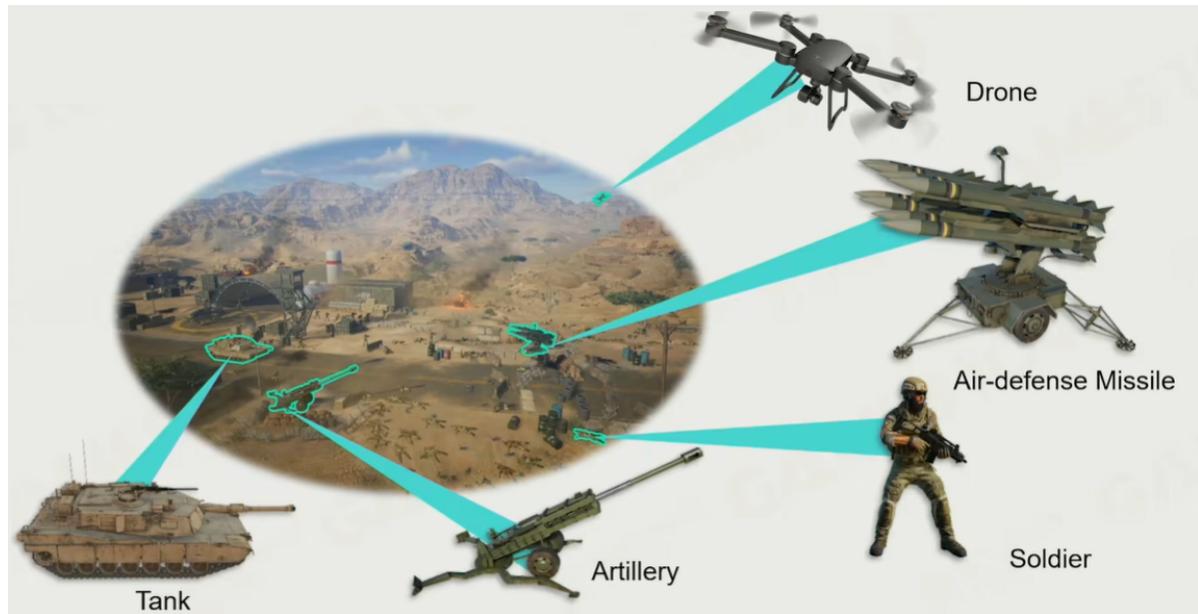


Lecture3 How to Build a Game World

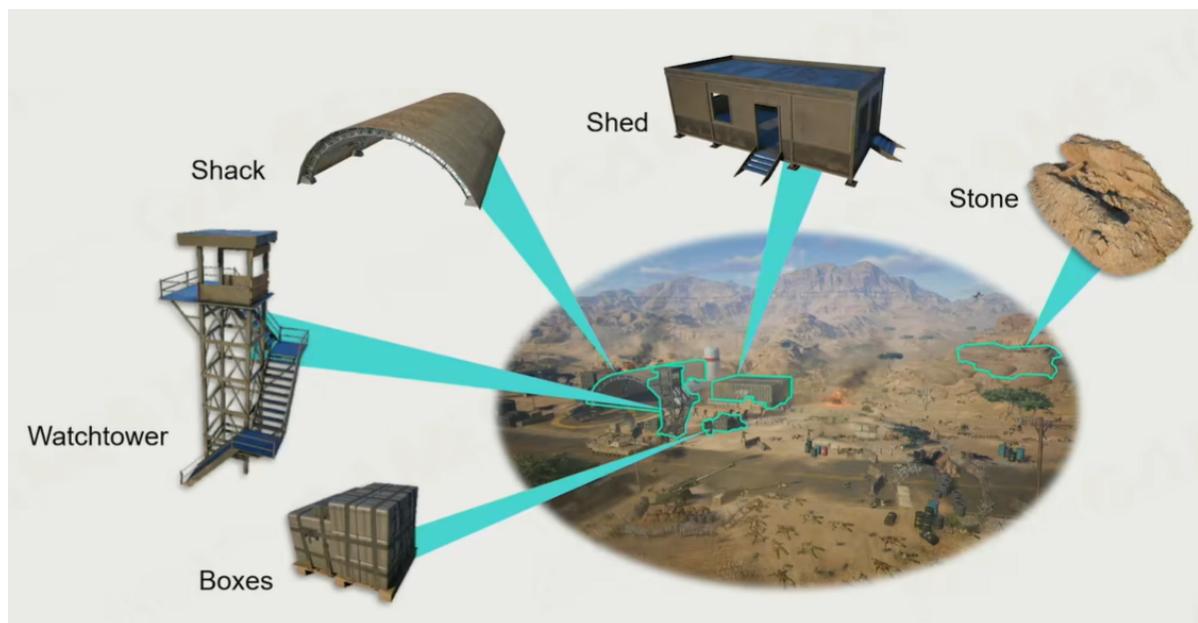
1. Game World Objects

Dynamic Game Objects



- Interactable

Static Game Objects

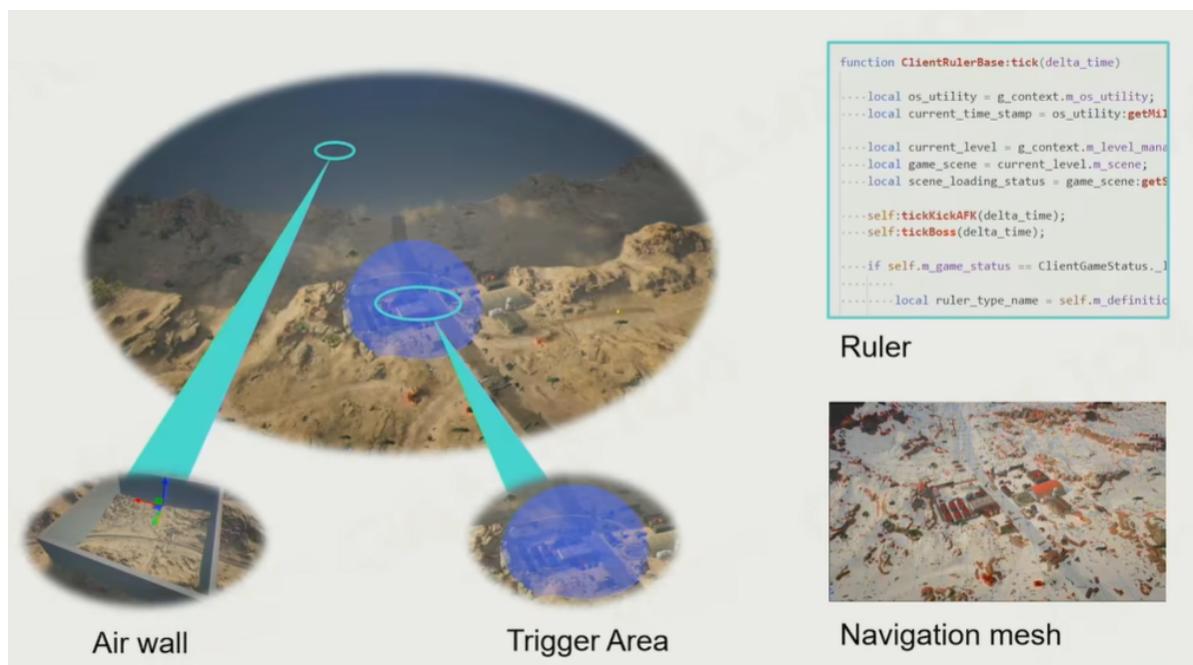


- Non-Interactable

Environments

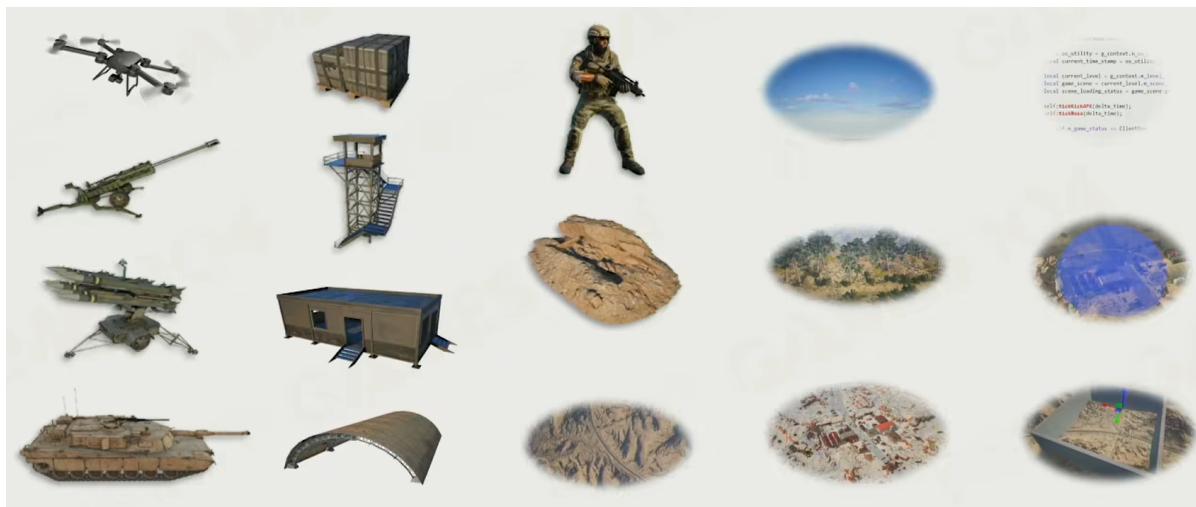


Other Game Objects



Everything is a Game Object

- Game Object (GO)



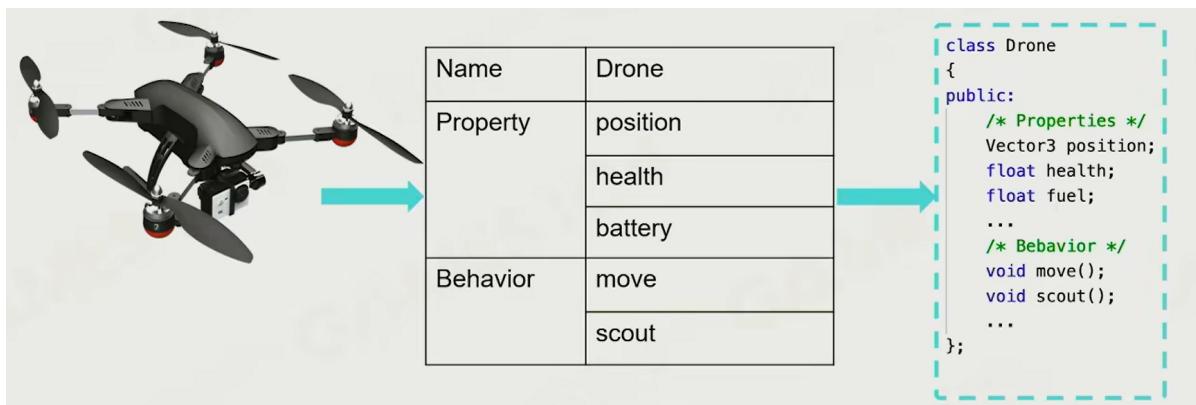
2. How to Describe a Game Object

Want to build a drone



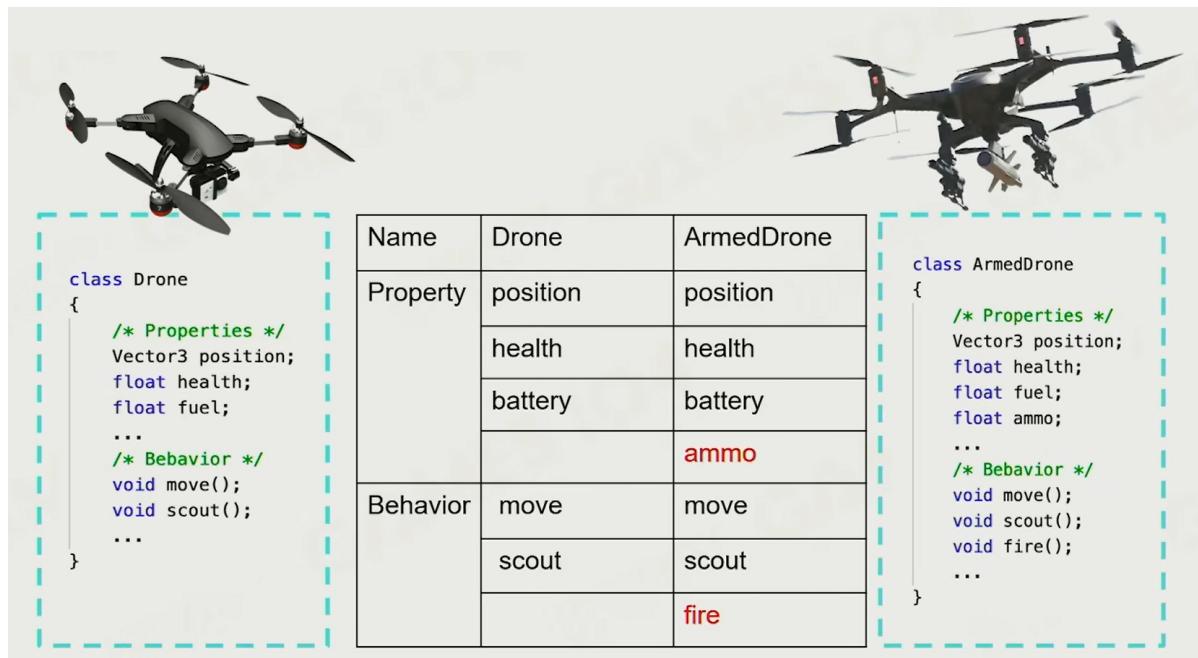
Properties and Behavior

- Property: Shape, Position, Capacity of battery, etc.
- Behavior: Movement

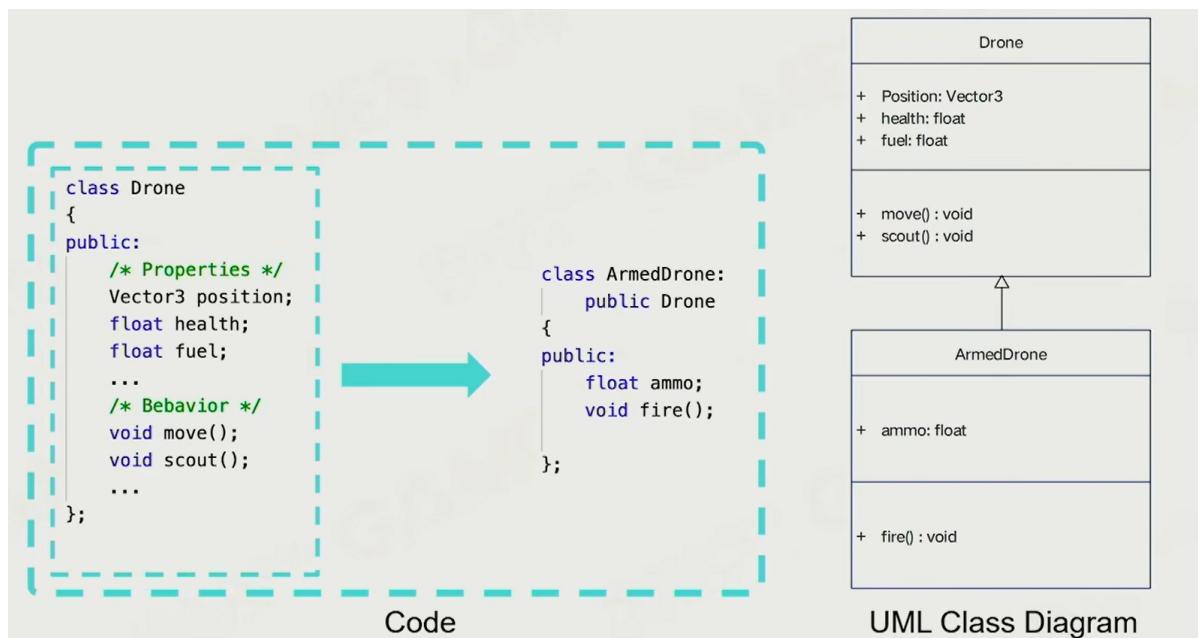


Inheritance / OOP

Drone vs. Armed Drone



Code Example



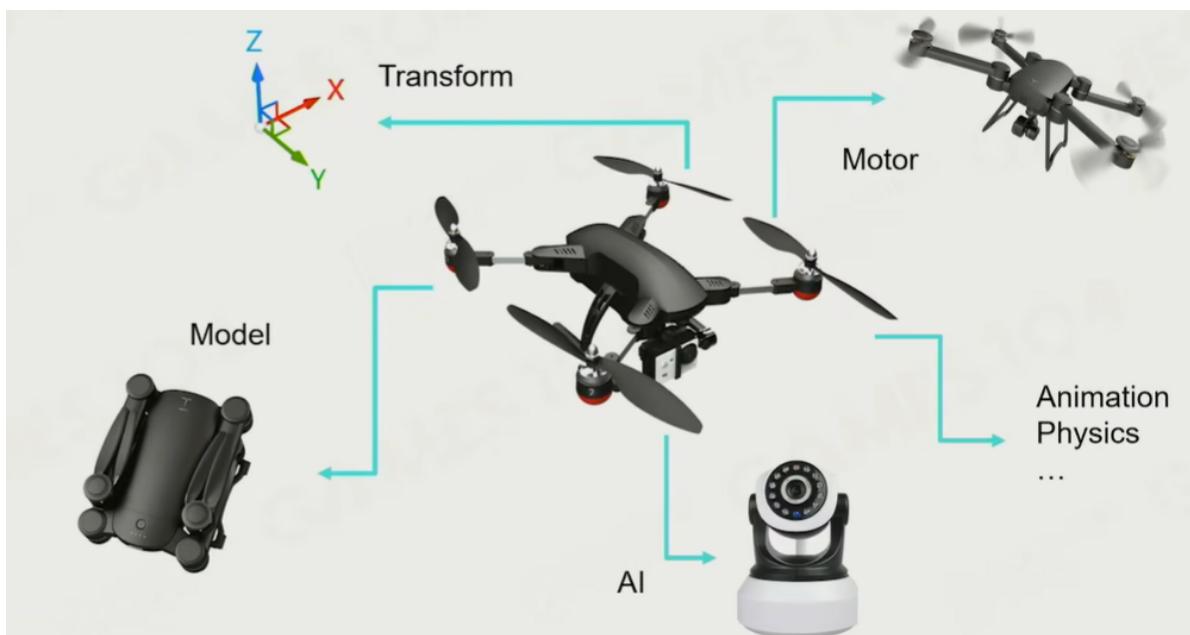
Cons

No perfect classification in the game world



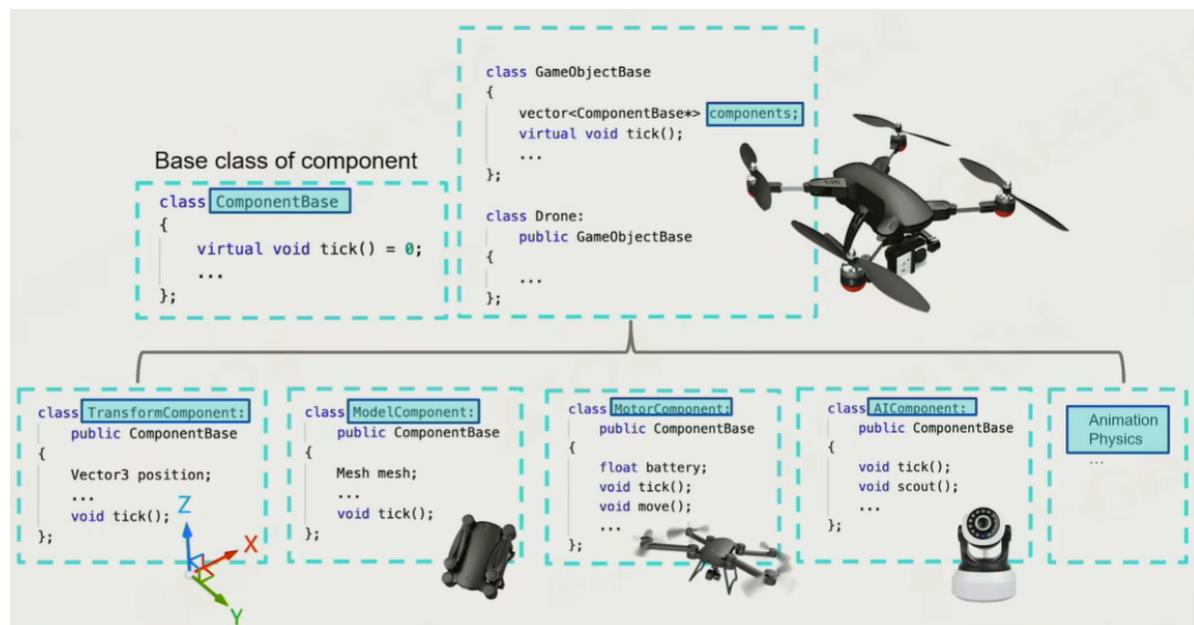
Component Based

- Component Composition in the real world

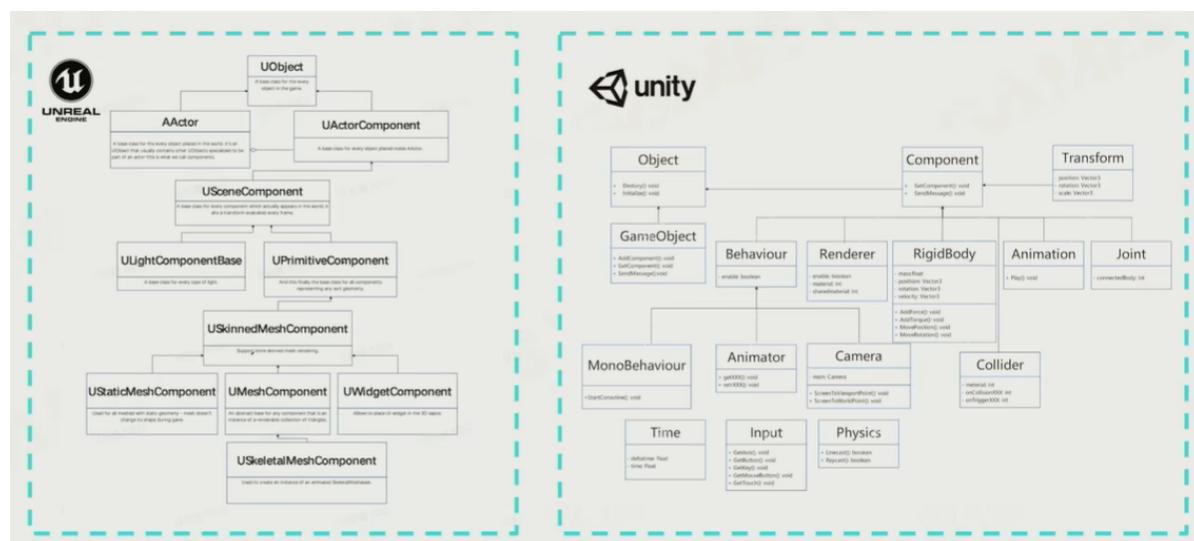




Code Example

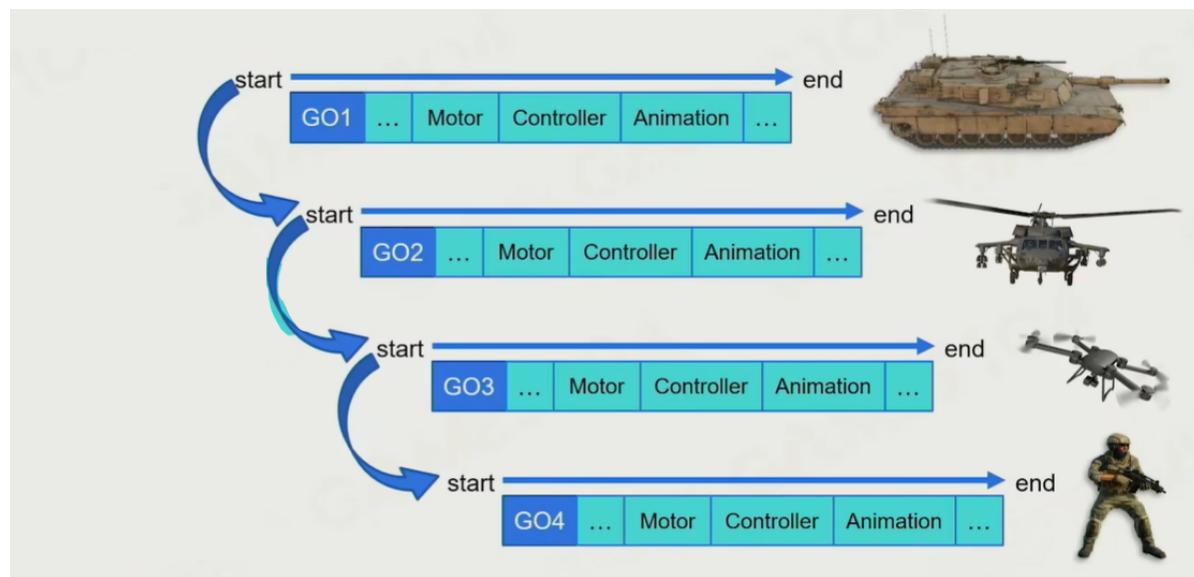


Components in Commercial Engines



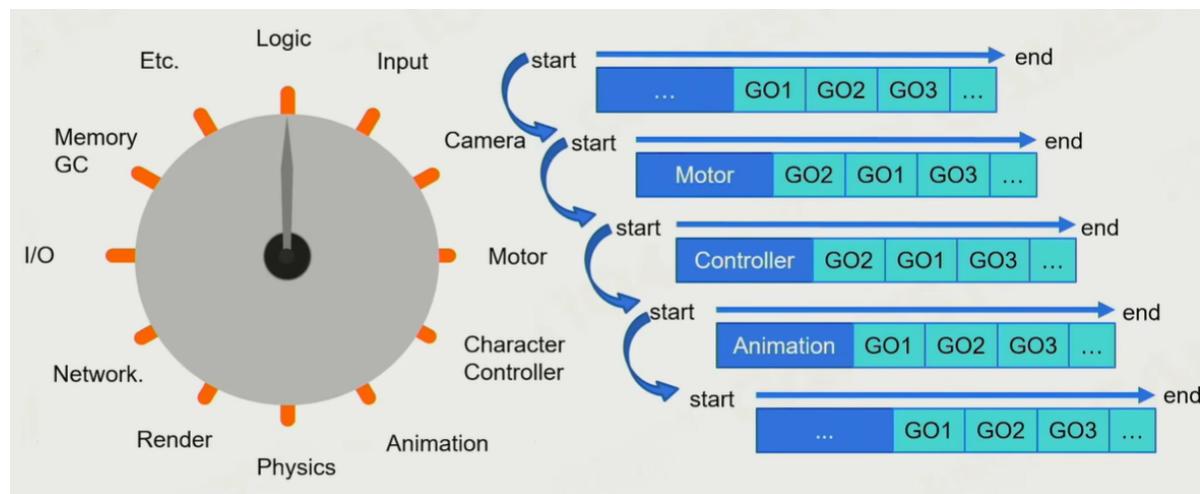
3. How to Make the World Alive

Object-based Tick

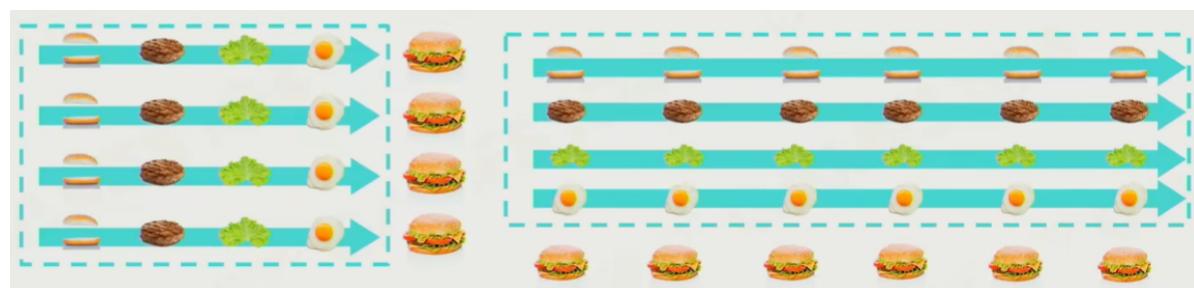


- Simple and Intuitive
- Easy to debug

Component-based Tick

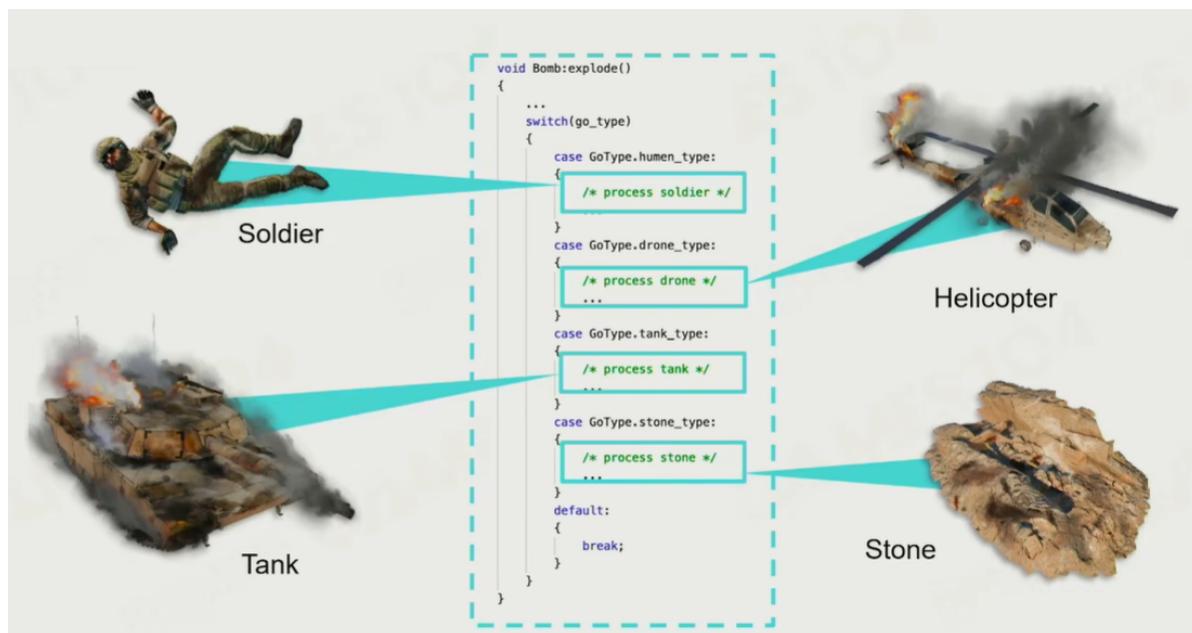


- Parallelized processing
- Reduced cache miss
- More efficient

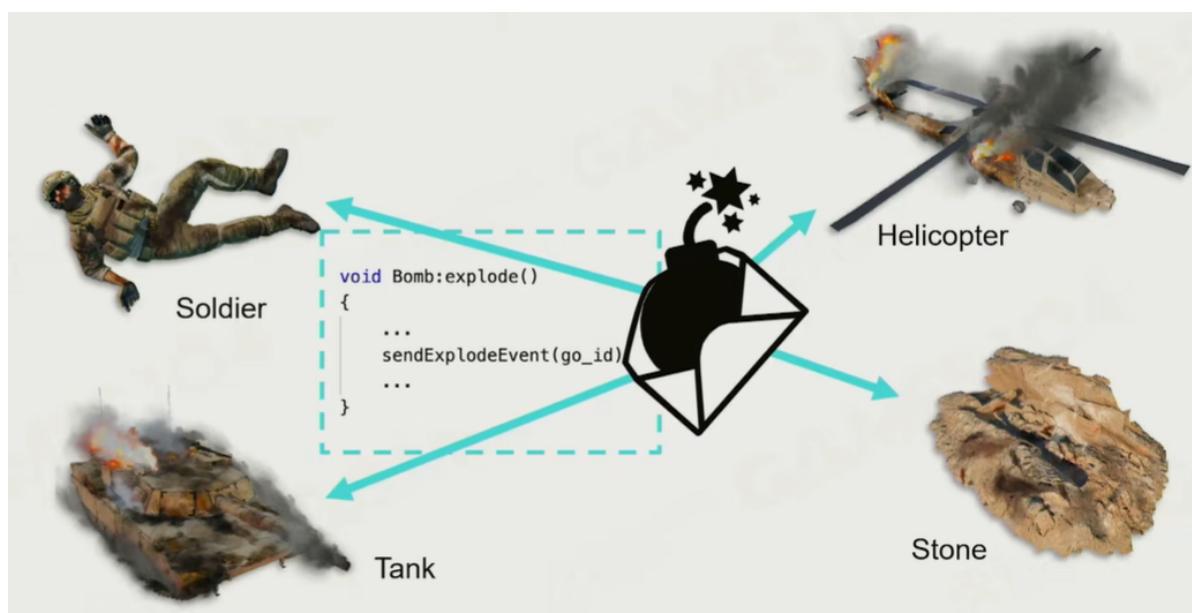


4. How to Interact between Game Objects

Hardcode



Events



- Message sending and handling
- Decoupling event sending and handling
- Encapsulate the message into an event
- Decoupling

Event Mechanism in Commercial Engines




```

using UnityEngine;

public class Example : MonoBehaviour
{
    void Start()
    {
        // Calls the function ApplyDamage with a value of 5
        // Every script attached to the game object
        // that has an ApplyDamage function will be called.
        gameObject.SendMessage("ApplyDamage", 5.0);
    }
}

public class Example2 : MonoBehaviour
{
    public void ApplyDamage(float damage)
    {
        print(damage);
    }
}

```

```

/** Event-for-when-collections-are-created */
DECLARE_EVENT_OneParam(ICollectionManager, FCollectionCreatedEvent, const FCollectionNameType&);

/** Event-for-when-collections-are-destroyed */
DECLARE_EVENT_OneParam(ICollectionManager, FCollectionDestroyedEvent, const FCollectionNameType&);

/** Event-for-when-assets-are-added-to-a-collection */
DECLARE_EVENT_TwoParams(ICollectionManager, FAssetsAddedEvent, const FCollectionNameType&, const TArray< FName >&);

/** Event-for-when-assets-are-removed-from-a-collection */
DECLARE_EVENT_TwoParams(ICollectionManager, FAssetsRemovedEvent, const FCollectionNameType&, const TArray< FName >&);

/** Event-for-when-collections-are-renamed */
DECLARE_EVENT_TwoParams(ICollectionManager, FCollectionRenamedEvent, const FCollectionNameType&, const FCollectionNameType&);

/** Event-for-when-collections-are-re-parented (params: Collection, OldParent, NewParent) */
DECLARE_EVENT_ThreeParams(ICollectionManager, FCollectionReparentedEvent, const FCollectionNameType&, const TOptional< FCollectionReparentedEvent > OnCollectionReparented() => 0);

/** Event-for-when-collections-is-updated, or otherwise changed and we can't tell exactly how (eg, after updating fr */
DECLARE_EVENT_OneParam(ICollectionManager, FCollectionUpdatedEvent, const FCollectionNameType&);

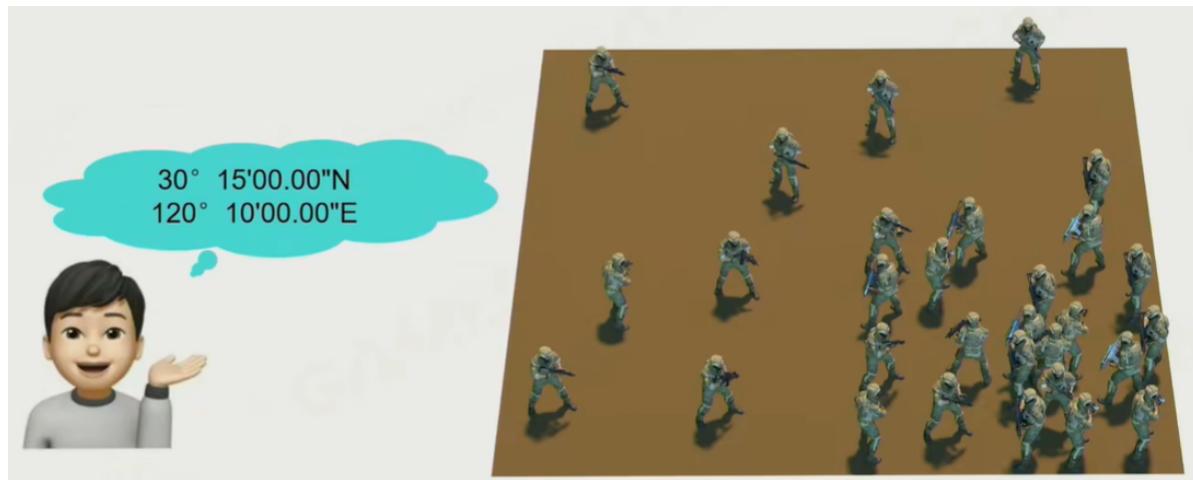
/** When a collection-checkin-happens, use this event to add additional text to the changelist-description */
DECLARE_EVENT_TwoParams(ICollectionManager, FAddToCollectionCheckinDescriptionEvent, const FName& /*CollectionName*/
virtual FAddToCollectionCheckinDescriptionEvent OnAddToCollectionCheckinDescriptionEvent() => 0;

```

5. How to Manage Game Objects

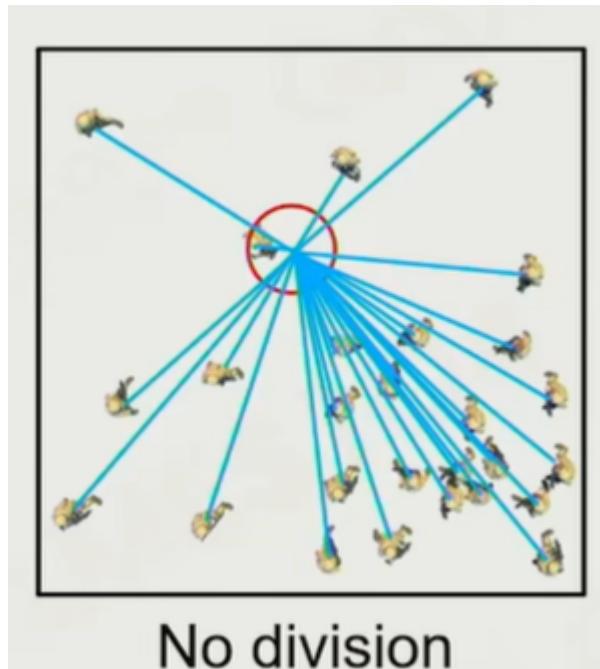
The bomb effects all the game objects in the scene

Scene Management



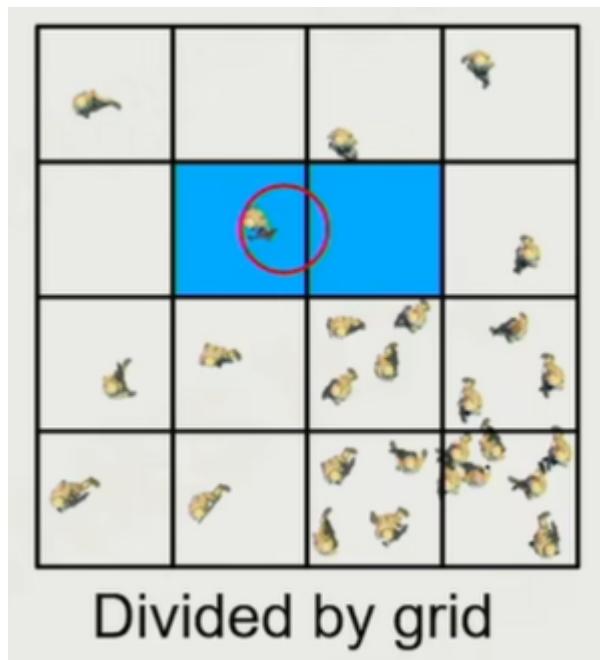
- Game objects are managed in a scene
- Game object query
 - By **unique game object ID**
 - By object position

No division



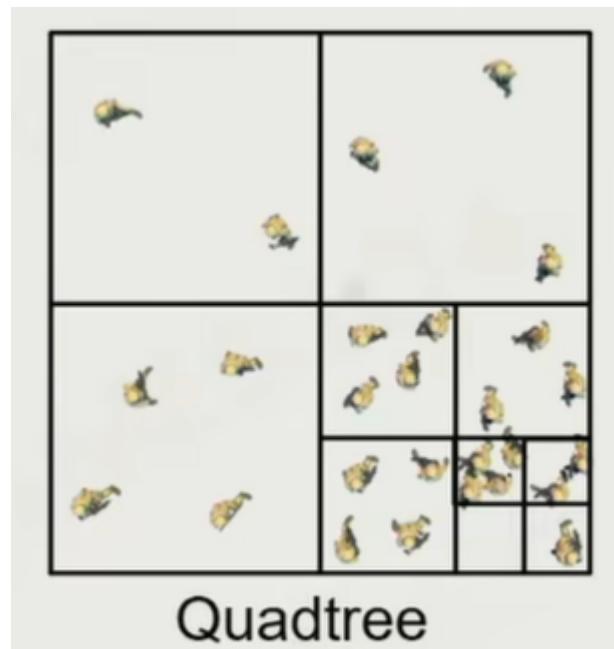
- Notify all the game objects in the scene
- $O(n^2)$ challenge

Divided by grid

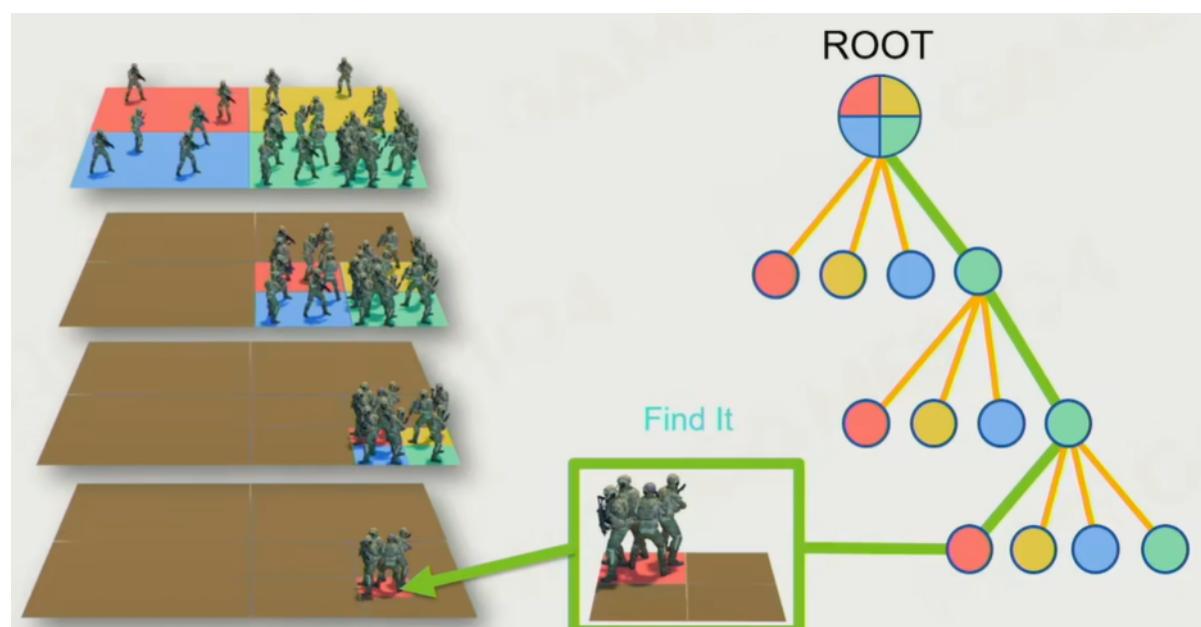
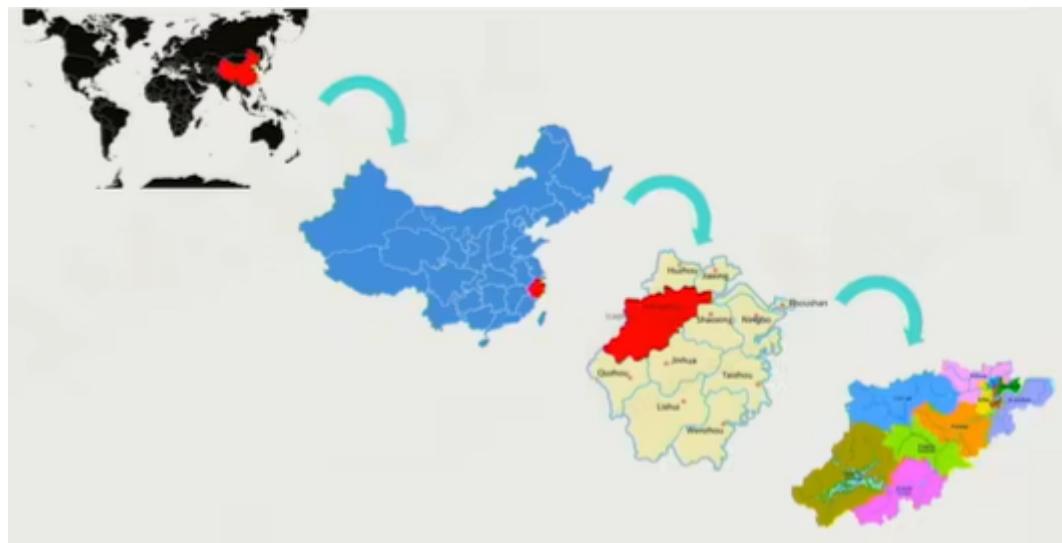


- Find the neighbor grid
- Cons: game objects in the scene are not equally distributed

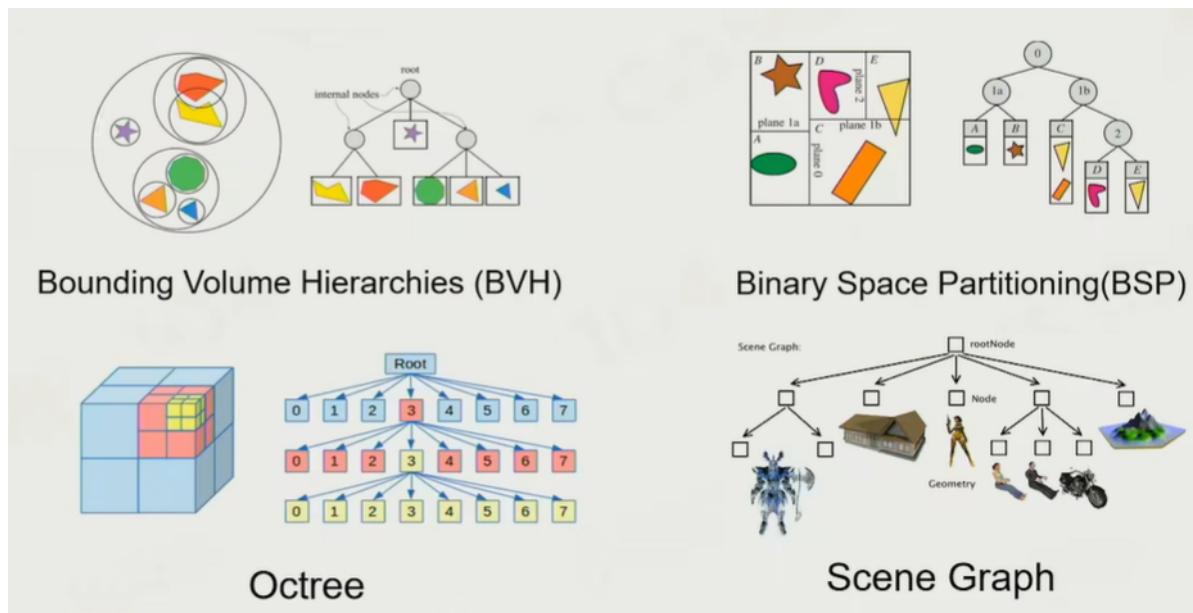
Hierarchical Segmentation



- Segmented space by object clusters



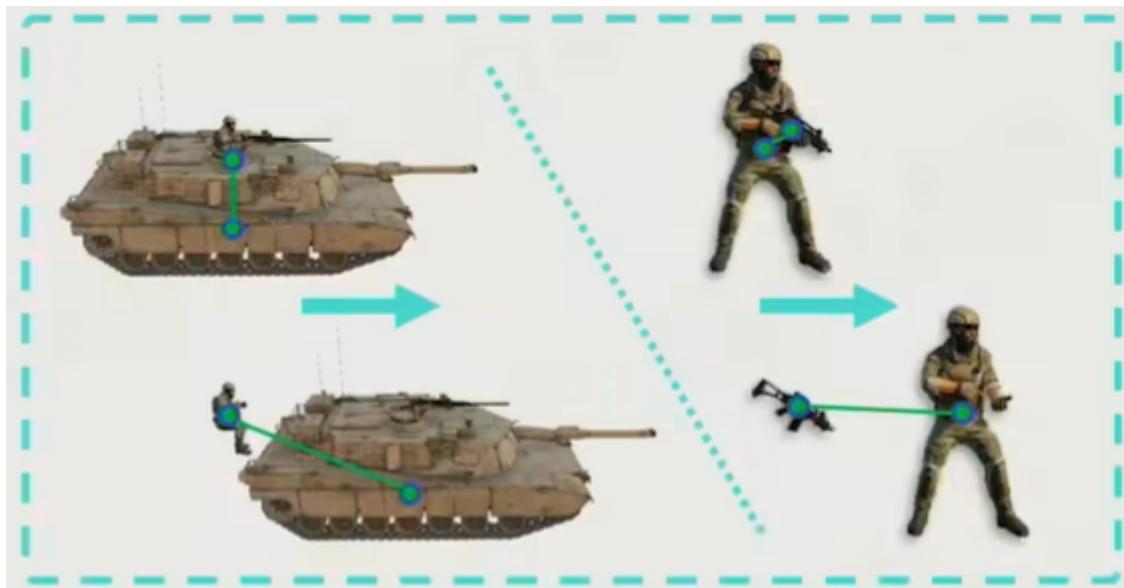
Spatial Data Structures



- Is the core in game engine and scene management
- Implement in different structure due to different game

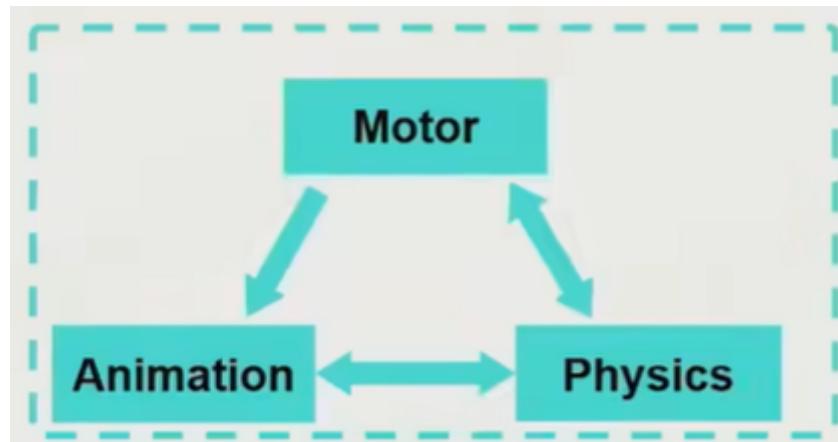
6. Other

GO Bindings



- When player is bound with a tank, how to tick
- Tick father game objects the first, and then tick the child game objects

Component Dependencies



- The sequence to tick the component is important (May in parallel execution)
- Different components can send message to each other simultaneously
- We want the game to be deterministic

Post Office



- Post office takes charge of all the messages
- Pre-tick and Post-tick to solve the sequential problem

7. Summary

- Everything is a game object in the game world
- Game object could be described in the component-based way
- States of game objects are updated in tick loops
- Game objects interact with each other via event mechanism
- Game objects are managed in a scene with efficient strategies