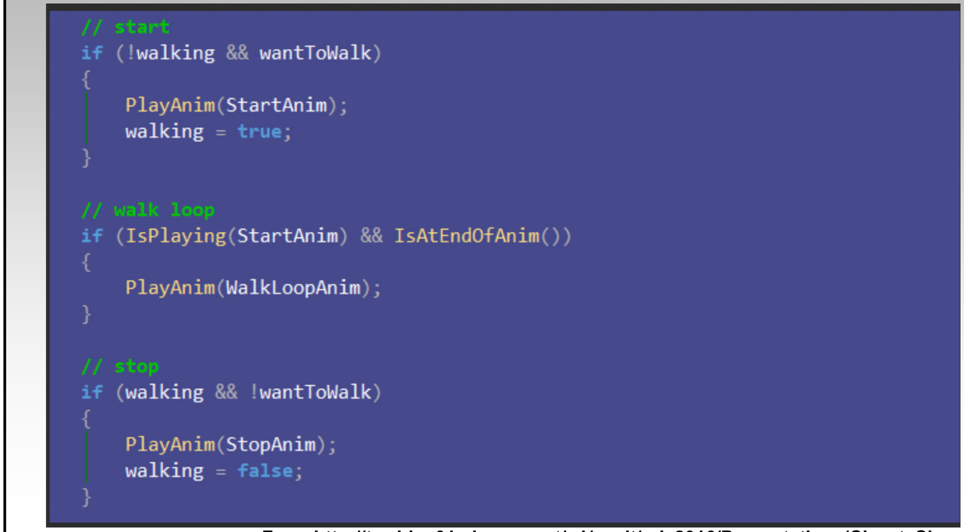
CPSC 436D
Video Game Programming

Basic Gameplay

© Alla Sheffer

Gameplay

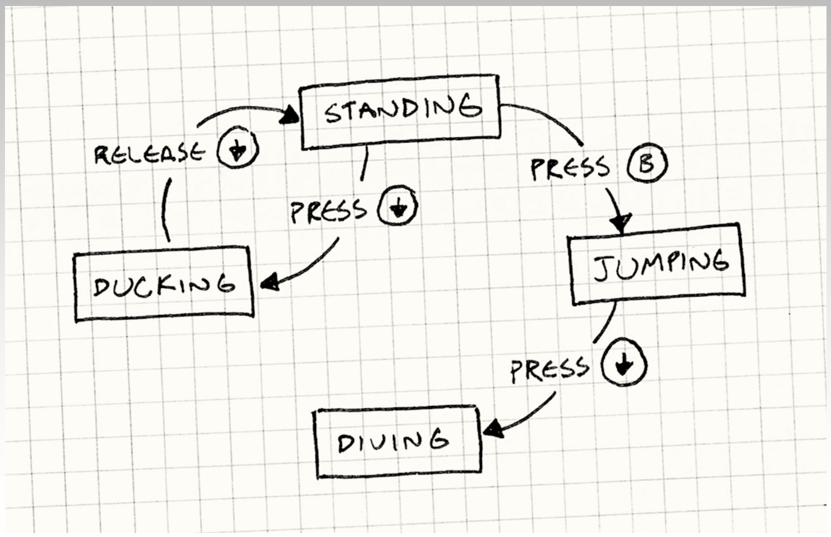


```
// start
if (!walking && wantToWalk)
{
    PlayAnim(StartAnim);
    walking = true;
}

// walk loop
if (IsPlaying(StartAnim) && IsAtEndOfAnim())
{
    PlayAnim(WalkLoopAnim);
}

// stop
if (walking && !wantToWalk)
{
    PlayAnim(StopAnim);
    walking = false;
}
```

From http://twvideo01.ubm-us.net/o1/vault/gdc2016/Presentations/Clavet_Simon_MotionMatching.pdf

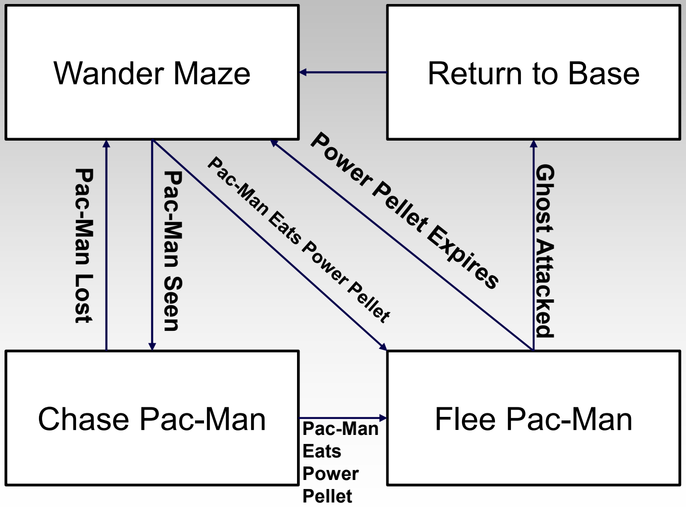
© Alla Sheffer

Finite State Machines: States + Transitions



FSM Example: Pac-Man Ghosts

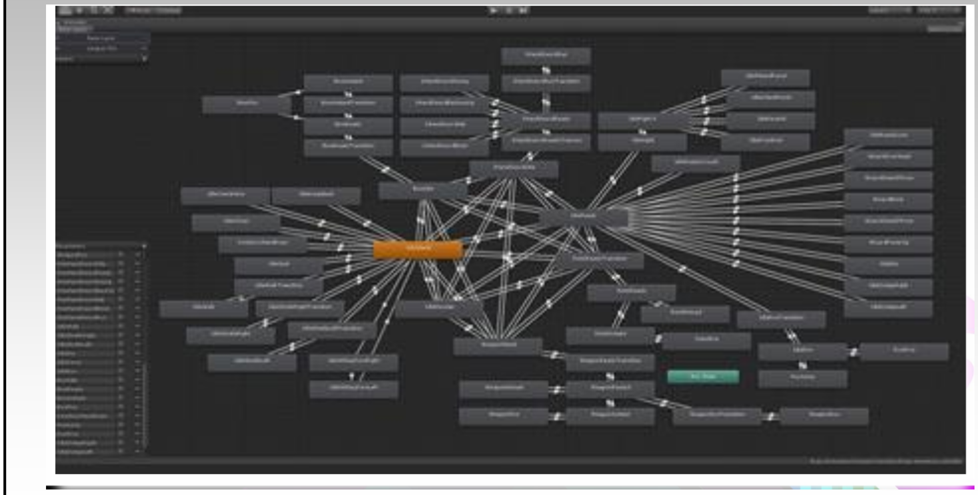
FSM Example: Pac-Man Ghosts



FSMs:

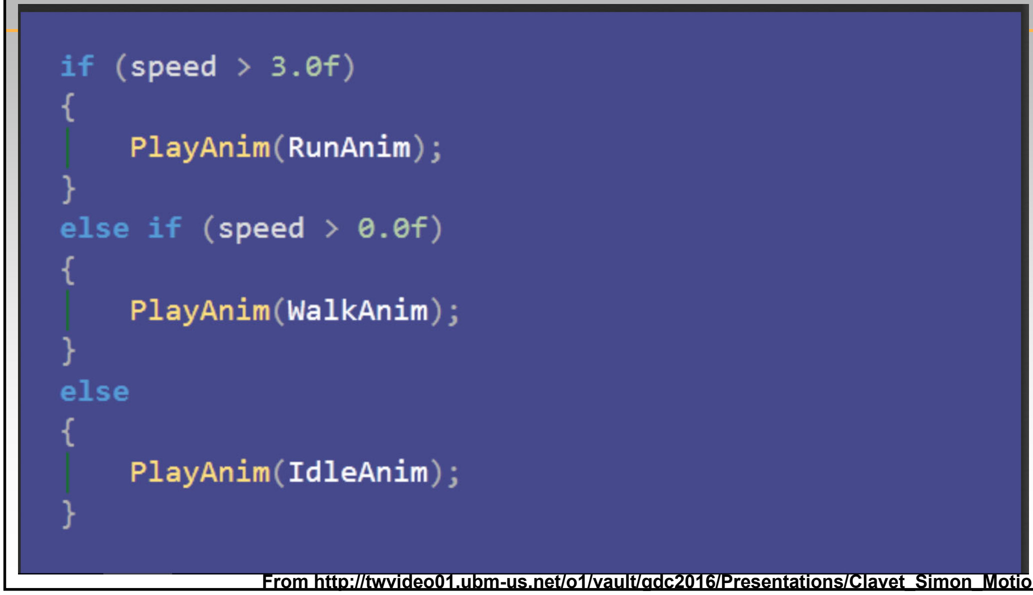
- ***Each frame:***
  - Something (the player, an enemy) does something in its state
  - It checks if it needs to transition to a new state
    - *If so, it does so for the next iteration*
    - *If not, it stays in the same state*
- ***Applications***
  - Managing input
  - Managing player state
  - Simple AI for entities/objects/monsters etc.

FSMs: States + Transitions



From http://twvideo01.ubm-us.net/o1/vault/gdc2016/Presentations/Clavet_Simon_MotionMatching.pdf

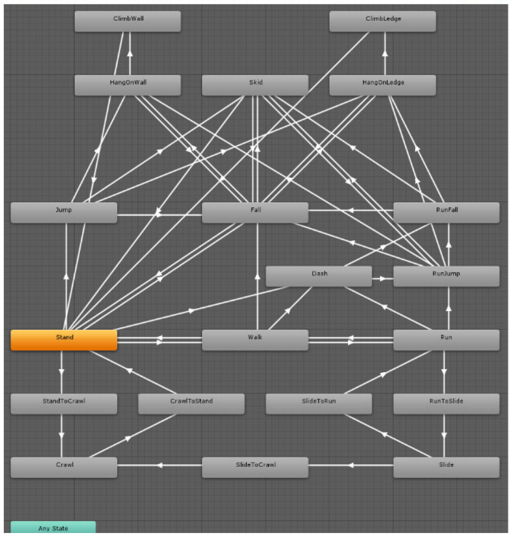
FSMs: States + Transitions



```
if (speed > 3.0f)
{
    PlayAnim(RunAnim);
}
else if (speed > 0.0f)
{
    PlayAnim(WalkAnim);
}
else
{
    PlayAnim(IdleAnim);
}
```

From http://twvideo01.ubm-us.net/o1/vault/gdc2016/Presentations/Clavet_Simon_MotionMatching.pdf

FSMs: Failure to Scale



No way to do long-term planning

No way to ask “How do I get here from there?”

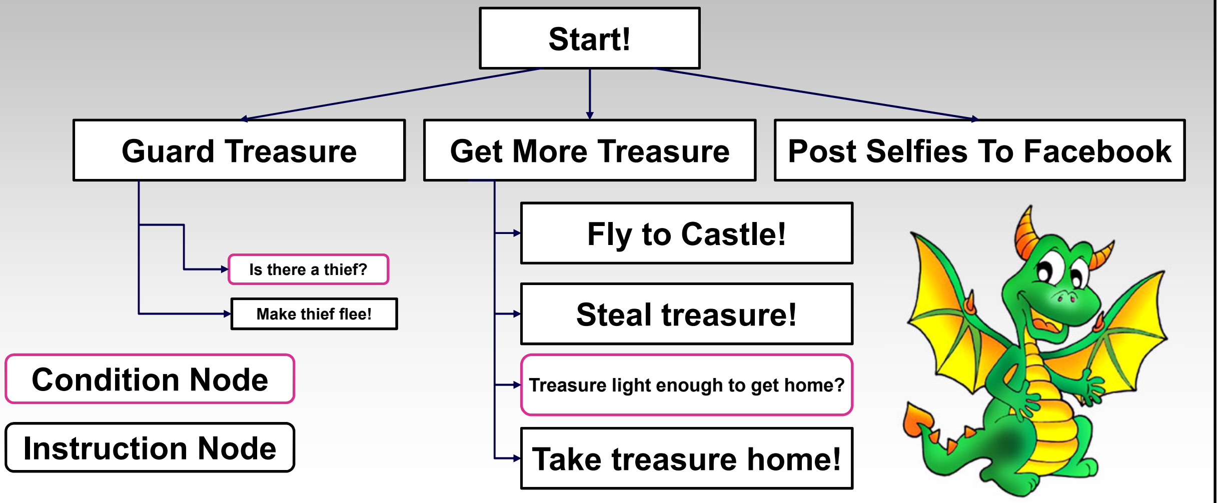
No way to reason about long-term goals

FSMs can get large and hard to follow

Can’t generalize for larger games

From http://twvideo01.ubm-us.net/o1/vault/gdc2016/Presentations/Clavet_Simon_MotionMatching.pdf

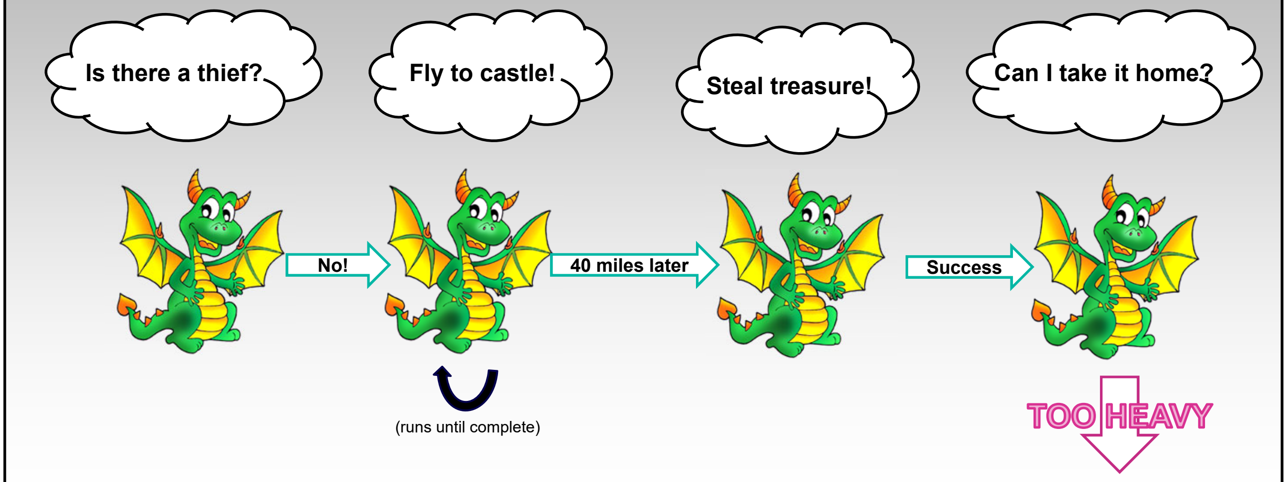
Behaviour Trees: How To Simulate Your Dragon

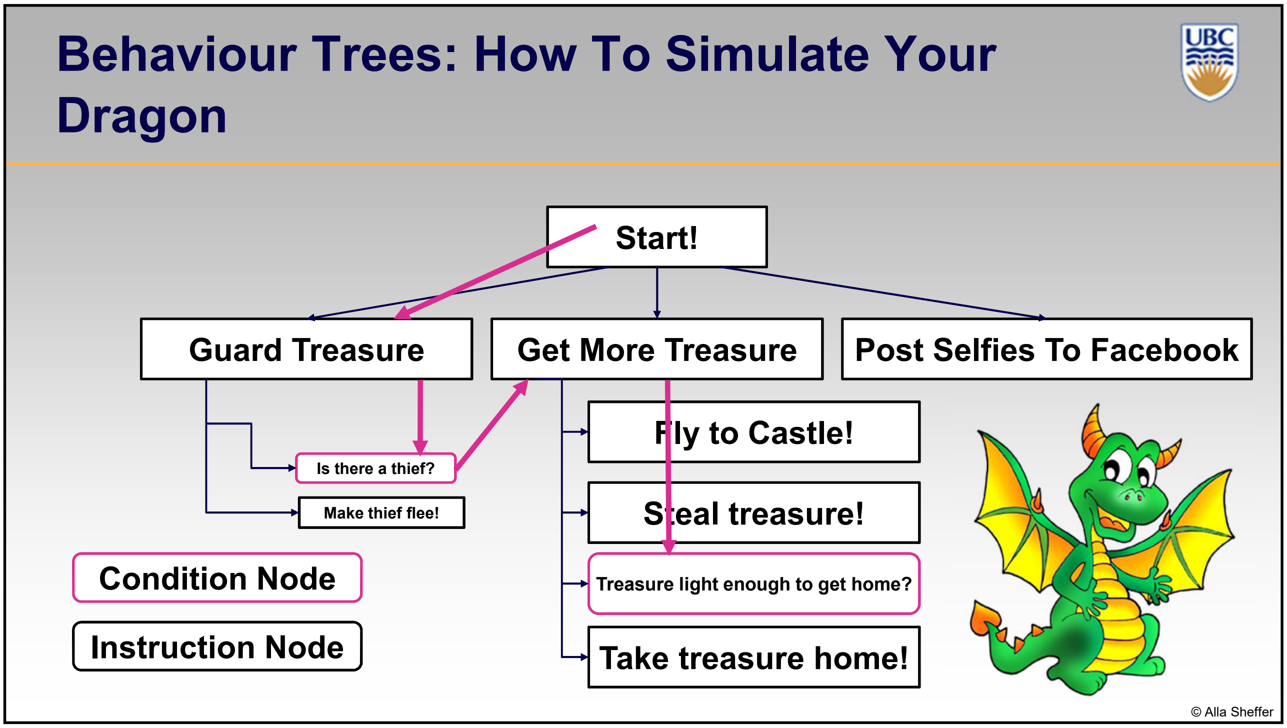


Behaviour Trees

- ***Each frame:***
  - Visit a node
  - See if any **higher priority** nodes now run
    - *If so, execute them instead*
  - See if my currently running node fails
    - *If so, return to the root of the behaviour tree! Start again!*
  - See if the currently running node is done
    - *If so, run the **lower priority** node in the current branch of the tree*

Start!





Behaviour Trees: How To Simulate Your Dragon

Start!

Guard Treasure

Get More Treasure

Post Selfies To Facebook

Is there a thief?

Make thief flee!

Fly to Castle!

Steal treasure!

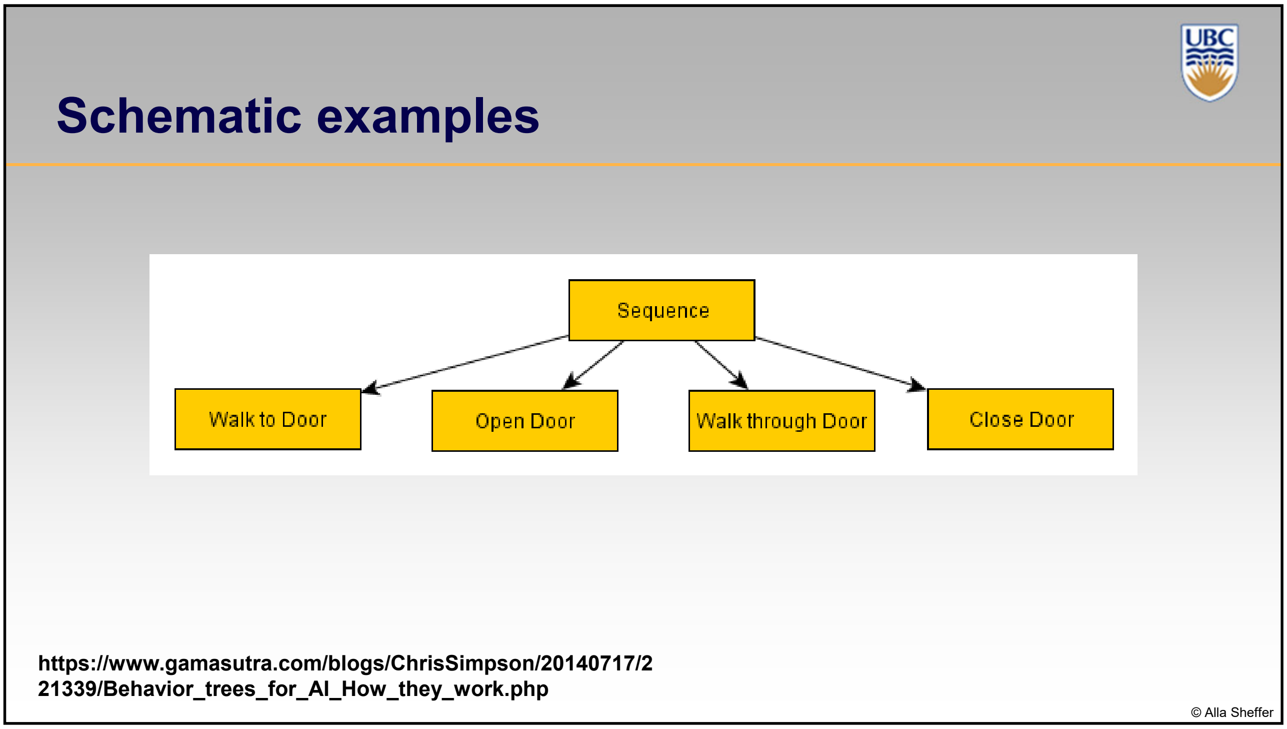
Treasure light enough to get home?

Take treasure home!

Condition Node

Instruction Node

© Alla Sheffer



Schematic examples

Sequence

Walk to Door

Open Door

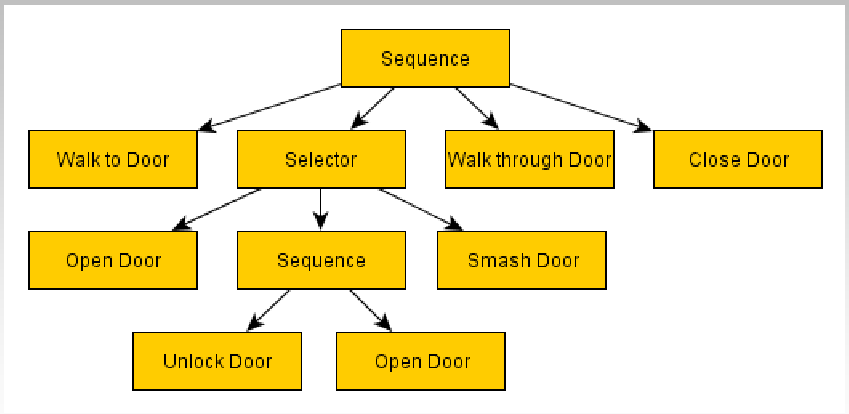
Walk through Door

Close Door

https://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php

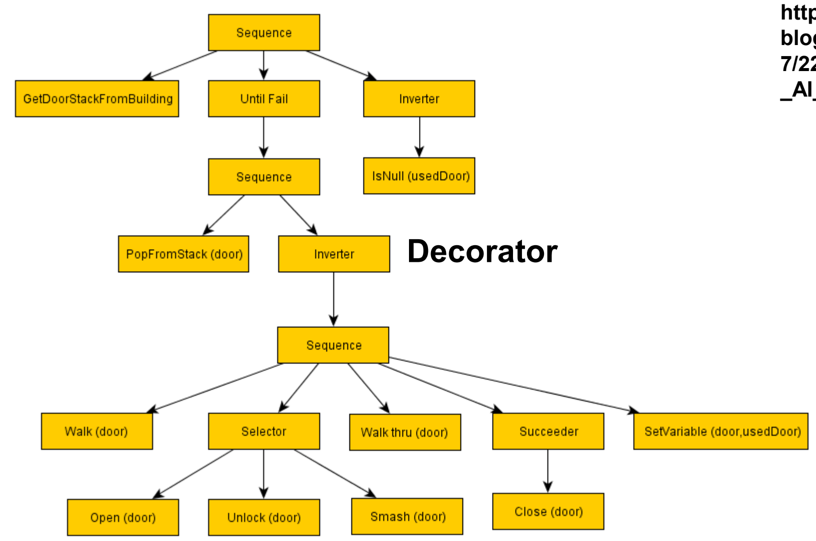
© Alla Sheffer

Schematic examples


Sequence
Walk to Door
Selector
Walk through Door
Close Door
Open Door
Sequence
Smash Door
Unlock Door
Open Door

https://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php

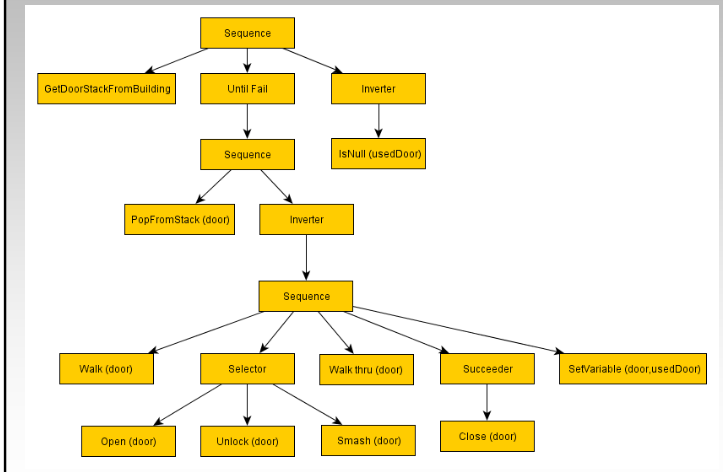
And more complex…

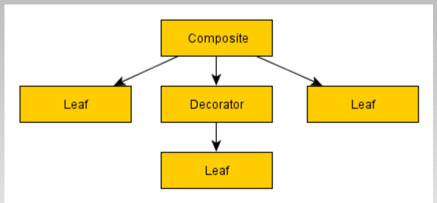

Sequence
GetDoorStackFromBuilding
Until Fail
Inverter
Sequence
IsNull (usedDoor)
PopFromStack (door)
Inverter
Sequence
Walk (door)
Selector
Walk thru (door)
Succeeder
SetVariable (door,usedDoor)
Open (door)
Unlock (door)
Smash (door)
Close (door)

Decorator

https://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php

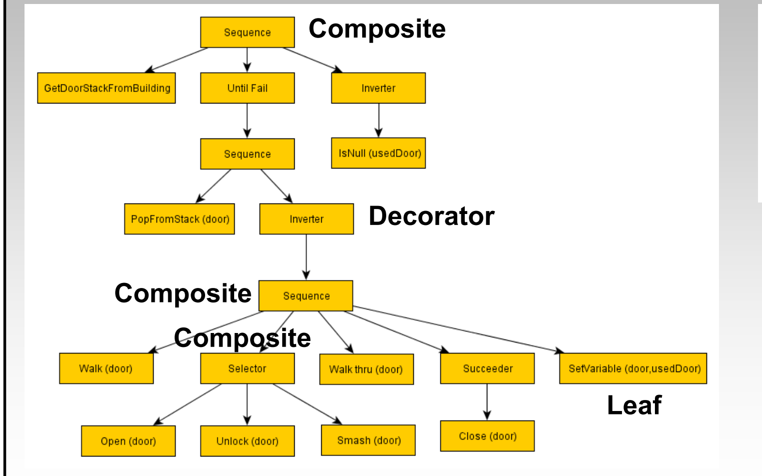
Types





© Alla Sheffer

Types



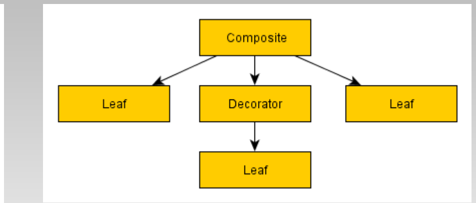
Composite

Decorator

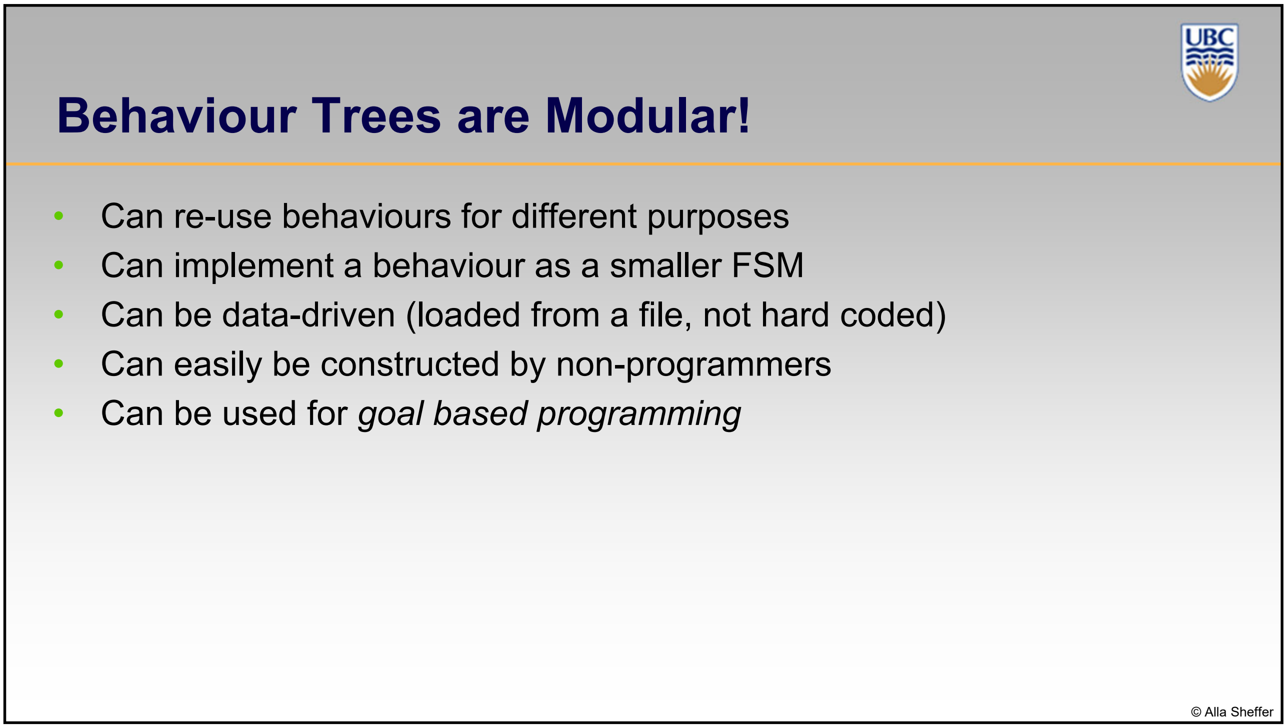
Composite

Composite

Leaf



© Alla Sheffer



Behaviour Trees are Modular!

- Can re-use behaviours for different purposes
- Can implement a behaviour as a smaller FSM
- Can be data-driven (loaded from a file, not hard coded)
- Can easily be constructed by non-programmers
- Can be used for *goal based programming*