

AI Behavior Editing and Debugging in Tom Clancy's The Division

Jonas Gillberg
Technical Lead AI Programmer
Massive Entertainment - A Ubisoft Studio



This presentation

- More context
 - Archetypes
 - Architecture
- Evolution
 - Intentional workflow change
 - New workflow almost reverted
- Examples
 - Debug Intro
 - BT Intro
 - BT Edit
 - Real Debug



Combat NPC Overview

- 36 unique combat AI split into 11 archetypes
- 5 enemy factions and 1 friendly faction
- Detection system, threat system, scripted control, reaction to player Abilities, dynamic objective-handling, reactions to civilians and wildlife, idle activities
- Used in both systemic open world and linear, scripted story missions
- Behavior Tree architecture

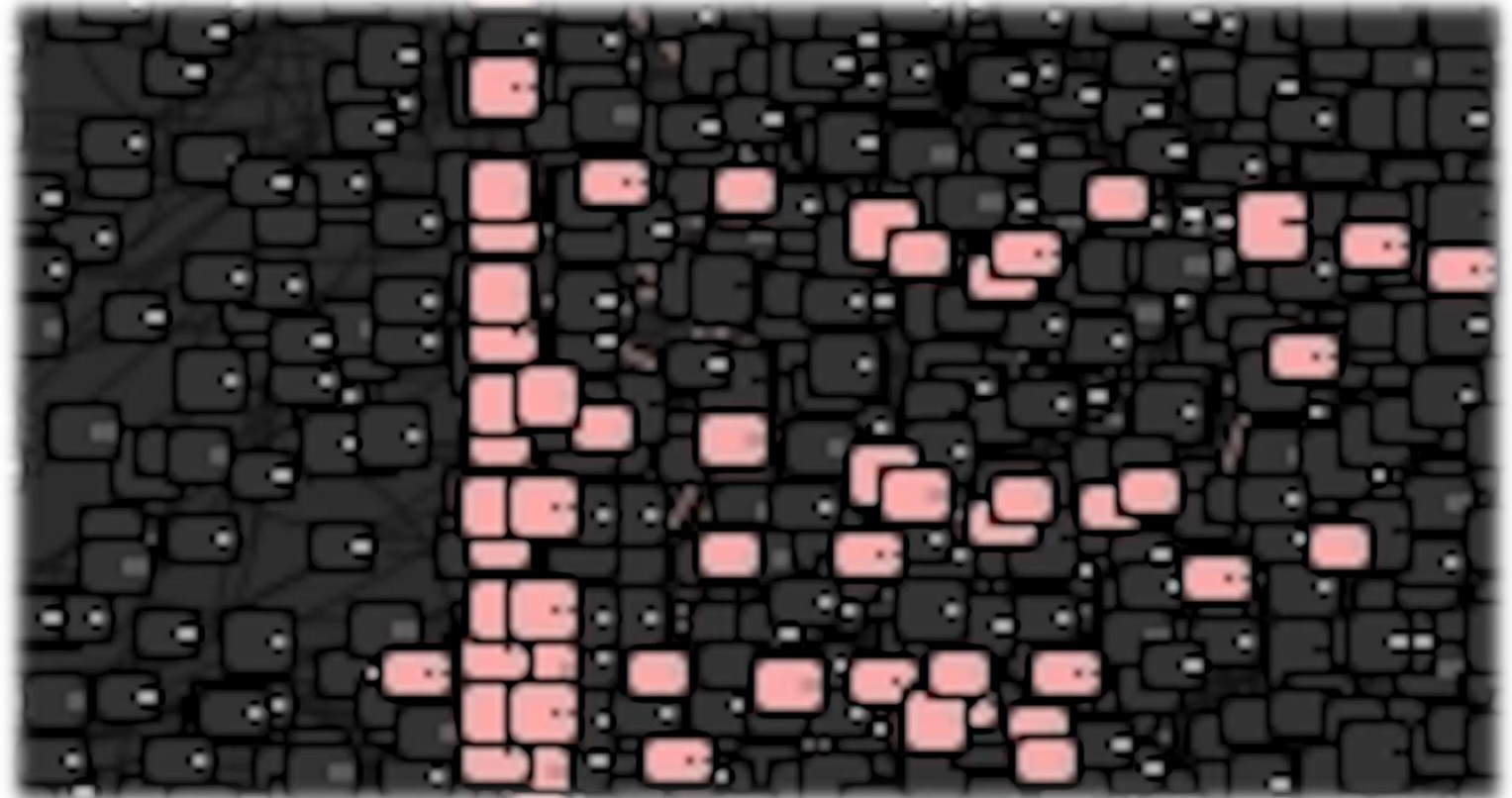


POWERED BY *snow drop*™



Architecture

- Behavior Tree architecture
 - Good info out there
- Server side
 - Different constraints - for another talk
- Right to Left
 - There are reasons
 - Do get used to it
 - Not good enough
 - First impression
 - When to make the change
 - Ask me next time





Evolution





In the beginning

- AI is for programmers
 - Creates, links and manages state
- Designers design
 - Might tweak inputs



Enter Visual Scripting

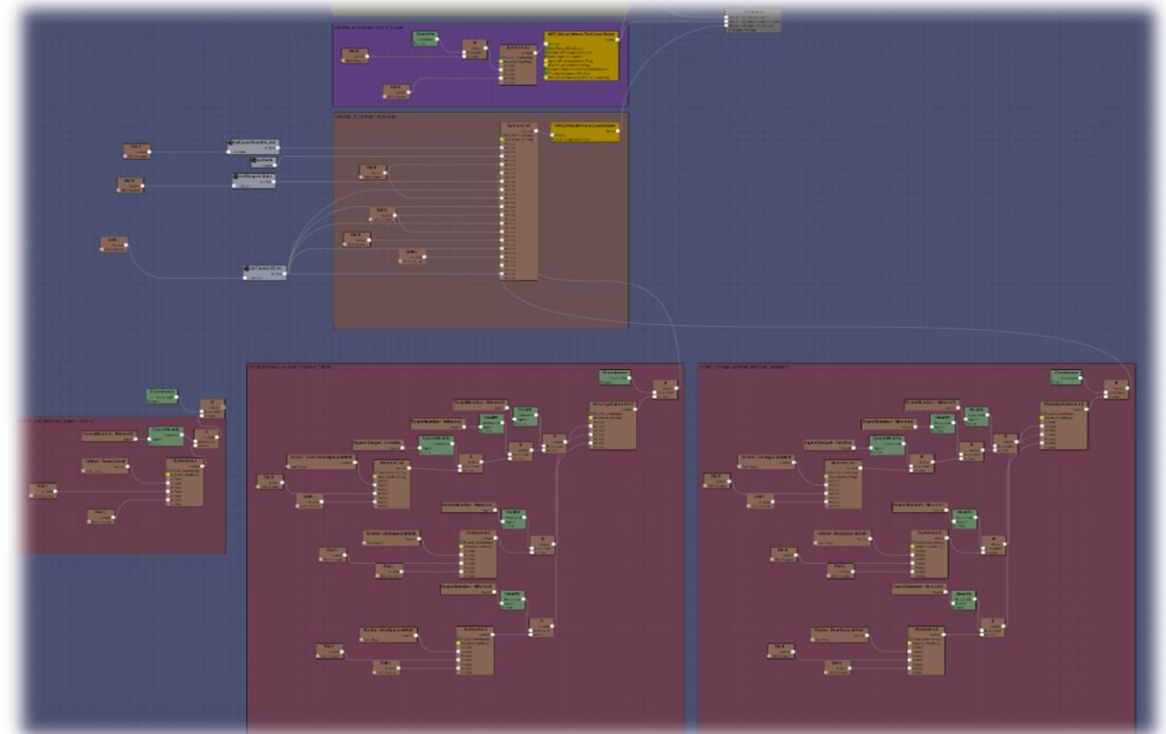
- Node graphs for art
- Empower content creators
- Without programmers
- Empower designers
 - Free up code time



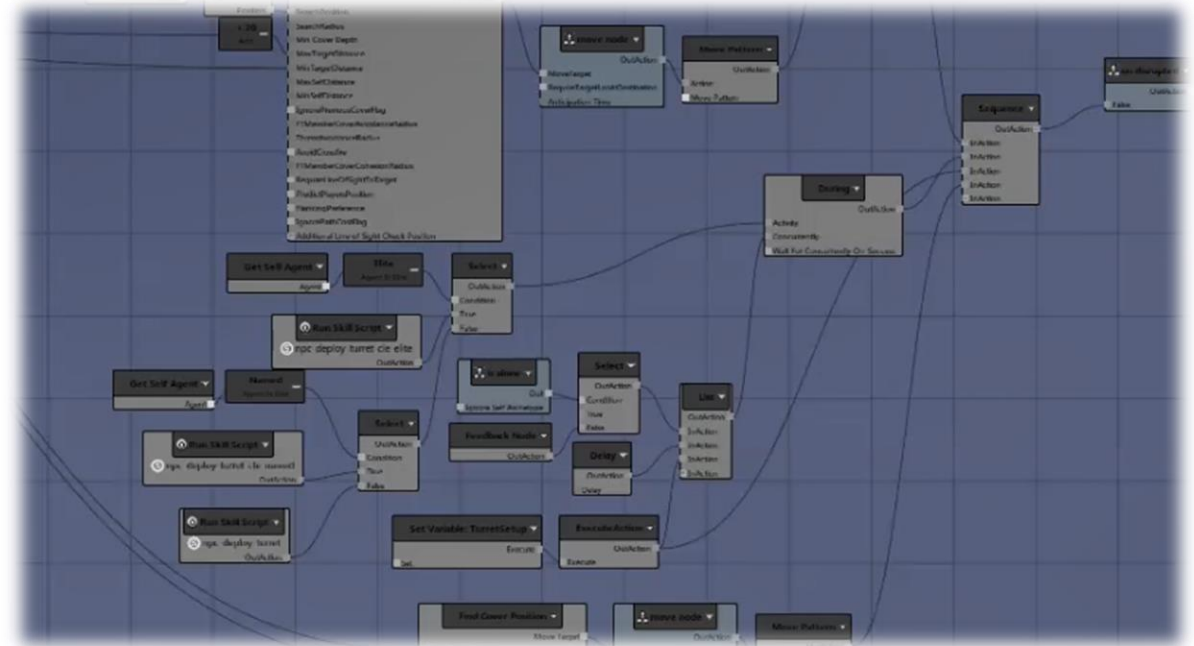


Early days

- Edit behaviors as node graphs
- Left to designers
 - Asking for nuts and bolts
- Visual edit not enough
 - No shared code - Reimplement
 - No reusable subgraphs - Copy pasta
- Restart on edit
 - Not fun with complex graphs
- Increasingly hard to manage



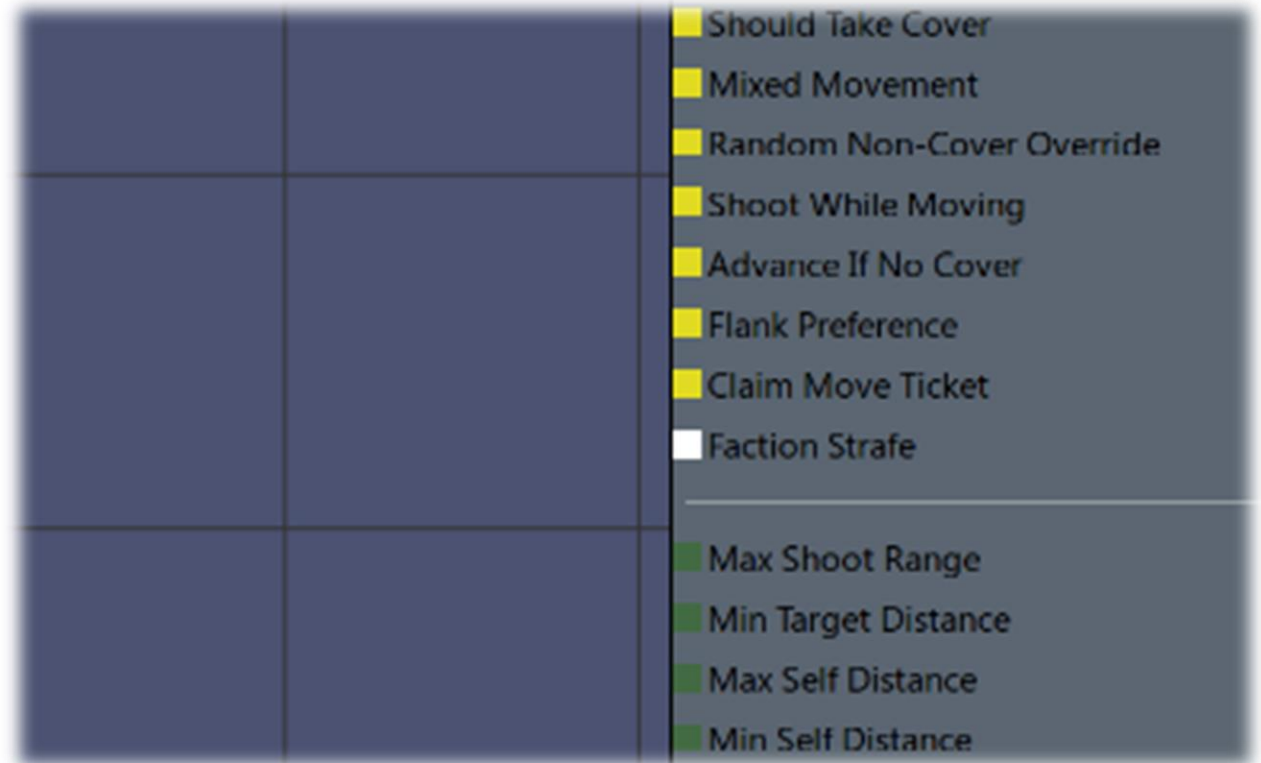
- Proper node graph
- Use common nodes
 - Less redundant work
- Reusable subgraphs - Compounds
 - Less copy paste
- Reloading
 - Faster trial & error





Still not enough

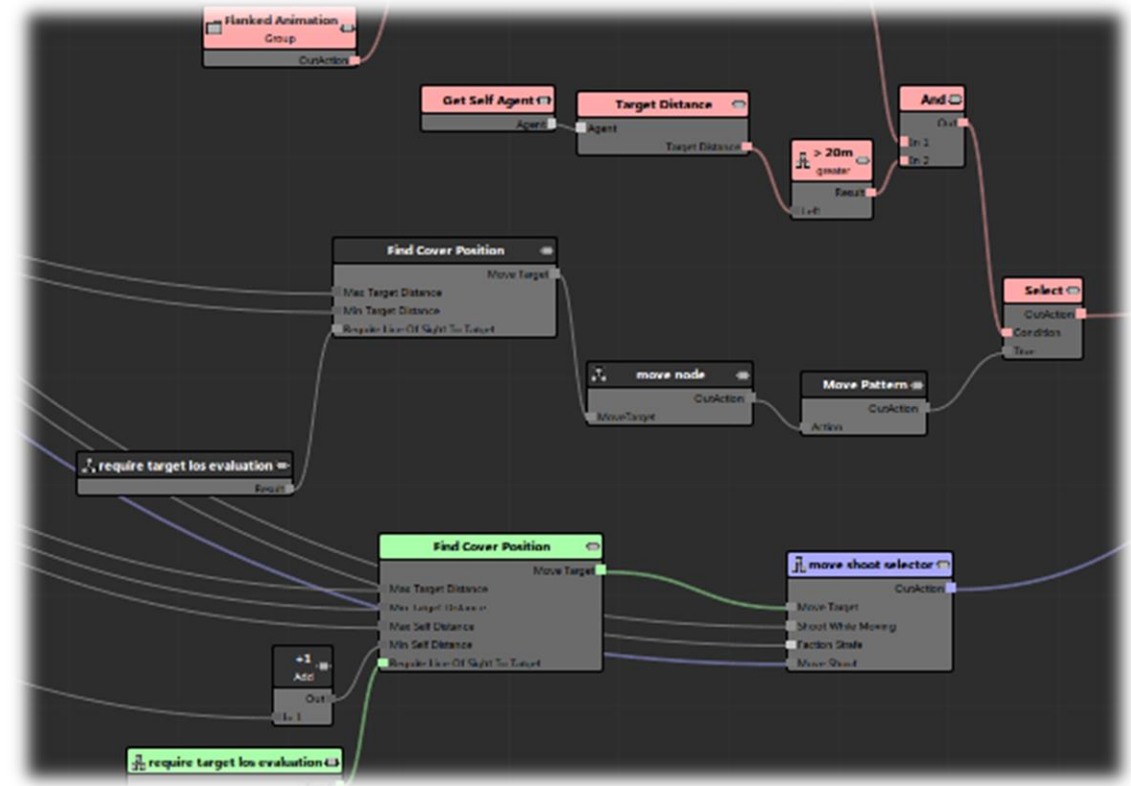
- Rewrite a bit late
- BT Graphs unreadable mess for programmers
- Despite best efforts - square one
- Nodes are for programmers
- Designers tweak inputs
 - Just slightly more of them
- Managed to work through it





A visual behavior debugger

- Programmers need debuggers
- Own workflow gets a bit more love
- Success - Green
- Failure - Red
- In Progress - Blue
- Inactive - Gray
- Interrupted - Orange





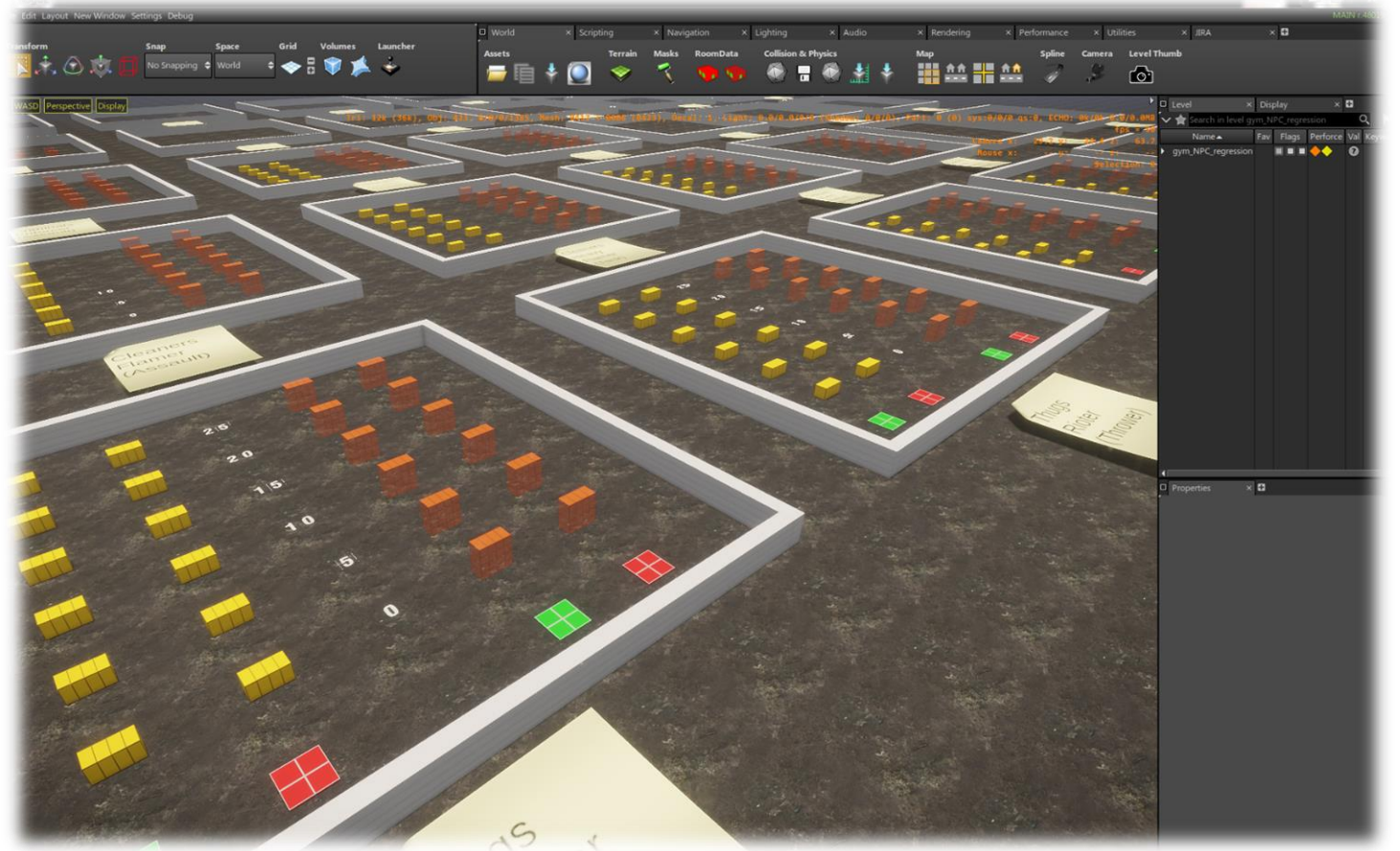
Examples





Debug Intro

- Regression test map
- Area for archetypes
- Easy bug repro
- Rioter Sniper





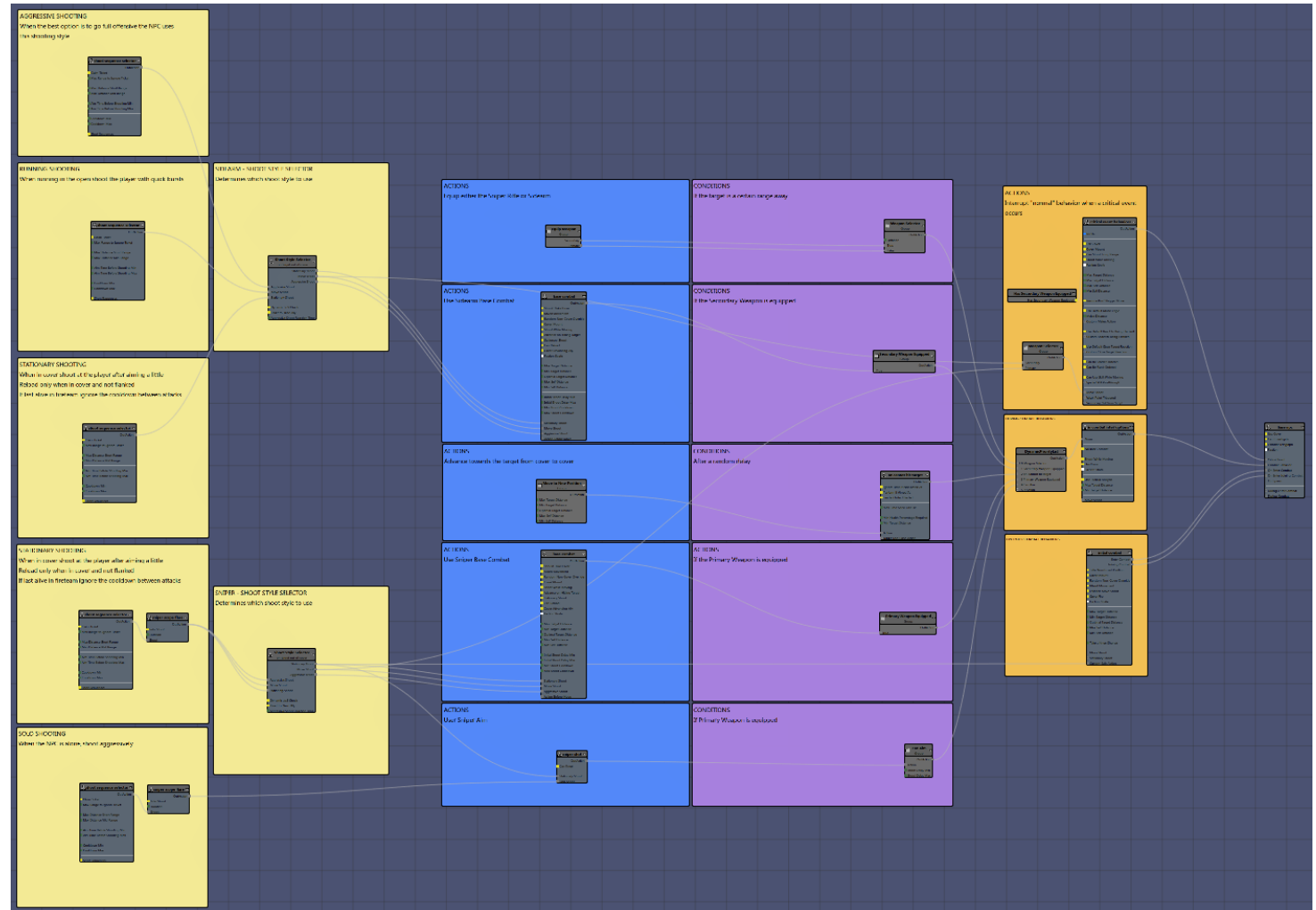
☐ Contextual Properties



☐ Contextual Properties

Behavior Tree System

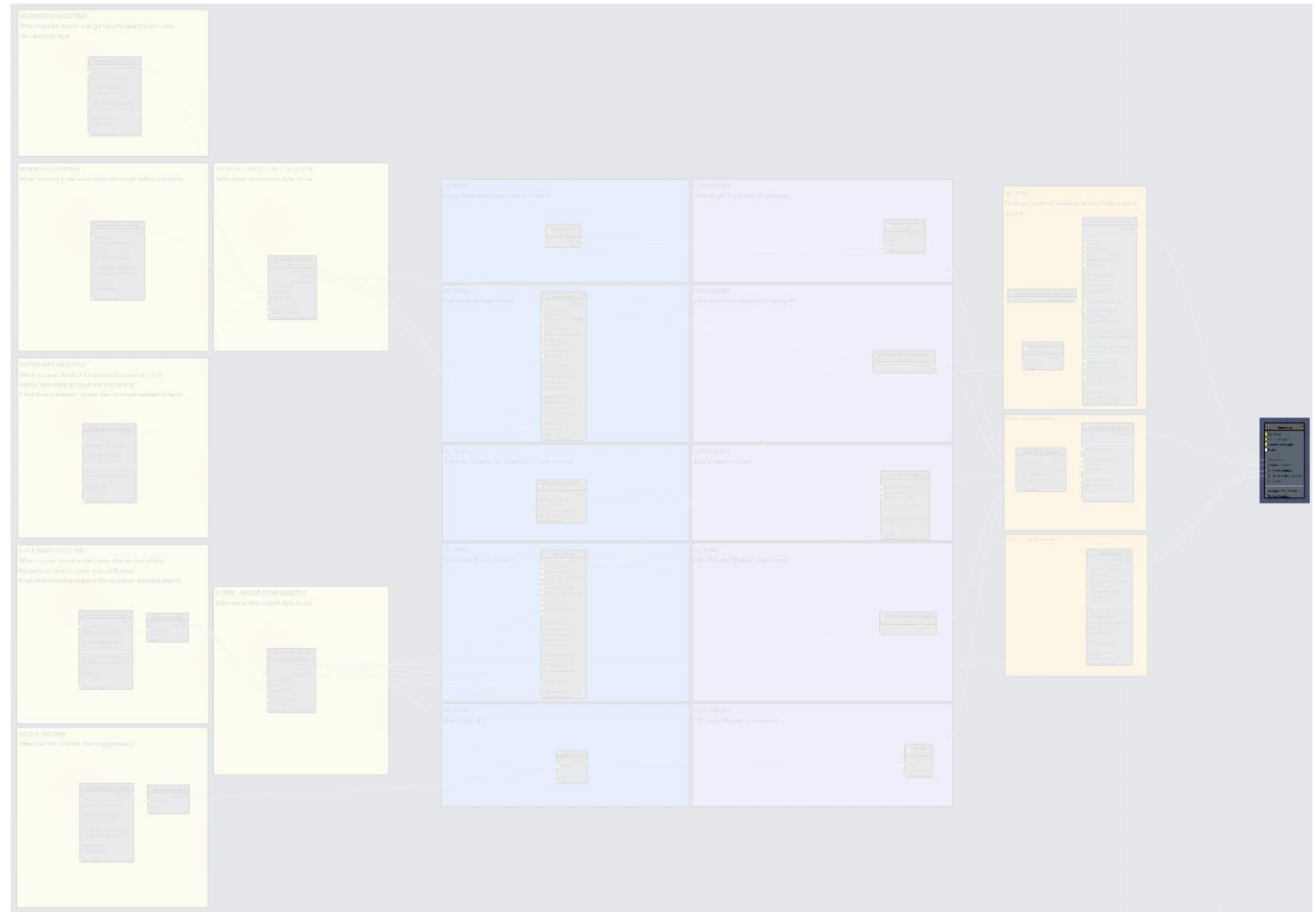
- Rioter Sniper
- Right to Left
 - Top to Bottom
- Color coded
 - By design
- Layer connections
 - Multiple parents
 - Slimmer bottom
- Not the details
 - But clear intent





Base Npc

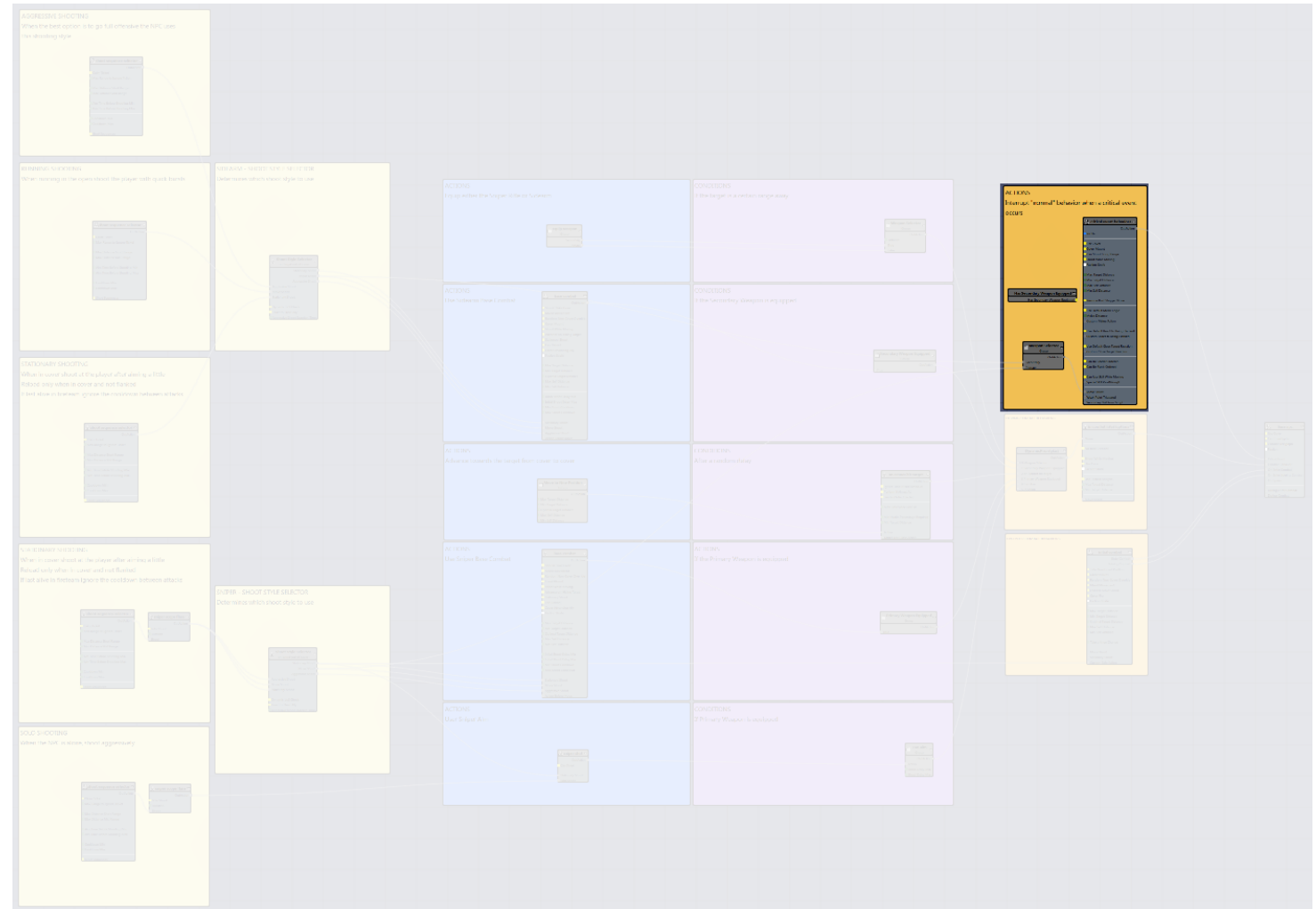
- Far right
- Used for all human combat NPCs





Critical Events

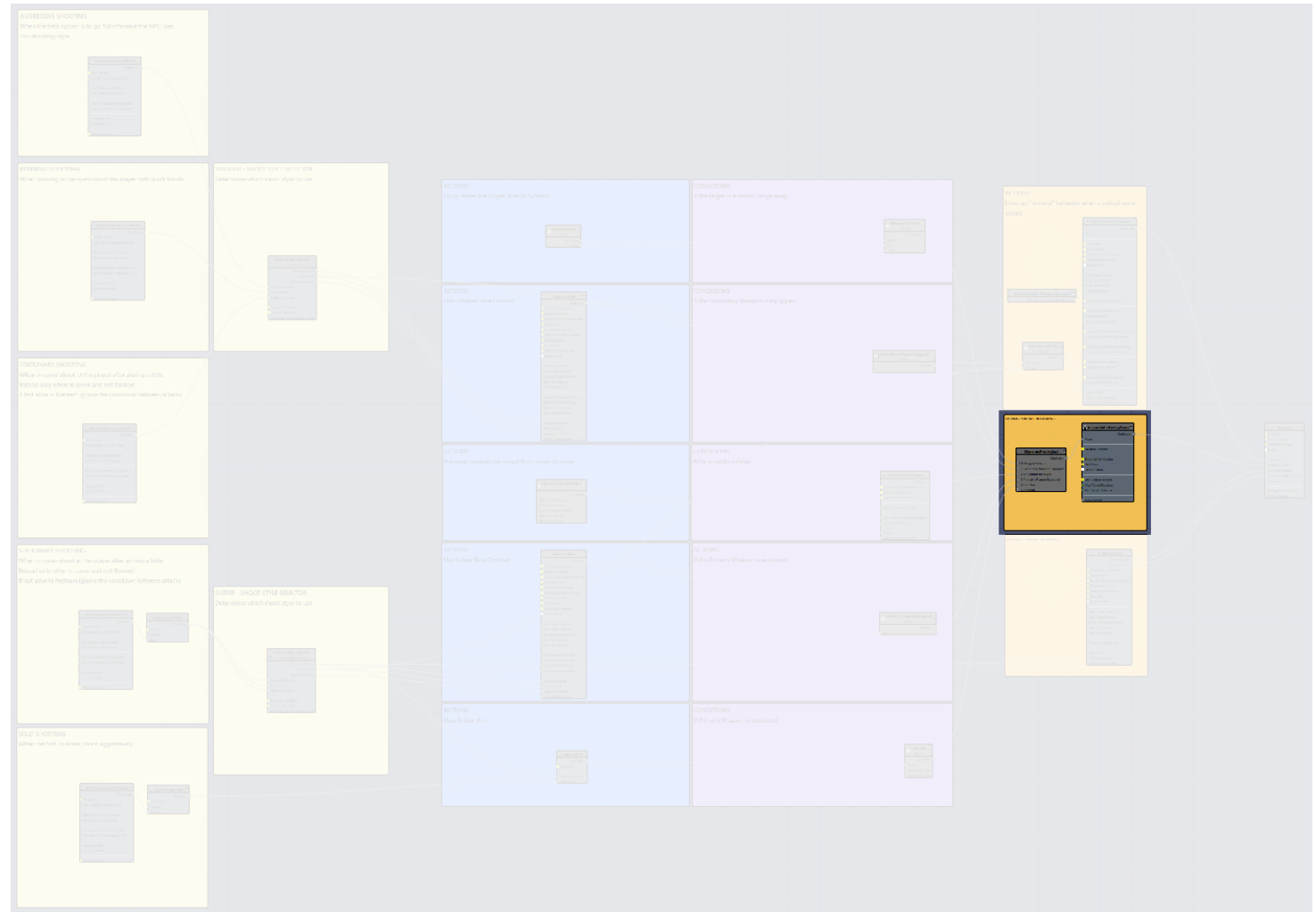
- Top input for Base Npc
- Higher priority than normal combat
- From Direct orders to Seeker Mines and Team Mate reactions





Normal Combat

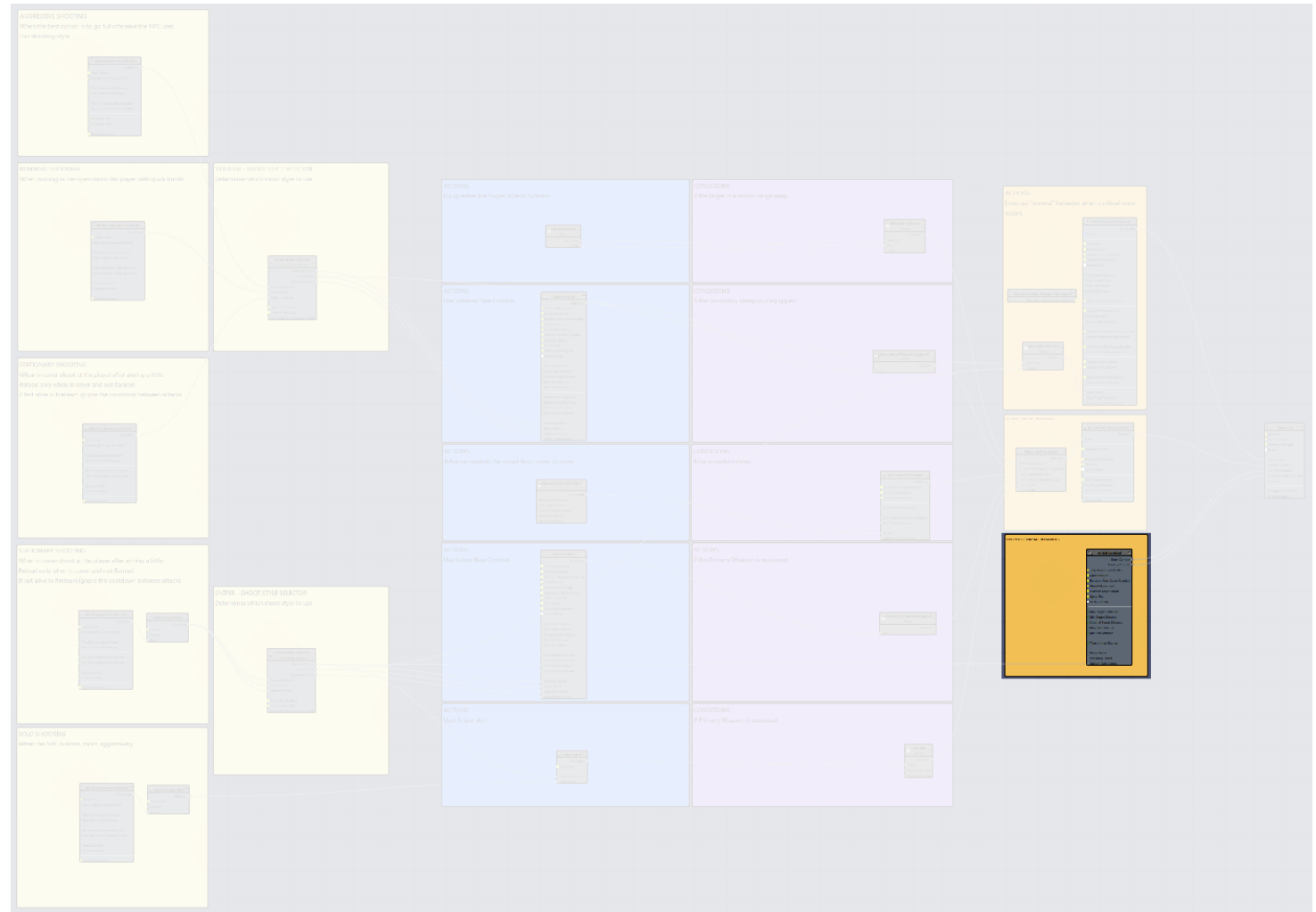
- Connects to the right conditions





Initial Combat

- Often special
- Logically executes before, but slightly less important and the bottom





Conditions

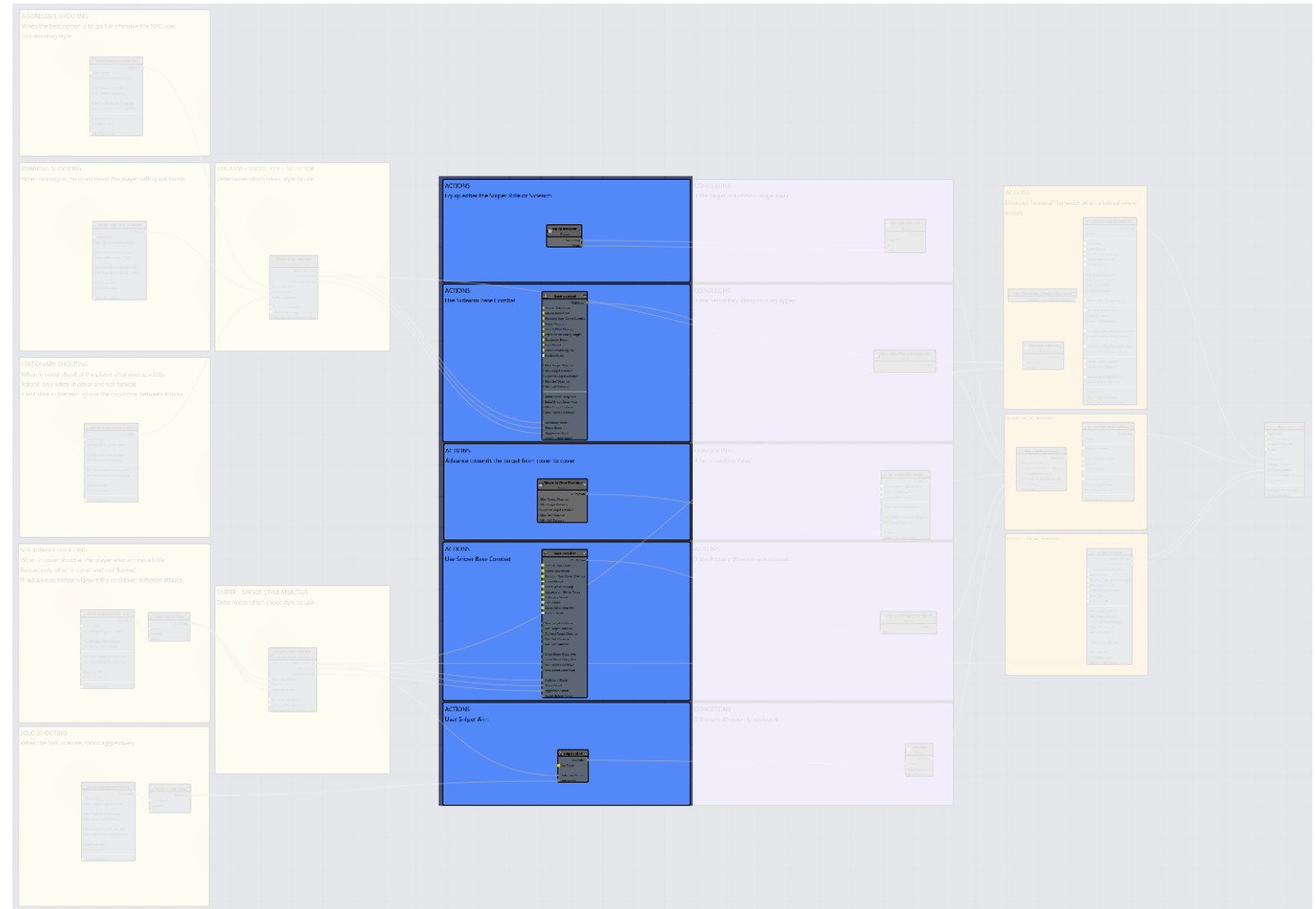
- Picks from actions (next layer)
- Allows reuse instead of reimplementation in Critical Events/Normal Combat





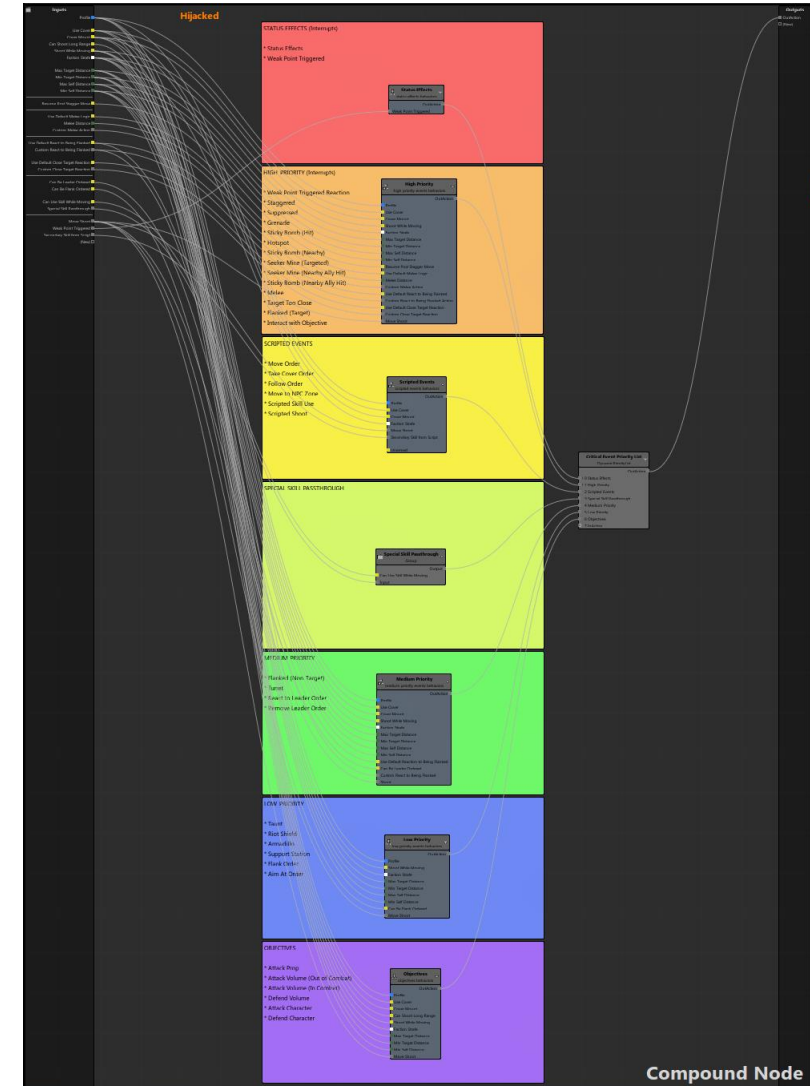
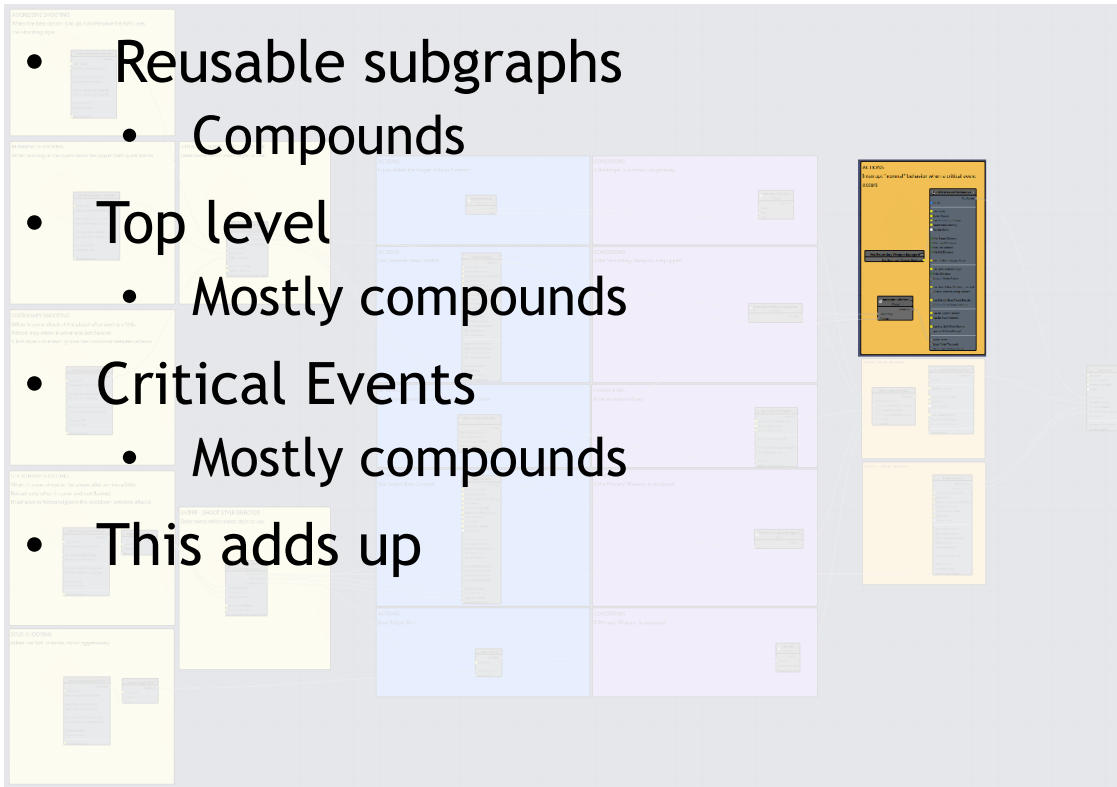
Actions

- What they do
- Also link to shooting



Graphs in graphs

- Reusable subgraphs
 - Compounds
- Top level
 - Mostly compounds
- Critical Events
 - Mostly compounds
- This adds up



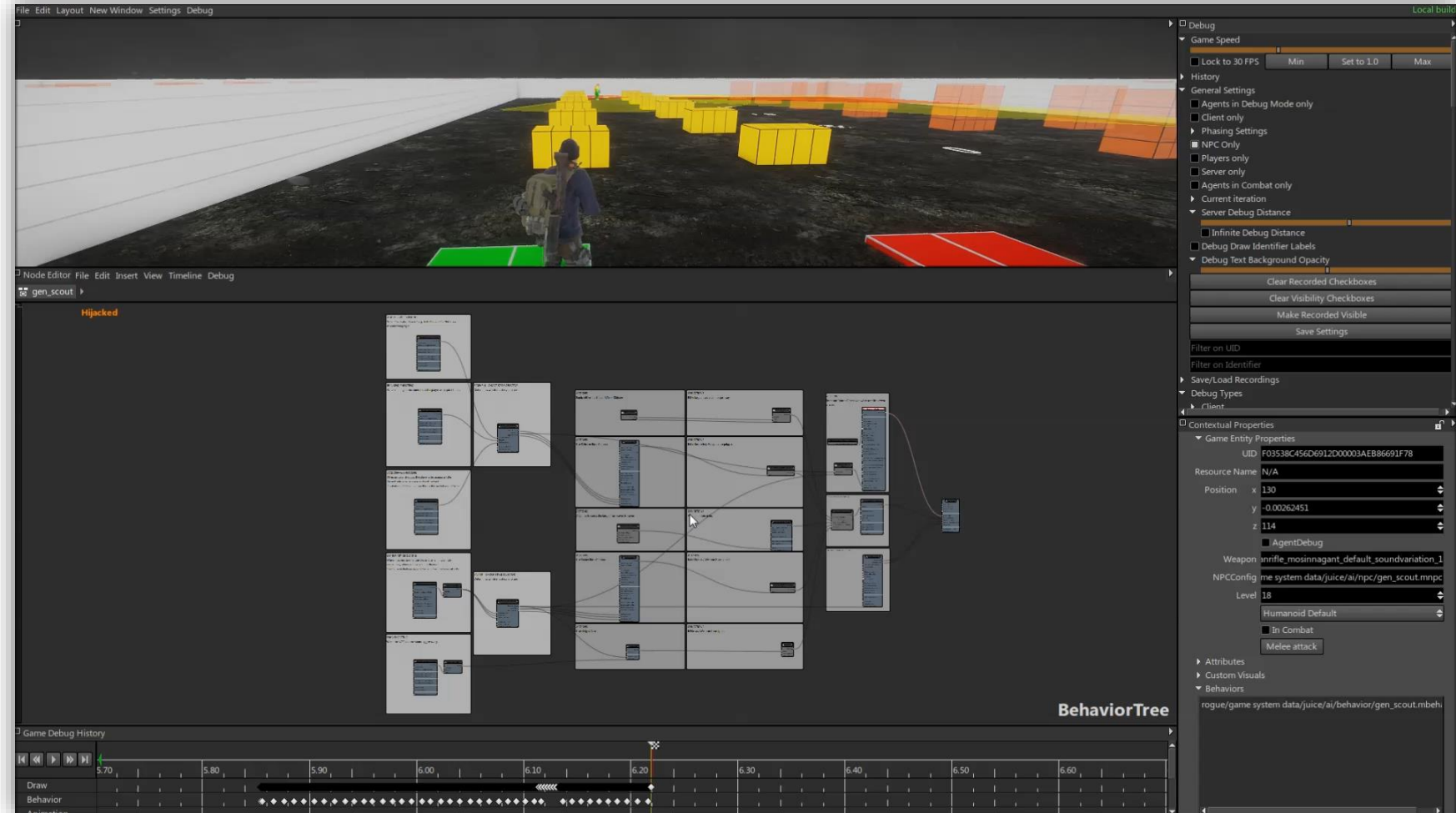






Rapid iterations

- More iterations
 - 360 spider sense
- Quick/Easy edit
- Easy to debug
- Regain confidence
 - One edit at a time
- Not enough





Not enough

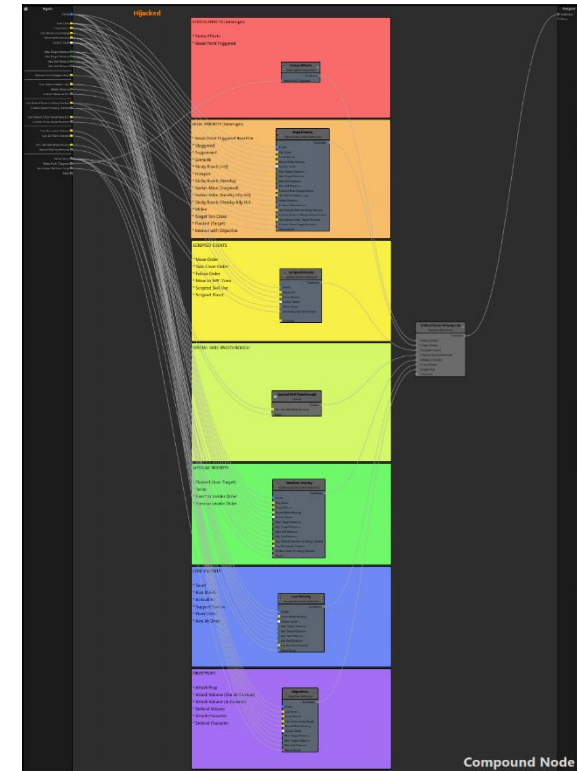
- Break old habits
- Improve communication
 - Seating
 - Working closely together
- Iterate on responsibilities/layout
- Flexible graphs
 - Adapt abstraction level to the team
- Now who does what?





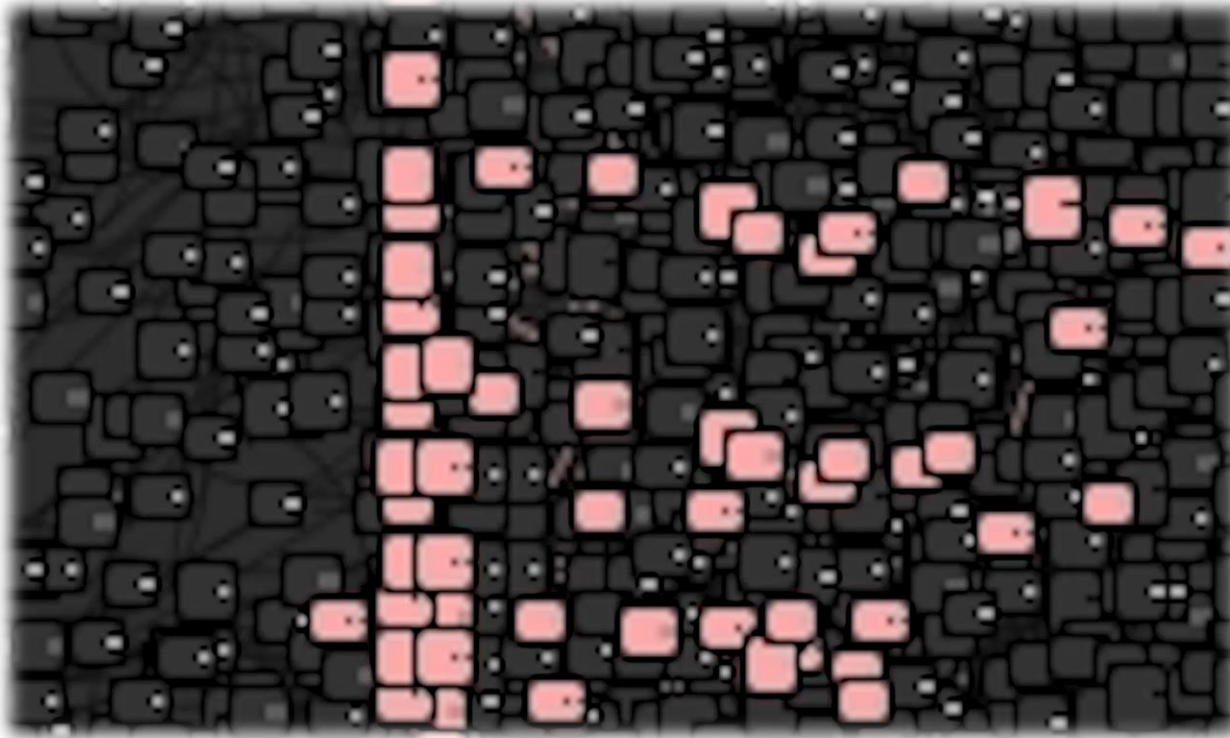
Now who does what?

- Powerusers wrapped in compounds
- What nodes to make?
- High level of detail



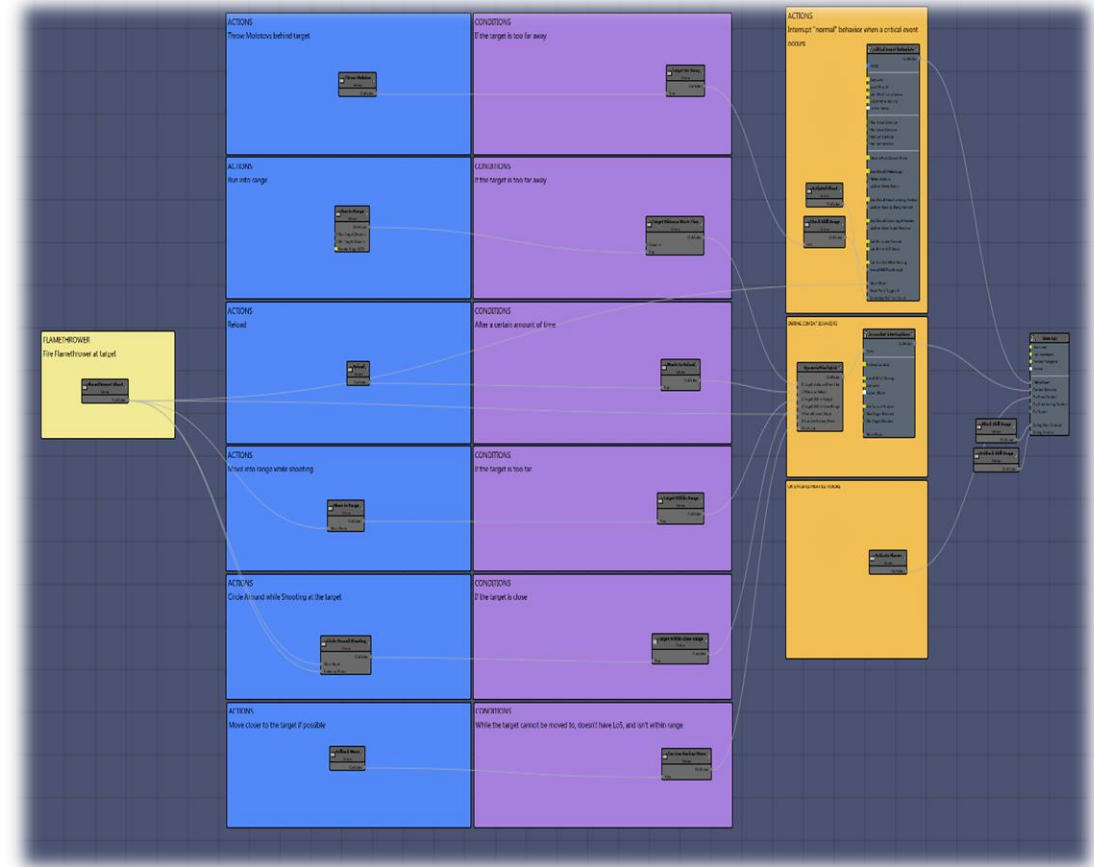
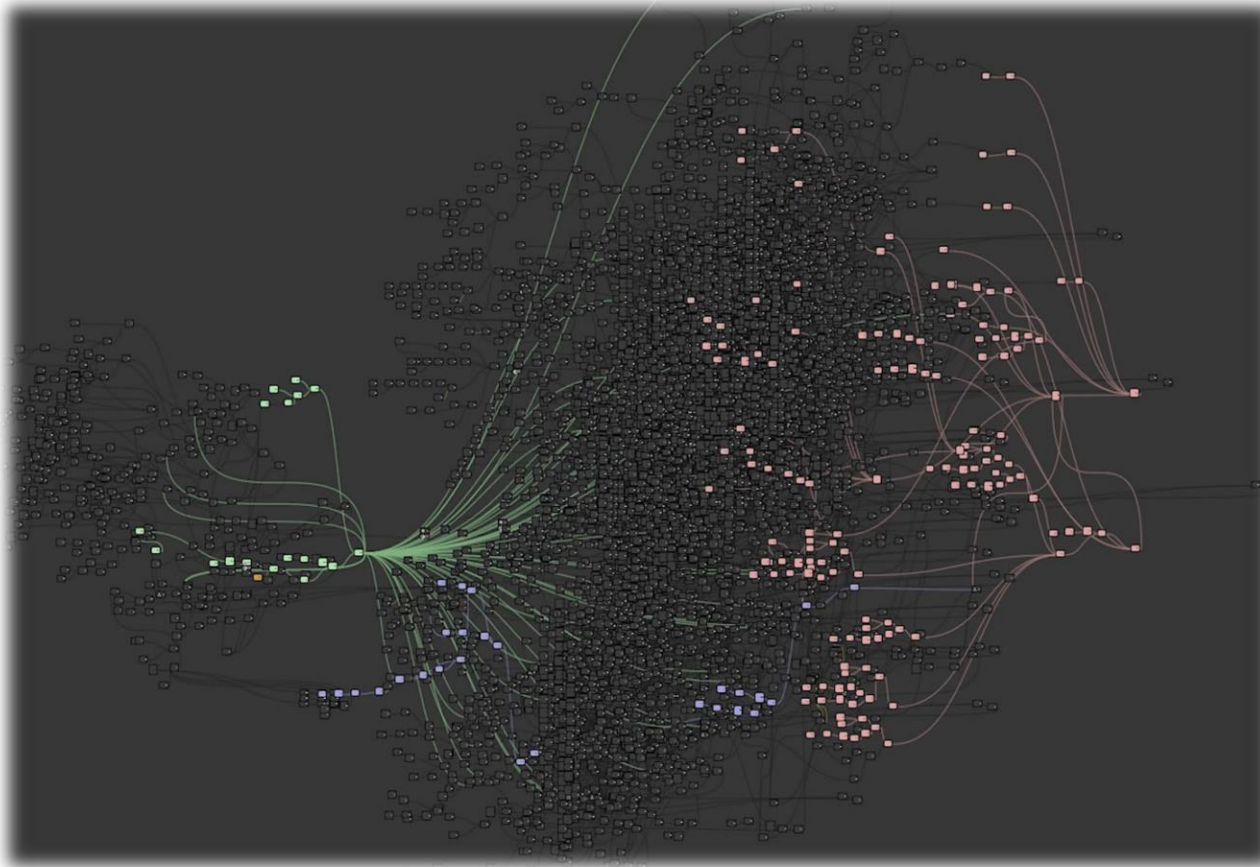


Remember this?



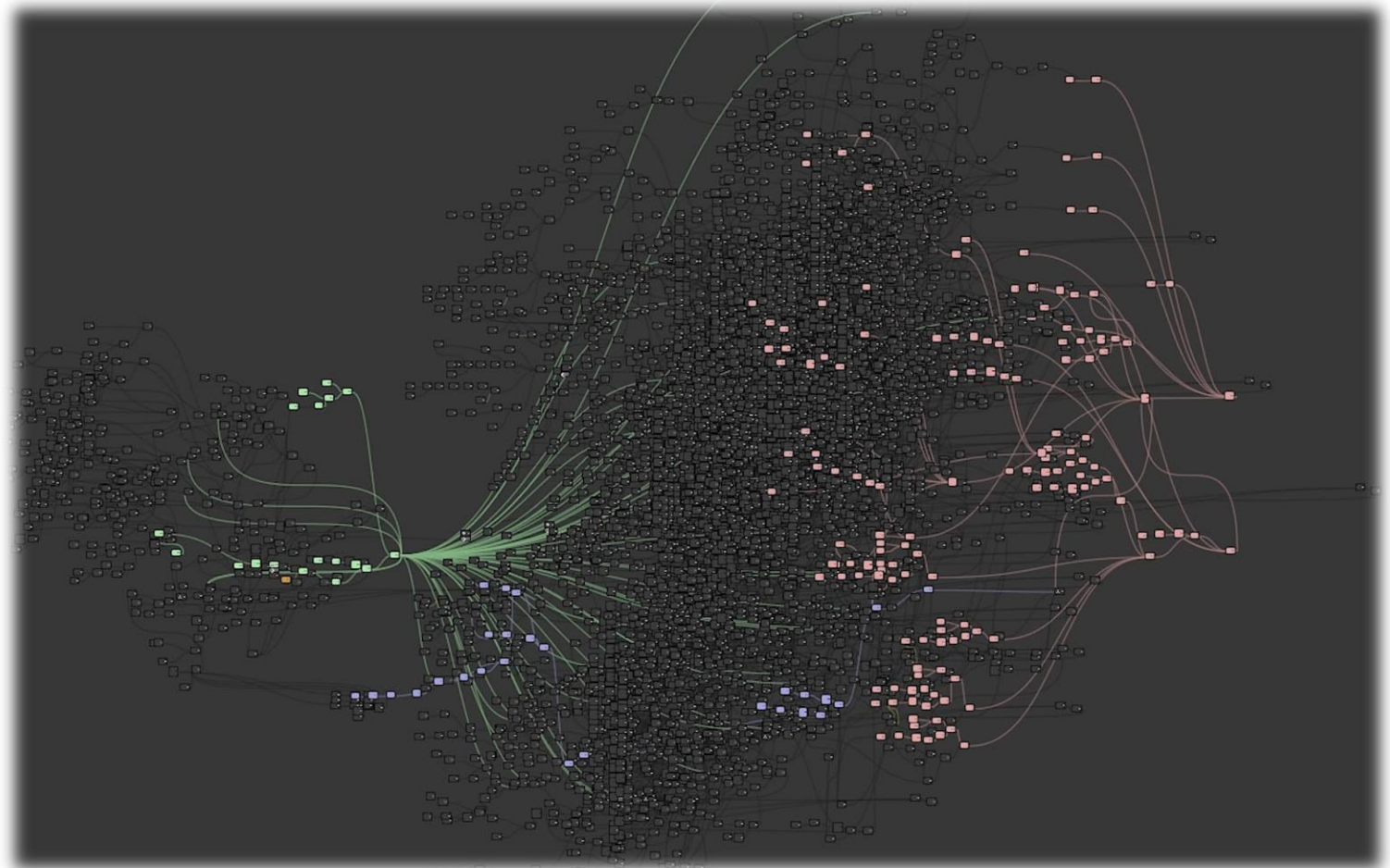


Not quite true



▼
With great power...comes reviews...

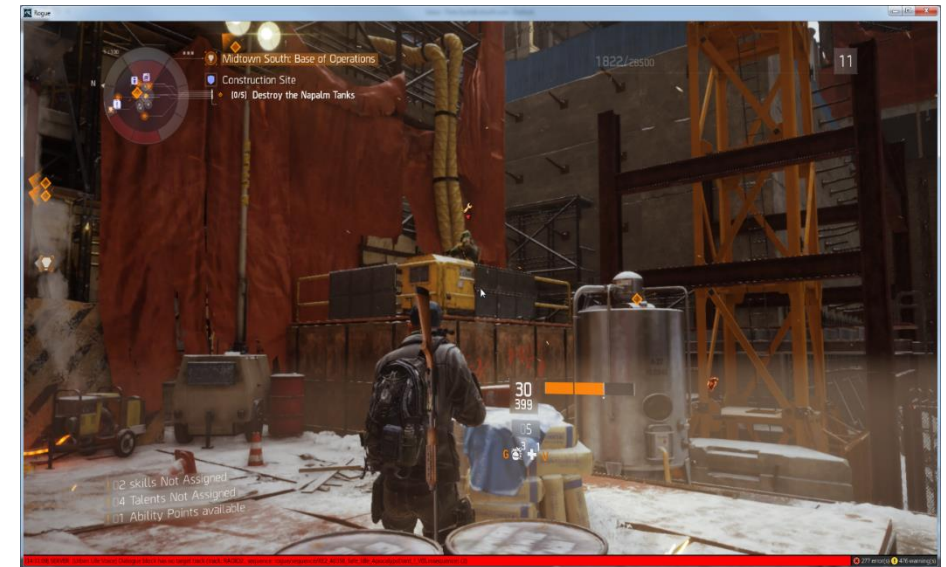
- Spaghetti
- Impossible like that
- Node Reviews
 - Code - Design
 - Design - Code
- Input sanitation
 - Easy edit
 - Easy error
- Real Example



▼ Cleaner Mechanic doesn't deploy the turret

We've created a little vantage point for it in Hope it would provide a focused turret gameplay section to the beat but sadly the NPC never places a turret.

- Archetype that spawns a Mech
- How to set up for real debugging
- Combine multiple debug sources







Game Debug History

	6.40	6.50	6.60	6.70	6.80	6.90	7.00	7.10	7.20	7.30
Draw										
Behavior										
Animation										
Script										

Debug

Game Speed

☐ Lock to 30 FPS

Min

Set to 1.0

Max

History

General Settings

Save/Load Recordings

Debug Types

Debug Log

Open log window

Contextual Properties

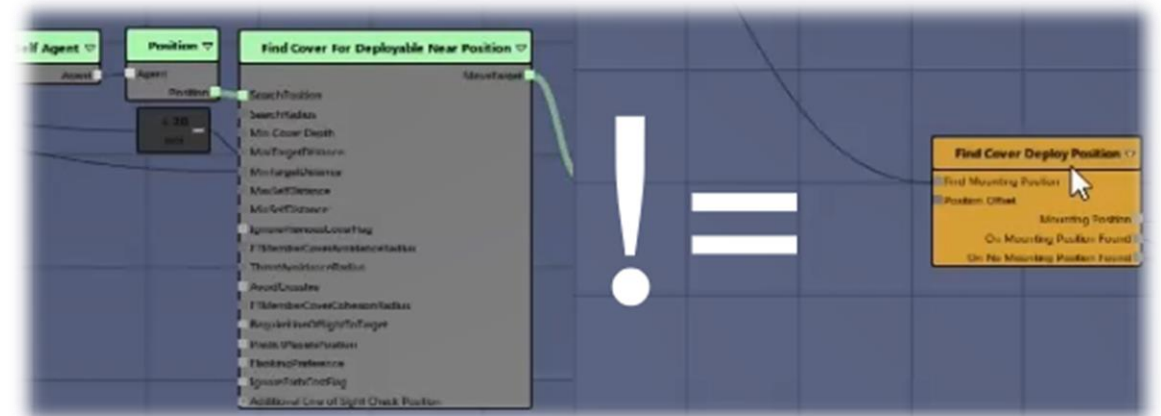


Turret Bug

- First issue
- Cover Depth Calculation
 - Includes the the napalm tank
- Why such a big issue?



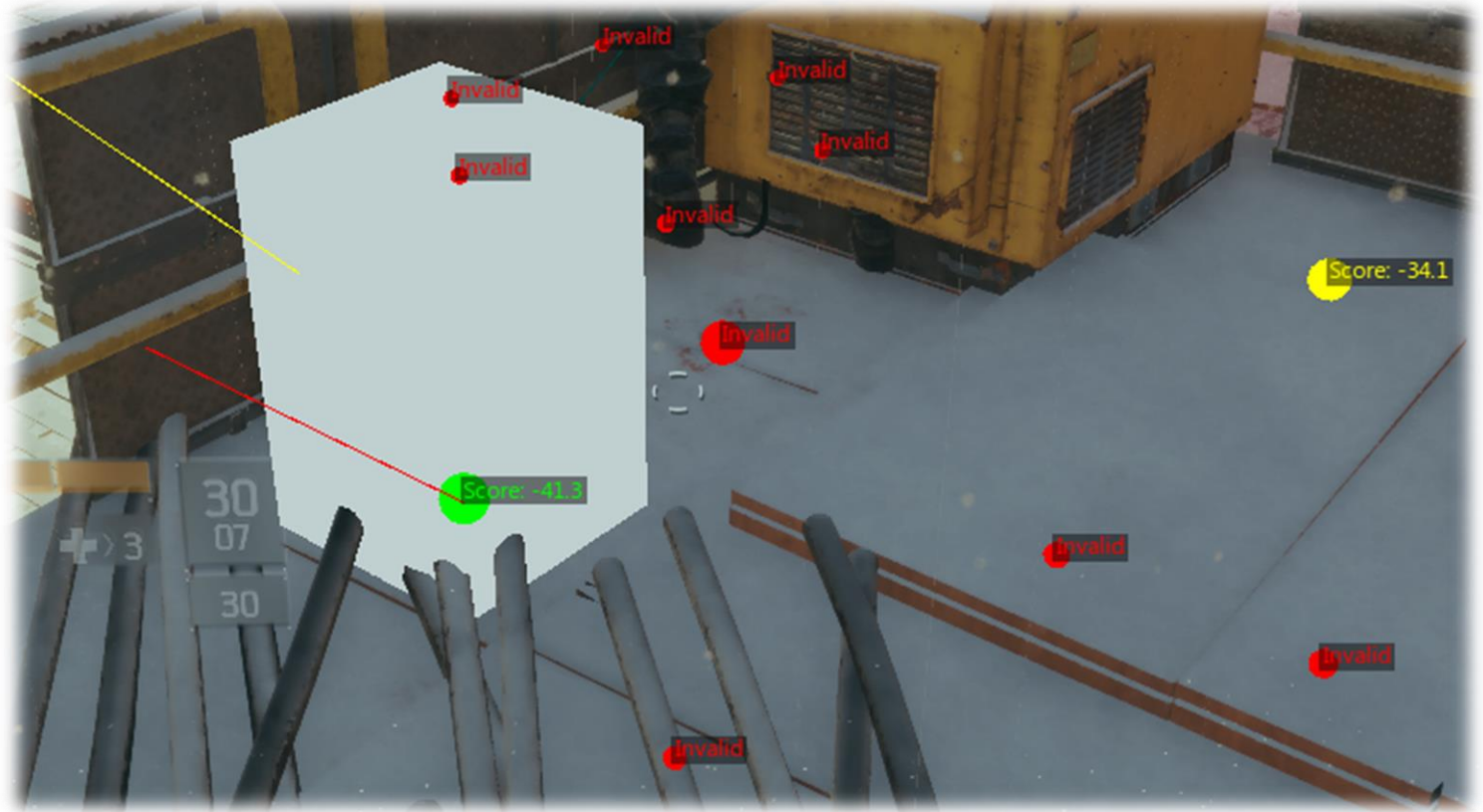
- Second issue
- Two different tests
 - At different times
 - Assumed match
 - When it works, yes





Turret Bug

- What does this mean?
- Hides the real position
- Behavior picks the wrong one - and fails.
- Pragmatic solution
- Future iterations





Conclusion

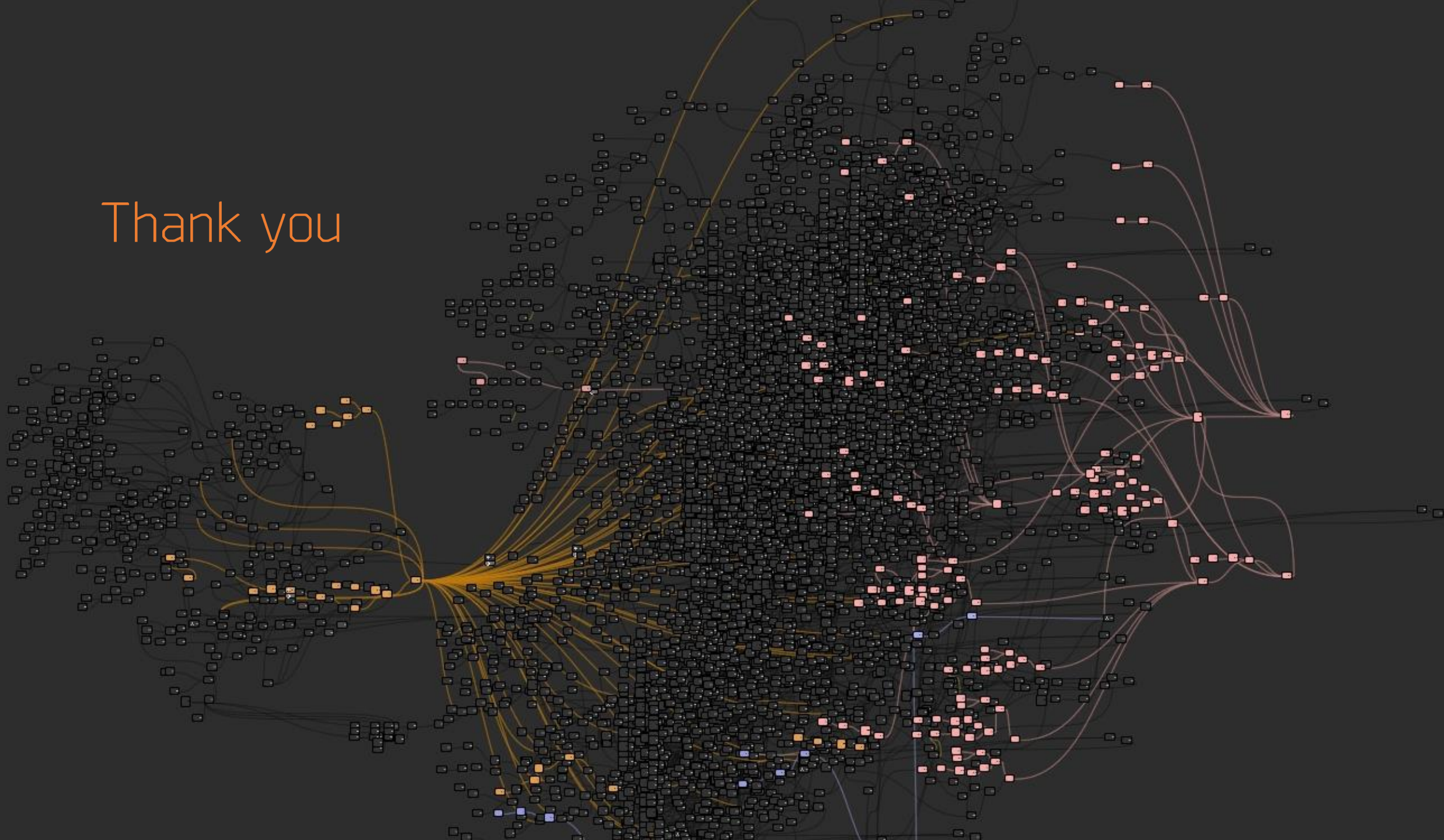


Own it

Iterate

Teamwork

Thank you



Friday March 18, 2016
1:30pm – 2:30pm
Room 2016, West Hall

Global Illumination in Tom Clancy's The Division

Nikolay Stefanov
Technical Lead, Ubisoft Massive



MORE QUESTIONS?

MEET ME ON THE UBISOFT LOUNGE

on

WEDNESDAY

from

11:00AM

to

12.30PM

WEST HALL, 2ND FLOOR

@UBISOFTCAREERS #UBIGDC