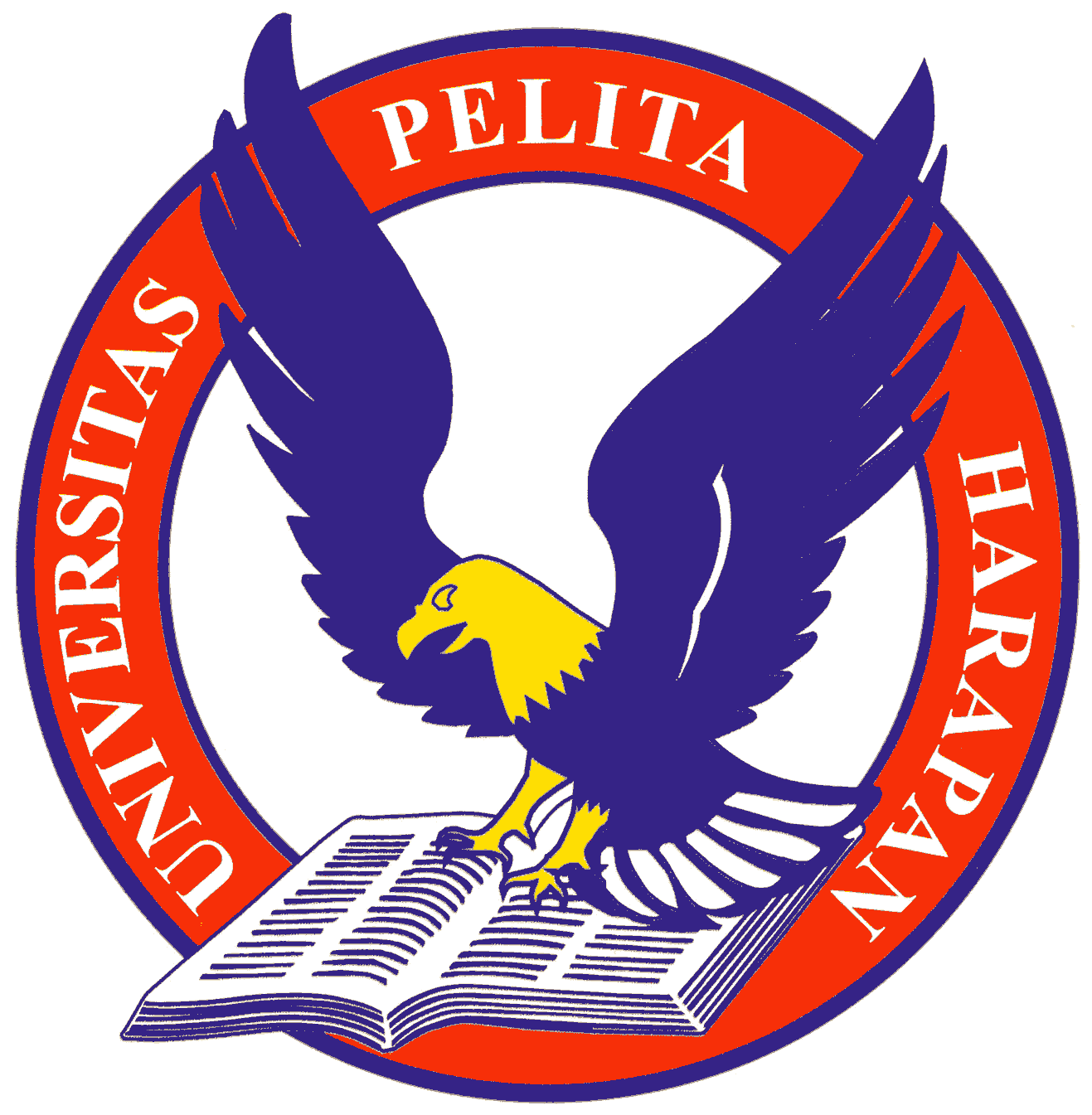
**MAZEBALL**

**PEMROGRAMAN MOBILE**

****

**Disusun oleh:**

Dustin Pradipta 00000011798

Williem Citralin 00000014292

**TEKNIK INFORMATIKA**

**UNIVERSITAS PELITA HARAPAN**

**2017**

**Daftar Isi**

[BAB I](#_gjdgxs)

[PENDAHULUAN 2](#_30j0zll)

[1.1. Latar Belakang 2](#_1fob9te)

[1.2. Tujuan 2](#_3znysh7)

[1.3. Batasan Masalah 3](#_2et92p0)

[BAB II](#_tyjcwt)

[ANALISA DAN PERANCANGAN APLIKASI 4](#_3dy6vkm)

[2.1. Deskripsi Aplikasi 4](#_1t3h5sf)

[2.2. Fitur 4](#_2s8eyo1)

[2.3. Tahapan Penggunaan Aplikasi 5](#_17dp8vu)

[2.4. Flowchart 6](#_3rdcrjn)

[2.5. Database 6](#_26in1rg)

[BAB III](#_lnxbz9)

[IMPLEMENTASI APLIKASI 9](#_35nkun2)

[3.1. Implementasi Source Code 9](#_1ksv4uv)

[3.2. Tampilan Aplikasi 13](#_44sinio)

[BAB IV](#_2jxsxqh)

[PENUTUP 18](#_z337ya)

[4.1. Kesimpulan 18](#_3j2qqm3)

[4.2. Saran Pengembangan 18](#_1y810tw)

[REFERENSI 19](#_4i7ojhp)

# BAB I

# PENDAHULUAN

## 1.1. Latar Belakang

Di zaman yang semakin berkembang, teknologi adalah salah satu aspek yang mengena hampir di seluruh aspek kehidupan lainnya. Teknologi masuk ke dalam sehari-hari, menjadi jembatan yang mempermudah kegiatan yang manusia lakukan. Sekarang ini hampir setiap orang memiliki perangkat *mobile* karena kegunaan dan kemudahannya untuk di bawa kemana-mana.

Perkembangan teknologi tentu termasuk ke dalam perkembangan perangkat *mobile*. Baik perangkat kerasnya yang semakin lama semakin canggih, perangkat lunak juga turut dikembangkan sedemikian rupa untuk membantu kehidupan manusia. Dalam perkembangannya saat ini, perangkat *mobile* umumnya memiliki sensor untuk mendeteksi perubahan yang tertangkap oleh perangkat tersebut.

Kenyataan yang ada sekarang ini adalah sensorterdapat pada perangkat *mobile*, namun belum tentu pengguna dari perangkat *mobile* sadar akan keberadaan sensor tersebut. Salah satu sensor yang umumnya ada pada perangkat *mobile* adalah sensor *accelerometer.* Sensor ini menghitung akselerasi yang terjadi pada perangkat.

Untuk menjawab permasalahan di atas, diperlukan adanya peningkatan kesadaran terhadap pengguna baik secara langsung maupun tidak langsung terhadap sensor tersebut. Sensor yang ditekankan pada proyek ini adalah sensor *accelerometer.* Maka dari itu, penulis membangun *prototype* dari aplikasi yang mengimplementasikan sensor *accelerometer.* Aplikasi dibuat dengan bentuk permainan (*game*) dengan desain yang interaktif sehingga pengguna tertarik untuk memainkannya dan secara tidak langsung diharapkan rasa penasaran dan kesadaran pengguna terhadap sensor *accelerometer* dalam perangkat *mobile*-nya meningkat.

## 1.2. Tujuan

Tujuan pembuatan aplikasi MazeBall adalah sebagai berikut.

1. Membuktikan implementasi sensor *accelerometer* dalam permainan yang dibuat pada perangkat *mobile.*
2. Meningkatkan kesadaran pengguna perangkat *mobile* secara tidak langsung akan sensor yang menjadi fitur dari perangkatnya.
3. Memenuhi syarat Ujian Akhir Semester dari Mata Kuliah *Mobile Programming* Teknik Informatika Universitas Pelita Harapan semester 7.

## 1.3. Batasan Masalah

Batasan masalah dalam pembuatan aplikasi MariMakan ini adalah sebagai berikut.

1. Aplikasi dibuat dengan bantuan *tools* Unity dan Visual Studio.
2. *Asset* pada aplikasi dibantu dengan gabungan *asset* yang dicari dari beberapa sumber penyedia *asset* dalam Unity *asset store*.
3. Aplikasi yang dibuat untuk dijalankan pada sistem operasi Android versi 5.0 ke atas.
4. Aplikasi yang dibuat berupa *prototype game* yang belum memiliki pilihan menu.
5. *Scene* gameover tidak memiliki delay yang kasat di mata.
6. *Input accelerometer* masih belum sempurna.
7. Terdapat *bug jump* yang menyebabkan bola keluar dari labirin dan apabila keluar dari lapangan, maka permainanakan *stuck* di mana bola akan terus melayang dan tidak bisa digerakkan*.*

# BAB II

# GARIS BESAR PERMAINAN

## 2.1. Konsep Permainan

MazeBall adalah sebuah aplikasi berbasis Android yang berupa *prototype game.* Konsep dari *game* ini adalah mencari jalan keluar dari labirin dan mendapatkan coin yang ada di garis akhirnya dengan menggerakan objek bola memakai implementasi sensor *accelerometer.*

## 2.2. *Genre* Permainan

*Genre* utama dari permainan ini adalah petualangan, di mana pengguna akan menggerakan bola untuk mencari jalan keluar dari labirin. Permainan ini bersifat kasual yang diselesaikan dalam waktu yang sebentar dan digunakan sebagai relaksasi.

## 2.3. Target Pengguna

Target pengguna yang ditujukan dalam permainan ini adalah semua usia.

## 2.4. Mekanika Permainan

Berikut adalah mekanika yang dimiliki oleh aplikasi MazeBall.

1. **Menampilkan (*Spawn*) bola**

Saat aplikasi dijalankan, bola akan di-*draw* sehingga terlihat wujud oleh pengguna.

1. **Membuat bola melakukan aksi lompat**

Di dalam permainan, *canvas* yang bersifat statik ditampilkan. *Canvas* tersebut berguna untuk menyimpan segala objek tidak bergerak supaya dapat terus dilihat dari sisi pengguna. Pada *canvas* terdapat *jump button* sebagai *trigger* untuk dilakukannya *force* pada bola terhadap parameter vector3 ke arah atas yang membuat bola tampak melompat.

1. **Bertubrukan *(collide*) dengan objek penghalang**

Dalam pergerakannya, objek bola dapat terjadi *collision* dengan objek lain. *Collision* yang terjadi membuat bola tidak bisa bergerak keluar area yang dihalangi objek.

1. **Mendapatkan *coin* dan permainan selesai**

Coin merupakan objek *collision* yang menjadi objektif garis akhir bagi aplikasi MazeBall. Saat terjadi *collision*, objek koin akan di-*destroy* dan akan muncul *scene* gameover.

# 

# 

# BAB III

# IMPLEMENTASI APLIKASI

**3.1. Implementasi *Source Code***

Scene utama dari game ini adalah scene “maze” sedangkan ketika gameover maka scene “gameover” akan dipanggil.

Kami menggunakan fungsi Quaternion.Euler pada unity untuk merepresentasikan rotasi menggunakan accelerometer pada perangkat android.

**Quaternion:**

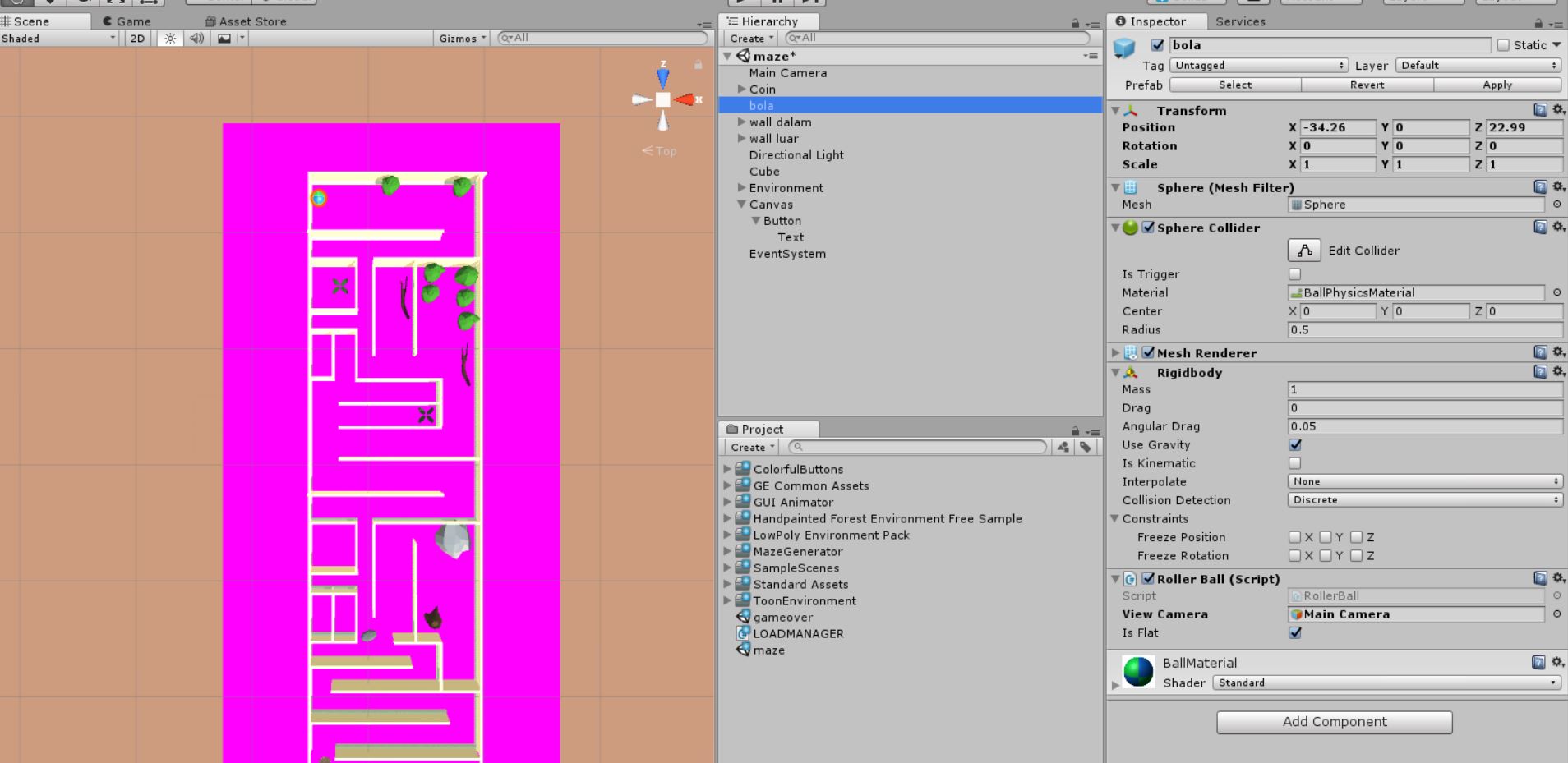
Quaternion adalah nomor kompleks dengan w sebagai *real part* dan x, y, z sebagai *imaginary parts.* Jika sebuah quaternion merepresentasikan sebuah rotasi, maka w=cos(theta / 2) di mana tetha adalah sudut rotasi dibanding axis dari quaternion. Quaternion digunakan sebagai representasi rotasi.

Untuk objek berada pada sudut A dan representasi axis (x, y, z), maka Quaternion (Q) adalah:

Q= [x, y, z, w]

Q = [sin(A/2\*x, sin(A/2)\*y, sin(A/2)\*z, cos(A/2)]

**3.1.1. Ball**

****

Objek bola adalah inti dari permainan ini, pada objek bola dipasang beberapa komponen penting seperti:

-Sphere collider: Agar bola dapat bertabrakan dengan objek lain.

-RigidBody : Agar bola mempunyai mass, dapat terkena gravitation , dan juga Angular Drag.

-Script RollerBall.cs dipasang di objek bola agar bola dapat melakukan beberapa hal yaitu(semua ini ada di fungsi update()) :

**-Agar kamera mengikuti pergerakan bola**

ViewCamera adalah objek gamecomponent camera. Transform.position adalah posisi dari objek transform(bola) dan position objek(bola).

if (ViewCamera != null) {

Vector3 direction = (Vector3.up\*2+Vector3.back)\*2;

Debug.DrawLine(transform.position,transform.position+direction,Color.red);

ViewCamera.transform.position = transform.position+direction;

ViewCamera.transform.LookAt(transform.position);

}

Debug.DrawRay (transform.position + Vector3.up, tilt, Color.cyan);

}

**-Dapat bergerak sesuai dengan gerak accelerometer**

public bool isFlat=true;

Vector3 tilt = Input.acceleration\*3;

if (isFlat)

tilt = Quaternion.Euler (90, 0, 0) \* tilt;

rigidbody.AddForce (tilt);

Vector3 merupakan sebuah *struct* di pustaka UnityEngine yang merepresentasikan vektor dan titik 3D.

Quaternion digunakan untuk merepresentasikan rotasi, sedangkan Euler adalah sudut. Parameter utama rotasi dengan sudut Euler yang digunakan ada 3, yaitu (x,y,z) dengan x untuk nilai rotasi di sumbu x, y untuk nilai rotasi di sumbu y, dan z untuk rotasi di sumbu z.

**-Dapat terlaksana ketika bola bersentuhan dengan *coin***

void OnTriggerEnter(Collider other) {

if (other.gameObject.tag.Equals ("Coin")) {

Destroy(other.gameObject);

SceneManager.LoadScene ("gameover");

}

Setelah koin terkena maka kelas loadmanager me-load scene baru yaitu scene “maze”, ini menyebabkan setelah koin terkena oleh bola maka bola akan kembali ke tempat awal permainan.

public class LOADMANAGER : MonoBehaviour {

void Start()

{

StartCoroutine(MyCoroutine());

}

IEnumerator MyCoroutine()

{

yield return 3; //Wait one frame

SceneManager.LoadScene ("maze");

}

}

**-Dapat lompat ketika tombol jump ditekan.**

public void onJumpClick()

{

if (rigidbody.position.y <=0.5) {

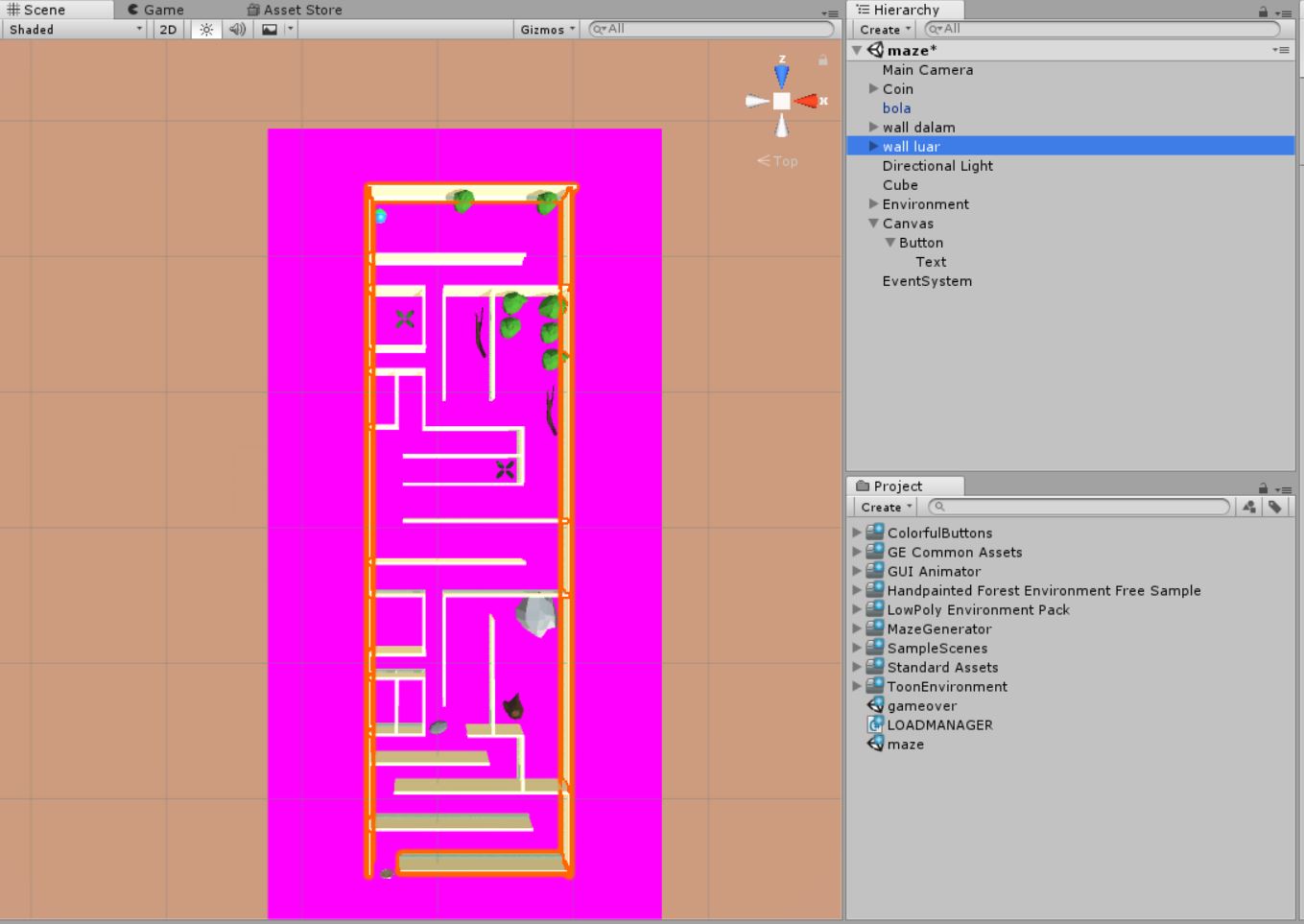
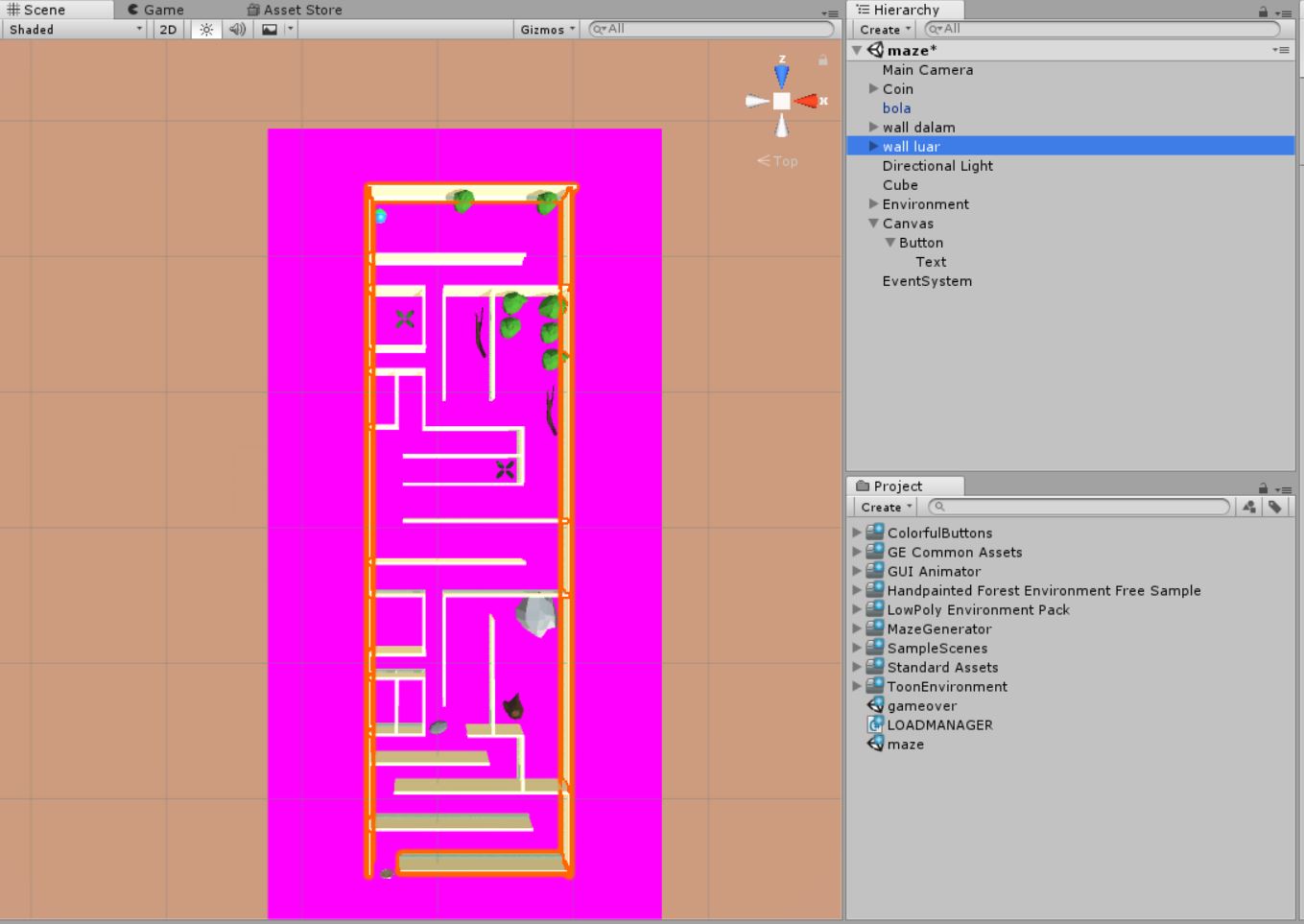
rigidbody.AddForce (Vector3.up \* 300);

}

}

**3.1.2. Map**

**-Labyrinth**

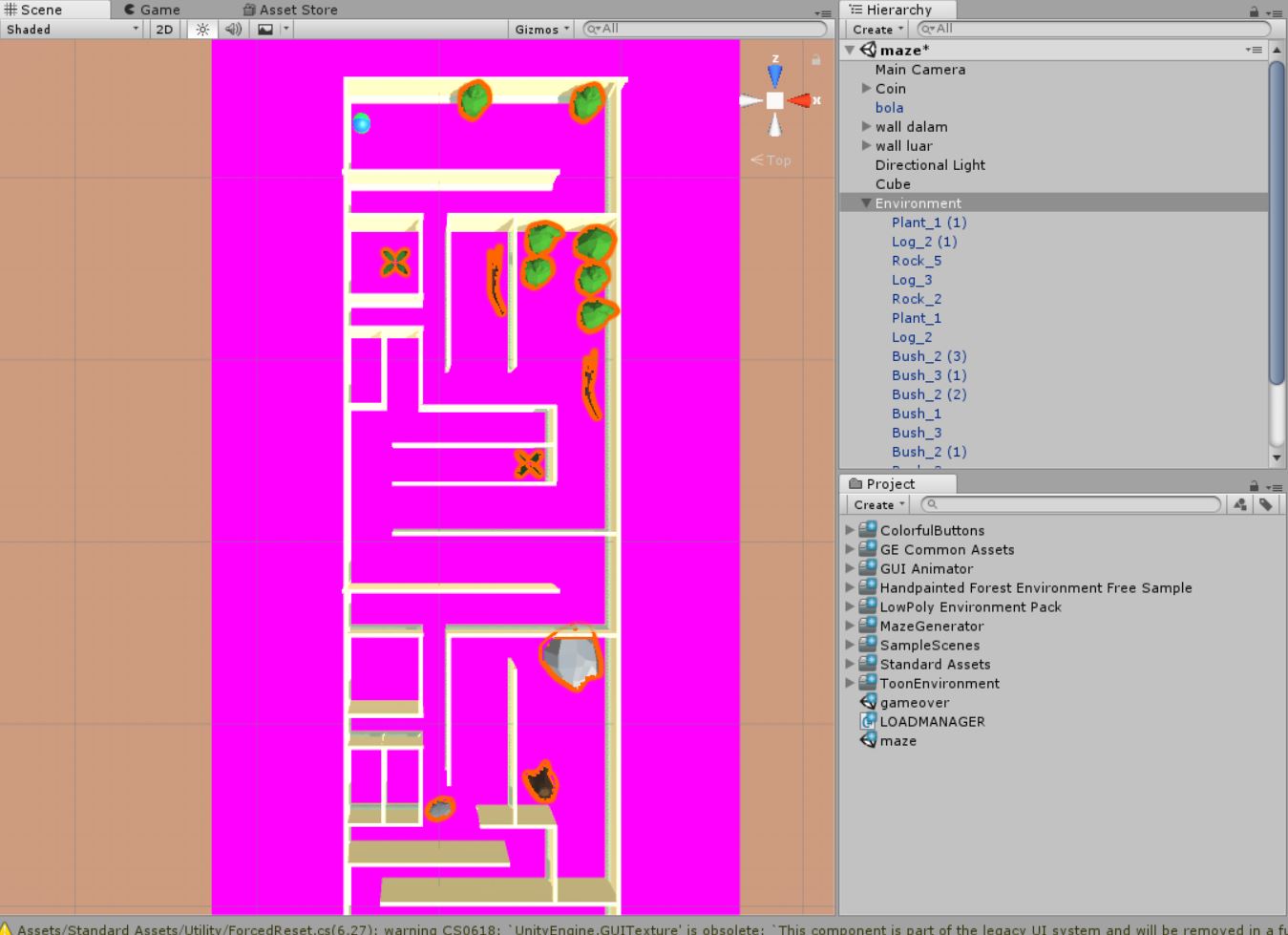
****

Labirin game ini, kami megambil asset tembok/wall dari unity store lalu kami taruh satu persatu di map, dan agar bola dapat bertabrakan dengan tembok tersebut ditambahkan mesh collider pada setiap objek tembok di dalam game.

**-Floor**

Untuk lantai dari game digunakan 3d objek yaitu box dan ditambahkan box collider agar bola tidak menembus ke lantai tersebut.

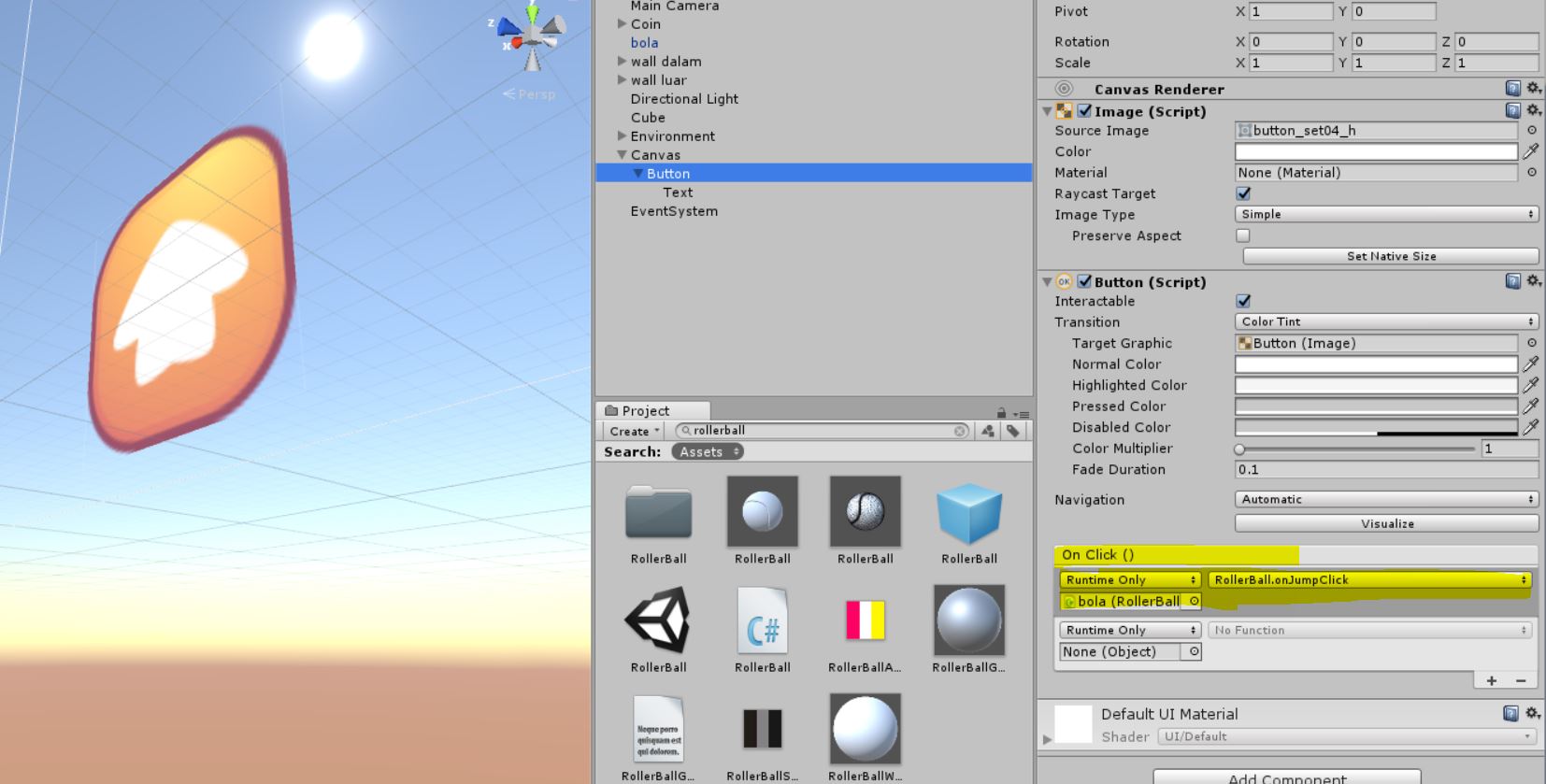
**-Environment(pohon, batu, batang kayu, dan daun)**

****

Semuanya kami ambil dari unity store yang bernama LowPolyEnvironmentpack lalu kami taruh satu-persatu di map game.

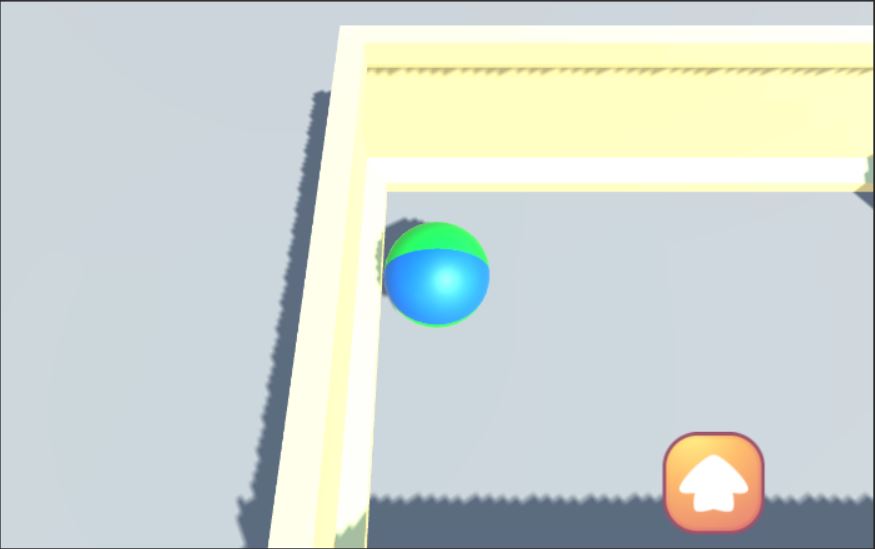
**3.1.3. Button**

Untuk tombol kami menggunakan GameObject tipe Button, lalu kali menambahkan list baru yaitu fungsi on Click() sehingga ketika tombol tersebut dipencet maka fungsi RollerBall.onJumpClick akan dilakukan.



**Screenshot:**

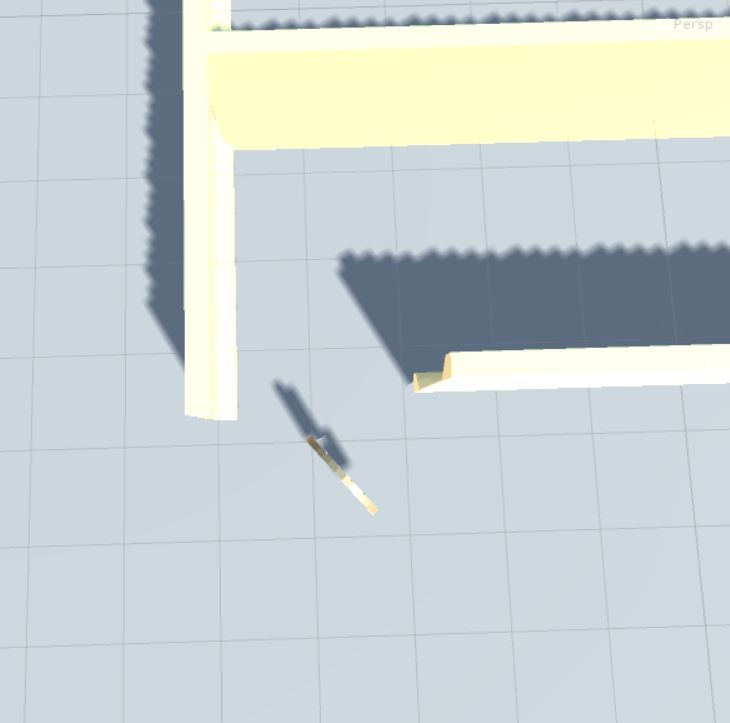
**Ball spawn location**

****

**Gameover scene**

****

**Coin**

****

# BAB IV

# PENUTUP

## 4.1. Kesimpulan

Aplikasi game MazeBall yang telah kami buat telah berhasil memenuhi tujuan dari pembuatan aplikasi game ini. Game ini menunjukan bahwa sensor *accelerometer* yang ada di perangkat mobile dapat digunakan untuk bermain game. Sebagai aplikasi permainan yang mengutamakan *accelerometer*, kami menyadari bahwa kontrol accelerometer haruslah baik dan responsif agar permainan dapat dimainkan dengan menyenangkan.

## 4.2. Saran Pengembangan

Aplikasi game MazeBall masih belum sempurna karena keterbatasan waktu yang ada. Maka dari itu, berikut merupakan saran pengembangan terhadap aplikasi yang telah kami buat.

1. Percantik design, tambahkan main menu, dan tingkatkan level *user-friendly* nya.
2. Implementasi *auto-generate* pada labirin sehingga *stage* tidak monoton.
3. Konfigurasi fungsi *jump* sehingga tidak terjadi *out-of-bond.*
4. Dapat lebih menunjukkan *accelerometer*nya.

# REFERENSI

Explanation of Quaternion - <https://answers.unity.com/questions/645903/please-explain-quaternions.html>

Quaternion.Euler - <https://docs.unity3d.com/ScriptReference/Quaternion.Euler.html>

Euler (gimbal lock) explained - <https://www.youtube.com/watch?v=zc8b2Jo7mno>

What is affected by the Q in Quaternion - <https://answers.unity.com/questions/147712/what-is-affected-by-the-w-in-quaternionxyzw.html>

Ball, coin, & wall assets - <https://www.assetstore.unity3d.com/en/#!/content/38689>

Environment assets - <https://www.assetstore.unity3d.com/en/#!/content/22320>