What changes you needed to make to accommodate the addition of one extra state? .

I added a new state in the enum TrollState called Idle and added a new case in the switch statement in the fsm() method. I also changed the current state to the idle state.

How are the changes dependent on the number of transitions from/to this newly added state? .

The changes are dependent. The new default state is Idle and it is the original state that needs to change in order for other states to be used.

What changes you predict you need to make to accommodate the removal of one state? .

I have to make sure that the state is not linked to other states. Meaning that they can’t be a dependent state. An example of this involves the realign waypoint, I have removed seek waypoint if I wanted to remove realignwaypoint because they are linked together. To accommodate this, I have to reevaluate states and make it so that all the states are able to transition from each state properly

How are these changes you predict dependent on the number of transitions from/to the state to be removed? .

It is dependent on the state, if the state has numerous transitions, the process becomes more complicated. Each transition that is dependent on the state has to either be removed or changed to ensure a proper flow between the states.

Is this method sustainable if you have a prior knowledge that the lead game designer is fond of changing (adding/removing) states a lot (Not just a Yes/No answer, but also a bit of rationale)?

It is obviously less sustainable if the lead constantly adds and removes states. More transitions between states have to be reviewed and it may become an incredibly complicated problem. There should be a more efficient way to manage states like encapsulation and adding more flexibility using state classes.