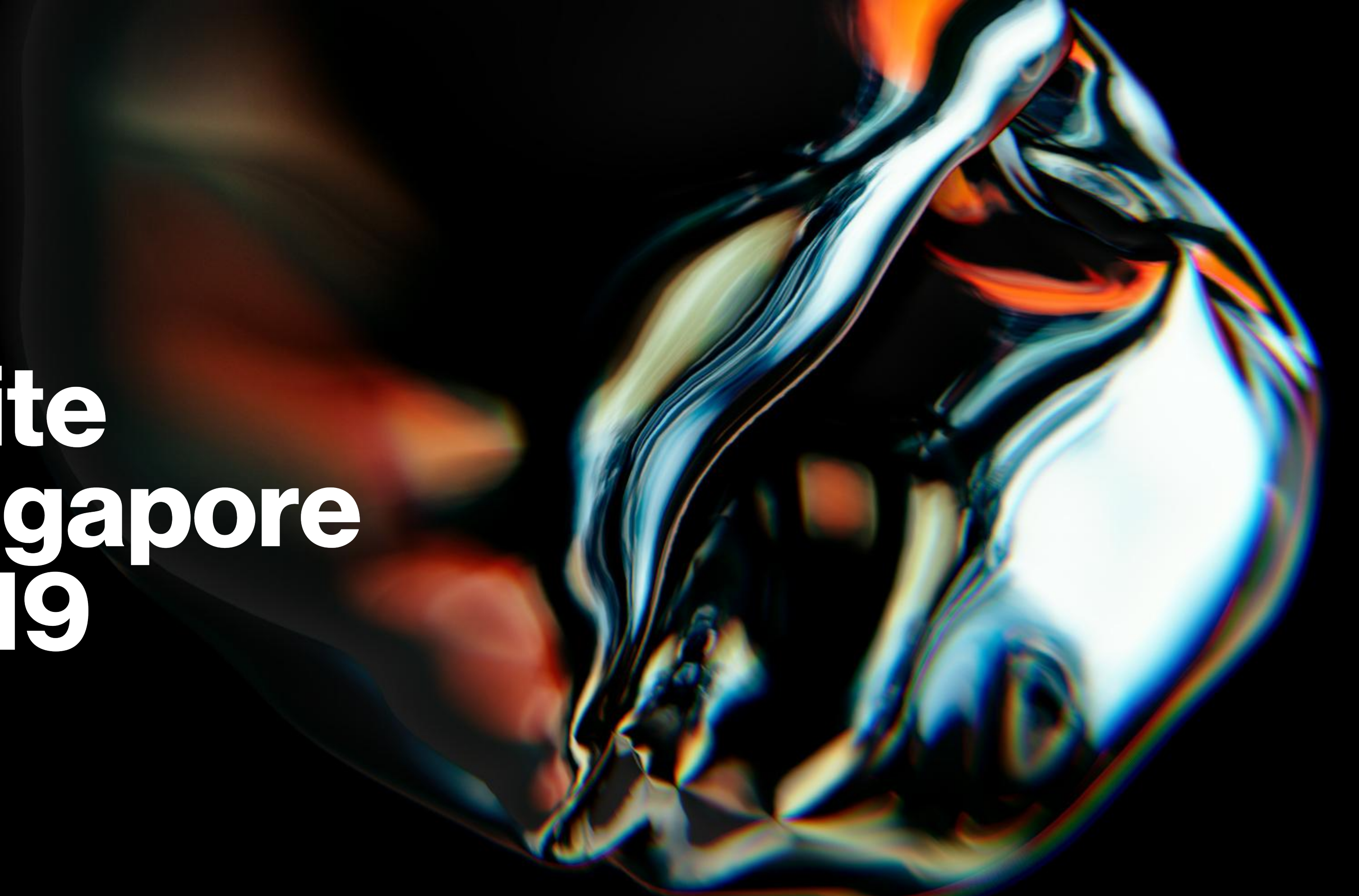
unity

Unite Singapore 2019

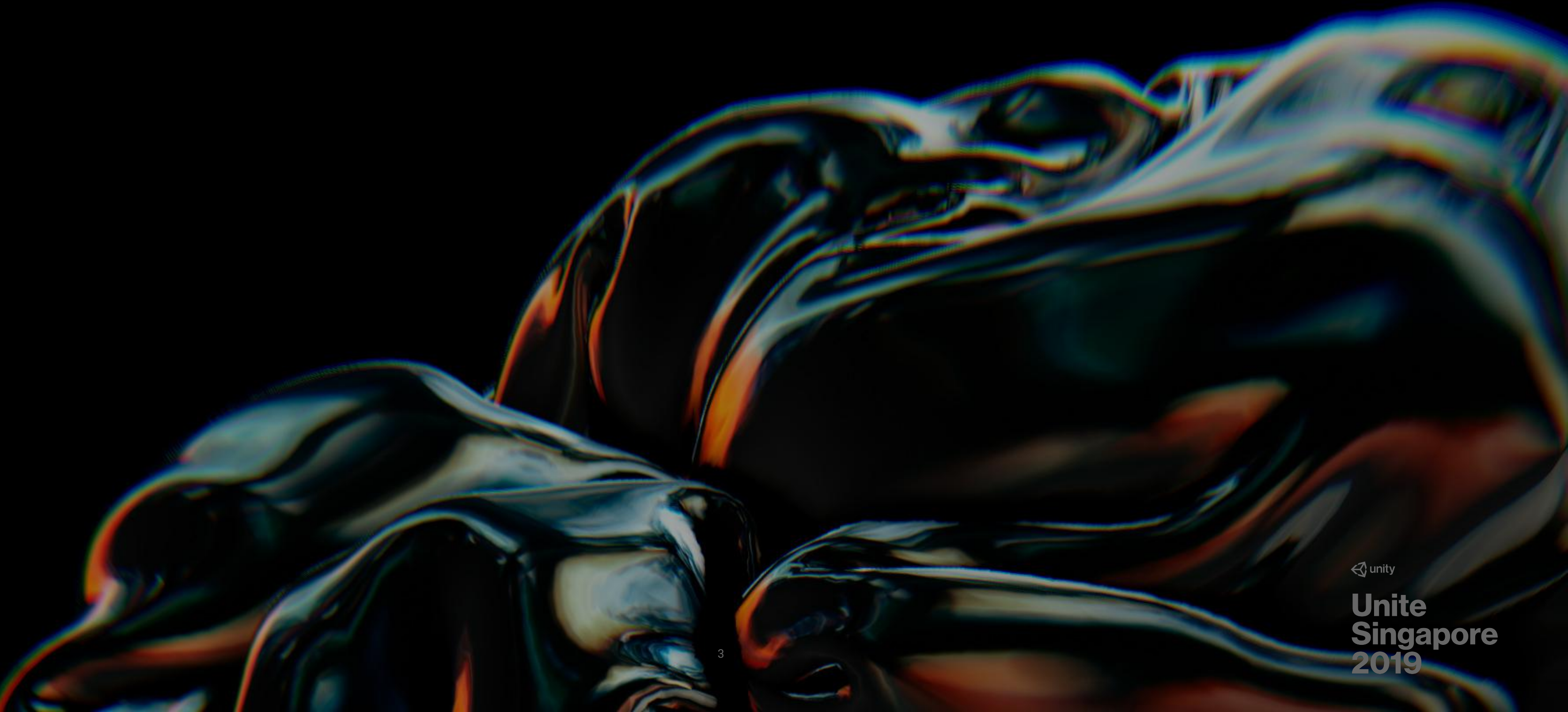


Creating the Next Location-Based AR Blockbuster Game

unity

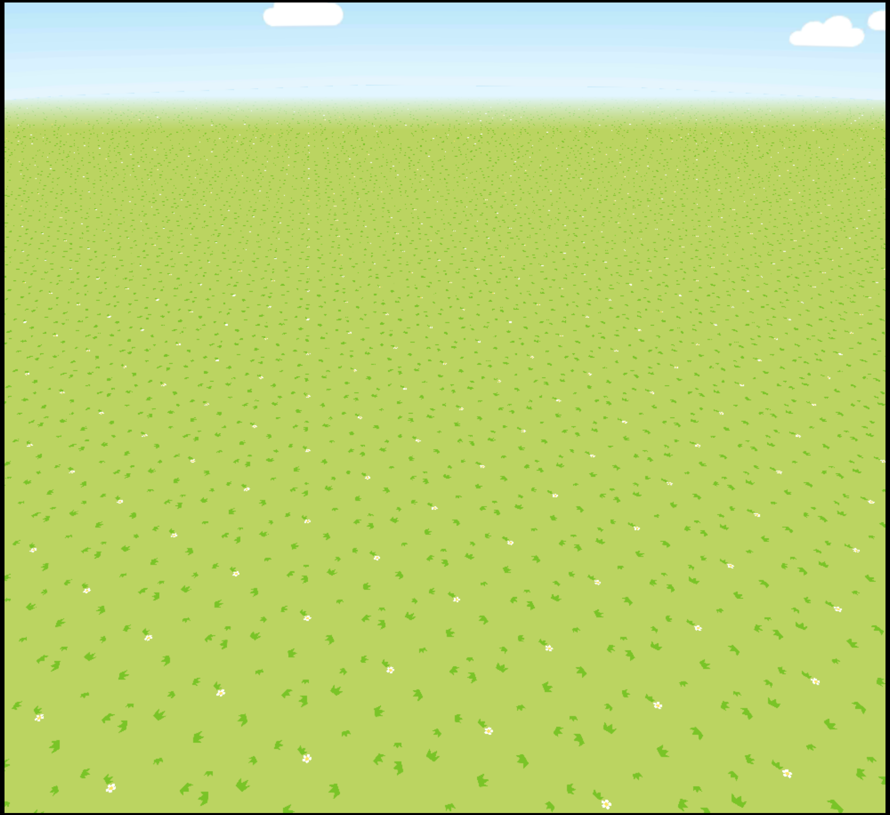
Unite Singapore 2019

Making The World



Ground

- Base layer of your World
- Tileable and infinite
- Fairly boring



Map Features

- Provide context
- Define feel of the game
  - Must be easy to customise

Scripting the Pipeline: Requirements

- Allow the art team to modify parameters easily
- Don’t pollute the Scene
- Self-contained
- Allow modifications pre/post object creation

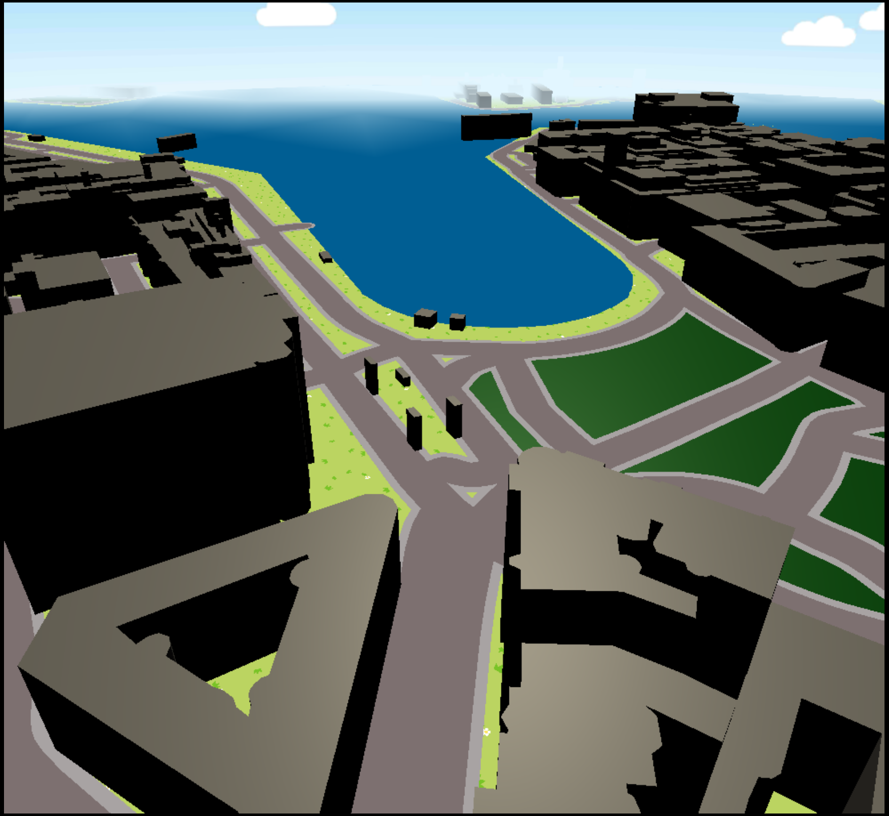
Scripting the Pipeline: Solutions

- Scriptable Objects as pipeline segments
  - Apply linearly
  - Receive and modify settings
  - Hook into scene only via event listeners

Change the Material

```csharp
public abstract class BeforeCreation<T> : ScriptableObject where T : MapCreateSettings
{
    internal abstract void Apply(T settings);
}

public abstract class BeforeWaterCreationSegment : BeforeCreation<WaterCreateSettings> { }

[CreateAssetMenu(menuName = "Map/Pipeline/BeforeWaterMaterialModifier")]
public sealed class BeforeWaterMaterialModifier : BeforeWaterCreationSegment
{
    [SerializeField]
    private Material m;

    internal override void Apply(WaterCreateSettings settings) => settings.material = m;
}
```

Before...



Better Water

Better World

Decorating the World: Requirements

- The world is more than buildings and roads
  - We want to make the game world feel alive
- But we also want performance

Decorating the World: Solutions

- ECS
- Job System

Spawning an Entity as a Job

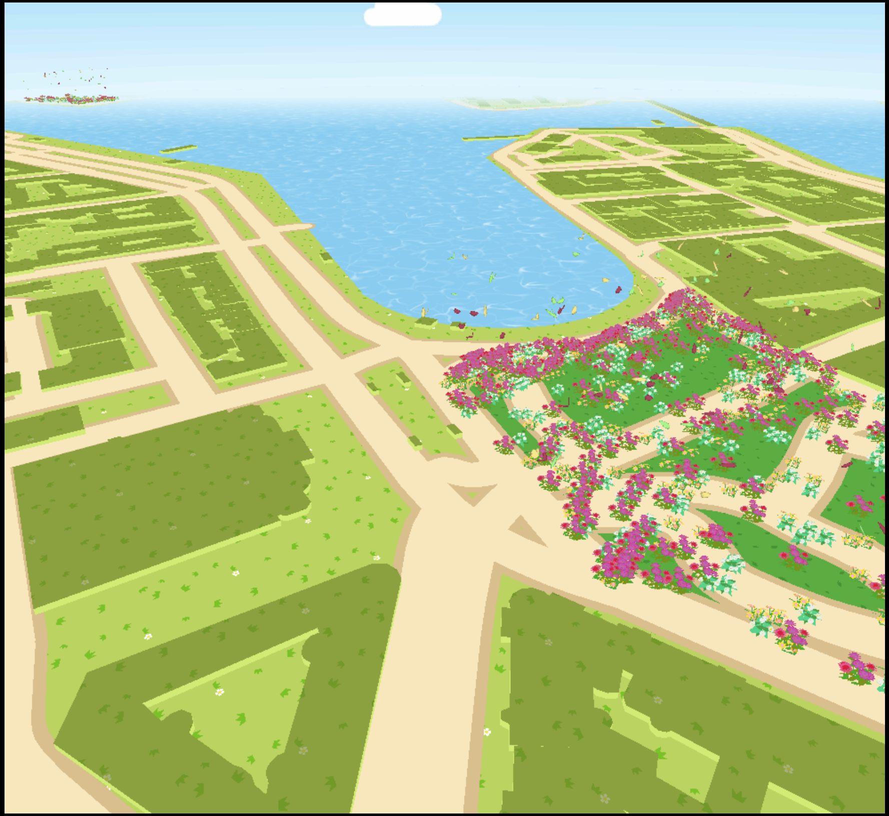
```
public struct EntitySpawnJob : IJob
{
    public EntityCommandBuffer.Concurrent commandBuffer;

    [NativeSetThreadIndex] public int threadIndex;

    [ReadOnly] public Entity entityPrefab;
    [ReadOnly] public float3 randomPosition;
    [ReadOnly] public float3 randomRotation;

    [BurstCompile]
    public void Execute()
    {
        Entity entity = commandBuffer.Instantiate(threadIndex, entityPrefab);
        commandBuffer.SetComponent(threadIndex, entity,
            new Translation{ Value = randomPosition });

        commandBuffer.SetComponent(threadIndex, entity,
            new Rotation
            {
                Value = quaternion.LookRotationSafe(randomRotation, new float3(0, 1, 0))
            });
    }
}

```

Before...

Flowers

- Spawned in Jobs
- Vertex Shader Animated
- GPU Instanced

Flowers And Butterflies



And More Flowers

More problems appear

- How do decorations move with the map?
  - If your map moves
- How and when to destroy them?
- Systems run OnUpdate every frame
  - What if you don’t want or can’t use that behaviour?

ECS != Job System

- Take what you need from both

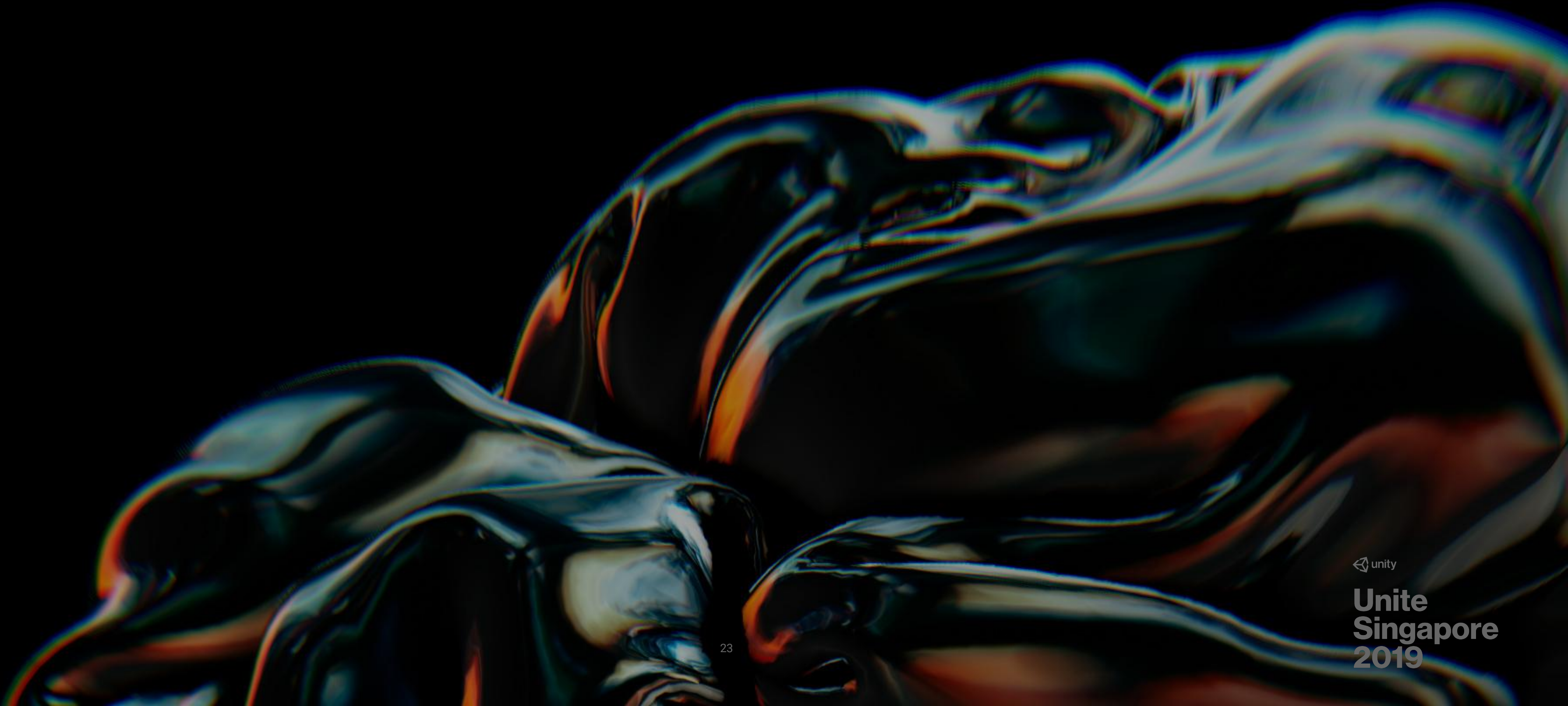
Components as tags

```csharp
[Serializable]
public struct FloatingObjectComponent : IComponentData { }

[BurstCompile]
public struct FloatingOriginJob : IJobForEach<Translation, FloatingObjectComponent>
{
    [ReadOnly]
    public float3 offset;

    public FloatingOriginJob(float3 offset)
    {
        this.offset = offset;
    }

    public void Execute(ref Translation translation, [ReadOnly] ref FloatingObjectComponent f)
    {
        translation.Value += offset;
    }
}
```

Entity Jobs from a MonoBehaviour

```
public class FloatingOriginListener : MonoBehaviour
{
        private EntityQuery entityQuery;
        private void Awake()
        {
                entityQuery = World.Active.EntityManager.CreateEntityQuery(
                   ComponentType.ReadWrite<Unity.Transforms.Translation>(),
                   ComponentType.ReadOnly<FloatingObjectComponent>());
       }

        public void OnJobTrigger(Vector3 offset)
        {
                var job = new FloatingOriginJob(offset);
                var jobHandle = job.Schedule(entityQuery);
                jobHandle.Complete();
        }
}
```

Playing in the World



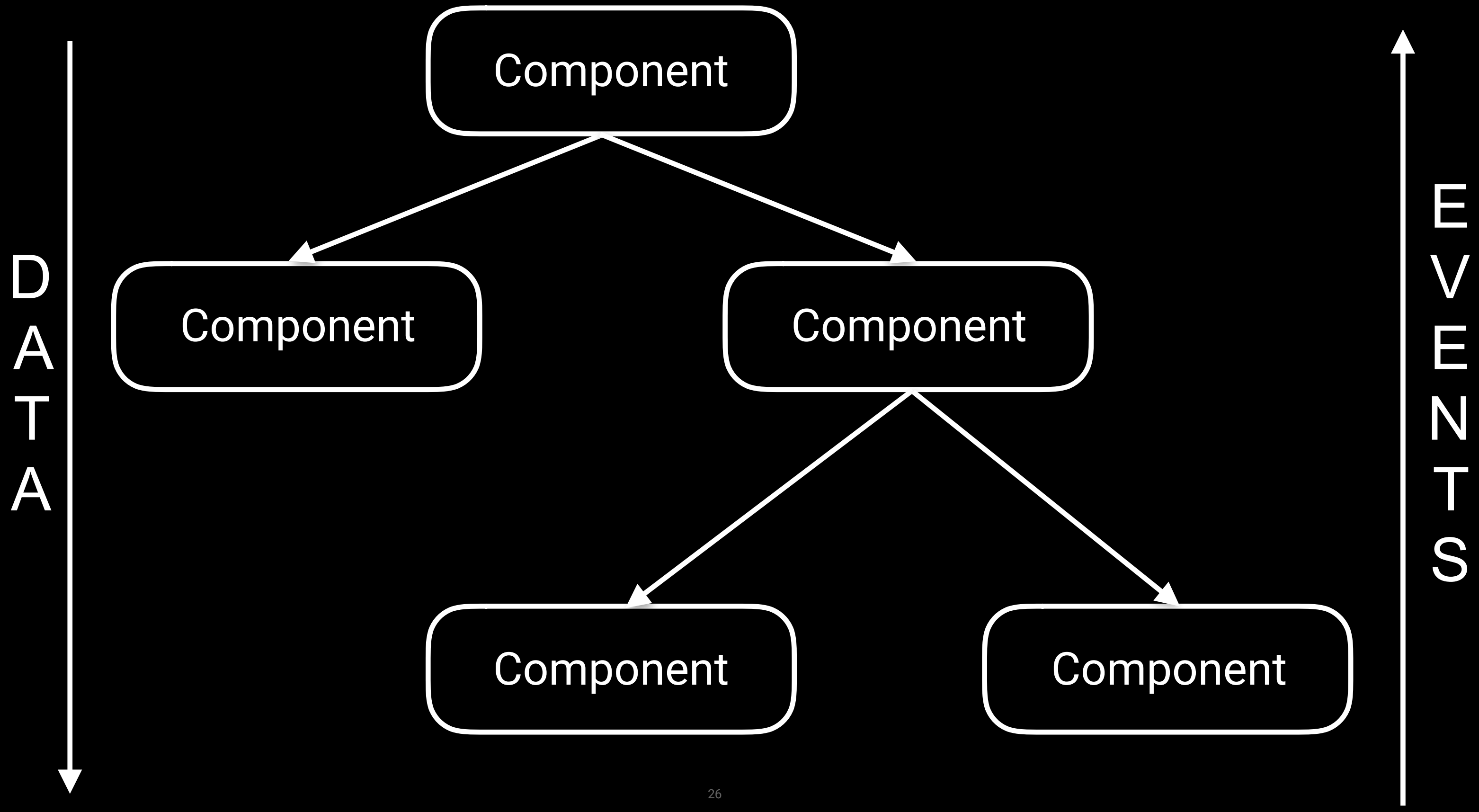
Enabling Play: Problems

- Populate the world with playable locations
- Separate art from logic
- Modular logic components

Enabling Play: Solutions

- Composition of functionality
  - Inspired by ECS and Golang
- One-way data flow
  - As seen in Elm, Vuex, Redux
- Spawning Jobs
  - Identical to decorations

One-way Data Flow



Initializable Component

```csharp
public class BaseInitializeComponent<T> : MonoBehaviour
{
    private T comp;
    public T ComponentData => comp;

    private event System.Action<T> _onInit;
    public event System.Action<T> OnInit
    {
        add
        {
            _onInit += value;
            if (comp != null)
                value(comp);
        }
        remove => _onInit -= value;
    }

    public void Initialize(T pl)
    {
        comp = pl;
        _onInit?.Invoke(pl);
    }
}
```

Base Data Component

```csharp
public struct PlayableLocation { public string ID; }

public class PlayableLocationComponent : BaseInitializeComponent<PlayableLocation> { }
```

Art Prefab Initialization

```
public class PlayableLocationArtPrefabComponent : BaseInitializeComponent<PlayableLocationComponent> { }

[RequireComponent(typeof(PlayableLocationComponent))]
public class PlayableLocationArtComponent : MonoBehaviour
{
    [SerializeField]
    private PlayableLocationArtPrefabComponent prefab;

    private PlayableLocationComponent playableLocationComponent;

    private void Awake() => playableLocationComponent = GetComponent<PlayableLocationComponent>();
    private void OnEnable() => playableLocationComponent.OnInit += OnInit;
    private void OnDisable() => playableLocationComponent.OnInit -= OnInit;

    private void OnInit(PlayableLocation obj)
    {
        var art = Instantiate<PlayableLocationArtPrefabComponent>(prefab);
        art.Initialize(playableLocationComponent);
    }
}
```

Events in the flow

- Readers choice
  - UniRx
    - ReactiveProperty
    - ReactiveCollections
    - Event Streams
  - Events And Delegates
    - The parent initialises the child and subscribes to its events

Before...

Playable Locations

- Component-based functionality

Dissecting the Vending Machine: Base

- Playable Location

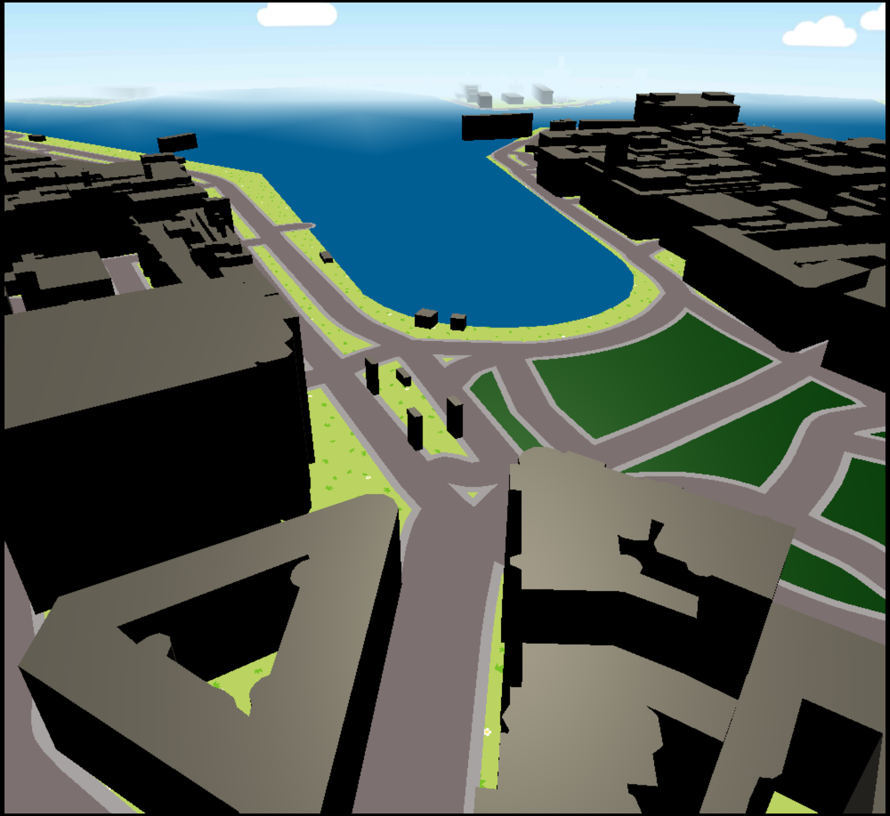
Dissecting the Vending Machine: Generic

- Playable Location
  - Distance Disable
  - Floating Object
  - Item Drop Rate Modifier
  - Cooldown Timer
  - Player Nearby Detection
  - Instance

Dissecting the Vending Machine: Instance

- Playable Location
  - Distance Disable
  - Floating Object
  - Item Drop Rate Modifier
  - Cooldown Timer
  - Player Nearby Detection
  - Instance
    - Art
      - Status Animator
    - Look At Player
    - Click Detection
    - Vending Machine Interaction

Before...

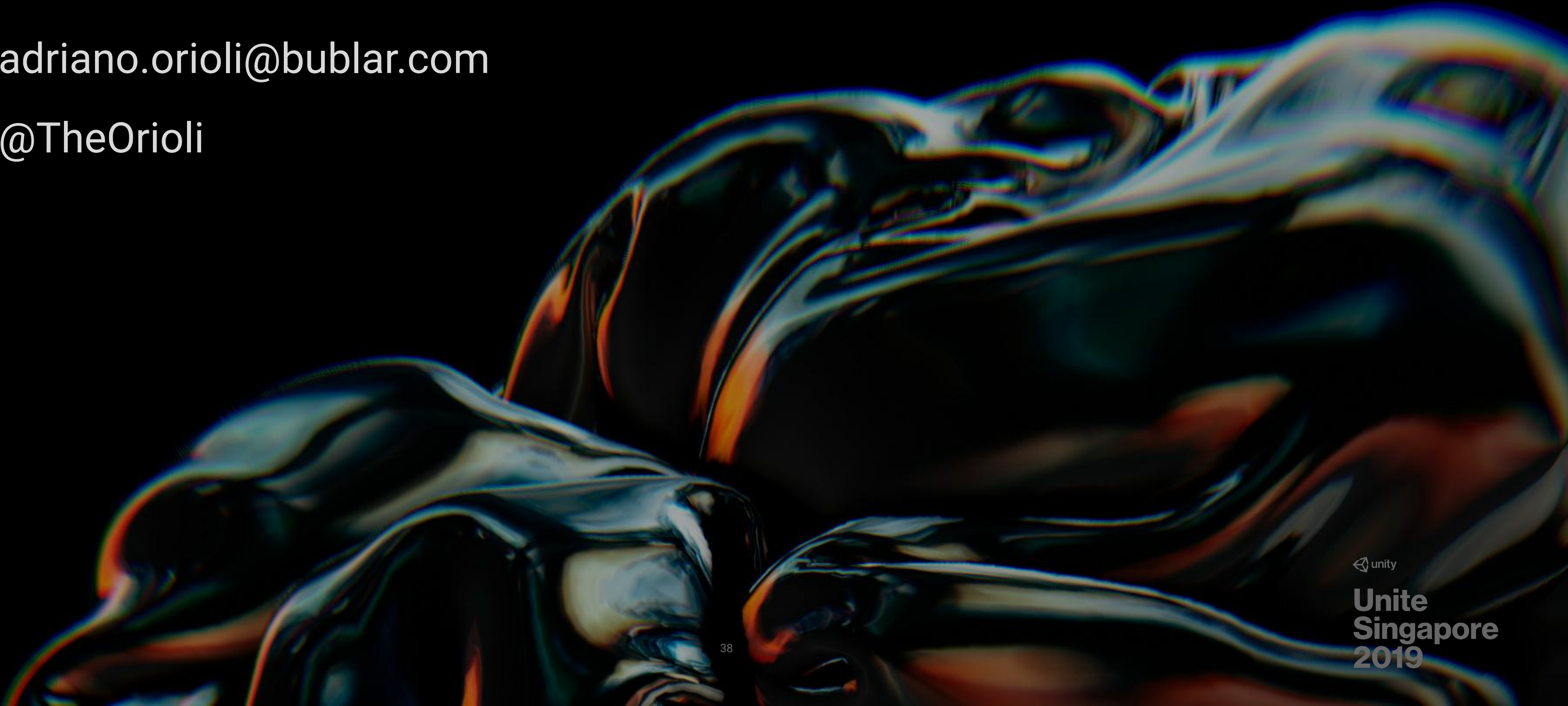


And Finally...

Thank You! Questions?

adriano.orioli@bublar.com

@TheOrioli



unity

Unite Singapore 2019