Game AI

COMP313 Computer Game Development

Simon McCallum

AI

- Two focuses
  - Understanding Intelligence **aI**
  - Replicating Intelligence behaviour **Ai**
- **Game AI**
  - Playing to lose
  - Creating an experience
  - Supporting the goals of the designer
  - Realtime
  - Controllable
  - Comprehensible
- Control
  - Challenges of Emergent AI
  - Limitations of Scripted AI

Middleware

- Using other systems to do the AI for you
- Advantages
  - Time
  - Cost
  - Technical support
- Disadvantages
  - Generic solutions
  - Problem solving techniques might not fit
  - Loss of control

Unity Store - AI tools **340**

Unreal Market AI **345**

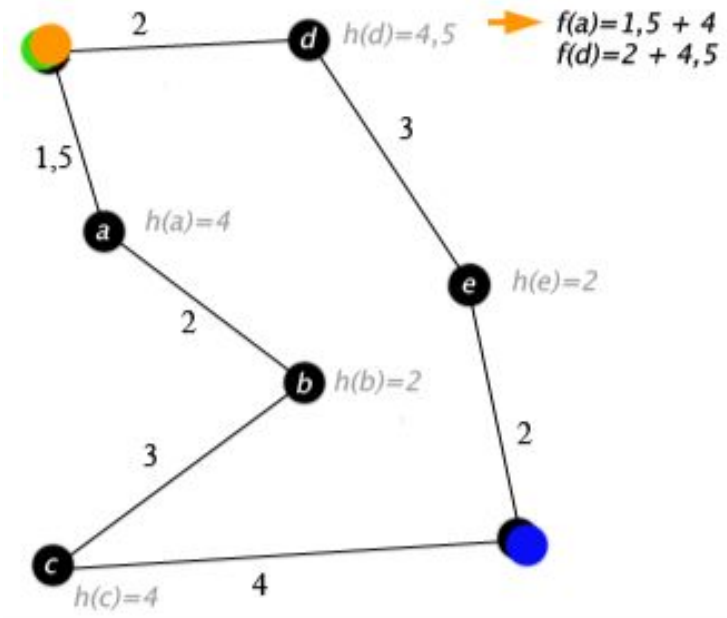
https://www.unrealengine.com/marketplace/en-US/assets?keywords=AI

Unreal Engine has build in Behavioural Trees as the AI system
https://docs.unrealengine.com/en-US/Engine/ArtificialIntelligence/index.html

Technologies

- Mostly simple
  - A*
  - FSM
  - Behavioural Trees
  - Scripts
  - Goal Oriented behaviour – Backward chaining
  - Simple multi-agent behaviour such as Boids
- More advanced
  - Subsumption Architecture
  - Physics based simulation
  - Neural Networks
  - Genetic Algorithms

A*

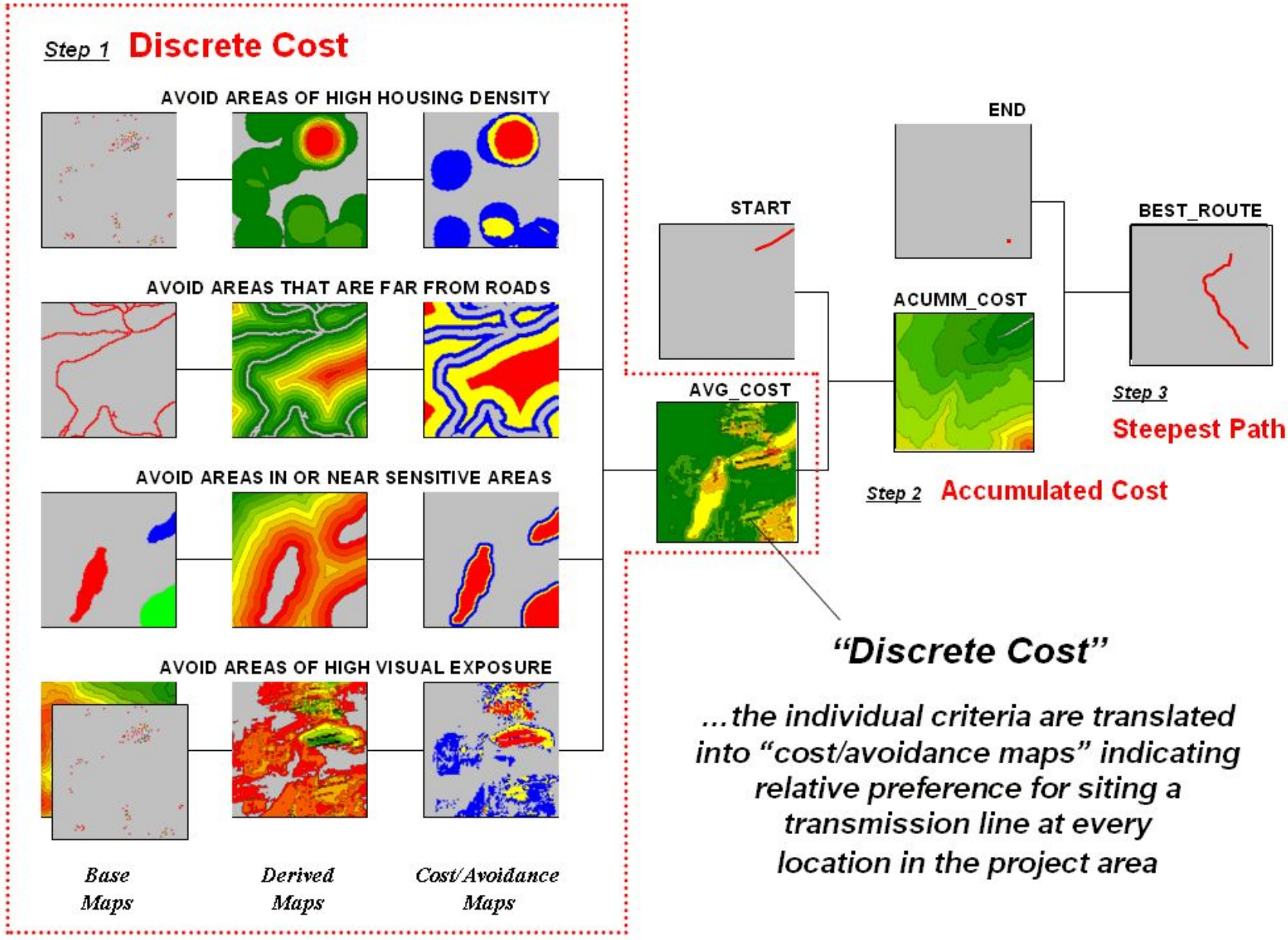


- Core search algorithm
- Used to find paths
  - Paths in space
  - Paths through decision nodes
  - Threat maps and motivation
- Frustration modelling
  - While blocked raise the cost on blocking area
- Problems
  - Unrealistic search
  - Slow
  - Does not deal well with deformations or alternative goal

A*

- Required
  - A system that has the concept of a path
  - Steps can be given a cost
  - Guess at the best path
- Halo 2
  - Defined actions as a graph
  - Define costs for selecting actions
  - Use A* to choose path through action graph

A*



"Discrete Cost"

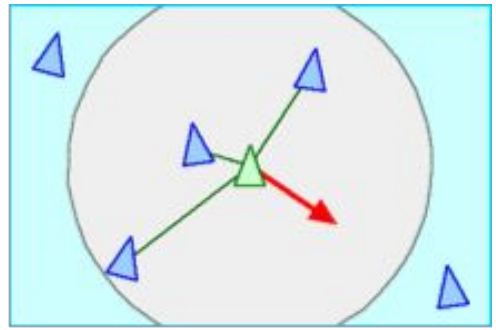
...the individual criteria are translated into "cost/avoidance maps" indicating relative preference for siting a transmission line at every location in the project area

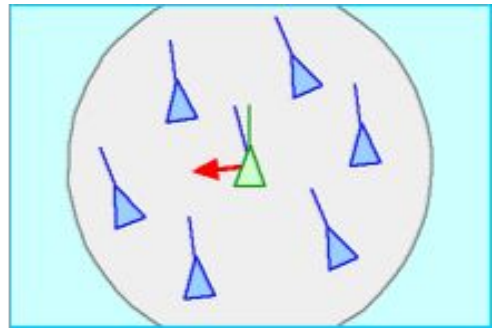
Scripts

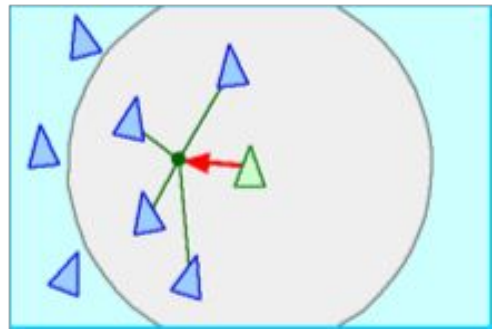
- AI technique
  - Scripted behaviours
  - The waiter script
    - Describe the sequence of behaviour
    - Transition points to new part of script
    - Fill in minor character details
- Scripting
  - Provide a language for designers
  - Functions – Player health, enemy health, closest enemy
  - Complexity of functions interesting

Emergent Behaviour

- Emergent behaviour
  - Simple rules, complex interactions
  - Boids
    - Flocking
    - Avoidance
- Designing for emergence
  - Very hard to plan the experience
  - Simplicity is very hard
  - Start with simple rules and explicit nerfing of combos
  - Playtest
  - Iterations







Behavioural Trees

A tree that is used to decide on behaviours.

- Internal nodes are **decisions**
- Leaf nodes are **actions**

Control node

- Selector - try children until a success
- Sequence - try to run all children until a failure
- Simple Parallel * (Unreal specific)

Compositions of trees to create complexity

- Like hierarchical finite state machines - but with tasks
- Halo, Bioshock, Spore

Behavioural trees in Unreal

Build in as the default AI system

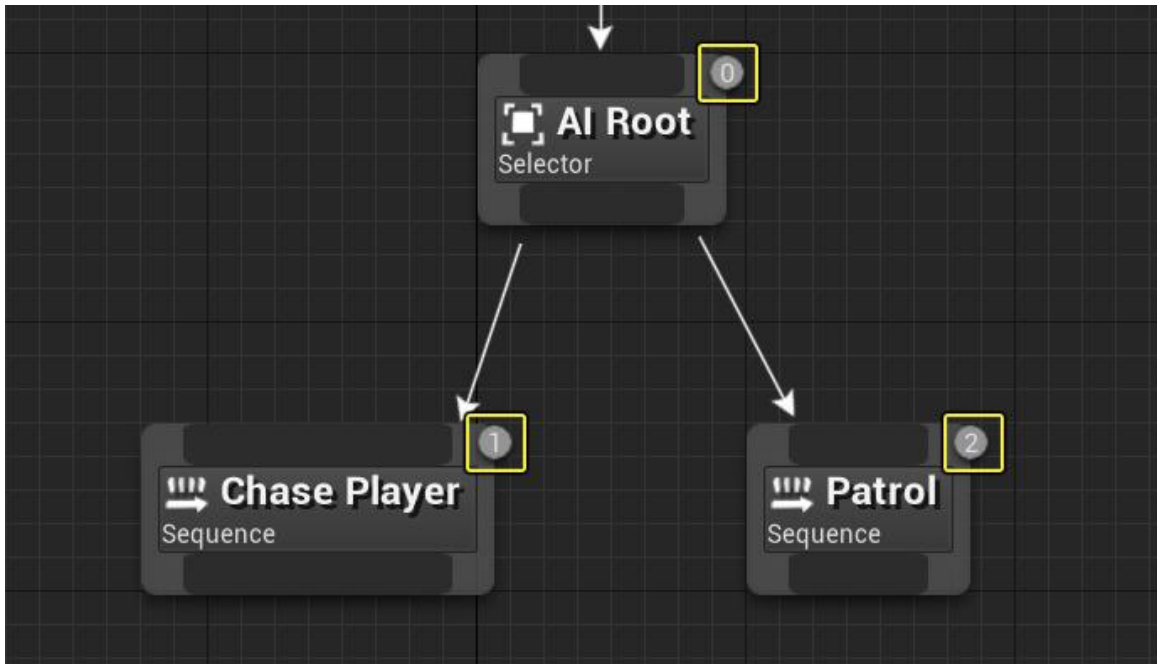
https://docs.unrealengine.com/en-US/Engine/ArtificialIntelligence/BehaviorTrees/index.html

Blackboard = memory of an agent

Visual creation in Unreal

Order L->R important

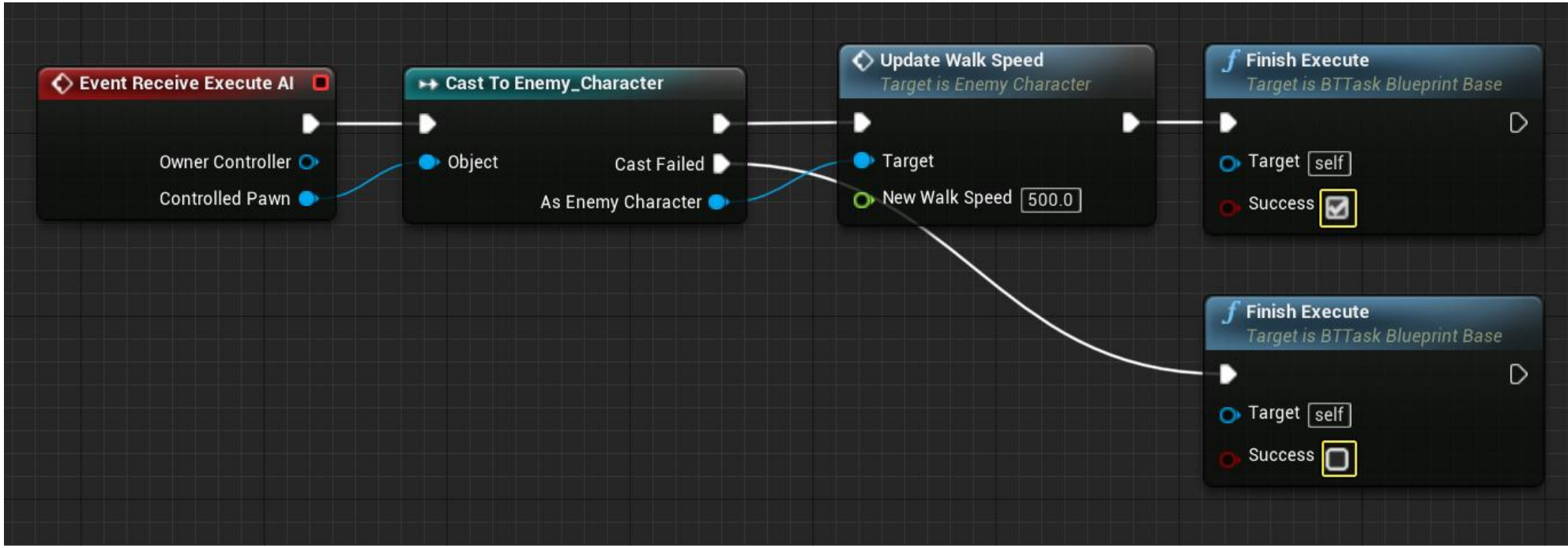
Tasks = Blueprints



Behaviour Tree Tasks

Unreal driven by Events - Execute AI

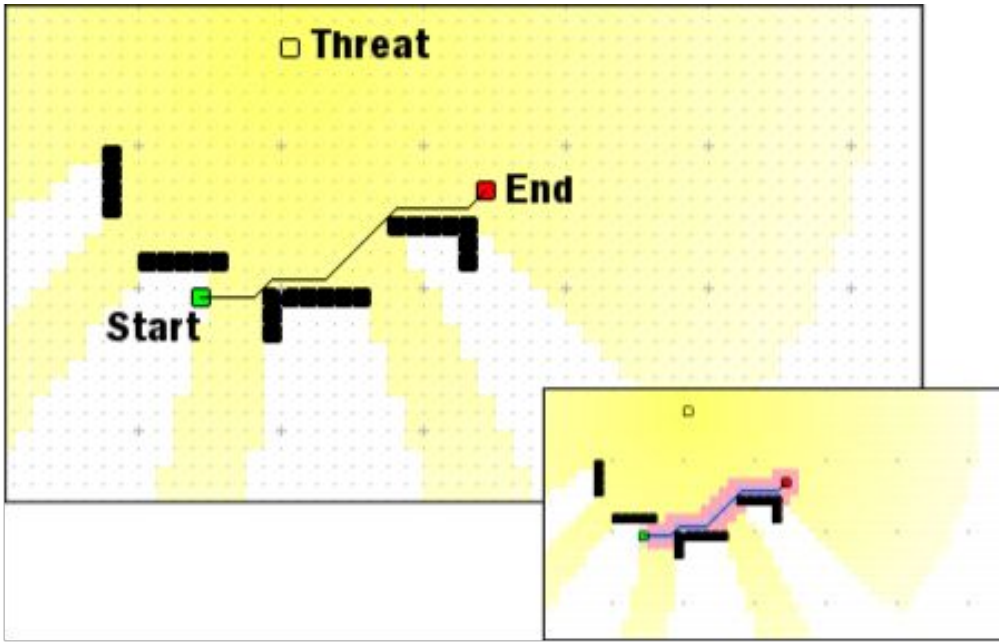
Must have finish Execute for the tree to work

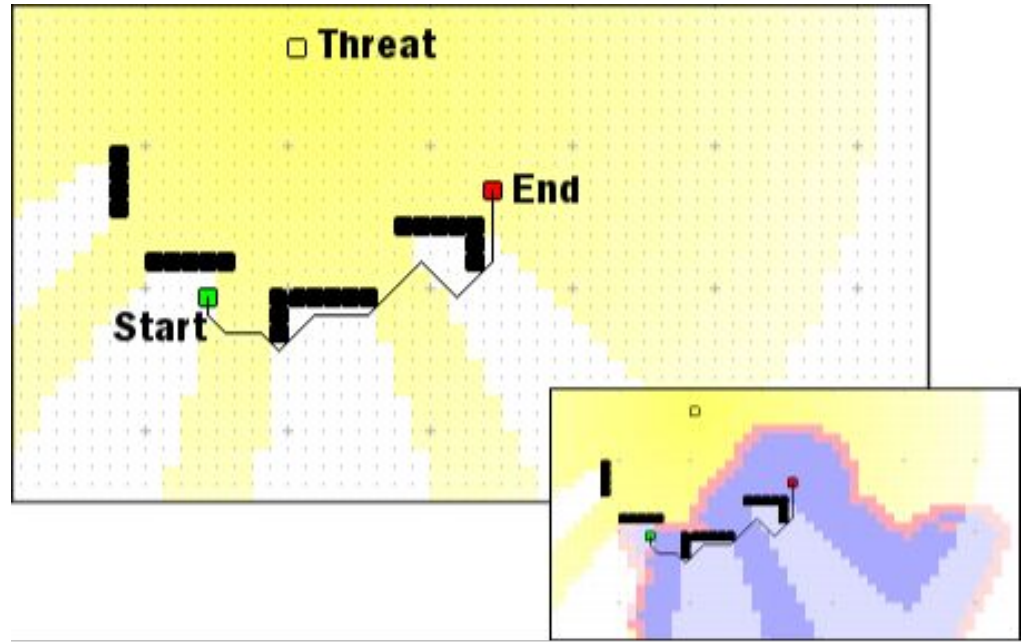


Intelligence Cheats

- Harder enemies are considered more intelligent
- Randomness
  - Players interpret randomness as intelligence
  - Tells a story to justify behaviour
  - Assumes motivation connected to player behaviour
- Make the environment intelligent not the character
  - Ants use sent trails
  - Leaving marks on the ground
  - Simple stimulus response is often enough
- Fake complex processes
  - Line of sight – if player wants to be in cover just make them in cover

Killzone





- Lots of debugging views
- Data driven views for improving AI

How Smart ?

- Simple techniques
  - Tracking the players velocity and last location
    - Fire a rocket to intercept at next location
  - Record where the player is encountered
    - Use information to ambush player
- Feedback
  - Players **assume** the computer is cheating
    - Using information not available to the player
    - Given more health
    - Better weapons
  - Players need to **see the AI thinking** and action
  - Invisible/imperceptible actions do not improve player experience

Communication

- Player must be aware of AI
- Exaggerate AI actions
- Inspiration from Theatre
  - Make it over the top action
  - Obvious decision process
- Reward player for “**out thinking**” AI
  - Include the AI as part of the game mechanics
  - Player “support AI” has to be weaker than the player
  - Balance stupidity

Lua

- Lua still used in game AI
  - Angry Birds mostly Lua
  - Flame malware Lua
  - WoW UI Lua
- Functions
  - Pass variables in a stack
  - Get results as a stack
- C++ loads Lua - doLua(file) then calls functions
- Lua calls functions out of the C++ code

- https://www.lua.org/pil/1.html

Lua in C++

```
#pragma comment(lib, "lib/lua")
#pragma comment(lib, "lib/lualib")
extern "C"
    {
        #include "include/lua.h"
        #include "include/lualib.h"
        #include "include/lauxlib.h
"
    }

lua_State* L;

int main(int argc, char* argv[]){
    L=lua_open();

    luaopen_base(L);
    luaopen_io(L);

    lua_dofile(L,"my.lua");
    lua_close(L);
    return 0;
}
```

```
static int l_sin (lua_State *L) { double d =
luaL_checknumber(L, 1); lua_pushnumber(L, sin(d));
return 1; /* number of results */ }
```

Lua in Unreal

Plugin

LuaMachine

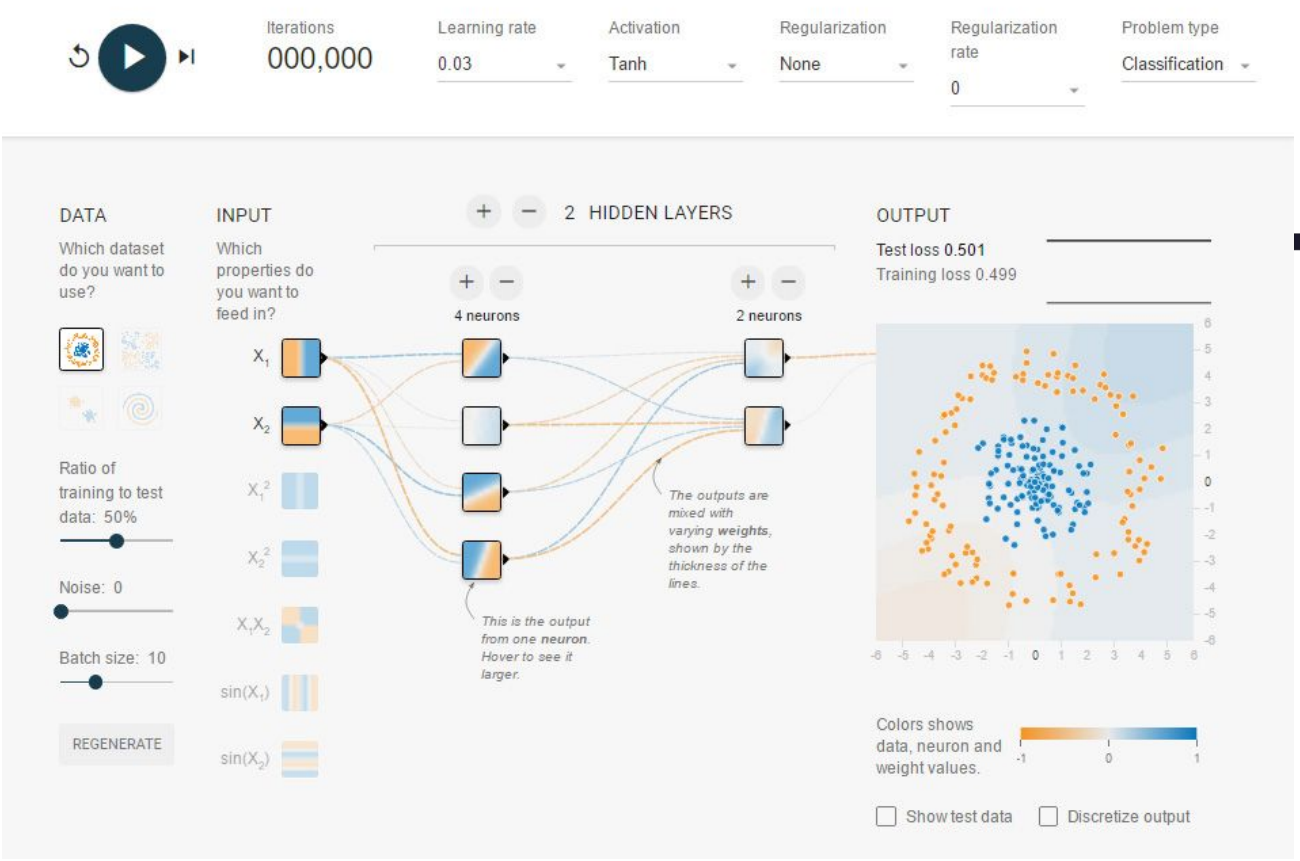
https://www.unrealengine.com/marketplace/en-US/product/luamachine

Manual addition through C++

https://michaeljcole.github.io/wiki.unrealengine.com/Integrating_Lua/

Tensor Flow

Google AI - network flow



TensorFlow
RESEARCH CLOUD

New AI

Used to model players

Used to extract income from players

Data visualisation

Debugging and playtesting the game

Balancing the game, AI playtesters.