

# Outline

- Objective
  - Understand POMDPs through Visualization
- Problem : UAV Navigation
- MDP
  - Value Iteration
  - Policy Execution
- POMDP
  - Belief State Search
  - Policy Comparison
    - 3 Step Horizon
    - 10 Step Horizon
  - AIMA (Russell & Norvig)
  - Coastal Observation
  - Fuel Gauge Observation

For my demo,

For my demo, I implemented a POMDP.

Mapped a small UAV navigation problem to a POMDP.

-scann

help you visualize how a pomdp works.

MDP

Mapped a small UAV problem to an MDP.

Used as base code for the MDP, a matlab script with some basic mdp functionality.

Mapped my problem to it.

Created the visualization of the value iteration, and the policy execution.

I took a small POMDP problem and mapped it to a UAV navigation problem.

To do this, I scanned in a map with an airport landing strip.

# Demonstrations

- MATLAB

- MDP

- Map to Problem Domain
- Value Iteration
- Policy Execution
- Visualization

- POMDP

- Map to Problem Domain
- Belief Space Search
- Policy Execution
- Visualization

\* All coding is mine except for some basic MDP value iteration functionality.

\*\* Visualizations Run In line with code.

<http://www.cs.ubc.ca/~murphyk/Software/MDP/mdp.html>

\* I implemented everything that I will be showing except for the MDP value iteration, in which I found some code with basic MDP value iteration functionality.

For my demo,

For my demo, I implemented a POMDP.

Mapped a small UAV navigation problem to a POMDP.

-scann

help you visualize how a pomdp works.

MDP

Mapped a small UAV problem to an MDP.

Used as base code for the MDP, a matlab script with some basic mdp functionality.

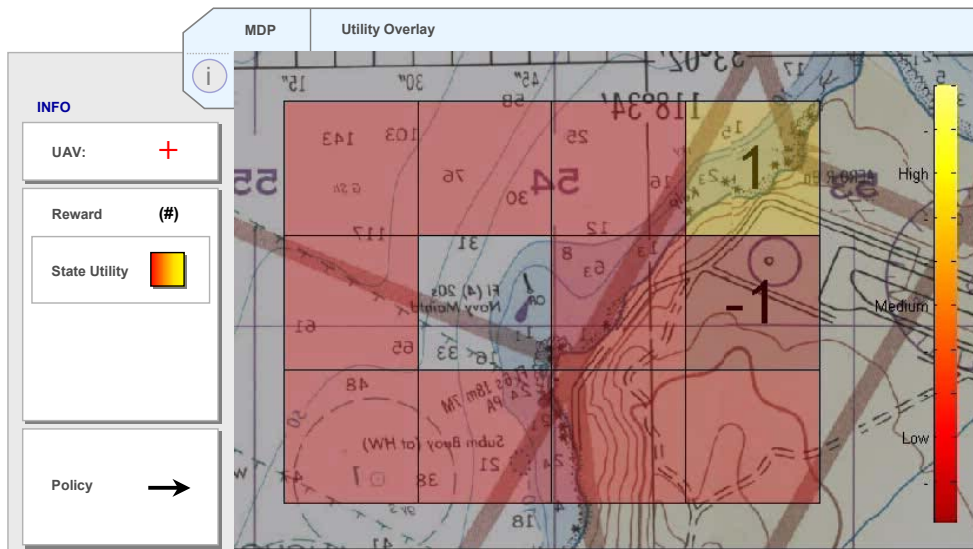
Mapped my problem to it.

Created the visualization of the value iteration, and the policy execution

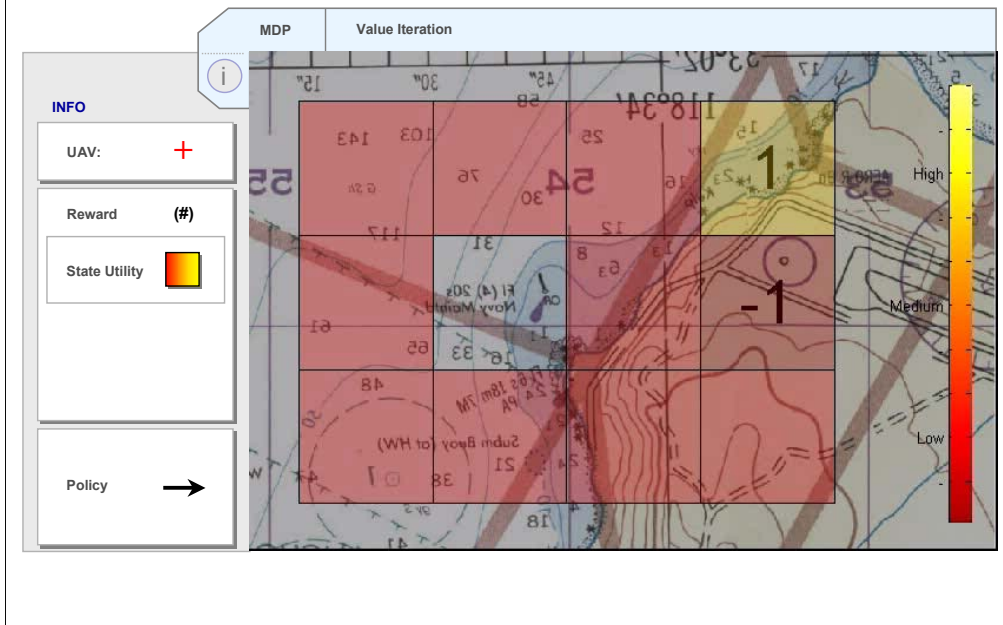
# Problem : Introduction



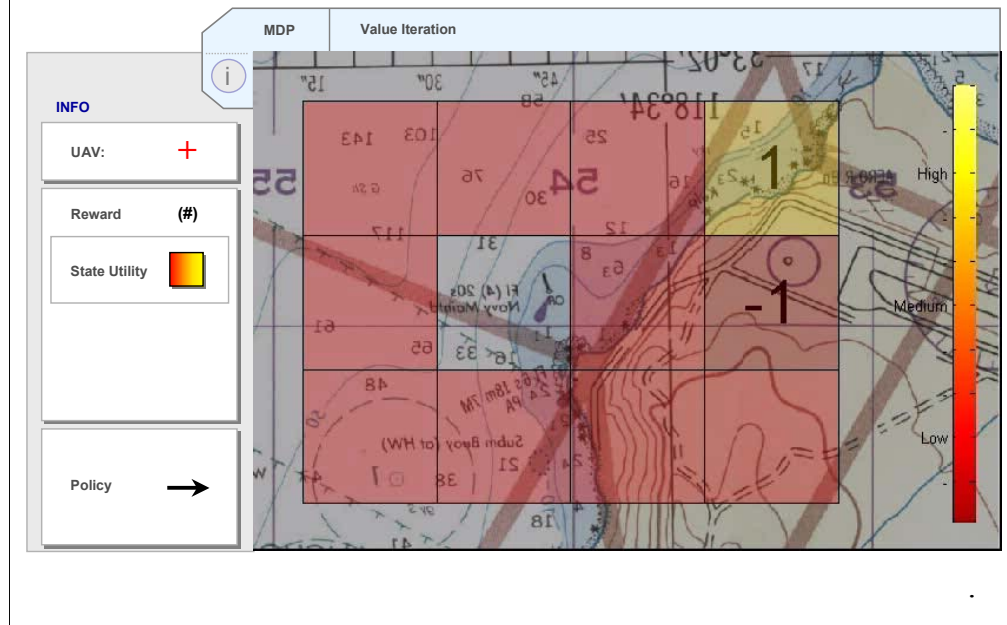
# Problem : Utility Overlay



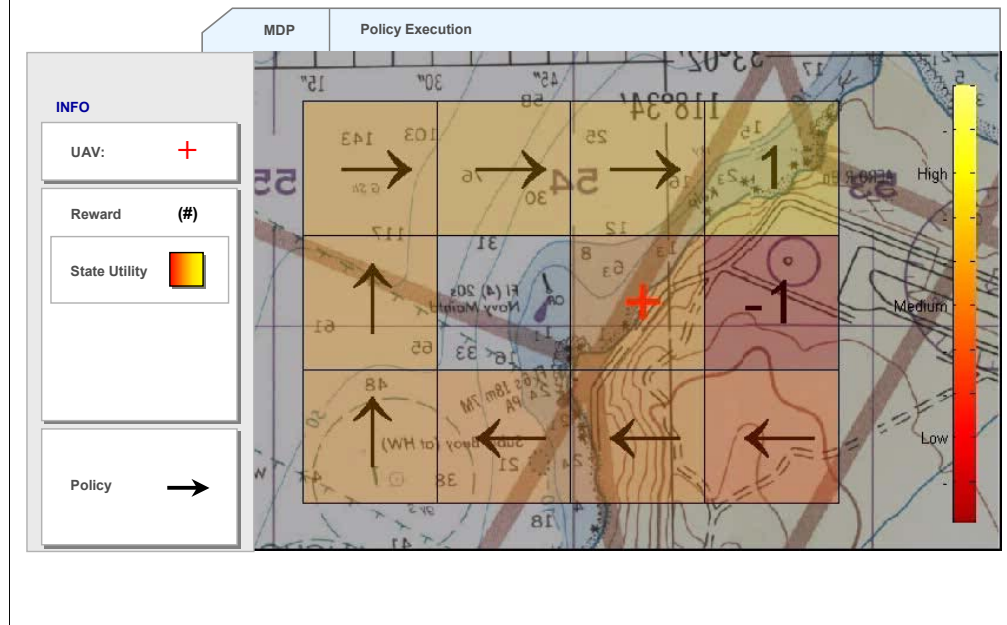
# MDP Value Iteration



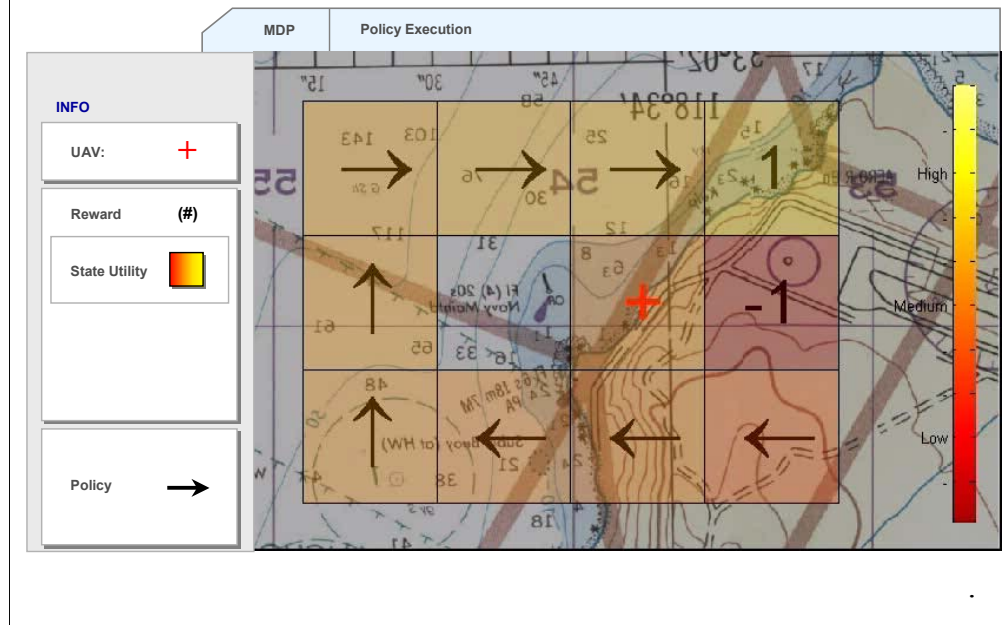
# MDP Value Iteration



# MDP Policy Execution

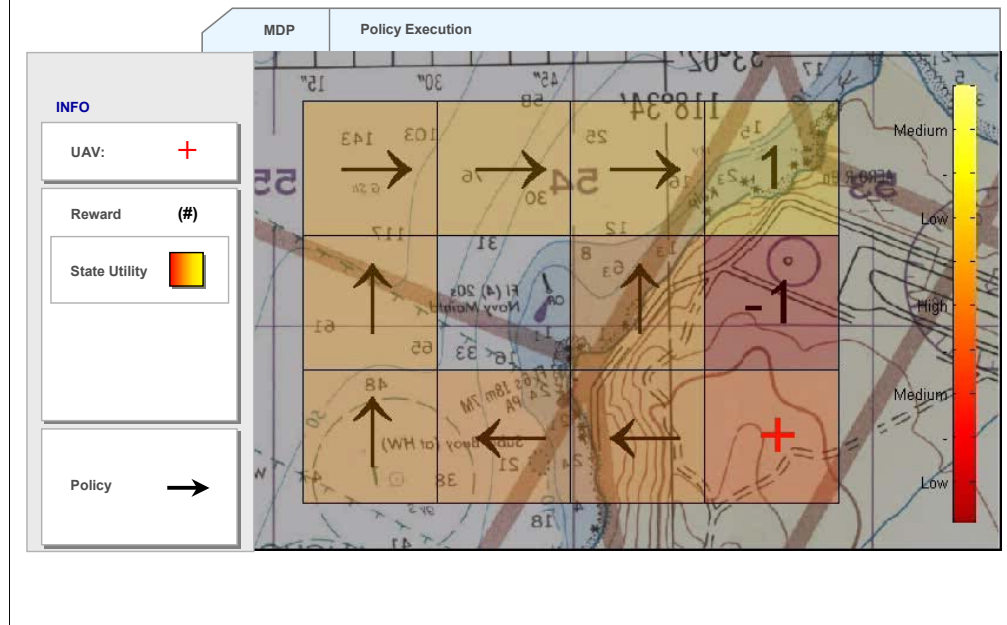


# MDP Policy Execution

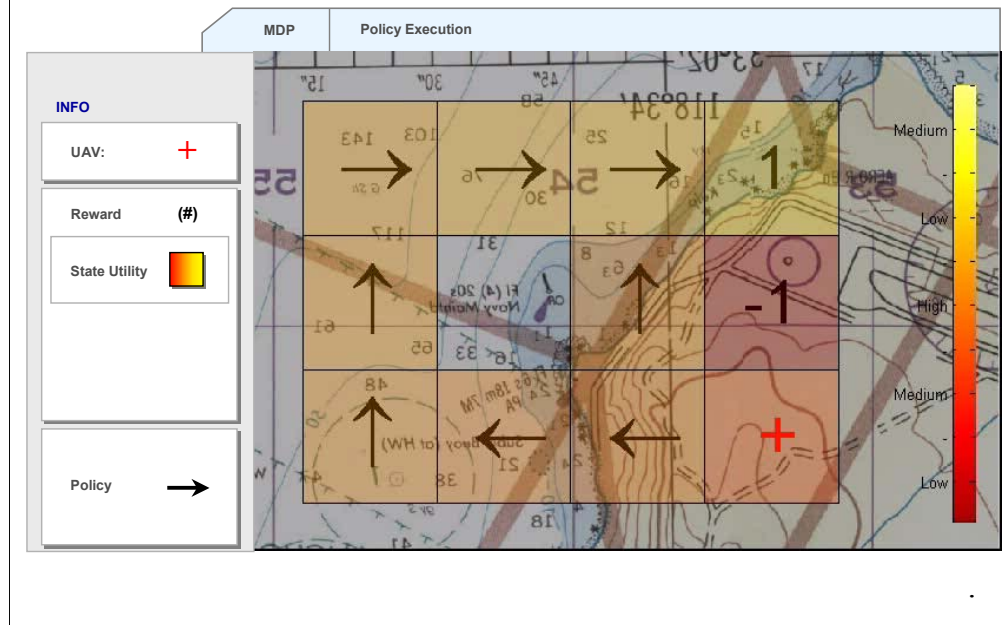




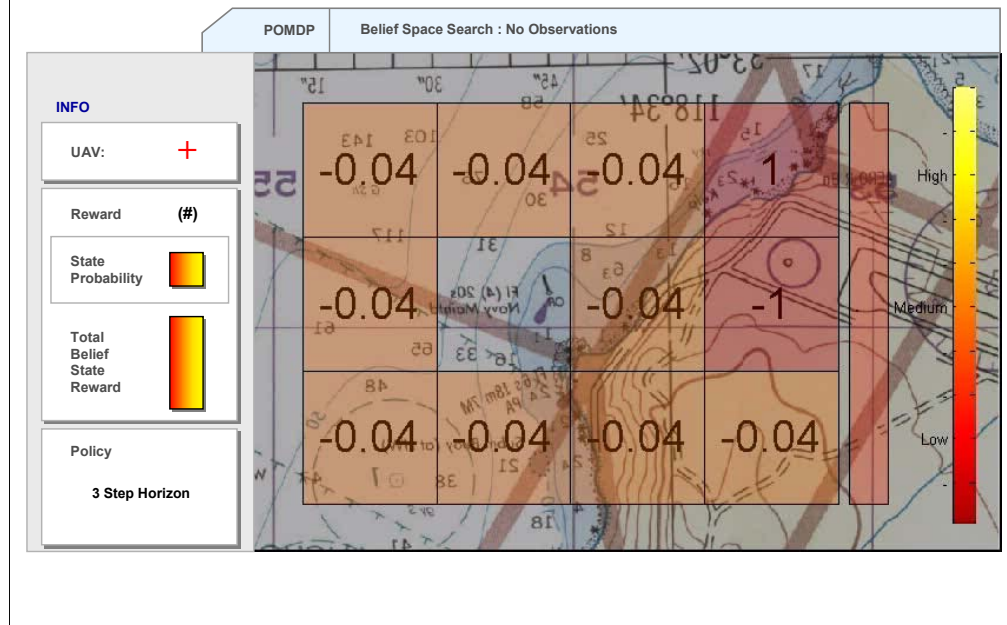
# MDP Policy Execution



# MDP Policy Execution



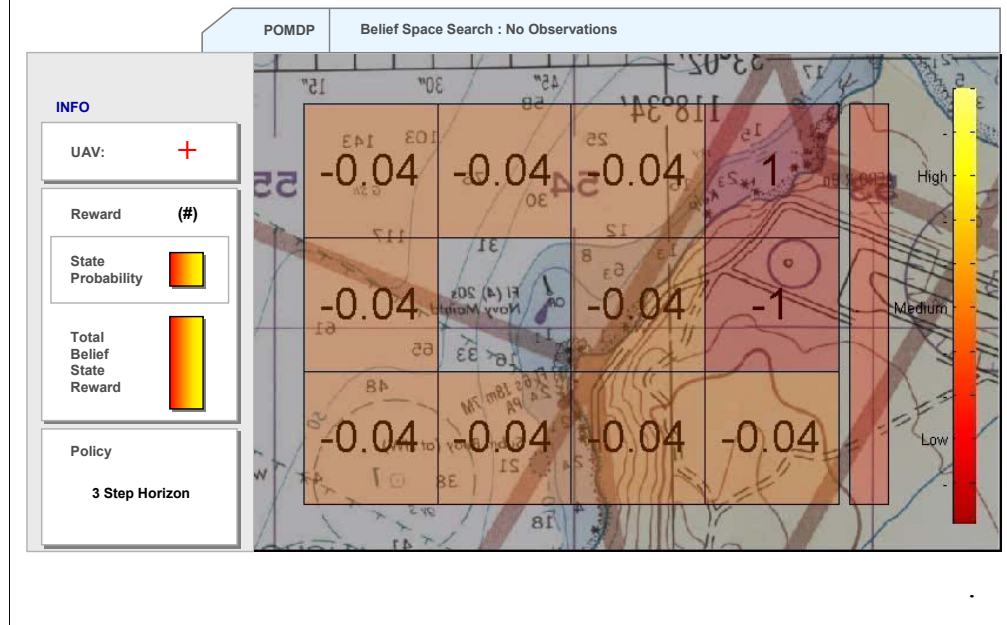
# POMDP Belief Space Search



For a POMDP, the policy creation is different. For a POMDP, we do not assume absolute positional knowledge. Our position is represented in our belief space. Therefore, in order to assess the value of a given action, we have to assess the value of all of the best possible subsequent actions. When we execute an action, we generate a new belief space given our current belief space and the particular action. With each action, (for example 'up') we have a probability of ending up in the next block up. There is also a probability of ending up in other blocks. (This was also the case with the MDP.)

However, with the POMDP, we do not assume absolute positional (or belief space) knowledge, therefore, we must propagate forward the chosen action for each possible belief. (assuming a 1 time-step look-ahead) We then get a matrix of values for each possible action-belief combination. We then multiply our current belief distribution by this matrix to find out which action has the best outcome. When we find out which action has the best outcome, we set our policy to that reward. However, if we have a 3 step look-ahead, we have to search ahead every combination of action sequences. For the 3 step look-ahead shown here (4 actions), there are 64 combinations which result in 85 possible futures (or beliefs) each with 13 states. For a 10 step look-ahead, there are 1.4 million possible futures which have to be assessed. The number represents the cost or reward associated with that positions. The color indicates the value of that position given the action sequence. The color bar at the right indicates the total value of that belief state, give the action sequence. We want the policy which generates the best possible outcome after 3 steps.

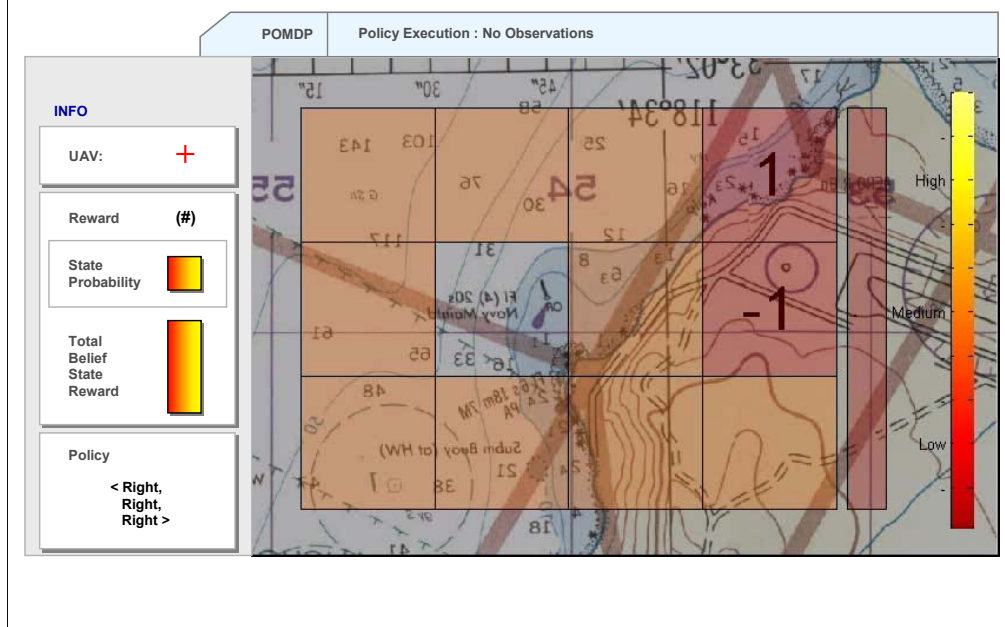
# POMDP Belief Space Search



For a POMDP, the policy creation is different. For a POMDP, we do not assume absolute positional knowledge. Our position is represented in our belief space. Therefore, in order to assess the value of a given action, we have to assess the value of all of the best possible subsequent actions. When we execute an action, we generate a new belief space given our current belief space and the particular action. With each action, (for example 'up') we have a probability of ending up in the next block up. There is also a probability of ending up in other blocks. (This was also the case with the MDP.)

However, with the POMDP, we do not assume absolute positional (or belief space) knowledge, therefore, we must propagate forward the chosen action for each possible belief. (assuming a 1 time-step look-ahead) We then get a matrix of values for each possible action-belief combination. We then multiply our current belief distribution by this matrix to find out which action has the best outcome. When we find out which action has the best outcome, we set our policy to that reward. However, if we have a 3 step look-ahead, we have to search ahead every combination of action sequences. For the 3 step look-ahead shown here (4 actions), there are 64 combinations which result in 85 possible futures (or beliefs) each with 13 states. For a 10 step look-ahead, there are 1.4 million possible futures which have to be assessed. The number represents the cost or reward associated with that positions. The color indicates the value of that position given the action sequence. The color bar at the right indicates the total value of that belief state, give the action sequence. We want the policy which generates the best possible outcome after 3 steps.

# POMDP Policy (3 Iterations)



This slide demonstrates the 3 step policy. <Right, Right, Right>

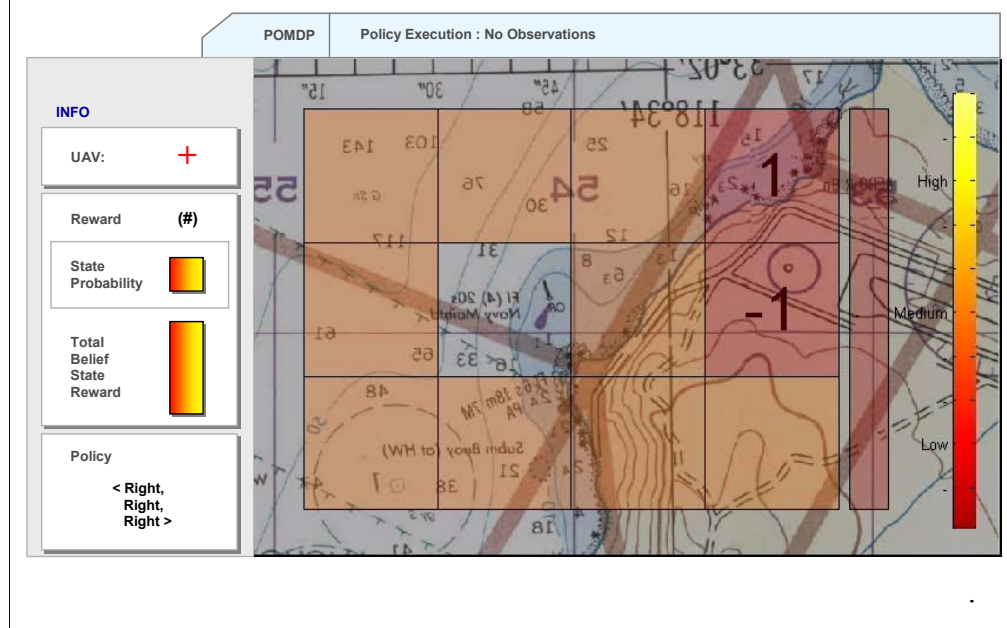
The color scheme indicates the probability that we are in a given location.

The policy isn't very good, because it is very likely that we end up in the bottom right corner, or on the take-off runway.

This is because our look-ahead was too short.

In the next slide, we will see the policy that is generated from a 10-step look-ahead.

# POMDP Policy (3 Iterations)



This slide demonstrates the 3 step policy. <Right, Right, Right>

The color scheme indicates the probability that we are in a given location.

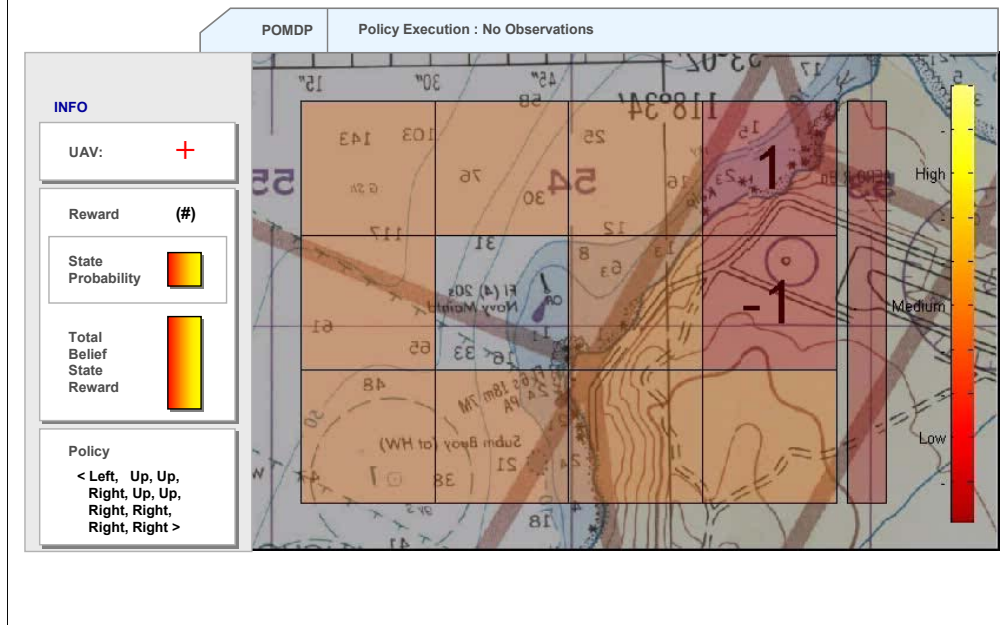
The policy isn't very good, because it is very likely that we end up in the bottom right corner, or on the take-off runway.

This is because our look-ahead was too short.

In the next slide, we will see the policy that is generated from a 10-step look-ahead.



# POMDP Policy (10 Iterations)



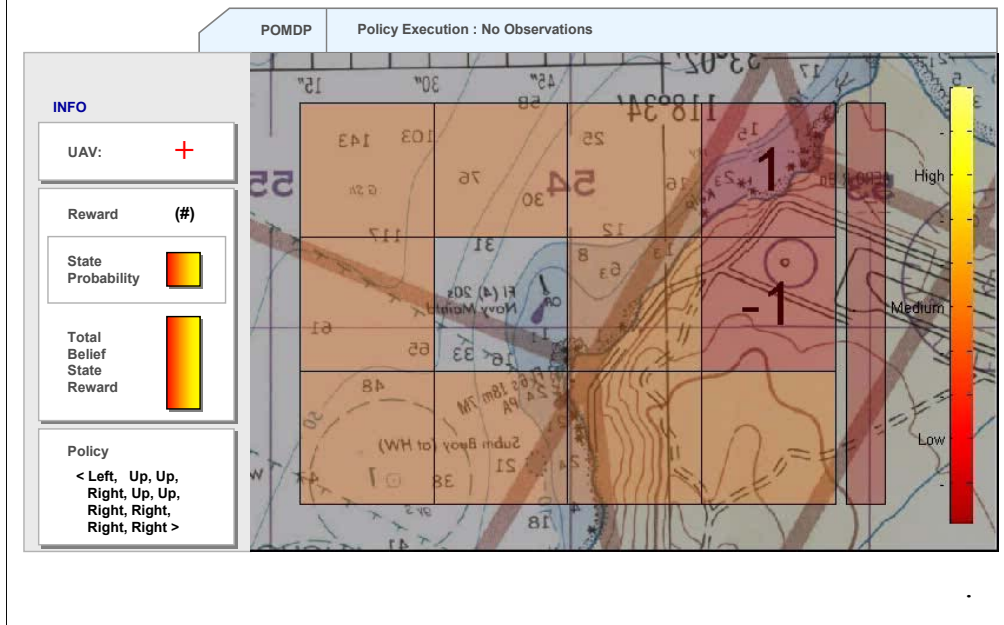
This slide shows the execution of a policy generated from a 10-step look-ahead. Remember that we haven't discussed observations yet. Therefore, this policy assumes an initial position distribution. Our belief space currently only represents our position distribution.

Our action cost is relatively low here. Therefore, the policy is to move left first. This gets us out of the lower right hand corner (if we were there, we don't know exactly).

The we go up... < Left, Up, Up, Right, Up, Up, Right, Right, Right, Right >

With each action, we gain positional accuracy. You can see this from the color scheme. The color scheme indicates the probability that we are in a given location. It is initially uniform for the 9 possible starting positions. It then gets more yellow to the left. The Yellow high probability trail then makes it's way up and over to the runway. The policy exploits the bounded-ness of the grid (we can't go too far to the left) to gain positional accuracy, and generate a better policy. This policy causes us to end up at the desired destination with high probability. Note that the representation has a sink. Once we hit the runway, we go to the sink in the next iteration. That is why the yellow disappears from the runway after it arrives. Note that the yellow high probability disappears from the grid. This problem is the exact same problem represented in the Russel and Norvig (AIMA) book. The book gives a different policy for this POMDP with no observations. The policy we generated is better, as I will show you.

# POMDP Policy (10 Iterations)



This slide shows the execution of a policy generated from a 10-step look-ahead. Remember that we haven't discussed observations yet. Therefore, this policy assumes an initial position distribution. Our belief space currently only represents our position distribution.

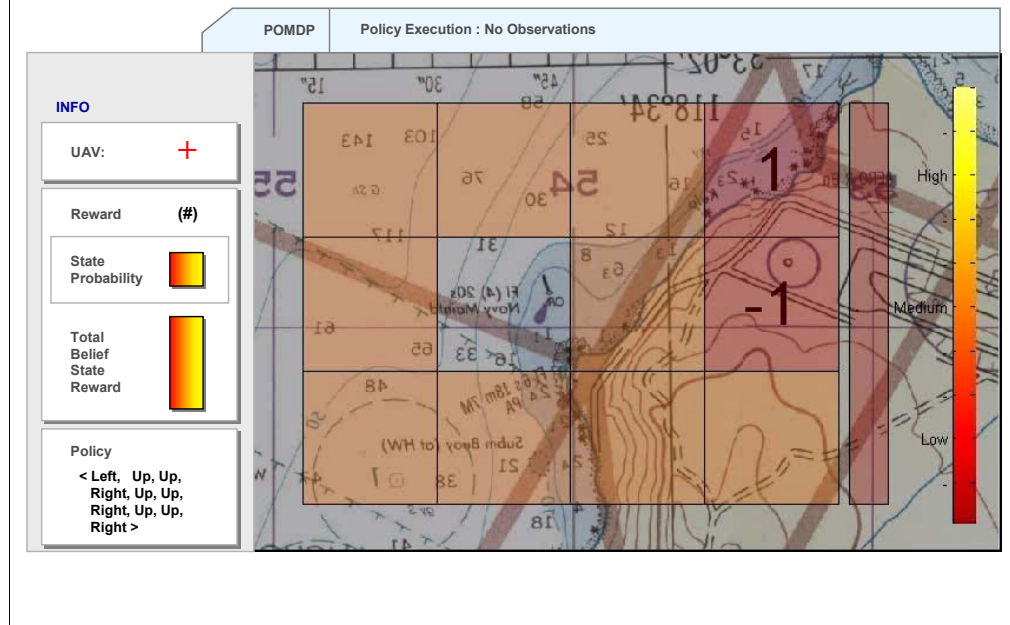
Our action cost is relatively low here. Therefore, the policy is to move left first. This gets us out of the lower right hand corner (if we were there, we don't know exactly).

The we go up... < Left, Up, Up, Right, Up, Up, Right, Right, Right, Right >

With each action, we gain positional accuracy. You can see this from the color scheme. The color scheme indicates the probability that we are in a given location. It is initially uniform for the 9 possible starting positions. It then gets more yellow to the left. The Yellow high probability trail then makes it's way up and over to the runway. The policy exploits the bounded-ness of the grid (we can't go too far to the left) to gain positional accuracy, and generate a better policy. This policy causes us to end up at the desired destination with high probability. Note that the representation has a sink. Once we hit the runway, we go to the sink in the next iteration. That is why the yellow disappears from the runway after it arrives. Note that the yellow high probability disappears from the grid. This problem is the exact same problem represented in the Russel and Norvig (AIMA) book. The book gives a different policy for this POMDP with no observations. The policy we generated is better, as I will show you.



# POMDP Policy (AIMA)



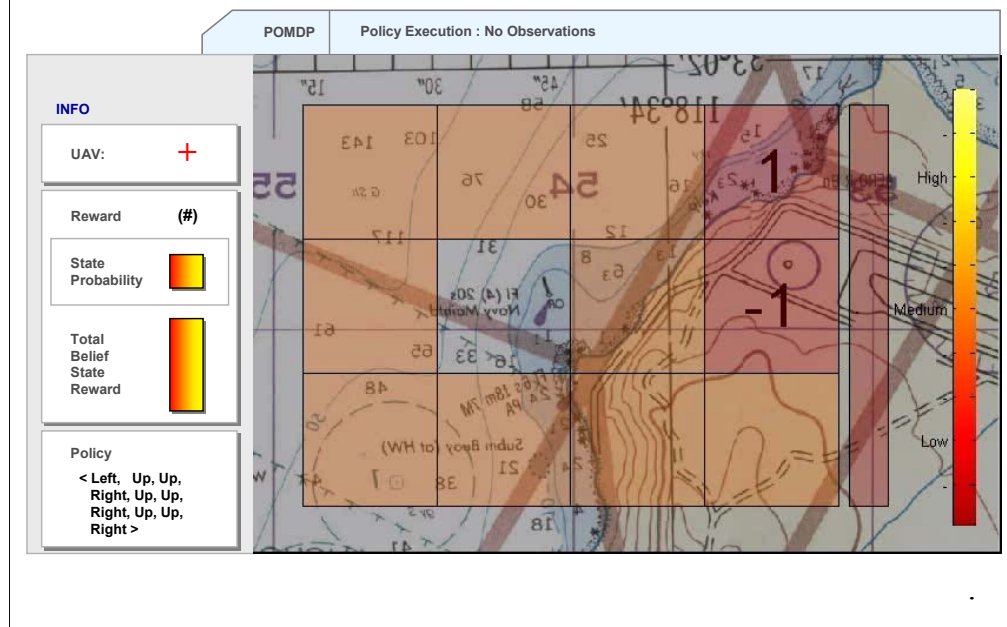
This is the AIMA policy.

It leaves some yellow on the field at the end.

AIMA < Left, Up, Up, Right, Up, Up, Right, Up, Up, Right >

I ran both policies for 10 steps.

# POMDP Policy (AIMA)



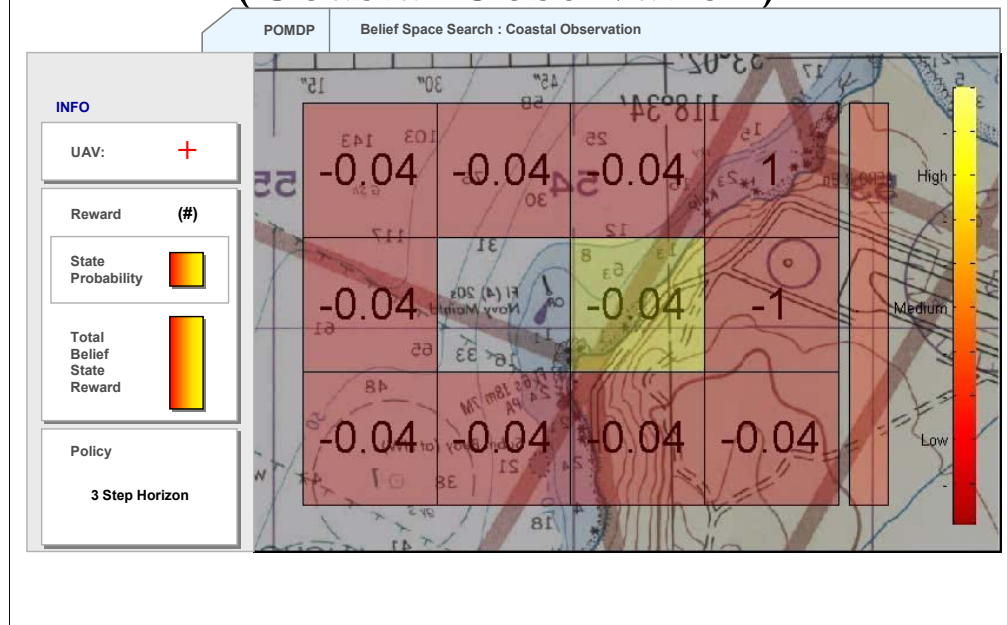
This is the AIMA policy.

It leaves some yellow on the field at the end.

AIMA < Left, Up, Up, Right, Up, Up, Right, Up, Up, Right >

I ran both policies for 10 steps.

# POMDP Belief Space Search (Coastal Observation)



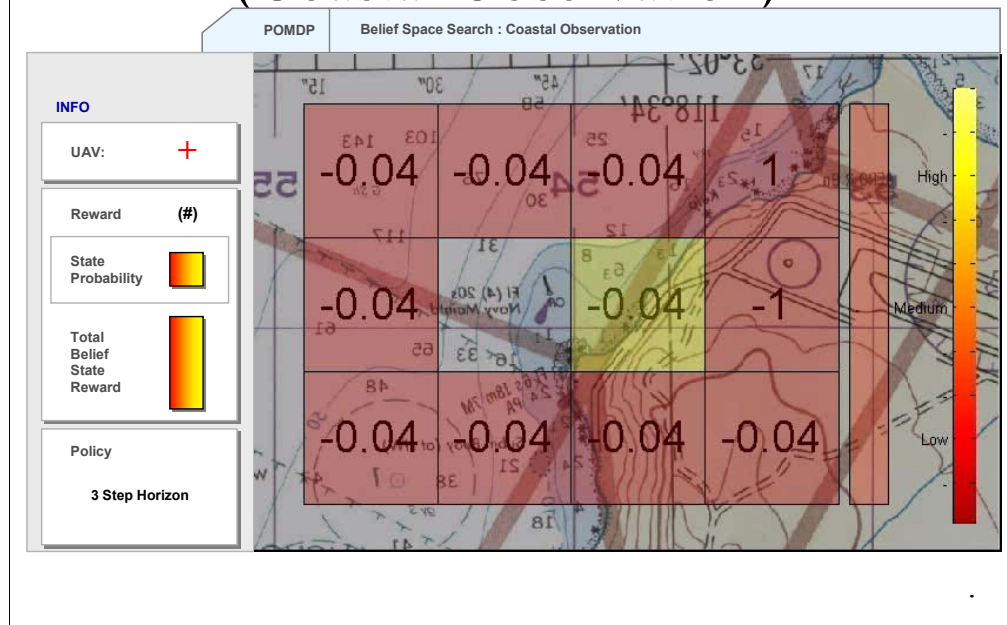
Next we introduce how observations play into POMDPs.

Let's say that if you are in position 8, you can clearly see the coast line.  
Consequently, you gain positional accuracy.

If that is a possibility, then we also have to generate a policy given that observation sequence.

This slide shows that policy being generated.

# POMDP Belief Space Search (Coastal Observation)



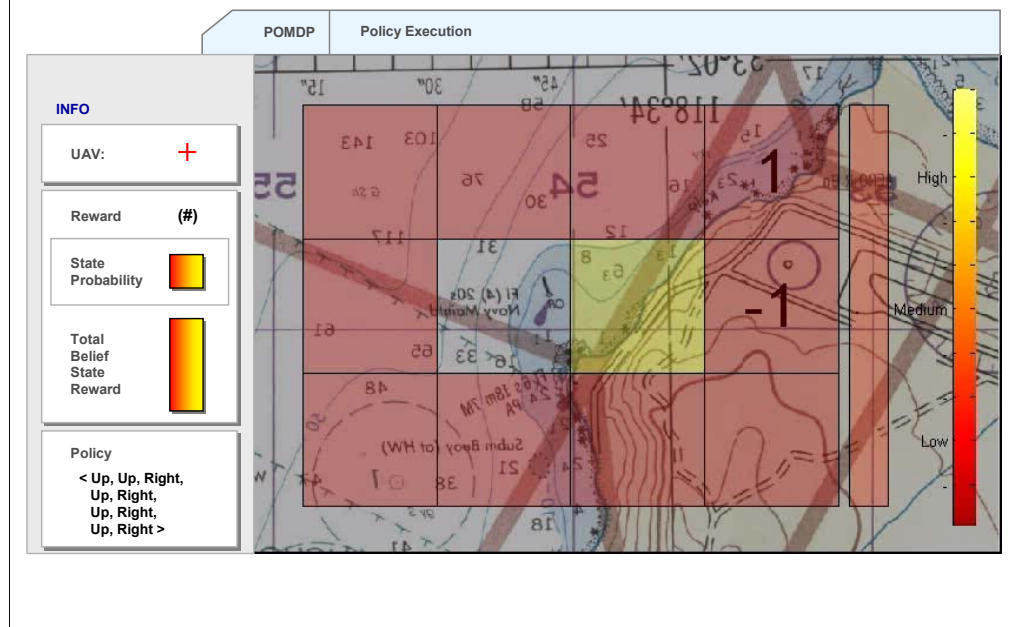
Next we introduce how observations play into POMDPs.

Let's say that if you are in position 8, you can clearly see the coast line.  
Consequently, you gain positional accuracy.

If that is a possibility, then we also have to generate a policy given that observation sequence.

This slide shows that policy being generated.

# POMDP Policy (Coastal Observation)



This slide shows the execution of a policy generated in the last slide.

It used a 3-step look-ahead and an initial coastal observation.

< Up, Up, Right, Up, Right, Up, Right, Up, Right >

Keep in mind that you may not always get the observation. For example, it may be slightly cloudy, which sometimes prevents the observation.

This means that we also need a policy for that observation sequence.

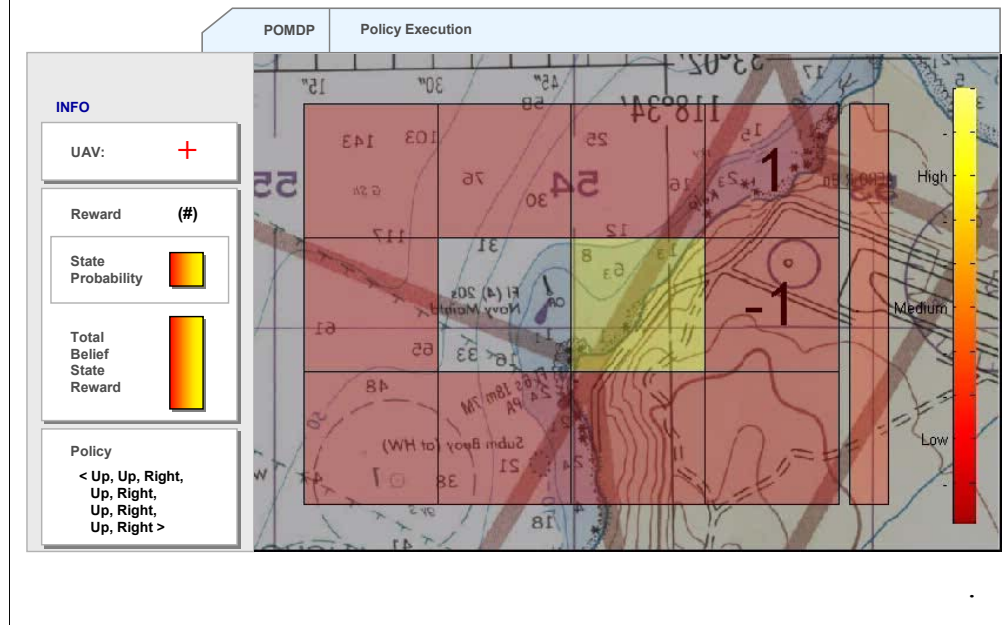
This is why POMDPs are so difficult to solve. Because there are so many belief states, due to positional uncertainty and many observation sequences.

This is why MDPs simply assume positional certainty even though it is an incorrect assumption.

MDPs can use, for example, our expected location (MLE).

This is fine under low uncertainty, however, it causes bad policies when there is high uncertainty.

# POMDP Policy (Coastal Observation)



This slide shows the execution of a policy generated in the last slide.

It used a 3-step look-ahead and an initial coastal observation.

< Up, Up, Right, Up, Right, Up, Right, Up, Right >

Keep in mind that you may not always get the observation. For example, it may be slightly cloudy, which sometimes prevents the observation.

This means that we also need a policy for that observation sequence.

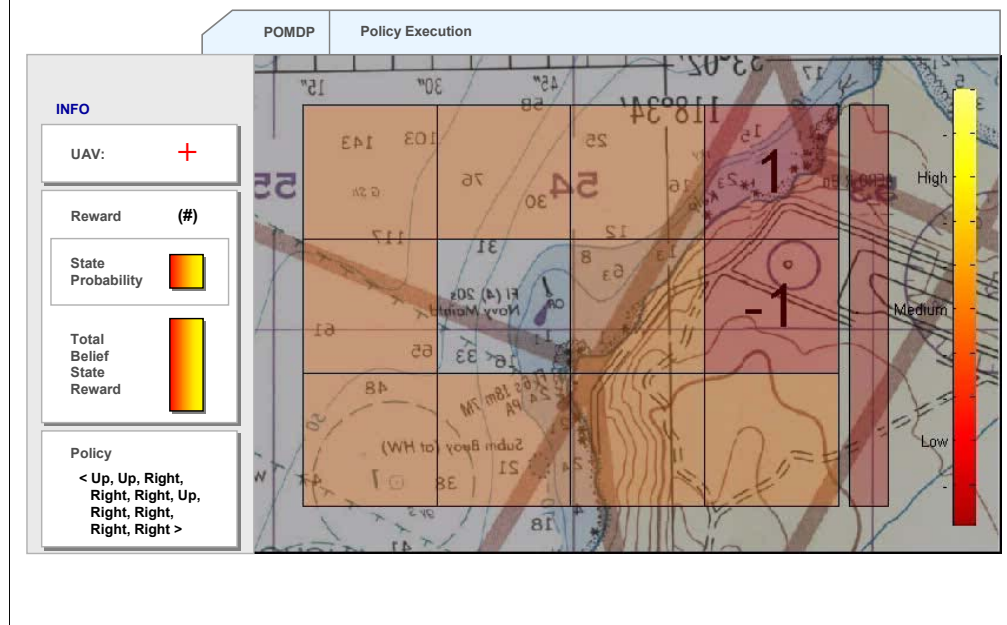
This is why POMDPs are so difficult to solve. Because there are so many belief states, due to positional uncertainty and many observation sequences.

This is why MDPs simply assume positional certainty even though it is an incorrect assumption.

MDPs can use, for example, our expected location (MLE).

This is fine under low uncertainty, however, it causes bad policies when there is high uncertainty.

# POMDP Policy (Fuel Observation)



Next we introduce another type of observation.

Let's say you have a fuel gauge and it is telling us that we are low on fuel. This will cause us to take a shorter route home, even if it has additional risk.

The policy generated here assumes we are low on fuel, which makes each action (relatively) more costly.

This causes the policy to be more risky, we may land on the take-off runway. However, our overall risk is minimized.

This slide shows that policy in action.

< Up, Up, Right, Right, Right, Up, Right, Right, Right, Right >

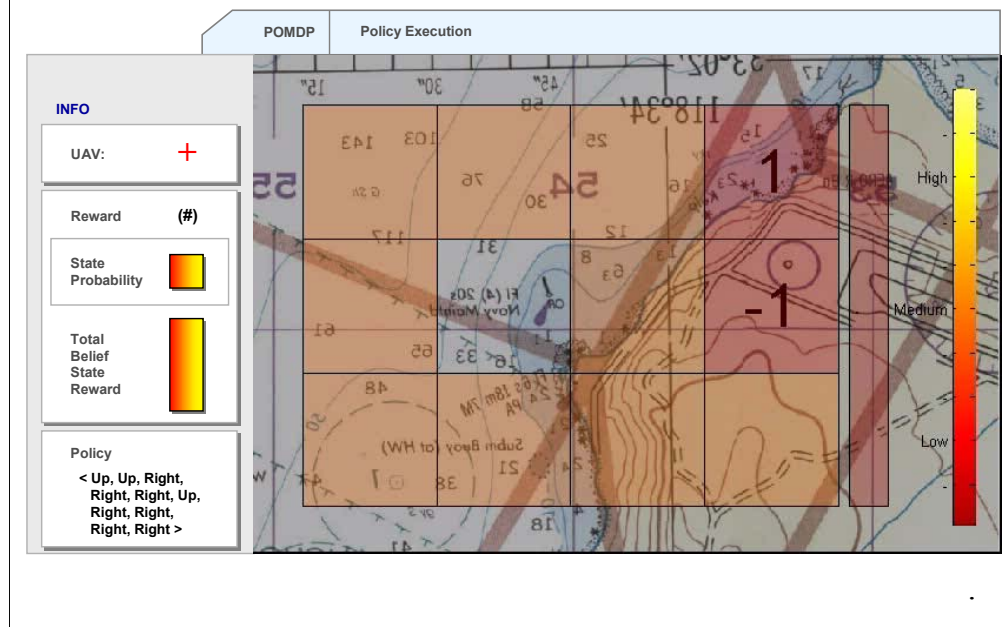
Tractability.

To summarize,

POMDPs are difficult to solve due to the combination of the belief space size, the the combination of observations.



# POMDP Policy (Fuel Observation)



Next we introduce another type of observation.

Let's say you have a fuel gauge and it is telling us that we are low on fuel. This will cause us to take a shorter route home, even if it has additional risk.

The policy generated here assumes we are low on fuel, which makes each action (relatively) more costly.

This causes the policy to be more risky, we may land on the take-off runway. However, our overall risk is minimized.

This slide shows that policy in action.

< Up, Up, Right, Right, Right, Up, Right, Right, Right, Right >

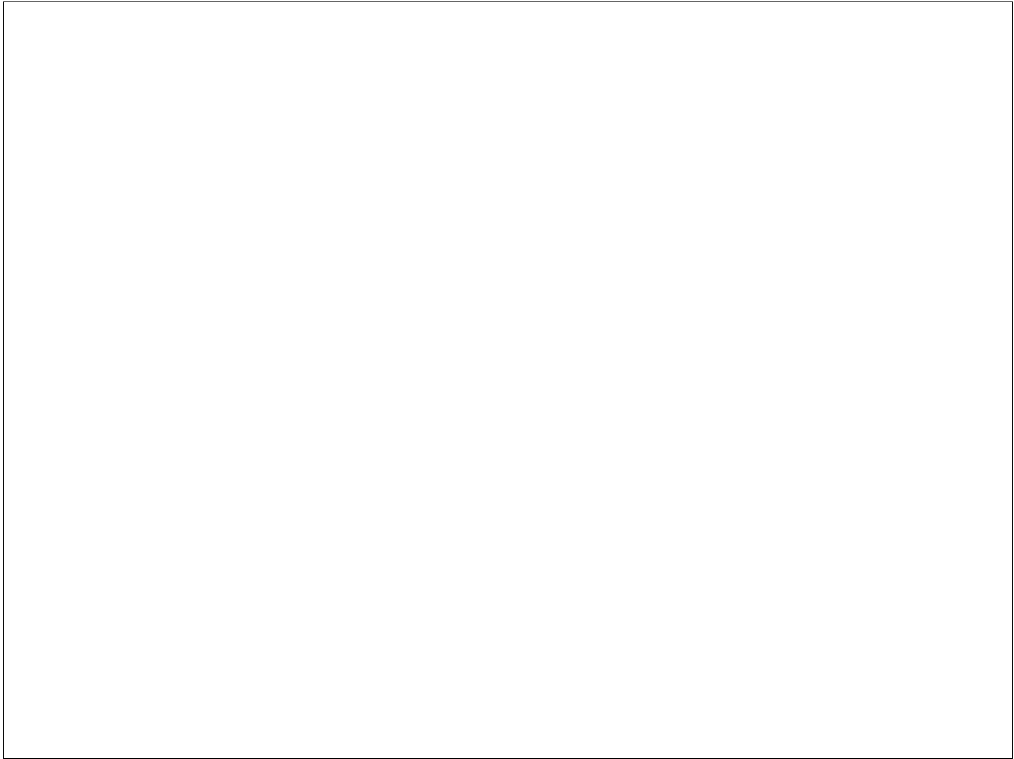
Tractability.

To summarize,

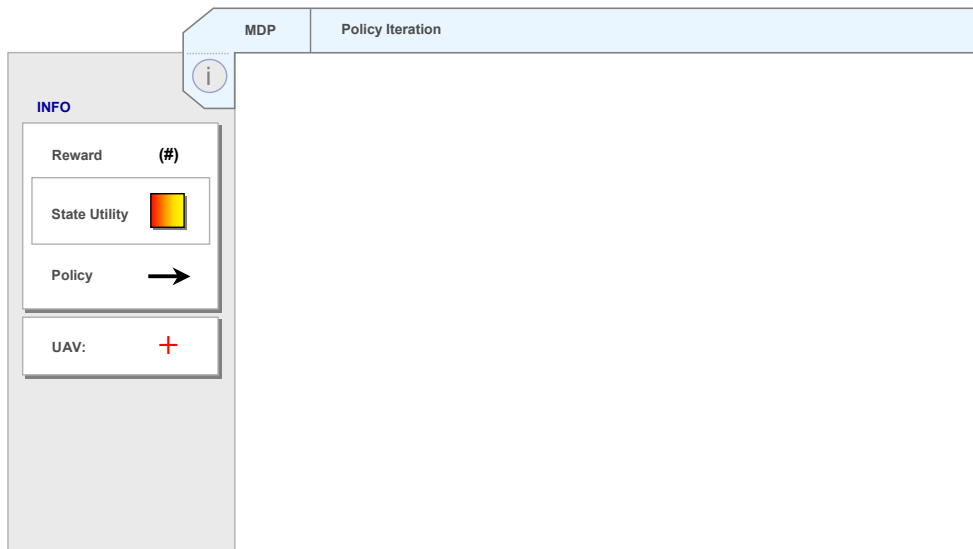
POMDPs are difficult to solve due to the combination of the belief space size, the the combination of observations.



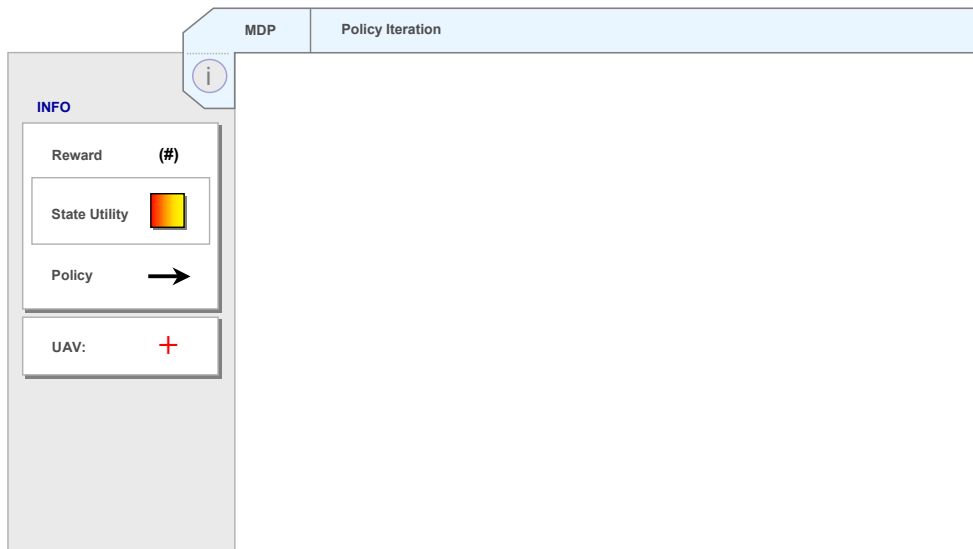
# Conclusion



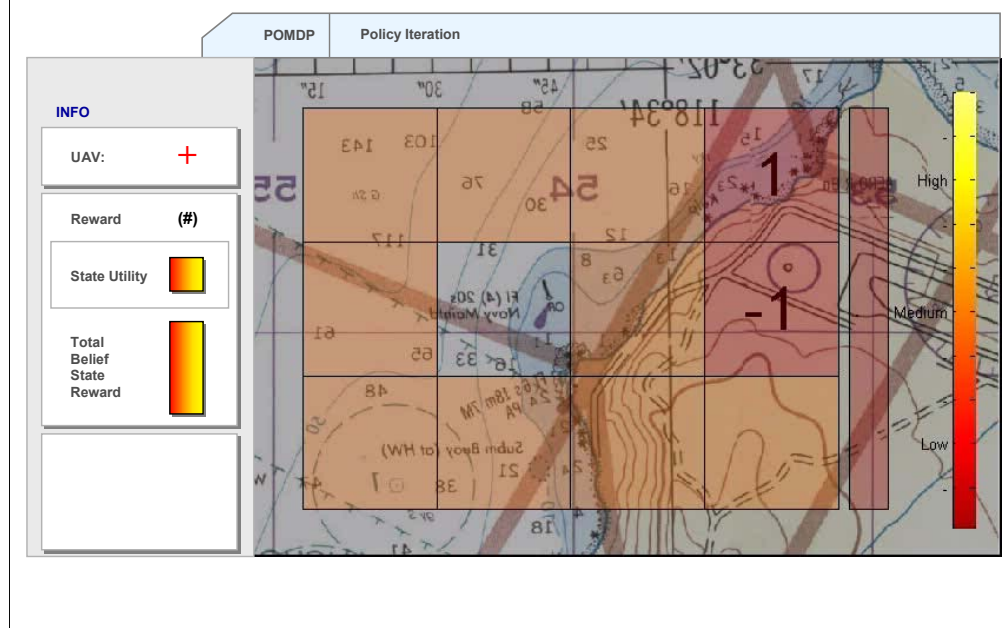
# MDP Value Iteration



# MDP Value Iteration



# POMDP Policy (Fuel Observation)



Next we introduce another type of observation.

Let's say you have a fuel gauge and it is telling us that we are low on fuel. This will cause us to take a shorter route home, even if it has additional risk.

The policy generated here assumes we are low on fuel, which makes each action (relatively) more costly.

This causes the policy to be more risky, we may land on the take-off runway. However, our overall risk is minimized.

This slide shows that policy in action.

< Up, Up, Right, Right, Right, Up, Right, Right, Right >

Tractability.

To summarize,

POMDPs are difficult to solve due to the combination of the belief space size, the the combination of observations.

# Problem : Utility Overlay

