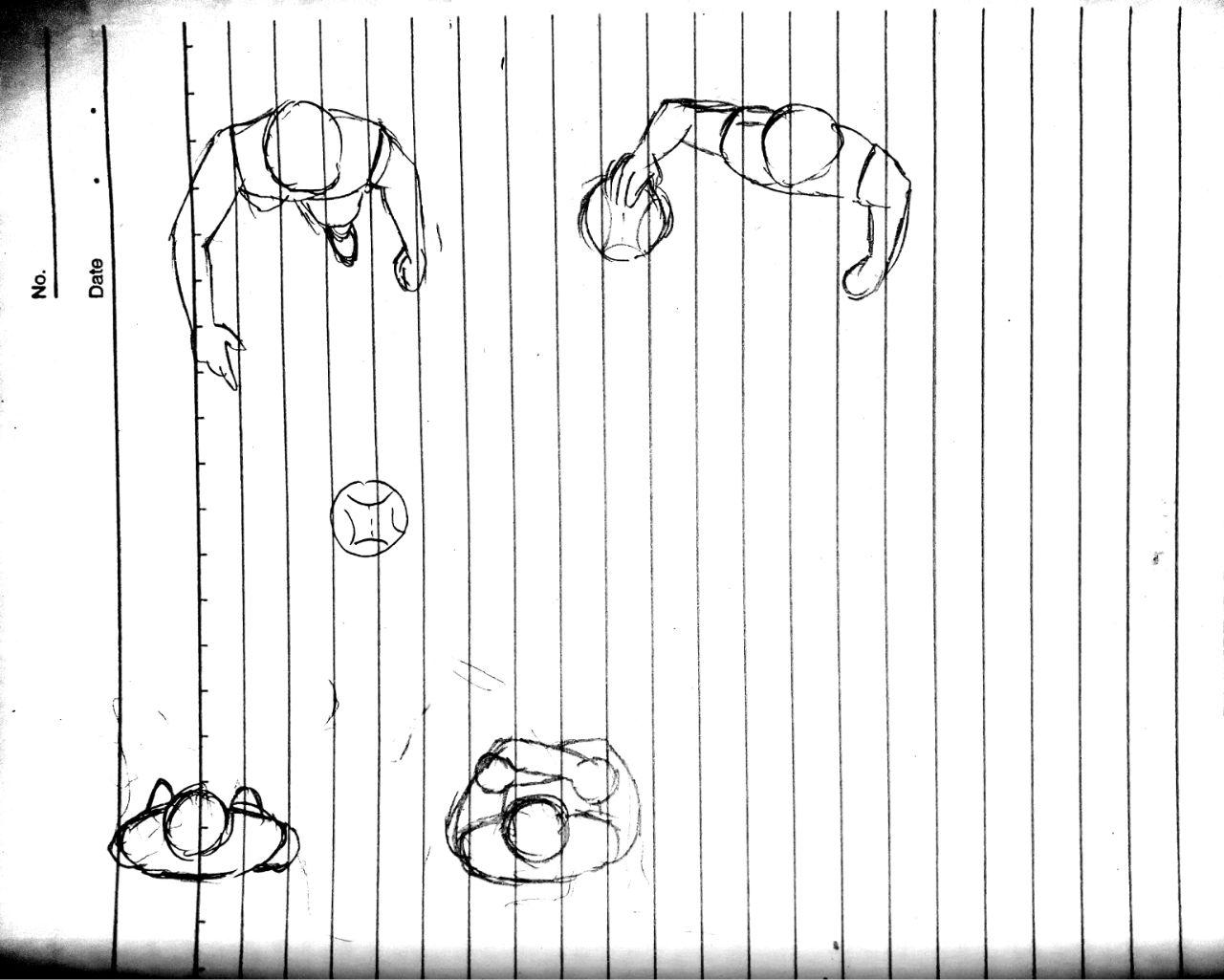
LOGBOOK  
1/October/2018

Team 1 : (Game Title Here)

Artist :

* Membuat sketsa kasar dan test art untuk game yang sudah direncanakan sesuai dengan rancangan game



* Game dibuat dengan model top view



* Teman-teman Suanton (*enemy*) yang suka membully. Dibuat dengan tampak wajah jahat



* Char utama, Suanton. Dibuat dengan wajah sedih dan lusuh karena penampilannya yang membuatnya di Bully.
* Brainstorming untuk art selanjutnya.

Programmer :

Player GameObject

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Player : MonoBehaviour {

[SerializeField]

private float speed = 0.1f;

private GameManager gameManager;

public int curDodge;

public int maxDodge;

private Queue<float> lastTimeDodge = new Queue<float>();

private float intervalDodge = 3;

// Use this for initialization

void Start()

{

gameManager = GameObject.Find("Game Manager").GetComponent<GameManager>();

curDodge = 3;

maxDodge = 3;

}

// Update is called once per frame

void Update()

{

//Debug.Log("X: " + transform.position.x);

// Keyboard Input

if (Input.GetKey("d"))

transform.Translate(speed, 0, 0);

if (Input.GetKey("a"))

transform.Translate(-speed, 0, 0);

// End of Keyboard Input

// Accelerometer Input

transform.Translate(Input.acceleration.x, 0, 0);

// End of Accelerometer Input

// Border of Movement

if (transform.position.x >= Screen.width / 100 + 4.3f)

transform.position = new Vector2(transform.position.x - 0.1f, transform.position.y);

if (transform.position.x <= -Screen.width / 100 - 4.3f)

transform.position = new Vector2(transform.position.x + 0.1f, transform.position.y);

// End of Border of Movement

if (curDodge < maxDodge)

{

if (Time.time - lastTimeDodge.Peek() > intervalDodge){

curDodge++;

lastTimeDodge.Dequeue();

}

}

}

public void DodgeRight()

{

if (curDodge > 0)

{

lastTimeDodge.Enqueue(Time.time);

transform.Translate(speed \* 20, 0, 0);

curDodge--;

}

}

public void DodgeLeft()

{

if (curDodge > 0)

{

lastTimeDodge.Enqueue(Time.time);

transform.Translate(-(speed \* 20), 0, 0);

curDodge--;

}

}

private void OnCollisionEnter2D(Collision2D collision)

{

if (collision.gameObject.tag == "Bullets")

{

gameManager.HealthPoint -= collision.gameObject.GetComponent<BulletPrefabs>().CurrentType.damage;

Destroy(collision.gameObject);

}

}

}

EnemiesPrefab.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

[System.Serializable]

public class EnemiesData

{

public GameObject visualization;

public BulletPrefabs bullets;

public float[] bulletsPercentage;

public float fireRate;

public float health;

public float score;

}

public class EnemiesPrefabs : MonoBehaviour {

public List<EnemiesData> types;

private EnemiesData currentType;

public EnemiesData CurrentType

{

get

{

return currentType;

}

set

{

currentType = value;

int currentTypesIndex = types.IndexOf(currentType);

GameObject typesVisualization = types[currentTypesIndex].visualization;

for (int i = 0; i < types.Count; i++)

{

if (typesVisualization != null)

{

if (i == currentTypesIndex)

{

types[i].visualization.SetActive(true);

}else

{

types[i].visualization.SetActive(false);

}

}

}

}

}

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

}

void OnEnable()

{

CurrentType = types[0];

}

public void setCurrentType(int typeSelected)

{

int currentTypeIndex = types.IndexOf(currentType);

if (currentTypeIndex < types.Count)

{

if(typeSelected < types.Count)

CurrentType = types[typeSelected];

}

}

}

Game Design :

- Game Stage Design  
Files in : ../Game Design Document/Game Stage Design – Chapter 1.docx

To do :

Chapter 1 Stages progressive story.

Adjusting game mechanics.