**BrickBreaker Game**



**Step 1: Create New Project “BrickBreaker”**

File -> New Project (check Standard Assets(Mobile).unityPackage)



Then save the scene following below instruction. Change name of scene to “MainScene”.

File -> Save Scene

**Step 2: Set Main Camera**

The position, rotation and scale of Main Camera are showed following figure.



Change background color whatever you want.

Change projection perspective -> orthographic

Size = 20

Clipping Planes : Near = 0.3 Far = 25



**Step 3: Create Directional Light**

GameObject-> Create Other -> Directional Light



**Step 4: Create Walls**

Create new cube following this instruction:

GameObject -> Create Other -> Cube

Change Cube Name to “ WallLeft”.

“WallLeft” position, rotation and scale are following below figure



Change “Main Color”.

Create material from the “Project” section.

Create -> Material

Name Material “wallMat” and change color. Then drag “wallMat” to the WallLeft.



Create new cube following this instruction:

GameObject -> Create Other -> Cube

Change Cube Name to “ WallRight”.

“WallRight” position, rotation and scale are following below figure



Change ‘Main Color”

Drag “wallMat” to the WallRight.



Create new cube following this instruction:

GameObject -> Create Other -> Cube

Change Cube Name to “ WallTop”.

“WallTop” position, rotation and scale are following below figure



Change ‘Main Color”

Then drag “wallMat” to the WallTop.



**Step 5: Create Bricks**

GameObject -> Create Other -> Cube

Change cube name to “brick”.

“brick” position, rotation and scale are following below figure.



Create new material from “Project” section and name “brickMat” and change color.

Then drag to “brick”.



Then copy “brick” 12 times following below figure.



**Step 6: Create Ball**

Windows -> Asset Store

Type ball pack and enter.

Then click download button and import button to import to project this model.



Then choose “eyeball” from the ball pack and drag to “scene”.

“EyeBall” position, rotation and scale are following below figure.



Add Rigidbody to “EyeBall” following below instruction.

Component -> Physics -> Rigidbody

Mass = 0.01

Uncheck “Use Gravity”

Check Freeze Position -> Z



Add Sphere Collider to “EyeBall” following below figure.

Component -> Physics -> Sphere Collider.

Then select Material None -> Bouncy



**Step 7: Create Paddle**

Window - > Asset Store

Type Hoverboard and click download button then click import button to import to project.



Drag hoverboard to “Scene”. Name hoverboard to “Paddle”.

Hoverboard position, rotation and scale are following below figure.



Add Box Collider to Paddle following below instruction

Component -> Physics -> Box Collider

Don’t change anything.



**Step 8: Create Score Text**

GameObject -> Create Other-> GUI Text

Name GUI Text to Score.



Change Text to “Score:”



**Step 9: Create PaddleController**

Create new javascript from “Project“ section following below instruction

Create -> JavaScript

Then drag to “Paddle” on Hierarchy

PaddleController.js

 function Update ()

 {

// paddle direction is just moved to x position right and left.

 transform.position.x = Camera.main.ScreenToWorldPoint (Input.mousePosition).x ;

// x direction limit is from -19 to 11.

 transform.position.x = Mathf.Clamp( transform.position.x, -19, 11);

 }

 function OnCollisionEnter(collision : Collision)

 {

// collision overall speed

 var velo = collision.rigidbody.velocity.magnitude;

// collision speed is changing to x position

 collision.rigidbody.velocity.x = (collision.transform.position.x - transform.position.x)\*8;

 if (collision.rigidbody.velocity.magnitude < velo)

 {

 collision.rigidbody.velocity \*= velo/collision.rigidbody.velocity.magnitude;

 }}

**Step 10: Create Ball direction**

Create new javascript from project section following below instruction

Create -> JavaScript

Then drag to “Eyeball” on Hierarchy

BallDirection.js

//initial value of speed multiply by 20.

var initialSpeed : float = 20;

function Start()

{

//xDir is x position range from -1.0 to 1.0 of Ball.

 var xDir : float = Random.Range(-1.0,1.0);

//y = -1 when game starts ball randomly falling down.

 rigidbody.AddForce(Vector3( xDir, -1, 0) \* initialSpeed );

}

 **Step 11: Create BrickDestroy**

Create new javascript from project section following below instruction

Create -> JavaScript

Then drag to “Eyeball” on Hierarchy

BrickDestroy.js

function OnCollisionEnter(myCol: Collision){

 if(myCol.gameObject.name == "brick"){

// when ball collide with gameObject “brick”, score will be added by 1.

 Score.score++;

// when ball collide with gameObject “brick” , brick will be destroyed.

 Destroy(myCol.gameObject);

 }

}

**Step 12: Create GameOver**

Create new scene following below instruction

File -> Scene

The Create new GUI Text following below instruction

GameObject -> Create Other -> GUI Text

Change Text to “Game Over”

Change Anchor to “middle center”

Change Font size to “71”



Then build 2 scenes following below instruction.

File -> Build Settings.

 Because after 1 scene the another scene has to work.



Create new javascript from project section following below instruction

Create -> JavaScript

Then drag to “Eyeball” on Hierarchy

GameOver.js

function Update () {

//if y position of ball is lower than -20, “GameOver” scene will be appeared.

 if(rigidbody.position.y < -20){

 Application.LoadLevel("gameOver");

 }

}

**Step 13: Create Score**

Create new javascript from project section following below instruction

Create -> JavaScript

Then drag to “Main Camera” on Hierarchy

Score.js

static var score : float = 0;

var scoreText: GUIText;

function Update(){

 //showing Score text

 scoreText.text = "Score : " + score;

}

Then Drag “Score” on Hierarchy to Score Text of “Main Camera”.



**Step 14: Create Restart**

Create new javascript from project section following below instruction

Create -> JavaScript

Then Open “GameOver” scene

Then drag to “Main Camera” on Hierarchy

Restart.js

//if click mouse button on the scene, “MainScene” will be appeared.

 function Update() {

 if(Input.GetMouseButtonDown(0))

 Application.LoadLevel("MainScene");

}