



PUMPKIN SEED FRAMEWORK

UAT GSIP

12.13.2021

Jake A O'Connor



What is it?



Turn-Based Game Framework

Cross-Platform

Genre-Agnostic

Highly Modular



What is the need?



Few Cross-Platform

Nothing Genre-Agnostic



Construction



.NET Core

Multi-Threaded

Software Patterns





Modular Construction

```
public Framework(  
    TGameState initialState,  
    IReadOnlyDictionary<uint, IActorController> characterControllers,  
    ITurnController<TTurnPhase> turnController,  
    IStandardActionGenerator standardActionGenerator,  
    IWinningPartyCalculator winningPartyCalculator,  
    IEntityChangeCalculator entityChangeCalculator,  
    IInitiativeQueue initiativeQueue,  
    IEventQueue eventQueue)
```




Basic Building Blocks

Entities

```
public interface IEntity : ICloneable
{
    /// <summary>
    /// Unique id used to track this entity.
    /// </summary>
     21 usages  2 implementations
    uint Id { get; }
}
```

Actors

```
public interface IActor : IEntity, ICloneable
{
    /// <summary>
    /// Party Id of the party which this Actor belongs to.
    /// </summary>
    10 usages 2 implementations
    uint PartyId { get; }

    /// <summary>
    /// Should this actor be considered during turns.
    /// </summary>
    6 usages 2 implementations
    bool IsActive { get; }

    /// <summary>
    /// Get this Actor's initiative step.
    /// </summary>
    /// <returns>Initiative step of the Actor.</returns>
    1 usage 2 implementations
    float GetInitiative();

    /// <summary>
    /// Get all Actions which this Actor can perform on their turn.
    /// </summary>
    /// <param name="gameState">Current active gameState.</param>
    /// <returns>Collection of all available Actions for this Actor.</returns>
    1 usage 2 implementations
    IReadOnlyCollection<IAction> GetAllActions<TTurnPhase>(IGameState<TTurnPhase> gameState) where TTurnP
}
```

Initiative

1 usage 1 implementation

```
uint InitiativeThreshold { get; }
```

```
/// Get the current state of the initiative queue, in Actor-Initiative pairs. ...
```

1 usage 1 implementation

```
IEnumerable<(uint actorId, float initiative)> GetCurrentQueue();
```

```
/// Get the next Actor in initiative queue. ...
```

4 usages 1 implementation

```
IActor GetNext();
```

```
/// Add a new Actor to the initiative queue. ...
```

1 implementation

```
bool Add(IActor actor, float initialValue);
```

```
/// Manually change an Actor's current initiative. ...
```

5 usages 1 implementation

```
void Update(uint actorId, uint changeValue);
```

```
/// Remove an Actor from the initiative queue. ...
```

1 implementation

```
bool Remove(uint actorId);
```

```
/// Generate a preview of the upcoming initiative order. ...
```

The background of the slide features a light green, textured surface with several overlapping, slightly crumpled white papers. Some of the papers have faint horizontal lines. On the left side, a solid red arrow points horizontally towards the right, partially overlapping the papers.

Examples

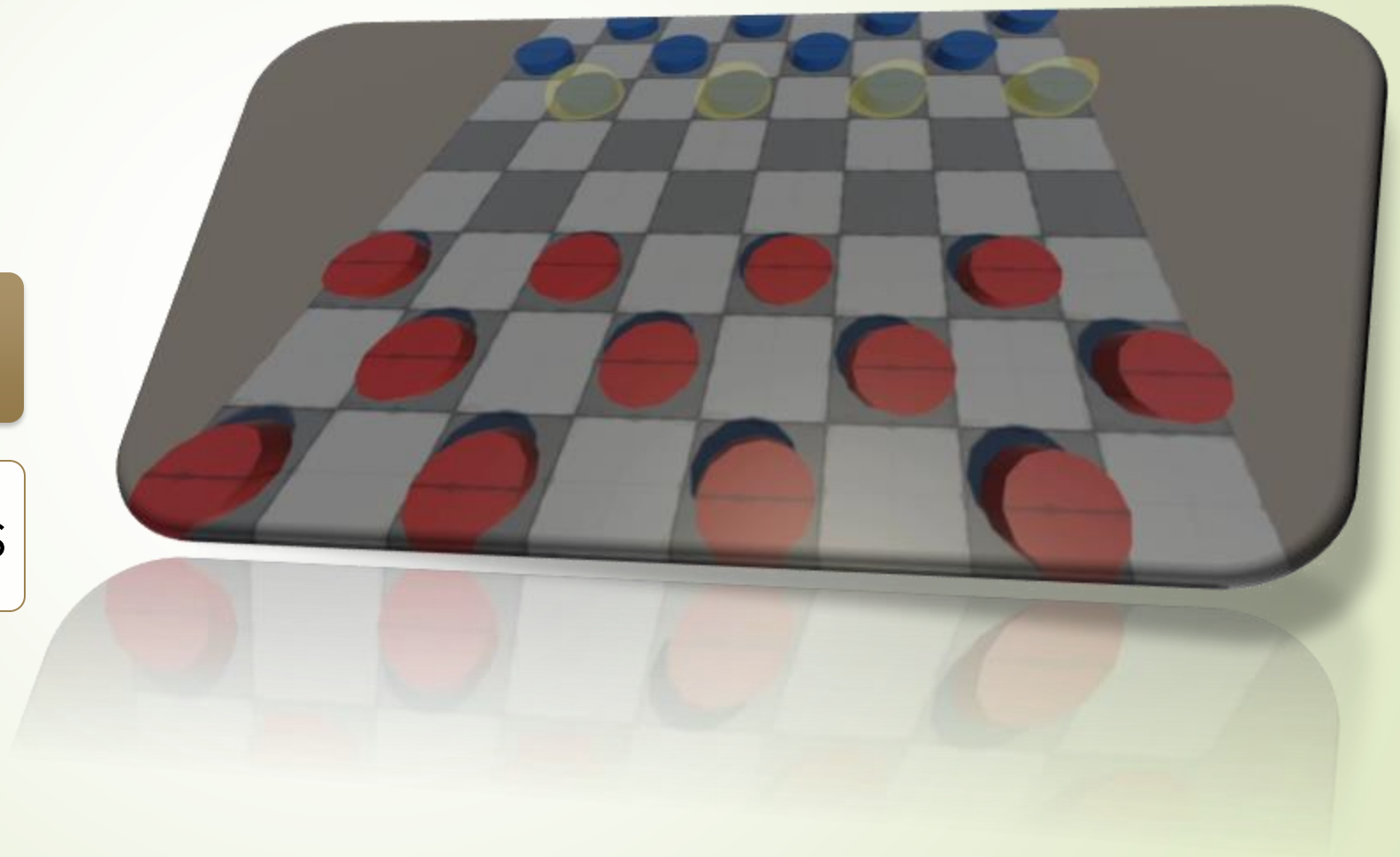
Checkers

Entities

Pieces

Actors

Players



Console RPG

Entities

Actors

Characters

Enemies



Web Game

Entities

Items

Equipment

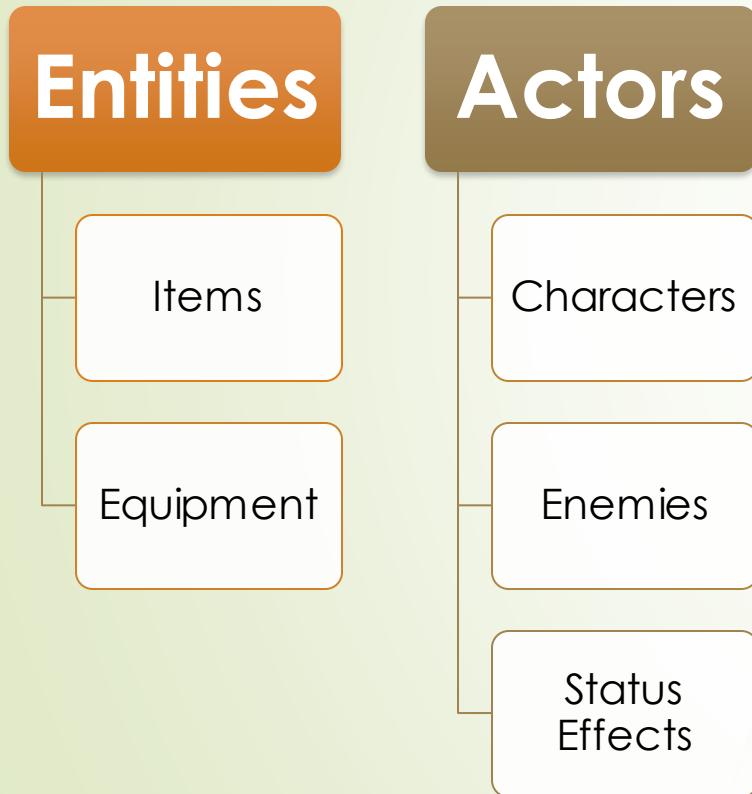
Actors

Characters

Enemies



Unity Game





Pumpkin Seed Framework