### 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

New Unity Project	3D 2D Add Asset Package
cation*	Fnable Unity Analytics (2)
C:\Users\Username\Downloads	

• 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

♥ 🔮 🗹 Camera Clear Flags Background Culling Mask	Skybox + Everything +	♥ 🚭 🗹 Camera Clear Flags Background Culling Mask	Skybox +
Projection Field of View	Perspective + 60	Projection Size	Orthographic +
Cipping Planes Viewport Rect	Near         0.3           Far         1000           X         0         Y           W         1         H	Viewport Rect	Far 1000 X 0 Y 0 W 1 H 1
Depth	-1	Depth	-1
Rendering Path	Use Graphics Settings +	Rendering Path	Use Graphics Settings \$
Target Texture	None (Render Texture) 0	Target Texture	None (Render Texture) O
Occlusion Culling	$\checkmark$	Occlusion Culling	
Allow HDR	✓	Allow HDR	
Allow MSAA		Allow MSAA	
Target Display	Display 1 +	Target Display	Display 1 +

#### 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)

O Inspector Service	s						<b>∂</b> •≡
👕 🗹 Player				s	tatic 🔻		
Tag Untagged	ag Untagged + Layer Default					+	
🔨 🙏 Transform 🛛 🔯 🔅							
Position	Х	0	Y	0	Z	0	
Rotation	Х	0	Y	0	Z	0	
Scale	Х	1	Y	1	Z	1	
🔻 🔍 🗹 Sprite Renderer 🛛 🔯 🌣,							
Sprite	isurvivor o			0			
Color	1						
Flip							
Material	Sprites-Default				0		
Draw Mode	Simple +						
Sorting Layer Default +					+		
Order in Layer	0						
Sprites-Default 🕅 🌣					<b>a</b> *,		
Shader Sprites/Default					•		
	Ad	d Compo	nent	:	]		



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/cl ass-SpriteRenderer.html

## 2D Physics

https://docs.unity3d.com/560/Documentation/Manual/Physics2DReference.h tml

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)

		🖉 Collab 🔻 🛛 📿	△ Account • Layers • Layout •
	*=	Inspector Services	<u></u> +≡
	Gizmos * Q*All	Player	Static 💌
			+ Laver Default +
		Prefab Select	Revert Apply
		Tuan cform	<b>a</b>
		Position	X 0.83 Y 0.321 Z 0
		Rotation	X 0 Y 0 Z 0
		Scale	X 1 Y 1 Z 1
		🔻 🔍 🗹 Sprite Renderer	💽 🌣.
		Sprite	⊠survivor ⊙
		Color	P
		Flip	
		Material	Sprites-Default 0
		Draw Mode	Simple +
		Sorting Layer	Default +
		Order in Layer	2
		🔻 🔶 Rigidbody 2D	[2] *.
		Body Type	Dynamic +
		Material	None (Physics Material 2D) •
		Simulated	
		Mass	
		Linear Drag	0
		Angular Drag	0.05
		Gravity Scale	0
		Collision Detection	Discrete +
		Sleeping Mode	Start Awake +
		Interpolate	None +
		Constraints	
		V 🔮 🗹 Polygon Collider :	
			The Edit Collider
		Material	None (Physics Material 2D) 0
		Used By Effector	
		Used By Composite	
Q	* 0 4	Auto Tiling	
		Offset	X 0 Y 0
		▶ Points	
		▶ Info	
		Sprites-Default	n ¢.
		Shader Sprites/De	fault
			Add Component

#### 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++) {</pre>
    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 = Ouaternion.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));
    }
}</pre>
```

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

C Project Settings		
		Q
Audio Editor	Tags and Layers	🔯 🖈
Graphics Input Physics	<ul> <li>Tags</li> <li>Sorting Layers</li> <li>Layers</li> </ul>	
Physics 2D Player Preset Manager Quality Script Execution Order Tags and Layers TextMesh Pro Time VFX	Builtin Layer 0 Builtin Layer 1 Builtin Layer 2 Builtin Layer 3 Builtin Layer 4 Builtin Layer 5 Builtin Layer 6 Builtin Layer 7 User Layer 8 User Layer 9	Default TransparentFX Ignore Raycast Water UI Player Zombie
	User Layer 10 User Layer 11 User Layer 12	Crates

#### 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);
   }
  }
```