# INFO8006: Project 3 - Report

**BARE Alexandre - s172388**
**RANSY Bastien - s174413**

December 8, 2019

## 1 Bayes filter

$E_t$ denotes the set of observable evidence variables at time t. It contains for each ghost, an approximate value of the manhattan distance between the ghost and pacman.
$X_t$ denotes the set of unobservable states at time t. It contains for each ghost its position $(x_{ghost}, y_{ghost})$.

a. The sensor model $(B = P(E_t|X_t))$ makes a first-order sensor Markov assumption: $P(E_t|X_{0:t}, E_{0:t-1}) = P(E_t|X_t)$. For each $E_t = e_t$, $P(E_t = e_t|X_t)$ corresponds to a random value of the normal distribution, where the mean is the true manhattan distance from the ghost to pacman and the variance is given. The sensor model also makes a stationarity assumption, leading to a sensor model that does not change over time.

b. As can be seen in the transition model $(T = P(X_t|X_{t-1}))$, we have a first-order Markov process: $P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$. $P(X_t|X_{t-1})$ is a list of matrices of size $M \times N$ (respectively the maze width and height), where each matrix of the list corresponds to a ghost and where each element $(l+k\times N, j+i\times N)$ of a matrix gives the probability of the corresponding ghost's presence at position (i, j), knowing its previous position (k, l). Therefore, $T$ is a list of matrices where $g$ is the $ghost\_id - 1$, $l + k \times N$ and $j + i \times N$ are the column and row indexes of the matrix $g$.

$$T_{g,l+k\times N,j+i\times N} = \begin{cases} 0 \text{ if } l + k \times N = j + i \times N \\ a \text{ otherwise} \end{cases}$$

where

$$a = \begin{cases} 2^{power} \text{ if } |i + j - k - l| == 1 \text{ and } i \geq 0 \text{ and } j \geq 0 \text{ and } i < M \text{ and } j < N \text{ and} \\ \text{ there's no wall at the initial position (k, l) and at the next position (i, j),} \\ \text{ i.e. the ghost makes a move of cost 1 from and to a legal position} \\ 0 \text{ otherwise} \end{cases}$$

where

$power = 0$ if $(i+j) - (x_{pacman} + y_{pacman}) < (k+l) - (x_{pacman} + y_{pacman})$

Otherwise,

$$power = \begin{cases} 0 \text{ if } ghost\_type = "confused" \\ 1 \text{ if } ghost\_type = "afraid" \\ 3 \text{ if } ghost\_type = "scared" \end{cases}$$

At the end of the computation of each row of the matrix, we also normalise the row by dividing each of its element by the sum of elements of the row The single free parameter is thus the ghost_type.

# 2 Implementation

a. See `bayesfilter.py`.

# 3 Experiment

a. The first measure, the uncertainty on the belief state, is obtained by summing for each ghost and for each position (i, j) of the grid, the probability of (i, j) not being the right position of the ghost according to the Bayes filter which is computed from pacman's belief state. We then divide the result by the number of cells of the grid and the number of ghosts. Uncertainty is included in the interval [0; 1]. 0 corresponds to no uncertainty : there is only one possible position for each ghost. 1 corresponds to the situation where pacman has no clue on the ghosts' position.

b. The second measure, the quality of the belief state(s), is obtained the following way :
For each ghost, the mean manhattan distance between him and each estimation of its position (each has as weight the probability of the estimation) is computed. They are all summed and then, divided by the maximum manhattan distance between a ghost and pacman (M + N - 2), and by the number of ghosts. We finally get a value in [0; 1]. We then do 1 minus this value to get the quality. Therefore, 0 corresponds to the worst quality while 1 is the best quality.

c. cfr. Figures 1 and 2 :

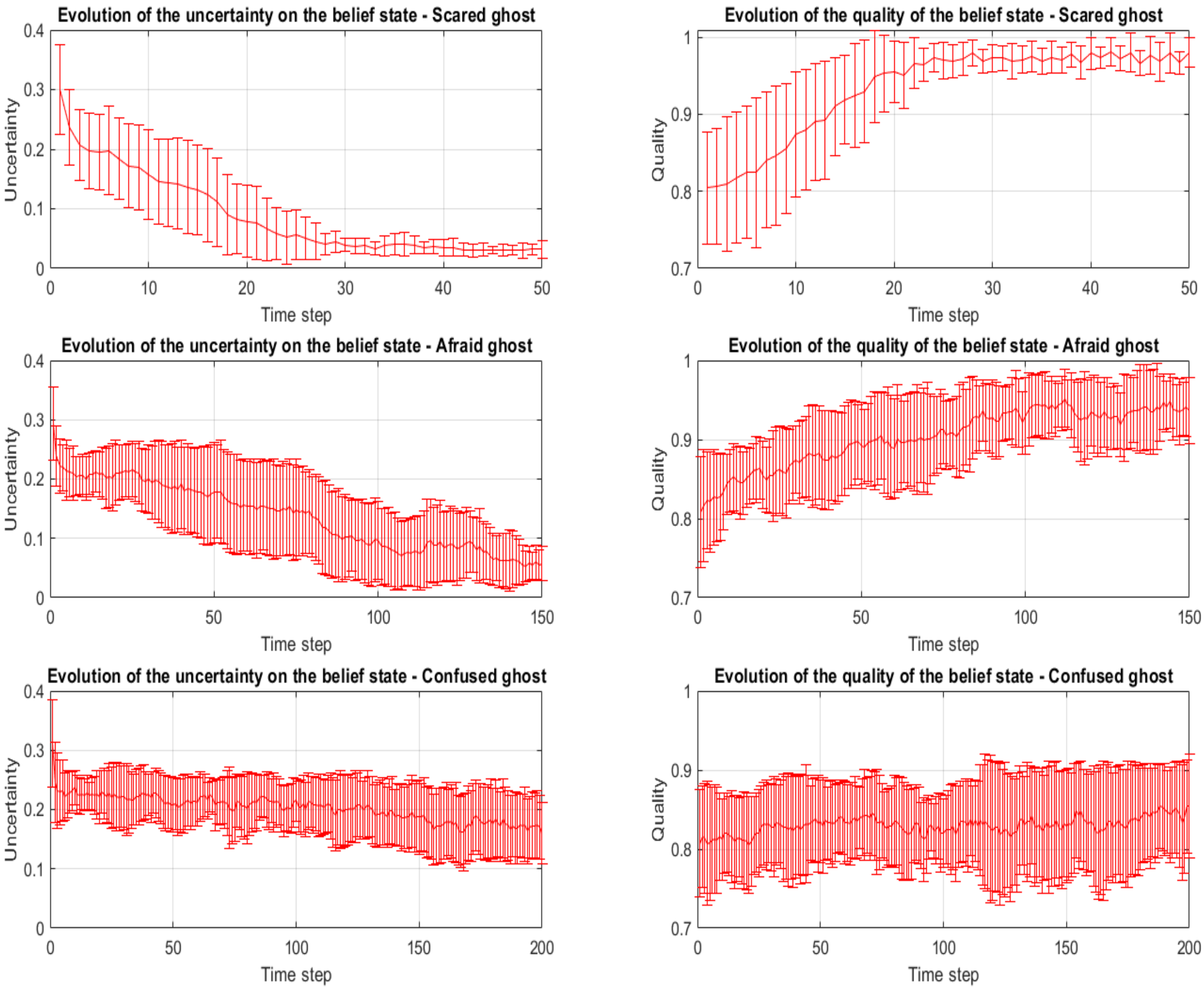


Figure 1: Comparison of the performances against the large_filter layout

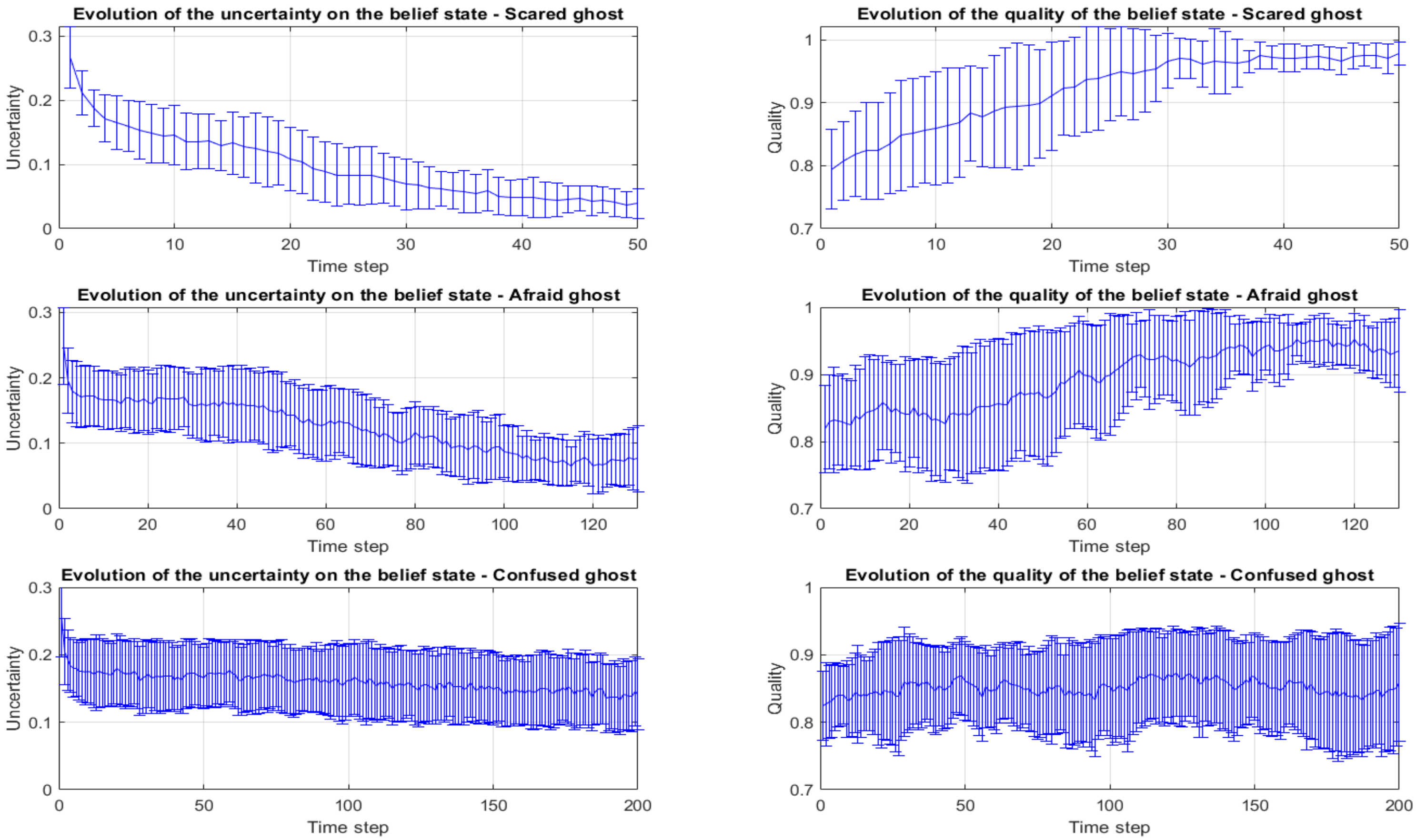


Figure 2: Comparison of the performances against the large_filter_walls layout

d. As said before, the ghost transition model parameter is the ghost_type. It can have 3 different values: "confused", "afraid" or "scared". They are placed in increasing order of information. The general "rule" would be: the more information pacman has, the faster the convergence, the better the quality of the belief states and the lower the uncertainty over time. But this affirmation must be nuanced. If the ghosts are "confused", the probability for them to move in any of the different directions (4 directions if there is no wall for each adjacent cell) is the same. Therefore, the filter can not converge to a precise uncertainty area. However, converging to a large uncertainty area requires less (convergence) time than to a small one. For "afraid" and even more for "scared", there is a higher probability that the ghosts will move in the direction(s) that draw them away from pacman. So, the order in convergence time is: "afraid" > "scared" > "confused". For "afraid" and "scared", the ghosts will progressively isolate themselves in corners (different corners for the large_filter_walls but only the furthest corner from pacman for the other layout) where there are less possibilities to move, and especially none to get further from pacman in one movement. The ghosts will then get stuck to small uncertainty areas. From the highest to the smallest uncertainty area at the convergence, we have "confused" > "afraid" > "scared". In the large_filter_walls, the convergence will be, in general, a bit slower because each ghost will often have a longer path (due to more walls that must be bypassed) to the corner to which they will get stuck at the convergence. The uncertainty and quality measures suffer from a higher standard deviation in the latter layout as each ghost can get stuck in different corners, thus increasing the number of possibilities for the ghost's position that are not closed to the real one.

e. The higher the sensor variance, the wider the uncertainty areas of the filter for each ghost. An increase of the variance thus decreases the quality of the belief states and increases the uncertainty measure.

f. As the ghosts' position are only known through a probability distribution held in the belief state, the environment is partially observable. We can use a partially observable Markov process decision (POMPD) that would maintain the belief state $b$ with a Bayes filter that takes the action of pacman into account. To reduce the POMPD to a Markov decision process (MDP), we would have to define a transition model

$$P(b'|b,a) = \sum_{e} P(b'|b,a,e) \sum_{s'} P(e'|s') \sum_{s} P(s'|s,a)b(s)$$

where

$$b^{'} = \alpha \times P(e|S') \times \sum_{s} P(S^{'}|s,a)b(s) = \alpha \times forward(b,a,e)$$

and

$$P(b^{'}|b,a,e) = \begin{cases} 1 \text{ if } b^{'} = forward(b,a,e) \\ 0 \text{ otherwise} \end{cases}$$

We would also have to define a reward function of the belief state $\rho(b) = \sum_{s} b(s)R(s)$.

We would finally solve our MDP with the policy iteration algorithm to define a policy mapping each belief state to an action. $e$ is the evidence (pacman's position), $a$ is a legal action, $s$ is the current (partially observed) state, $s^{'}$ is the next (partially observed) state, $S$ and $S^{'}$ are the current and next set of (partially observed) states, $\alpha$ is a normalisation constant and $R(s)$ is the reward function of the state s.