

The Simplest AI Trick in the Book

Rez Graham | Software Engineer III, Maxis

Kevin Dill | Chief Architect of the Game AI Architecture, Lockheed Martin

Brian Schwab | Senior AI/Gameplay Engineer II, Formerly of Blizzard Entertainment

Cloderic Mars | Lead Software Engineer, MASA Group

John Abercrombie | Lead Programmer, Irrational Games

Michael Dawe | Software Developer, Harmonix Music Systems

Jeet Shroff | Senior Game Programmer, Avalanche Studios

Simplest Trick

David “Rez” Graham

AI Programmer – Maxis – The Sims

AI Team



Play the Game

- Play the game once week
 - Production, design, engineering, and QA
- Find bugs
- Discuss design decisions
- Generate future work

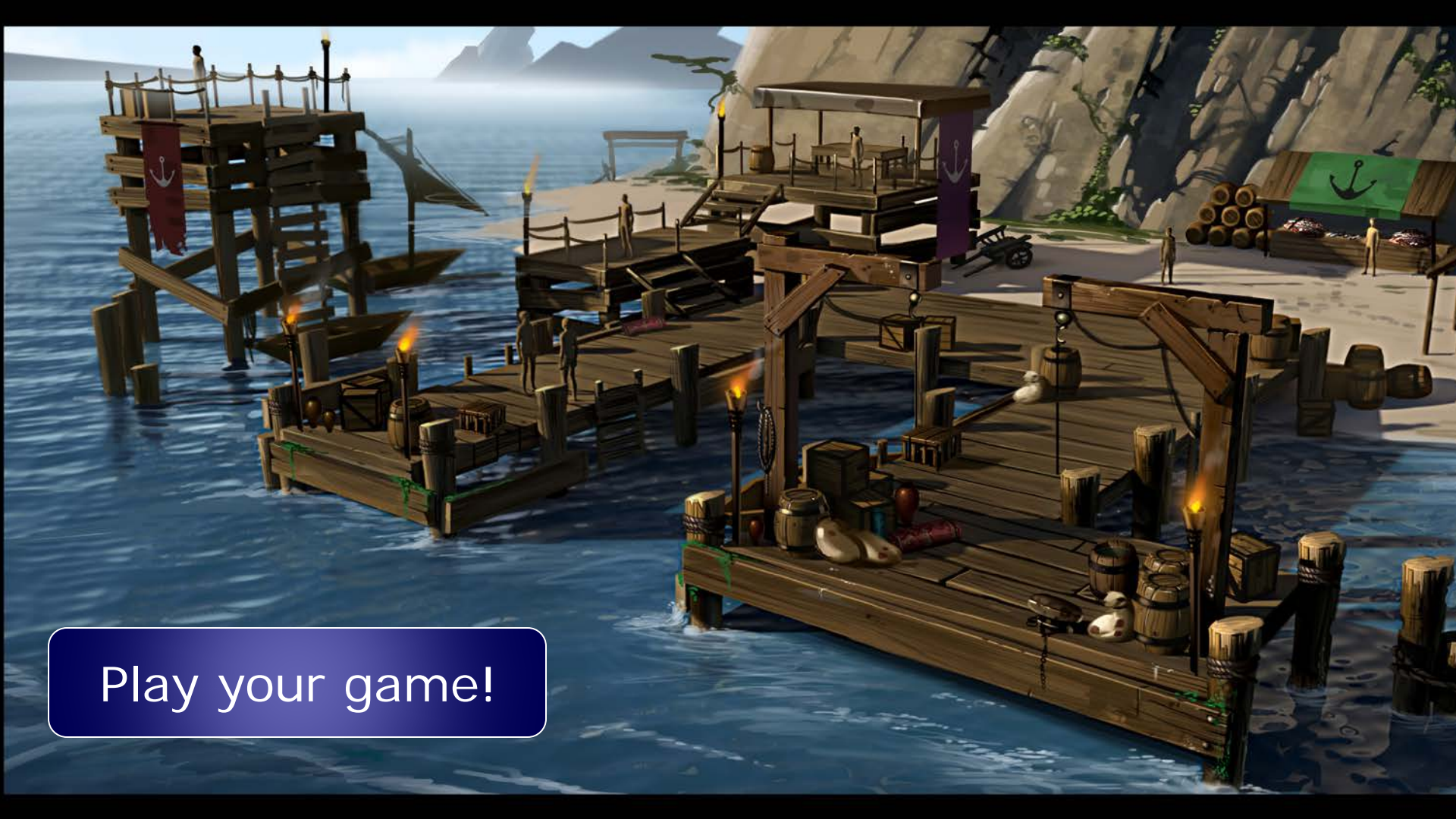
?

?



?





Play your game!

The Simplest AI Trick in the Book

MODULAR TARGETS

Target Interface

```
class AITargetBase : public AIBase
{
    virtual void Init(const AICreationData& cd)

    virtual const AIVector3& GetPosition();
    virtual const AIVector3& GetFacing();

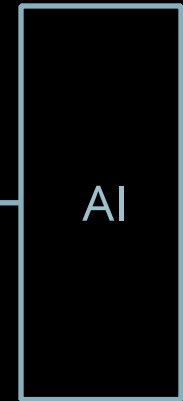
    virtual bool HasEntity();
    virtual AIEntity& GetEntity();

    virtual bool IsValid();
};
```

- ◎ This is what I configure
- ◎ Short and sweet



- ◎ This is what I work with in code



- ◎ This is where the complexity is!
- ◎ Implement exactly once
- ◎ Reuse over and over
 - ◎ Even across projects

Simple Examples

```
<Target Type="Self" />
```

```
<Target Type="Camera" />
```

```
<Target Type="NamedEntity" EntityName="Joe" />
```

```
<Target Type="Variable" VariableName="EnemySub">  
  <Offsets x="0" y="-500" />  
</Target>
```

Complex Example

```
<Target Type="EntityPicker">
  <Filters>
    <Filter Type="Side" Side="US" />
    <Filter Type="Geometry" .../>
  </Filters>
  <Picker Type="WeightBasedRandom">
    <Consideration Type="LineOfSight .../>
    <Consideration Type="Distance" .../>
    <Consideration Type="Rank" .../>
    <Consideration Type="Role" .../>
  </Picker>
</Target>
```

The Simplest AI Trick in the Book

MODULAR TARGETS

SIMPLEST TRICK

BRIAN SCHWAB

BUILD A BREAKPOINT AND COMMENT INTO YOUR SCRIPTING LANGUAGE

Conditional breakpoints are awesome

- Don't assert, trap in a loop and allow moving forward

Make commenting script part of checking it in



ASK “WHAT ARE YOU TRYING TO FIX?”

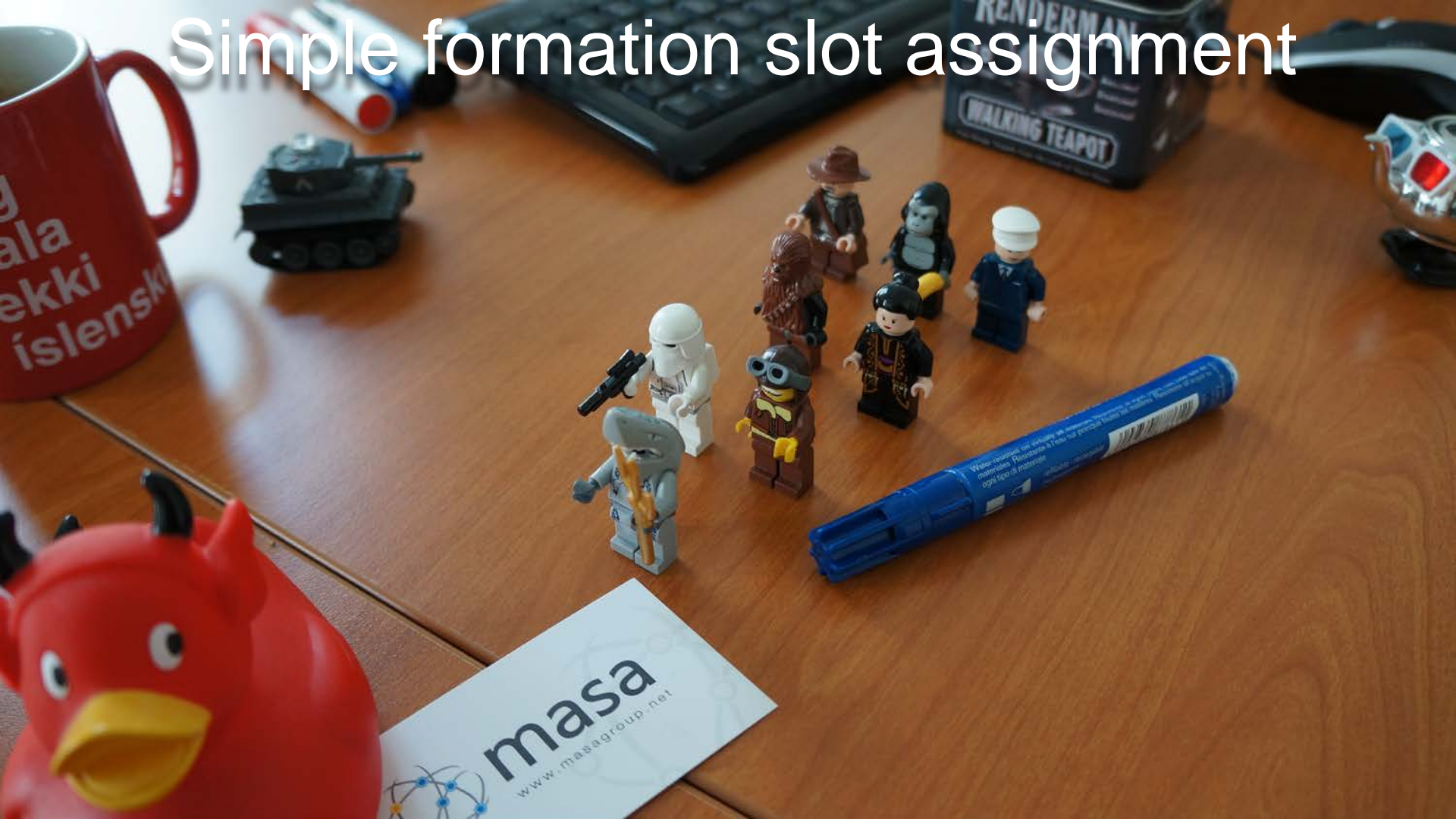
Cure the Disease, not the Symptom

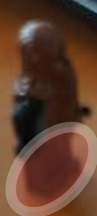


Simple formation slot assignment



Simple formation slot assignment



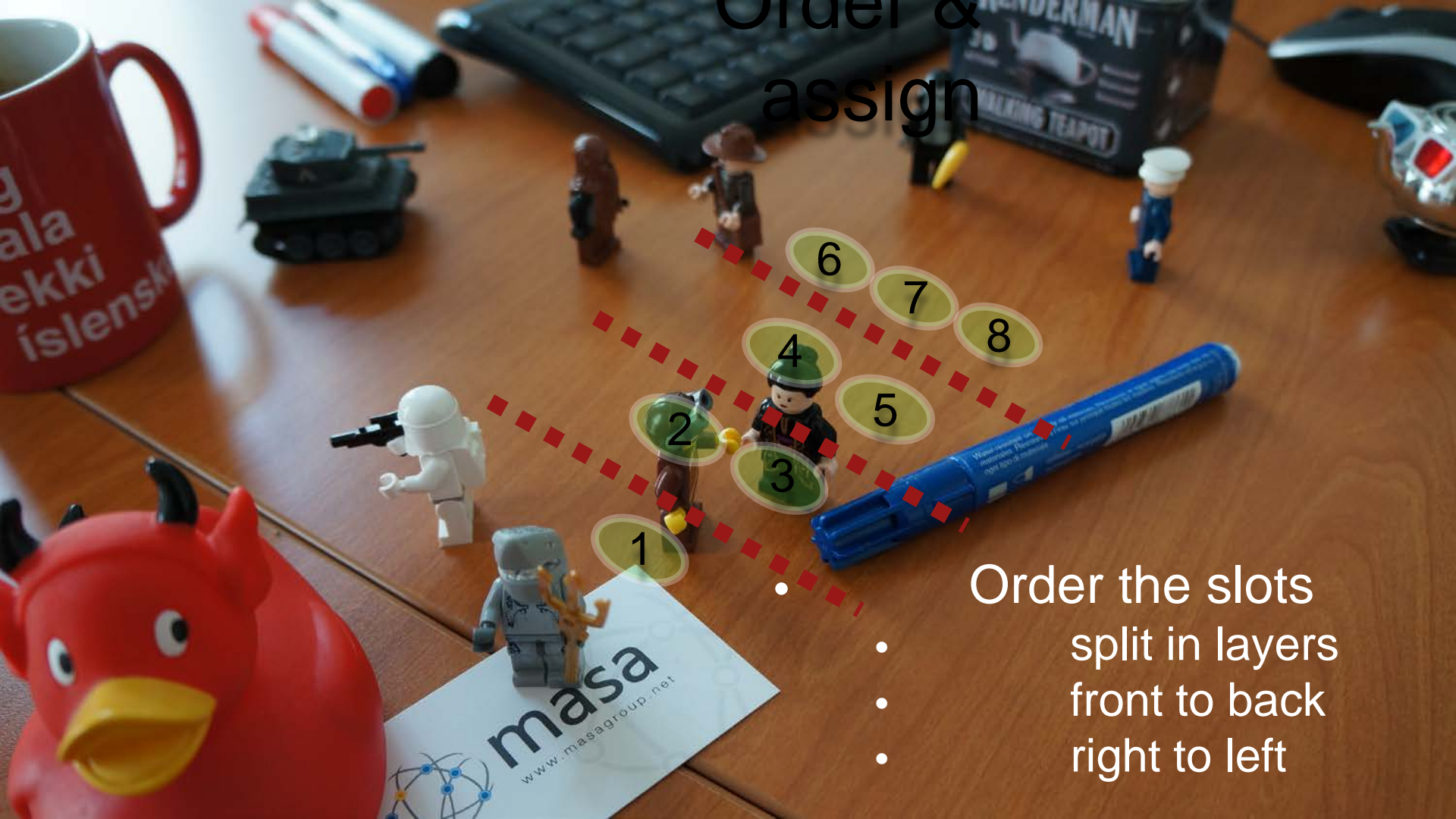




•
•

Greedy won't work
Optimal is NP-Hard

Order & assign



Order the slots
split in layers
front to back
right to left

Order & assign



Order the initial positions
target orientation
layers count
layers size
Assign n^{th} to n^{th}

Everything is awesome!



Clodéric Mars
cloderic.mars@masagroup.net
[@clodericmars](https://twitter.com/clodericmars)

AI PRODUCTION TRICKS I WISH SOMEONE HAD TOLD ME...

JOHN ABERCROMBIE

KIS(S) - Keep your implementation simple.

Know what you're trying to do!

Communicate well with others.

Don't make the same mistake again.

**Don't let others make the
mistakes you've made again.**

**If you see something wrong with the AI,
find a way to fix it.**

**When playtesting your work,
learn how to separate
your game developer self
from being a player.**

HAVE FUN!
This is game development!

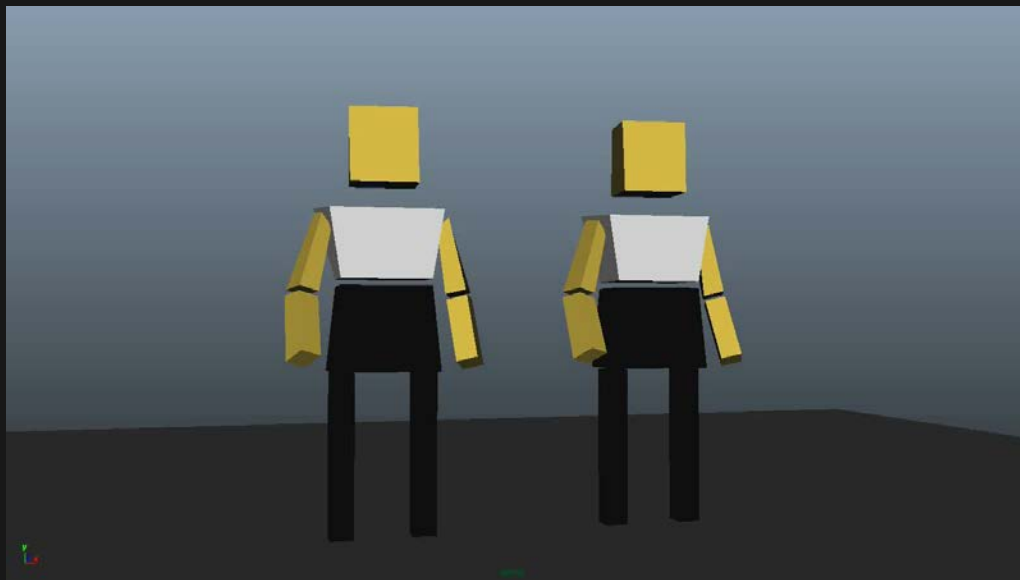
Stop your AI from looking so artificial with the world's simplest blackboard

Michael Dawe

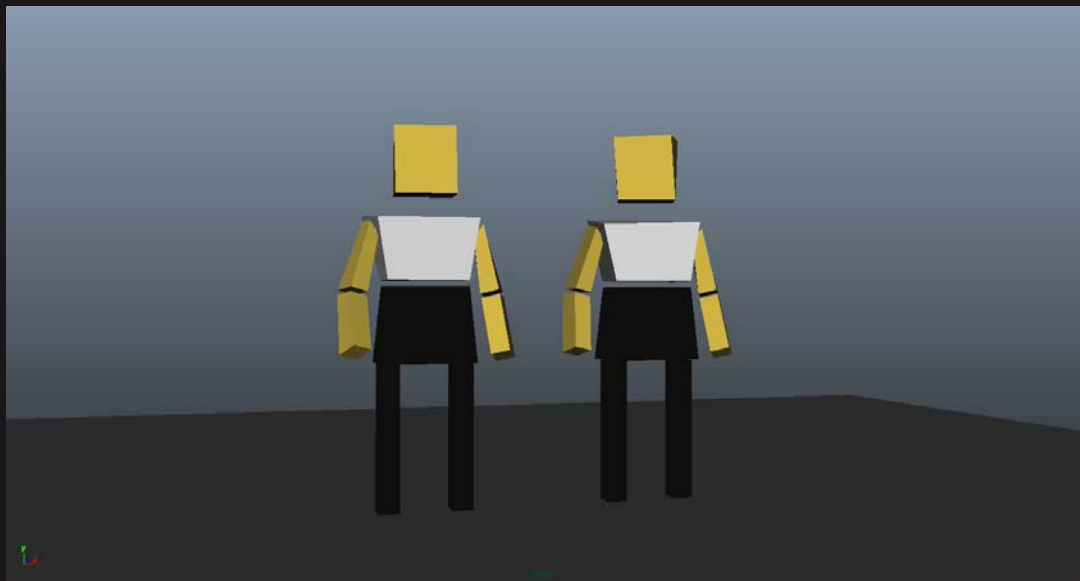
dawe@harmonixmusic.com

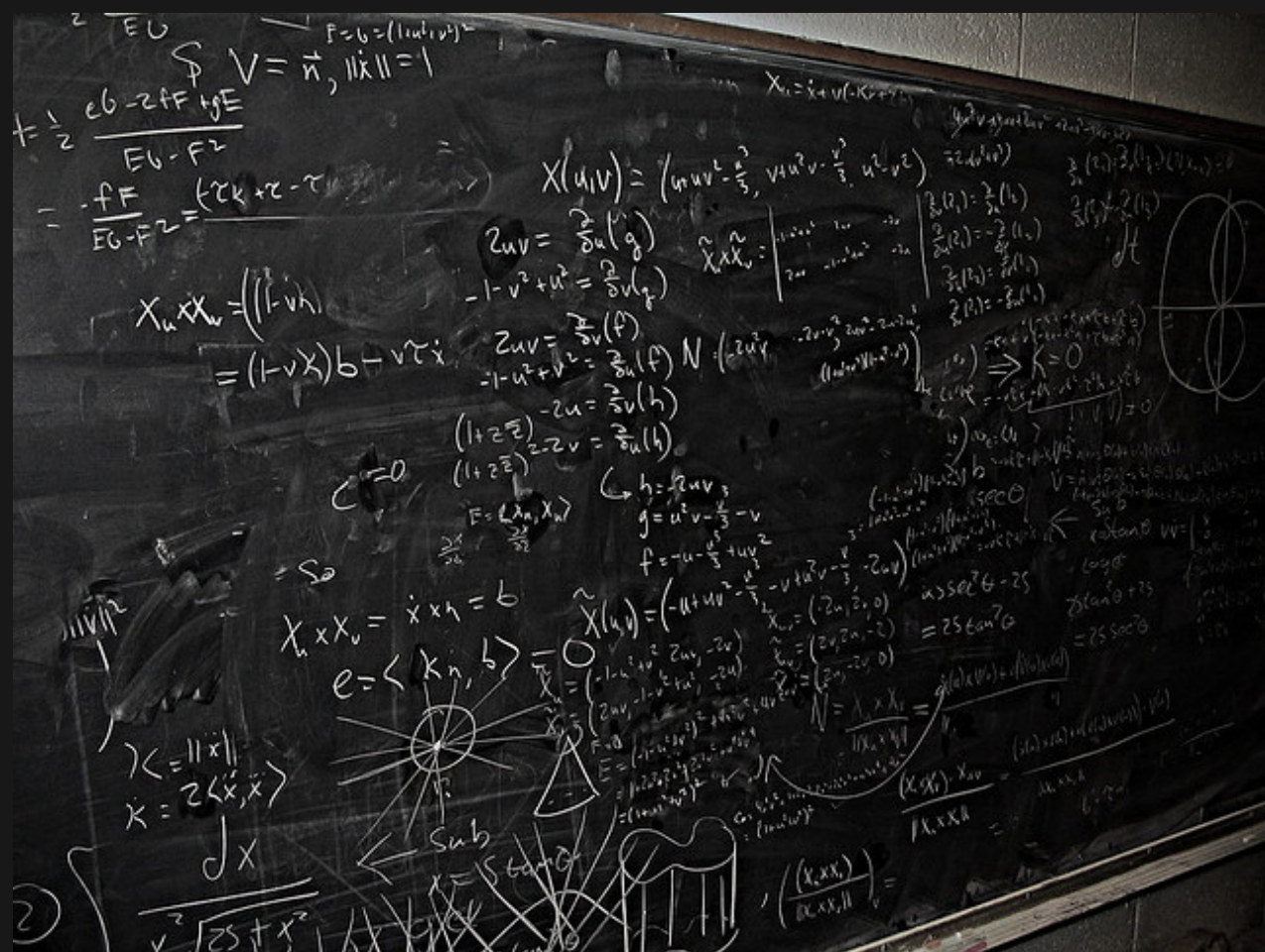
@mdawe

Problem: non-robotic characters looking like robots



Solution: small anim offset





Blackboard internals

```
class AnimBlackboardEntry {
```

```
    Hash animId;          // any unique id for the anim
```

```
    unsigned int time;     // game time the anim played
```

```
    vectors worldLocation; // optional, but useful
```

```
    AnimPriority priority; // What "type" of anim it is
```

```
};
```

```
class AnimBlackboard {
```

```
    AnimBlackboardEntry blackboard[32]; // Size based on # of NPCs
```

```
    size_t front, back; // Treat array as a circular queue
```

```
    unsigned int priorityTimes[]; // How long do we have to  
        wait to play an anim of this priority?
```

```
};
```

Blackboard internals

```
bool AnimBlackboard::CanPlayAnim(Hash anim) {  
    For each entry  
        If entry is too old, remove it  
        If entry isn't our anim, skip it  
        If entry is our anim:  
            If entry.time < min delay for entry.priority  
                return false  
            Else if entry.location too close  
                return false  
  
    // Could add the anim to the blackboard here, or do it afterwards  
    return true  
}
```



AnimPriority priority;

- `enum AnimPriority { NORMAL, REACTION };`
- ~1s delay for normal anims
- ~300ms delay for reaction anims

2010



19 10



Thanks!

- Thanks to Kristin Palach for animations!



Simplest Trick: Next is Best!

(and other simpler tricks)

Jeet Shroff

Lead Programmer



AVALANCHE STUDIOS



Befriend your QA!

- You know your AI behaviors best
- Design documents only go so far
- Empower them
 - let them know about your gameplay loops + AI behaviors + debugging visualization
- Help them help you! - they will play the game more than you will get a chance to



Result:
BETTER GAME 😊

ACT IT OUT!

- Talking to animators/artists sometimes can be challenging
- Don't get caught up in the terminology
- Use reference
- Actions speak LOUDER than words – so get up and act it out
- Exercise (we all know we need it)



Result:
BETTER GAME 😊

Next is Best!

- Problem: Selecting an animation from a pool
- First instinct for selection from a set is to use random heuristic or some variation
- “Next-is-Best” heuristic often gives the player a much more rewarding experience
- Simply have a reference index and update
- Add On-Screen realization



Result:
BETTER GAME 😊



Befriend your QA!

Act it Out!

Just NEXT it!