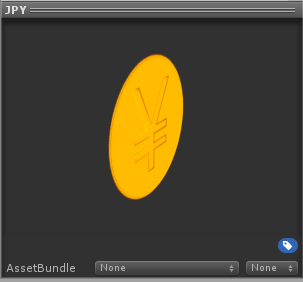
* Download asset dari asset store.
* Membuat game object dan scripting.
* Membuat game object particle.

### 2.2.1. Download Asset dari Asset Store

Dalam pembuatan game untuk tugas besar ini, kami menggunakan beberapa asset gratis yang ada di asset store Unity. Asset-asset gratis yang kami gunakan adalah:

* **Animated Coins**

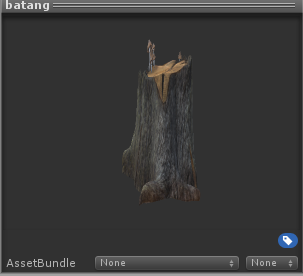
Asset ini menyediakan object koin yang digunakan untuk dikumpulkan player (gambar 221.1).



Gambar 221.1. Animated Coins

* **Speed Tree**

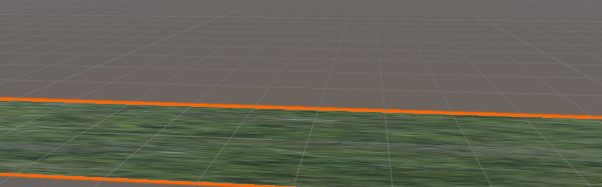
Asset ini menyediakan object pohon yang digunakan sebagai halang rintang dalam game ini.(gambar 221.2).



Gambar 221.2. Speed Tree

* **Surface Texture**

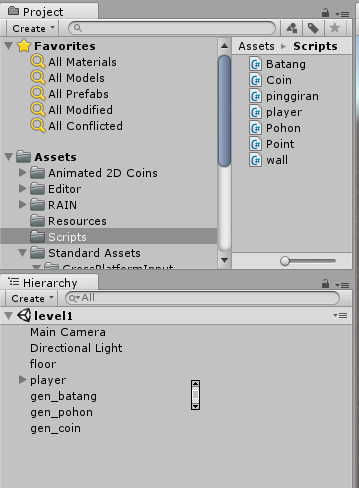
Asset ini menyediakan texture tanaman yang digunakan untuk area game (gambar 221.3).



Gambar 221.3. Surface Texture

### 2.2.2. Membuat Game Object dan Scripting

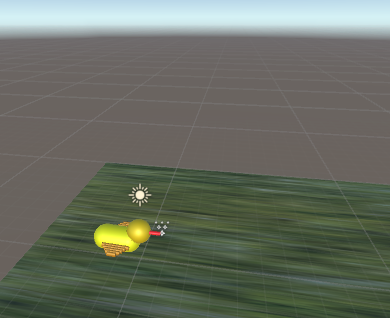
Pembuatan game menggunakan Unity-Game Engine, tentunya dibutuhkan object dan scripting. Disini kami akan menjelaskan object dan script apa saja yang digunakan di dalam game “Flying Bebek” (gambar 222.1).



Gambar 222.1. Object dan Script in Game

* **Directional Light**

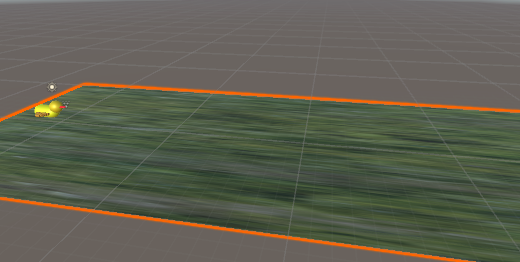
Untuk pencahayaan pada game. Directional Light ini akan berada tepat diatas kepala “Player” dan akan mengikuti kemanapun “Player” berpindah tempat (gambar 222.2).



Gambar 222.2. Directional Light

* **Plane**

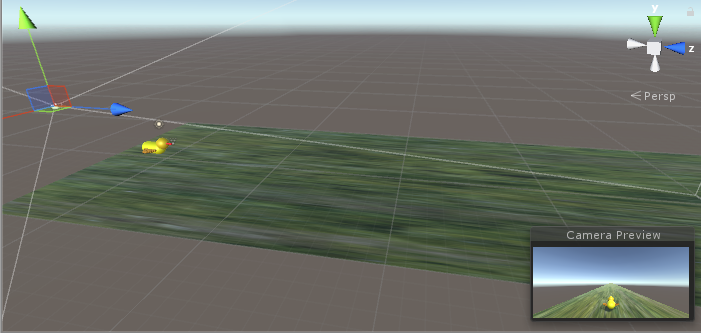
Untuk pijakan awal “Player” ketika game akan dimulai (gambar 222.3).



Gambar 222.3. Plane

* **Main Camera**

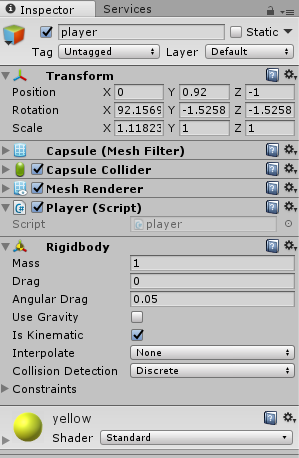
Untuk mengatur sudut pandang pemain ketika game dimainkan (gambar 222.4).



Gambar 222.4. Main Camera

* **Player**

Untuk object “Player” di dalam game. Object ini akan dapat pemain kendalikan, seperti bergerak keatas atau kebawah dan kearah kiri atau kanan. Di dalam object “Player” ini terdapat CharacterController dan beberapa script yang membuat object ini memiliki collision, dapat dikendalikan oleh pemain, dan memiliki efek particle ketika pemain menekan tombol tertentu atau object bersentuhan dengan object lainnya (gambar 222.5).



Gambar 222.5. Inspector Player

* + **Player**

using UnityEngine;  
using System.Collections;  
  
public class player : MonoBehaviour {  
  
    public AudioClip coinCollectSound;  
    private uint coins = 0;  
    public float speed = 5.0f;  
  
    *// Use this for initialization*  
    void Start() {  
          
    }  
      
    *// Update is called once per frame*  
    void Update () {  
        if(Input.GetKeyDown(KeyCode.RightArrow) && transform.position.x < 5){  
  
            transform.position += Vector3.right;  
        }  
  
        if(Input.GetKeyDown(KeyCode.LeftArrow) && transform.position.x > -5){  
  
            transform.position += Vector3.left;  
        }  
  
        if(Input.GetKeyDown(KeyCode.UpArrow)&& transform.position.y < 3){  
  
            transform.position += Vector3.up;  
        }  
  
        if(Input.GetKeyDown(KeyCode.DownArrow)&& transform.position.y > 1){  
  
            transform.position += Vector3.down;  
        }  
      
    }  
  
    void CollectCoin(Collider coinCollider)  
    {  
        coins++;  
  
        Destroy(coinCollider.gameObject);  
  
        AudioSource.PlayClipAtPoint(coinCollectSound, transform.position);  
    }  
}

* + **Script Batang**

using UnityEngine;  
using System.Collections;  
  
public class Batang : MonoBehaviour {  
  
    public int spawnTime;  
  
    private int timer;  
  
    *// Use this for initialization*  
    void Start () {  
      
        batang();  
    }  
      
    *// Update is called once per frame*  
    void Update () {  
  
        timer++;  
  
        if(timer >= spawnTime){  
  
            batang();  
            timer = 0;  
        }  
      
    }  
  
    void batang(){  
  
        Instantiate (Resources.Load ("batang"));  
    }  
          
}

* + **Script Coin**

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Coin : MonoBehaviour {  
    public int spawnTime;  
  
    private int timer;  
  
    *// Use this for initialization*  
    void Start () {  
  
        coin();  
    }  
  
    *// Update is called once per frame*  
    void Update () {  
  
        timer++;  
  
        if(timer >= spawnTime){  
  
            coin();  
            timer = 0;  
        }  
  
    }  
  
    void coin(){  
  
        Instantiate (Resources.Load ("coin"));  
    }  
}

* + **Script Pinggiran**

using UnityEngine;  
using System.Collections;  
  
public class pinggiran : MonoBehaviour {  
public Vector3 spawnPoint;  
  
    private Transform camPos;  
    *// Use this for initialization*  
    void Start () {  
  
        transform.position = new Vector3(7, spawnPoint.y, spawnPoint.z);  
          
    }  
      
    *// Update is called once per frame*  
    void Update () {  
  
        transform.position += Vector3.back\*0.5f;  
      
    }  
}

* + **Script Pohon**

using UnityEngine;  
using System.Collections;  
  
public class Pohon : MonoBehaviour {  
  
    public int spawnTime;  
  
    private int timer;  
  
    *// Use this for initialization*  
    void Start () {  
      
        batang();  
    }  
      
    *// Update is called once per frame*  
    void Update () {  
  
        timer++;  
  
        if(timer >= spawnTime){  
  
            batang();  
            timer = 0;  
        }  
      
    }  
  
    void batang(){  
  
        Instantiate (Resources.Load ("pohon"));  
    }  
}

* + **Script Wall**

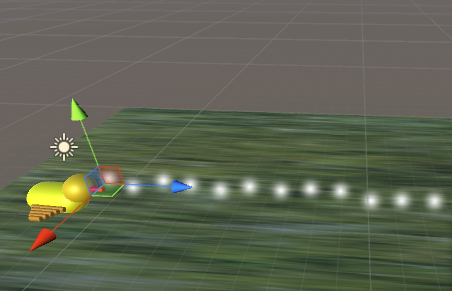
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class Point : MonoBehaviour {  
  
    public Vector3 spawnPoint;  
  
    private Transform camPos;  
    *// Use this for initialization*  
    void Start () {  
        int ranXPos = Random.Range(-5,5);  
        int ranYPos = Random.Range(2,5);  
        transform.position = new Vector3(ranXPos, ranYPos, spawnPoint.z);  
        camPos = GameObject.Find ("Main Camera").GetComponent<Transform> ();      
    }  
  
    *// Update is called once per frame*  
    void Update () {  
  
        transform.position += Vector3.back\*0.5f;  
  
        if(transform.position.z < camPos.position.z){  
  
        }  
  
    }  
}

* + **Script TesHit**

using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class testhit : MonoBehaviour {  
    public Transform canvas;  
  
    public void OnTriggerEnter(Collider other)  
    {          
        if(other.gameObject.tag == "batang")  
        Debug.Log("hit");  
        gameObject.SetActive(false);  
        canvas.gameObject.SetActive(true);  
    }  
}

### 2.2.3. Membuat Game Object Particle

Salah satu ilmu yang dipelajari pada matakuliah Game Programming II adalah object particle effects. Oleh karena itu, penulis menambahkan particle system pada game ini.



Gambar 223.1. Particle System

# BAB III IMPLEMENTASI DAN TAMPILAN AKHIR

## 3.1. Implementasi

Cara memainkan mini game “FLYING BEBEK” ini sangat sederhana. Pemain cukup menggerakan player ke arah kanan dan kiri serta ke atas dan ke bawah menggunakan arah pada keyboard.

## 3.2. Tampilan Akhir

Untuk tampilan akhir mini game “Flying Bebek” dapat dilihat di [youtube](https://youtu.be/6igON5tyQRM): https://youtu.be/W9EGmnXMJZA

# BAB IV PENUTUP

## 4.1. Kesimpulan

Setelah kami menyelesaikan tugas besar ini, kami dapat menyimpulkan beberapa point, yaitu:

* Unity-Game Engine adalah sebuah tools yang sangat powerfull dalam membangun sebuah game, namun tetap dibutuhkan investasi waktu untuk mempelajari cara penggunaannya.
* Metode pembelajaran dengan cara mempraktekan apa yang kita pelajari ke dalam sebuah project sangatlah efektif. Kami menjadi mengerti bagaimana mengimplementasikan teori yang sudah kami pelajari ke dalam sebuah project.
* Pentingnya Github (atau layanan sejenis) dalam mengerjakan sebuah project yang dikerjakan bersama-sama. Kami mengerjakan project ini tanpa menggunakan layanan Github atau sejenisnya, dan hasilnya kami harus bekerja dua kali, kami membuang banyak waktu serta tenaga hanya untuk menggabungkan pekerjaan kami.