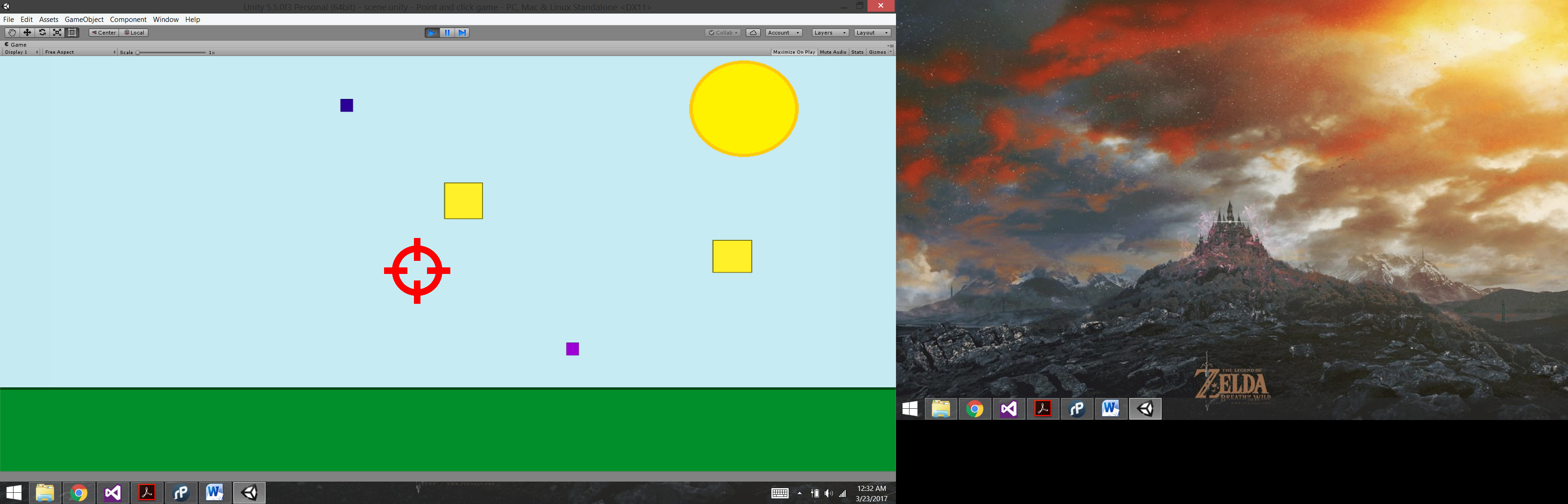
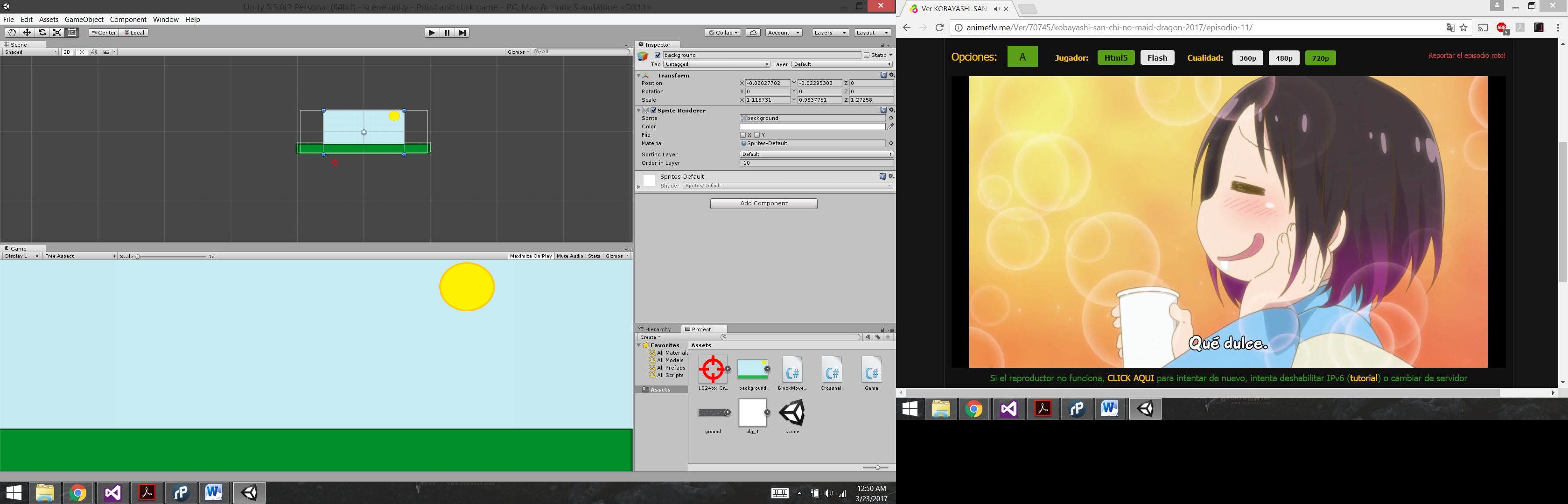
**Introduction to Point and Click**

Hello! In this short tutorial we will show you how to make a simple point and click game, we will focus on the actual action of making the point and click. At the end you should have something like the following picture. You don’t need prebuilt assets other than just a background drawn in paint and a square to make the colored squares that are showing. Let’s begin!



First you want to create your 2D project and name it as you please. Then once it has been created you will see it already has the main camera in there, if you already made you background image and your squares(they can be white there is no problem with that, as you will change the color from unity).

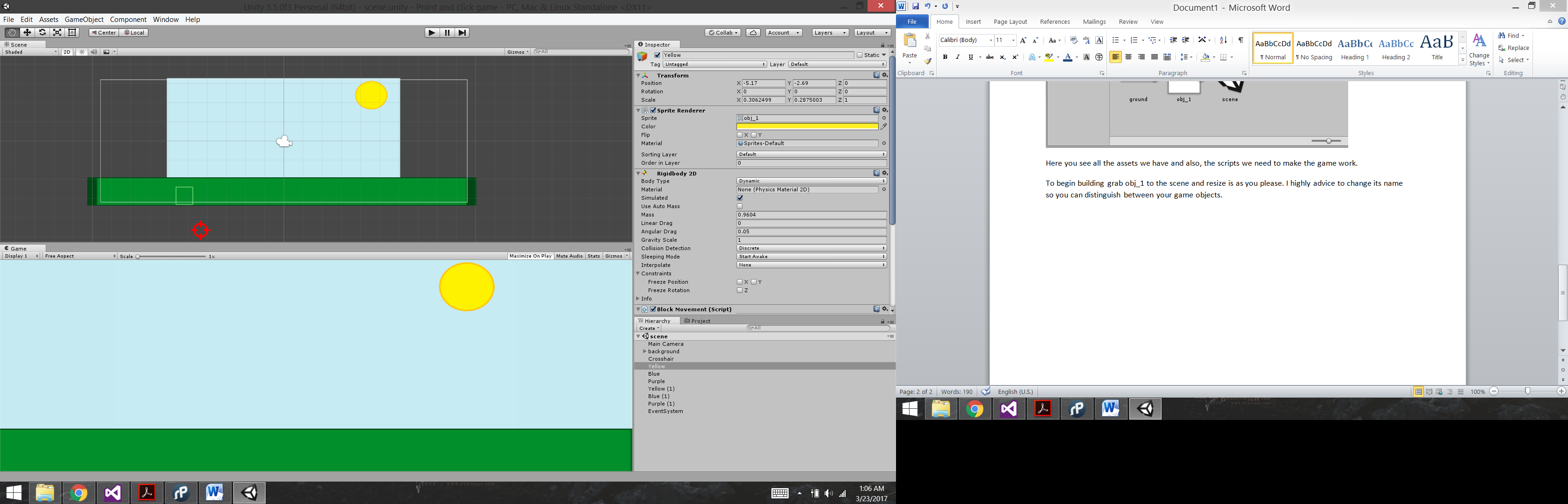
As always, you have your project window and it should look more or less like this:

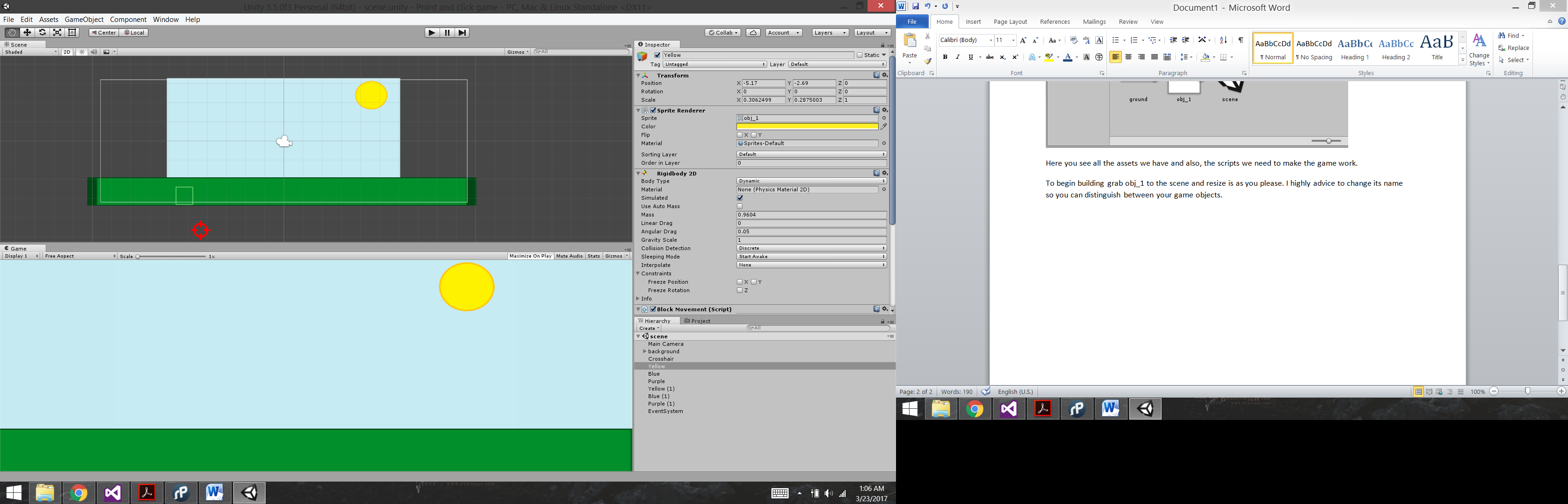
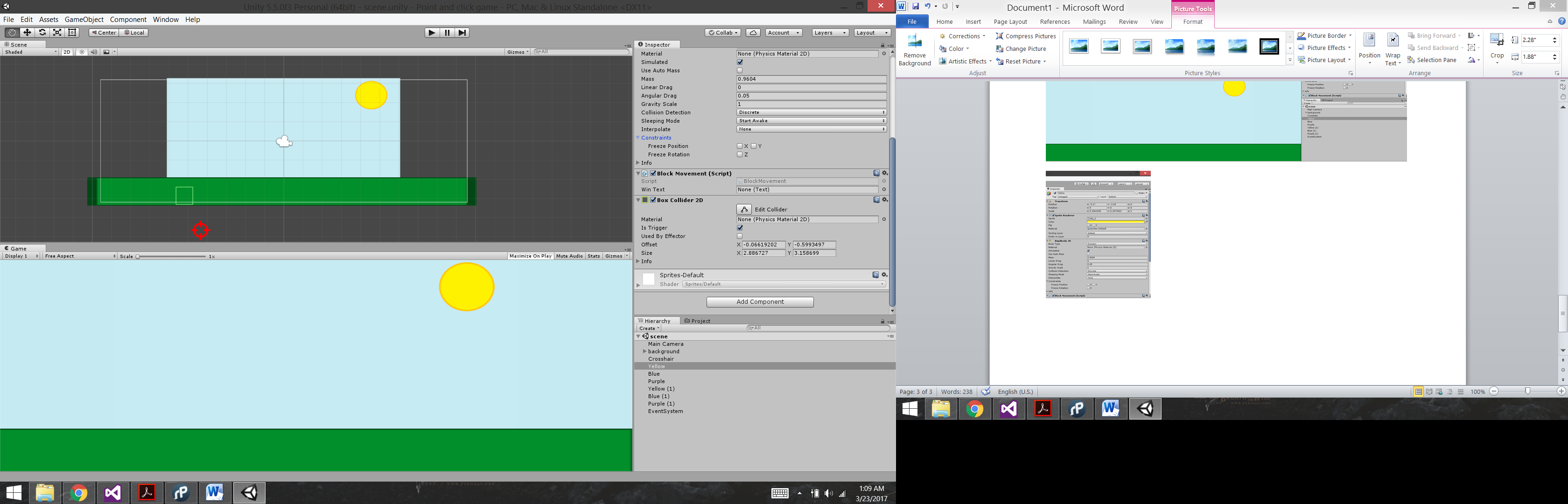


Here you see all the assets we have and also, the scripts we need to make the game work.

To begin building grab obj\_1 to the scene and resize is as you please. I highly advice to change its name so you can distinguish between your game objects.

In the next picture you can see that I have selected “Yellow” which is one of the boxes you will try to hit in the game. I will go in detail of what does it need to contain in order to work properly and dissapear when clicked on.



Here we have the inspector of Yellow and we see several things going on in here. First one, position doesn’t really matter but it would be better if the squares are hidden behind the green rectangle as they will jump and move arround. Then on the Sprite Renderer we can change the color and we choose yellow for it. Then we have the Rigidbody2D and look at the values it has for each field. Then we also have the script component and the script is called BlockMovement. Finally, in the right picture we have the Box Collider 2D and you just need to adjust that to be the same size as your square(sometimes it does it automatically).

Script for BlockMovement:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class BlockMovement : MonoBehaviour {

public Text winText;

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

float xPos = Random.Range(-7, 7); //changes position in x axis

float yVel = Random.Range(8, 14); // changes velocity in y-axis

float xVel = Random.Range(-4, 4); //changes velocity in x-axis

//this restrict the objects to leave a certain area

if(this.transform.position.y < -6)

{

xPos = Random.Range(-7, 7);

if (xPos < -3) xVel = Random.Range(-1, 6);

if (xPos > 3) xVel = Random.Range(-6, 1);

this.transform.position = new Vector2(xPos, -6);

this.GetComponent<Rigidbody2D>().velocity = new Vector2(xVel, yVel);

}

}

private void OnMouseOver()

{

//this destroys gameobjects when clicked

if (Input.GetMouseButtonDown(0))

{

Destroy(gameObject);

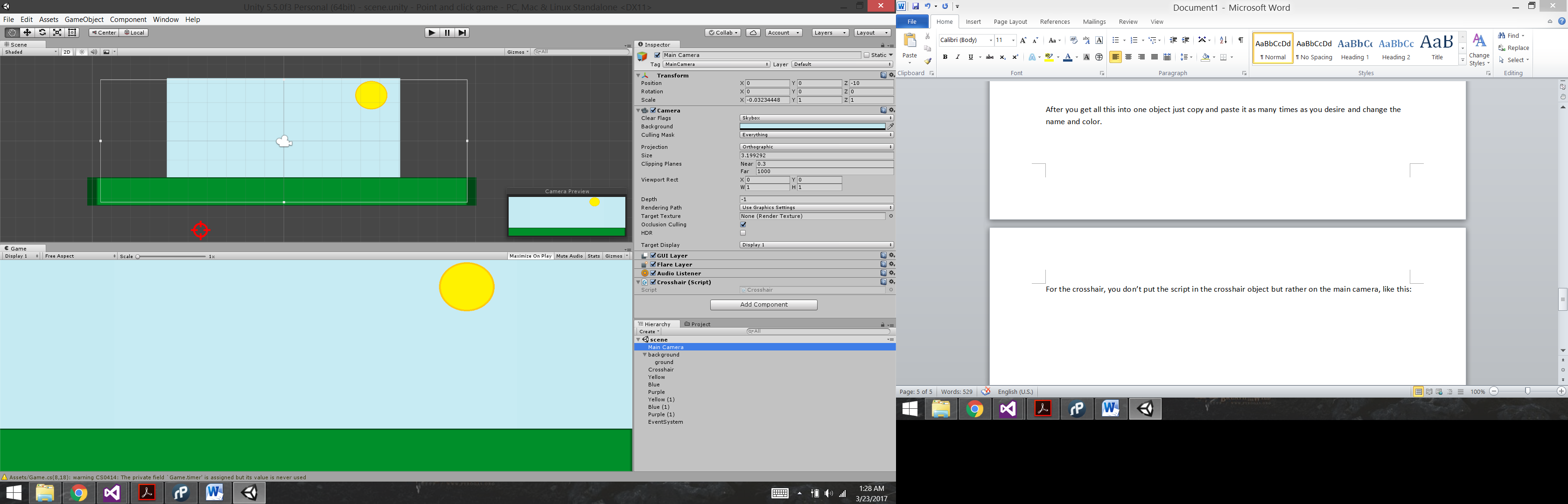
}

}

}

After you get all this into one object just copy and paste it as many times as you desire and change the name and color.

For the crosshair, you don’t put the script in the crosshair object but rather on the main camera, like this:



Script for Crosshair:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Crosshair : MonoBehaviour {

// Use this for initialization

void Start () {

Cursor.visible = false; //disables computer cursor from displaying

}

// Update is called once per frame

void Update () {

Vector3 pos = this.GetComponent<Camera>().ScreenToWorldPoint(Input.mousePosition); //gets position in the screen

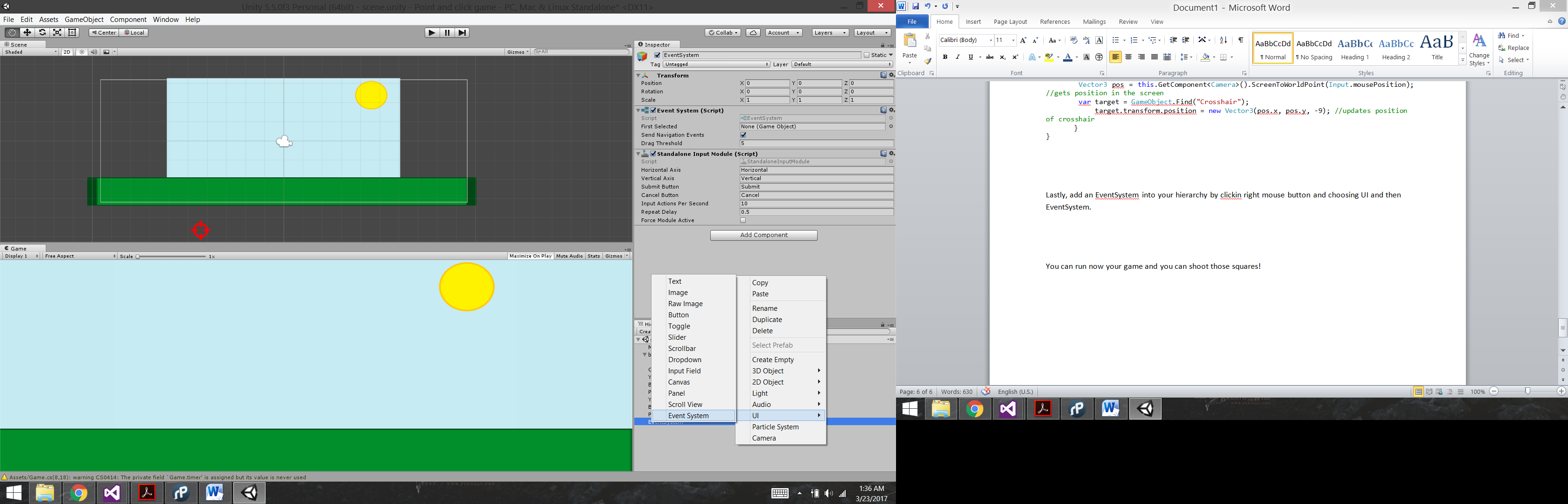
var target = GameObject.Find("Crosshair");

target.transform.position = new Vector3(pos.x, pos.y, -9); //updates position of crosshair

}

}

Lastly, add an EventSystem into your hierarchy by clickin right mouse button and choosing UI and then EventSystem:



You can run now your game and you can shoot those squares! Thanks for reading!