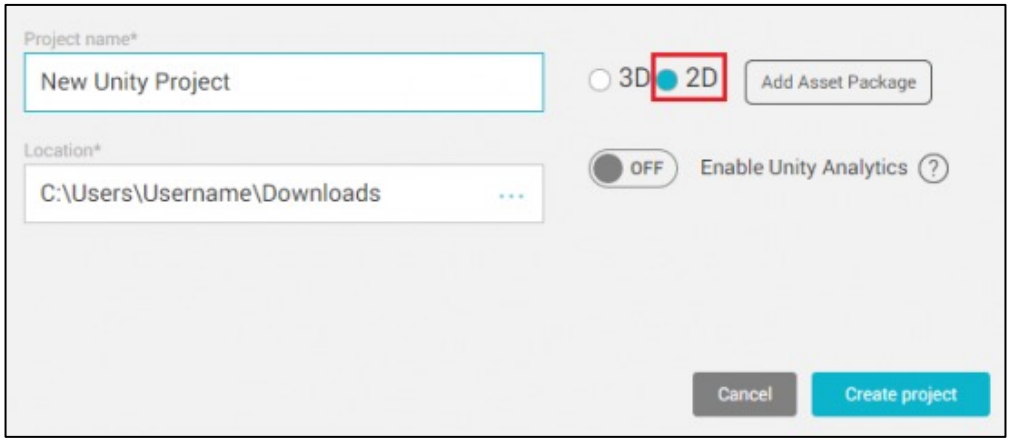
CT3536 Games Programming

Section 7
2D games in Unity

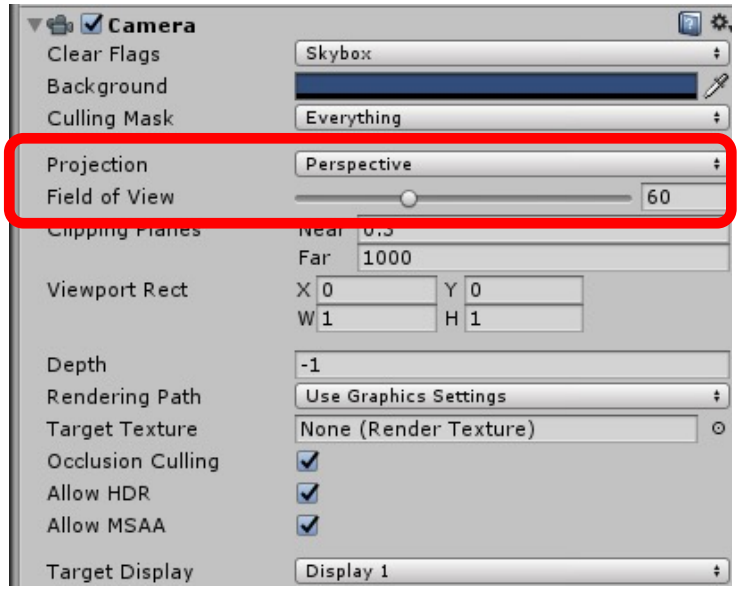
2D Games in Unity

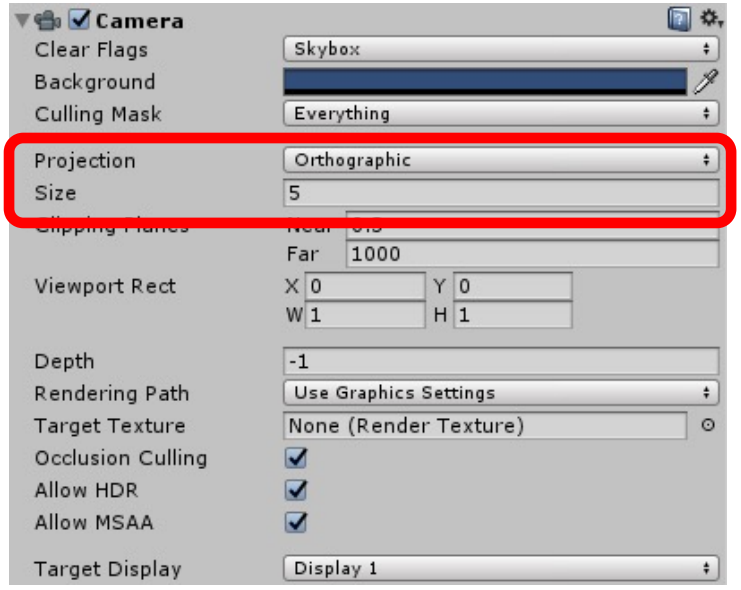
- When you start a new project, you can select 2D or 3D



- The main difference is that the Main Camera is set to Orthographic (rather than Perspective) for 2D games

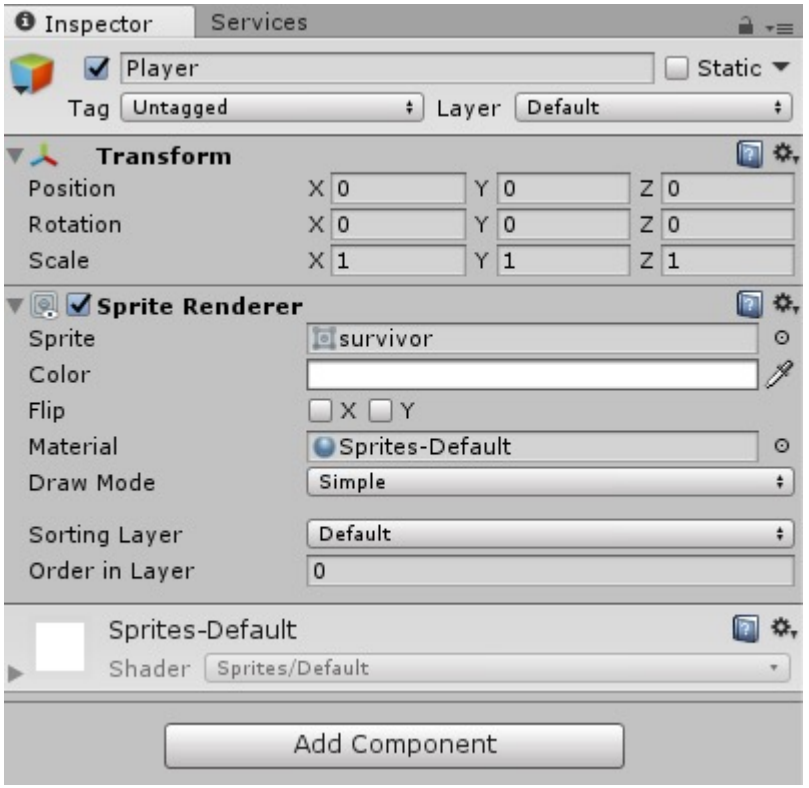
The Size property (referred to as orthographicSize in script) defines the space you want to display across the screen.. i.e. making the number smaller will zoom the camera in

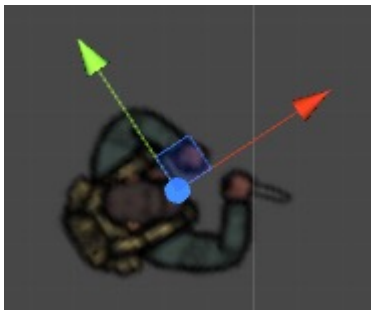




Sprites Assets and the SpriteRenderer Component

- Any png/jpg files you bring into your project are available as sprites, for display using the SpriteRenderer component
- To make a new GameObject which will display a sprite, right click in the hierarchy and choose 2D Object > Sprite
- Note: this is a Game Object in the world, rather than in a Canvas (not the same thing, different coordinate systems, no RectTransform)





Movement only makes sense on the x and y axes
Rotation (typically) only makes sense on the z axis

https://docs.unity3d.com/560/Documentation/Manual/class-SpriteRenderer.html

2D Physics

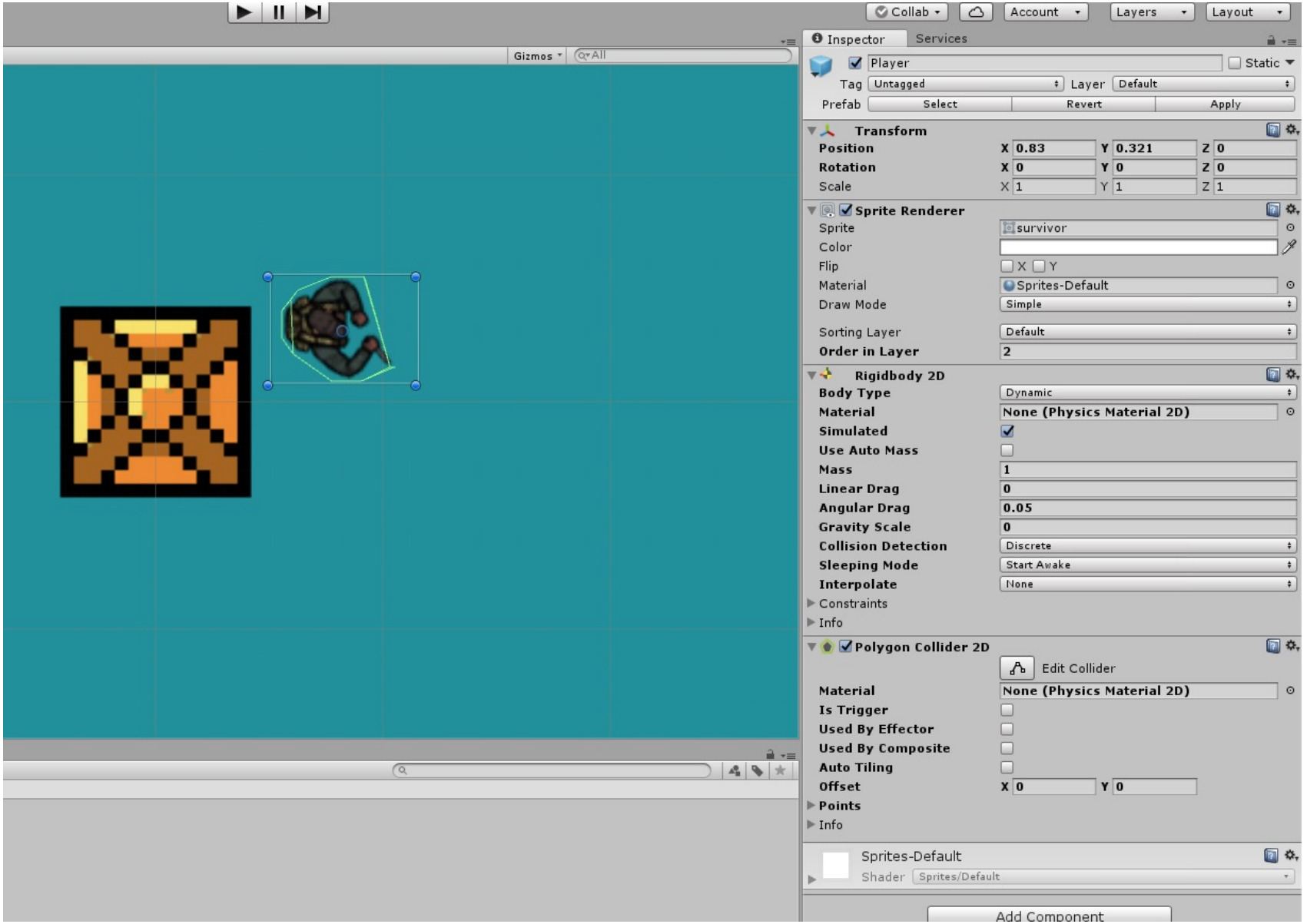
https://docs.unity3d.com/560/Documentation/Manual/Physics2DReference.html

2D games typically use Physics2D, Rigidbody2D, and Collider2D rather than the normal 3D versions

Collider types include BoxCollider2D, CircleCollider2D, PolygonCollider2D, and EdgeCollider2D

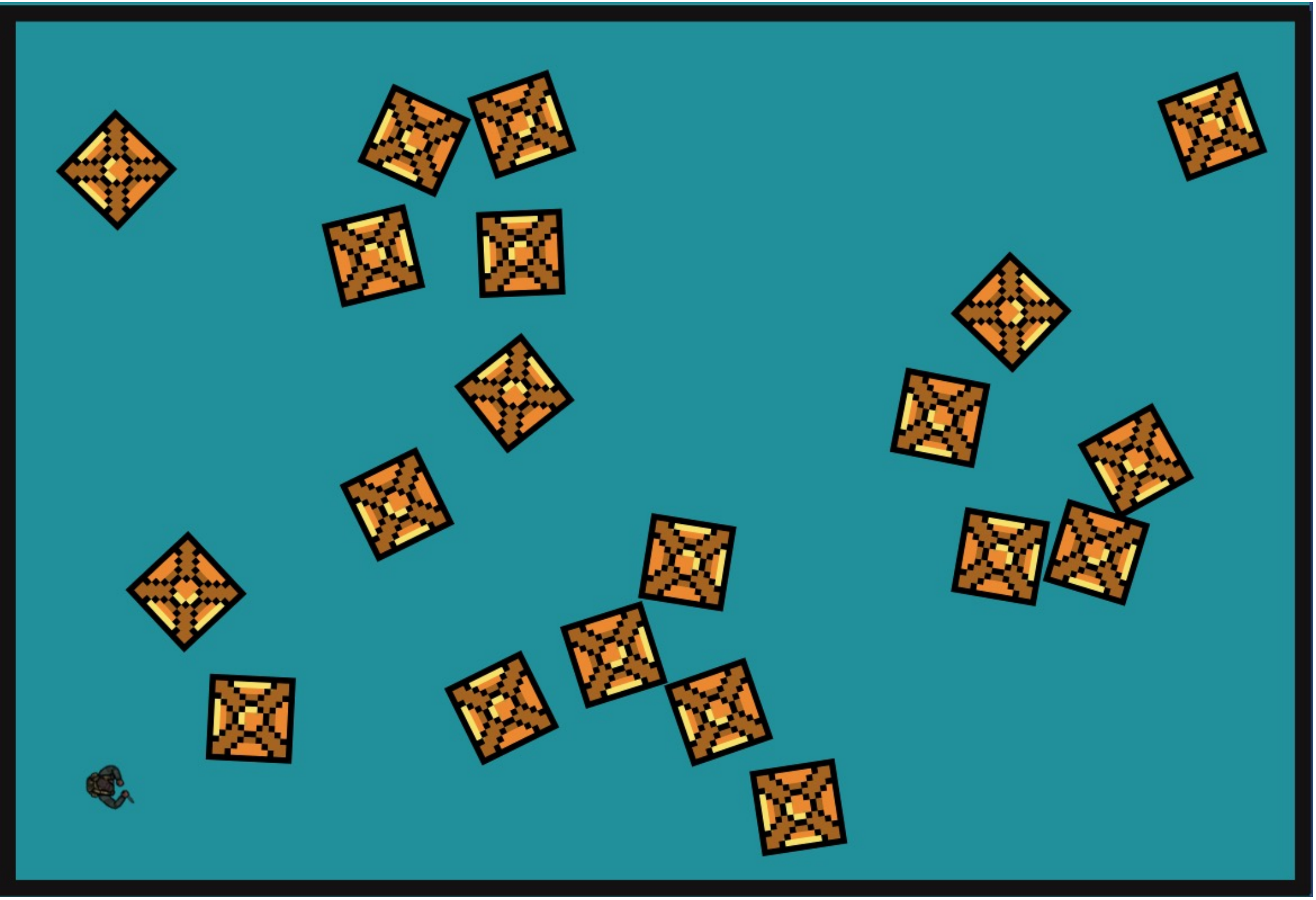
Unity uses a completely different physics engine for this, optimized for 2D

Example (see Zombies2D project on Blackboard)



Spawning some crates (and a player)

```
public class GameManager : MonoBehaviour {
  // inspector settings
  public GameObject playerPrefab, zombiePrefab, cratePrefab;

 // reference to the runtime-instantiated player object
  public static GameObject thePlayer;

  void Start () {
     // spawn the player and some crates
     thePlayer = Instantiate(playerPrefab);
     thePlayer.transform.position = FindSpawnPosition(0.5f);

     for (int i=0; i<20; i++) {
       GameObject go = Instantiate(cratePrefab);
       Vector2 pos = FindSpawnPosition(0.7f);
       go.transform.position = pos; // .z gets set to 0 when Vector2 is assigned to Vector3
       go.transform.rotation = Quaternion.AngleAxis(Random.Range(0f,360f), new Vector3(0f,0f,1f));
     }
  }

  private Vector2 FindSpawnPosition(float spaceNeeded) {
    Vector2 pos = Vector2.zero;
    Vector2 testBoxSize = new Vector2(spaceNeeded,spaceNeeded);
    do {
      pos.x = Random.Range(-5.5f, 5.5f);
      pos.y = Random.Range(-3.5f, 3.5f);
    } // Physics2D.OverlapBox returns the first overlapping collider found (if any)
    while (Physics2D.OverlapBox(pos, testBoxSize, 0f)!=null);

    return pos;
  }
}
```



Making the player move.

```
public class Player : MonoBehaviour {

  // inspector settings
  public Rigidbody2D rigid;

  void FixedUpdate () {
    // turn left/right with arrow keys
    if (Input.GetKey(KeyCode.LeftArrow))
      rigid.AddTorque(Time.fixedDeltaTime*20f);
    else if (Input.GetKey(KeyCode.RightArrow))
      rigid.AddTorque(-Time.fixedDeltaTime*20f);
    // move forward with up arrow key
    if (Input.GetKey(KeyCode.UpArrow))
      rigid.AddForce(Time.fixedDeltaTime*100f*transform.right);
  }
}
```

Note: the player's Rigidbody2D linear drag is 1 and angular drag is 5

Note: for Rigidbody2D, AddTorque takes just a scalar as its argument

Spawning some zombies

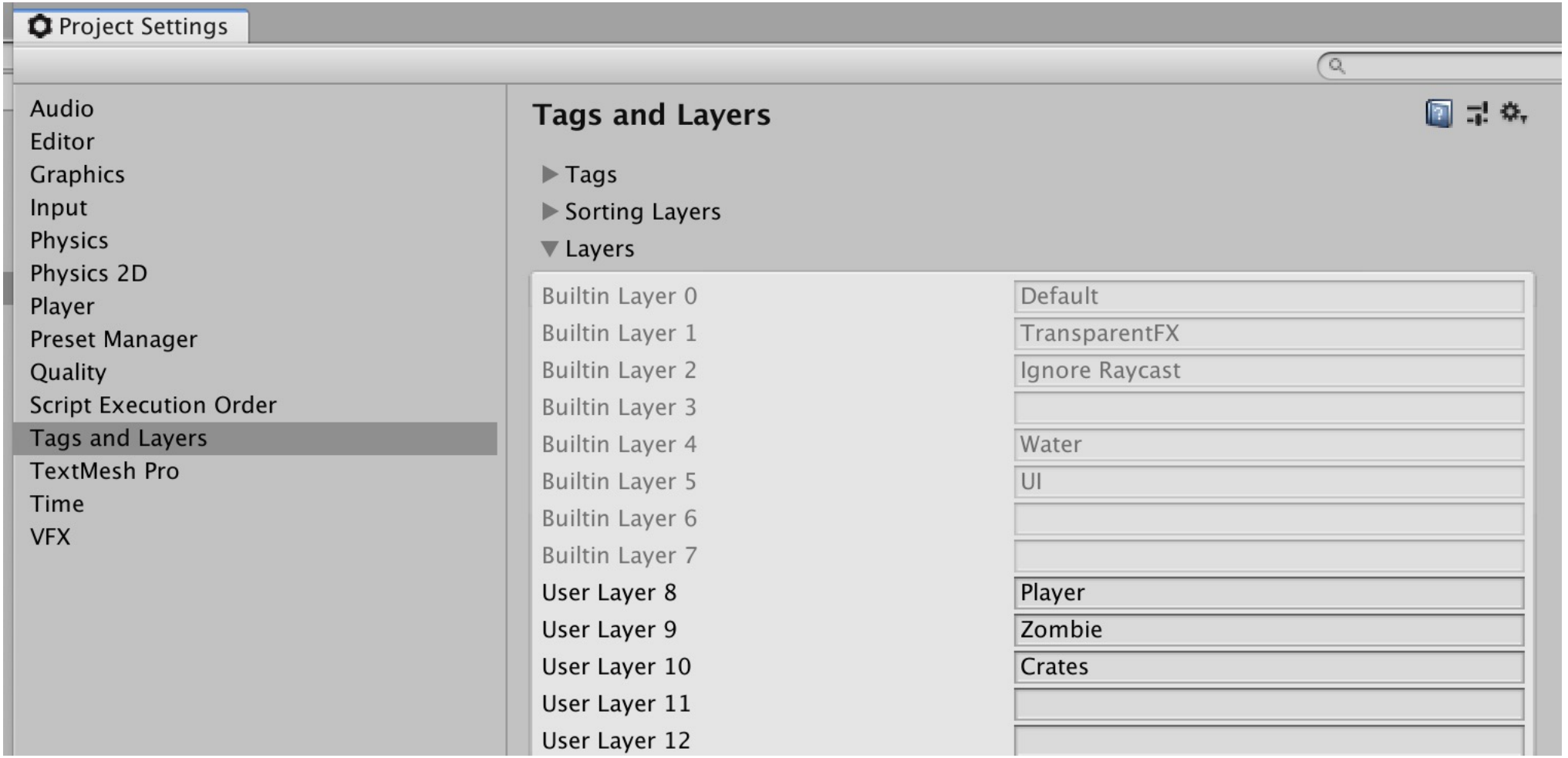
Add to the GameManager's Start() method:

```
// start spawning zombies
StartCoroutine( SpawnZombies() );
```

Add a new method to GameManager.cs:

```
private IEnumerator SpawnZombies() {
  for (int i=0; i<30; i++) {
    yield return new WaitForSeconds(3f);
    GameObject go = Instantiate(zombiePrefab);
    Vector3 pos = FindSpawnPosition(0.5f);
    while ((pos-thePlayer.transform.position).magnitude<2f)
      pos = FindSpawnPosition(0.5f);
    go.transform.position = pos;
    go.transform.rotation =
Quaternion.AngleAxis(Random.Range(0f,360f), new Vector3(0f,0f,1f));
  }
}
```

Create 3 layers and assign the Player, Zombie, and Crates to them



Movement code for Zombies

```
public class Zombie : MonoBehaviour {
  // inspector settings
  public Rigidbody2D rigid;

  void FixedUpdate () {
    Vector3 playerPos = GameManager.thePlayer.transform.position;
    Vector3 vecToPlayer = playerPos - transform.position;
    Vector2 dirToPlayer = vecToPlayer.normalized;
    float distToPlayer = vecToPlayer.magnitude;

    // Facing within approx 45 degrees of the player?
    float dot = Vector2.Dot(dirToPlayer, transform.right);
    if (dot>0.707f) {
      // Can see the player? (blocked by crates)
      int cratesMask = LayerMask.GetMask("Crates");
      if (!Physics2D.Raycast(transform.position, dirToPlayer, distToPlayer,
cratesMask)) {
        // face the player
        float turnToAngle = Mathf.Atan2(vecToPlayer.y, vecToPlayer.x);
        rigid.rotation = turnToAngle*Mathf.Rad2Deg; // change radians to degrees
        // move forwards
        rigid.AddForce(Time.fixedDeltaTime*50f*transform.right);
      }
    }
  }
}
```