CPSC 436D
Video Game Programming

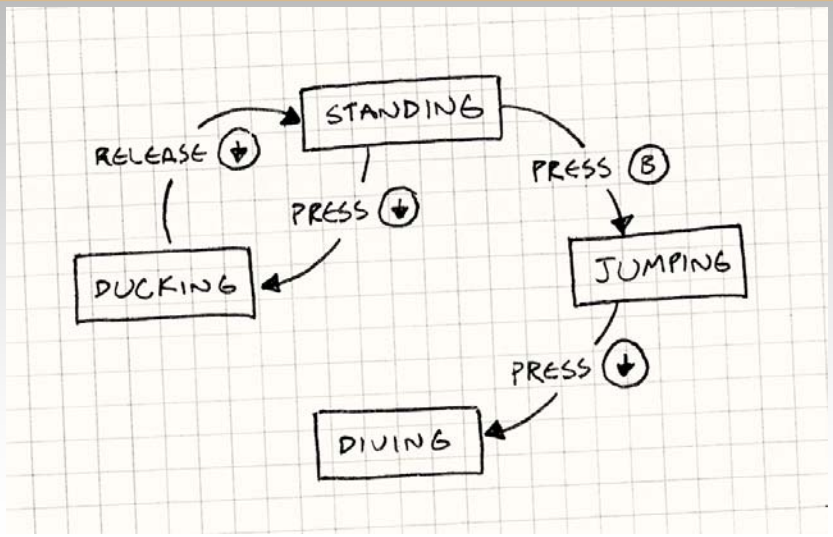
Basic Gameplay

Gameplay

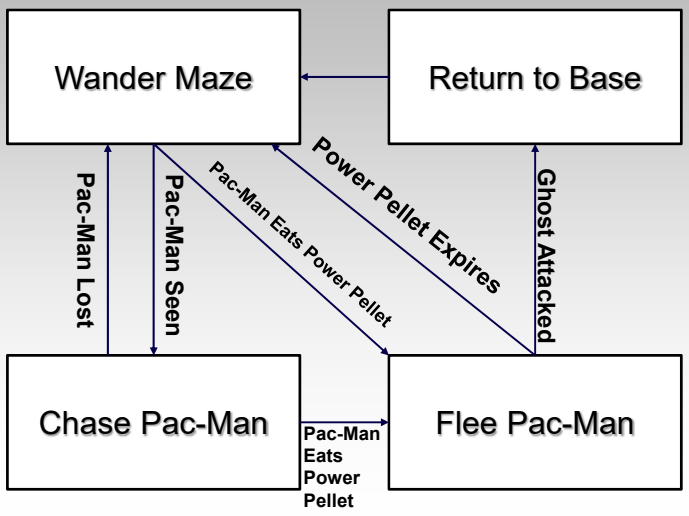
```
// start
if (!walking && wantToWalk)
{
    PlayAnim(StartAnim);
    walking = true;
}

// 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

Finite State Machines: States + Transitions



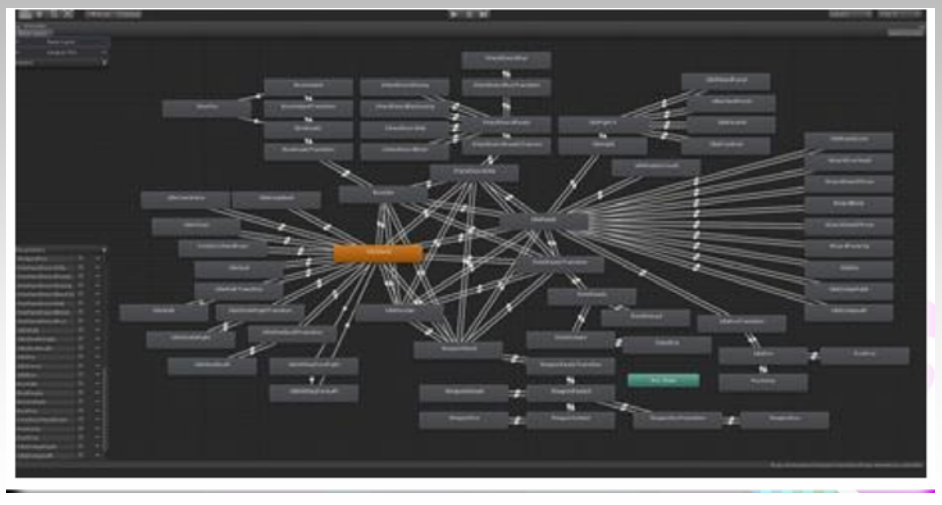
FSM Example: Pac-Man Ghosts



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



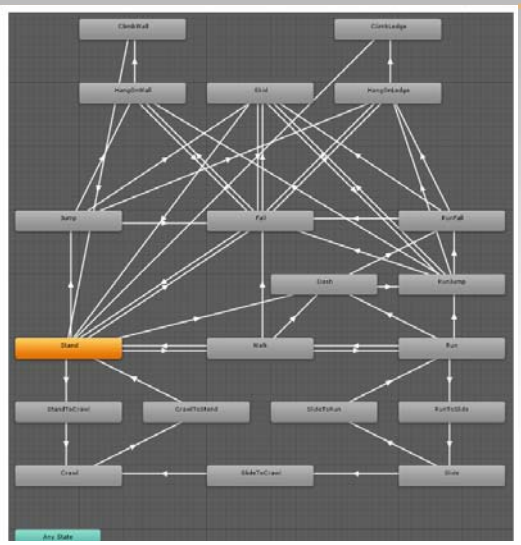
From http://twvideo01.ubm-us.net/o1/vault/gdc2016/Presentations/Clavet_Simon_MotionMatching.pdf

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?"

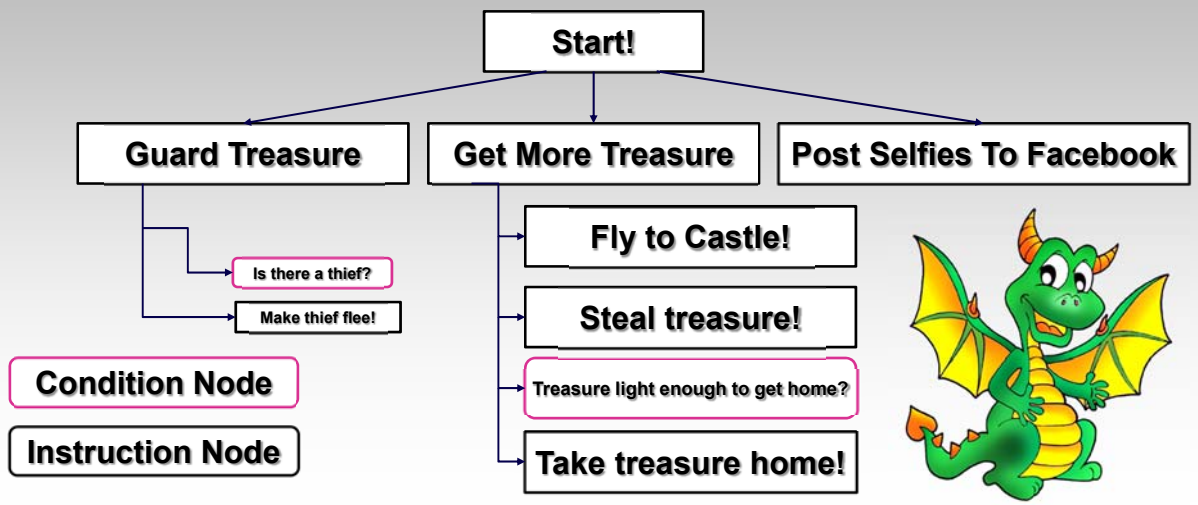
No way to reason about long-term goals

FSMs can get large and hard to follow

Can't generalize for larger games

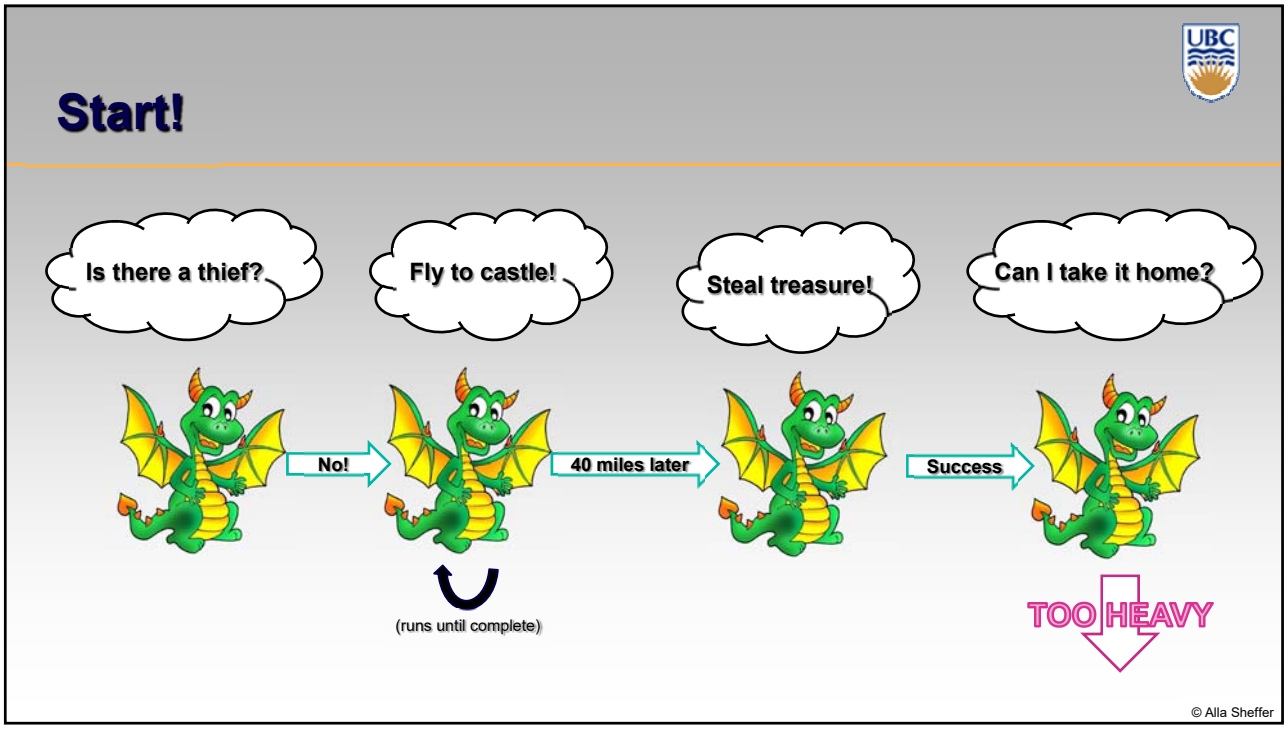
From http://twvideo01.ubm-us.net/o1/vault/gdc2016/Presentations/Clavet_Simon_MotionMatching.pdf

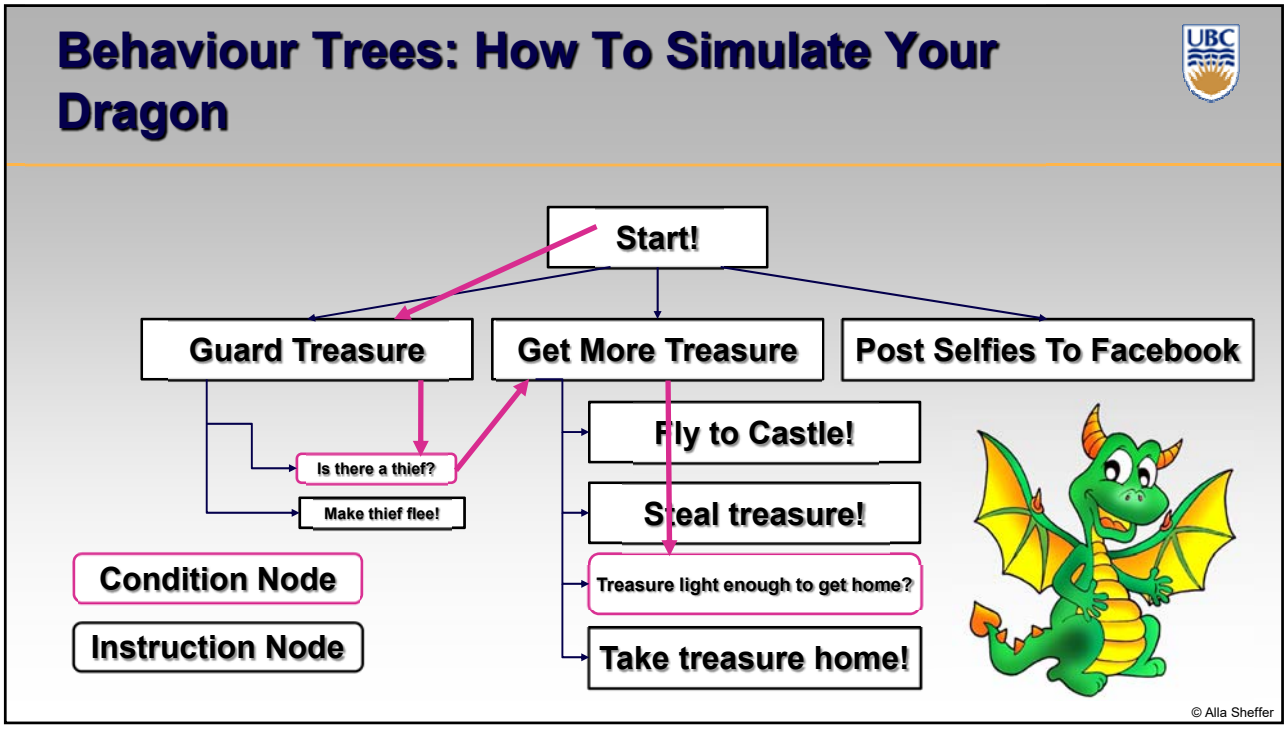
Behaviour Trees: How To Simulate Your Dragon



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!
Is there a thief?
Fly to castle!
Steal treasure!
Can I take it home?
No!
40 miles later
Success
(runs until complete)
TOO HEAVY
© Alla Sheffer


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

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*