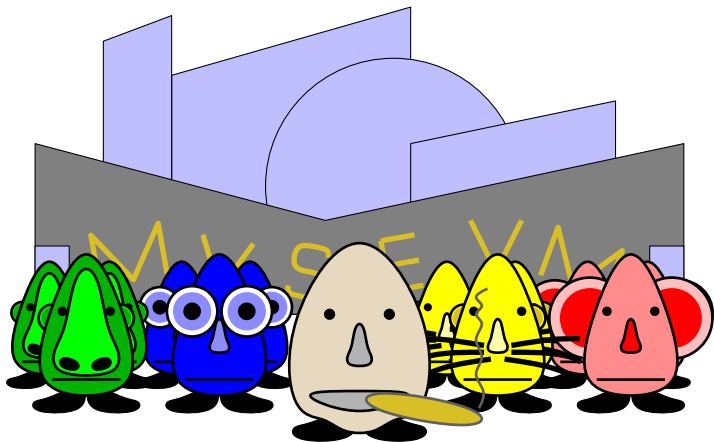
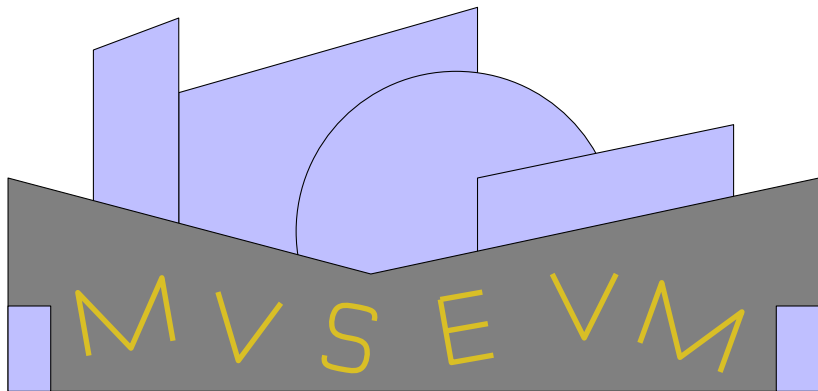
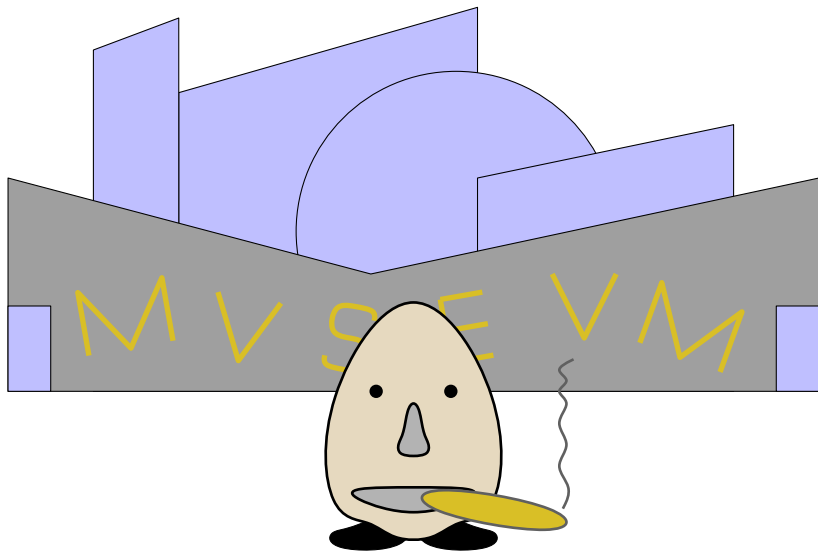


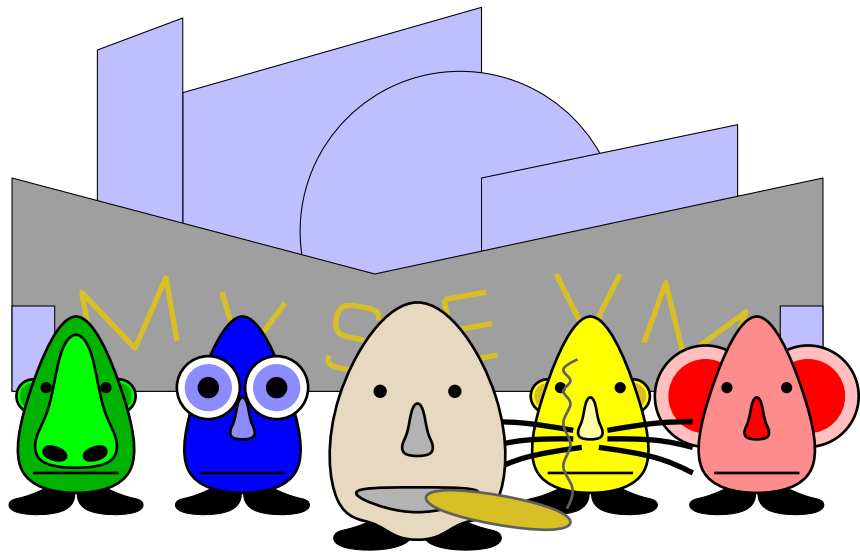
## colour patterns for polychromatic four-colourings of rectangular subdivisions

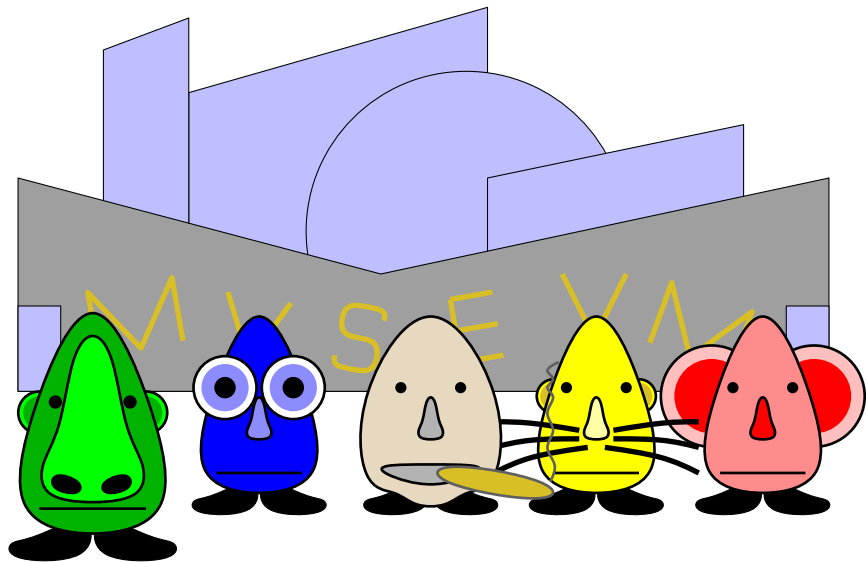


Herman Haverkort ◦ Maarten Löffler ◦ Elena Mumford  
Matthew O'Meara ◦ Jack Snoeyink ◦ Bettina Speckmann

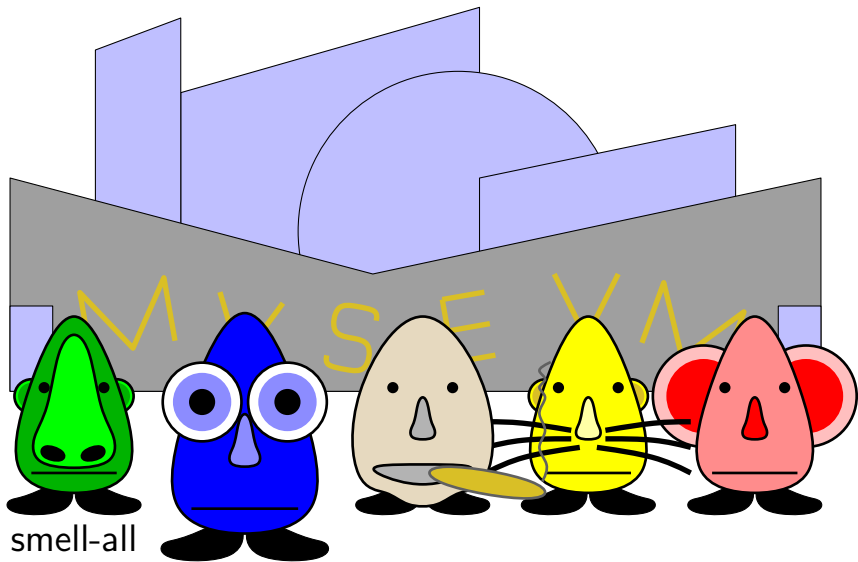






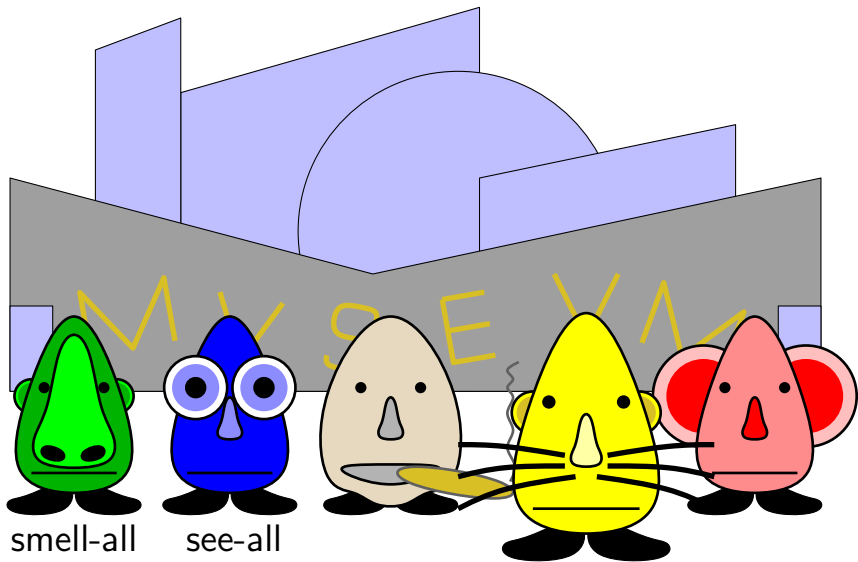


smell-all



smell-all

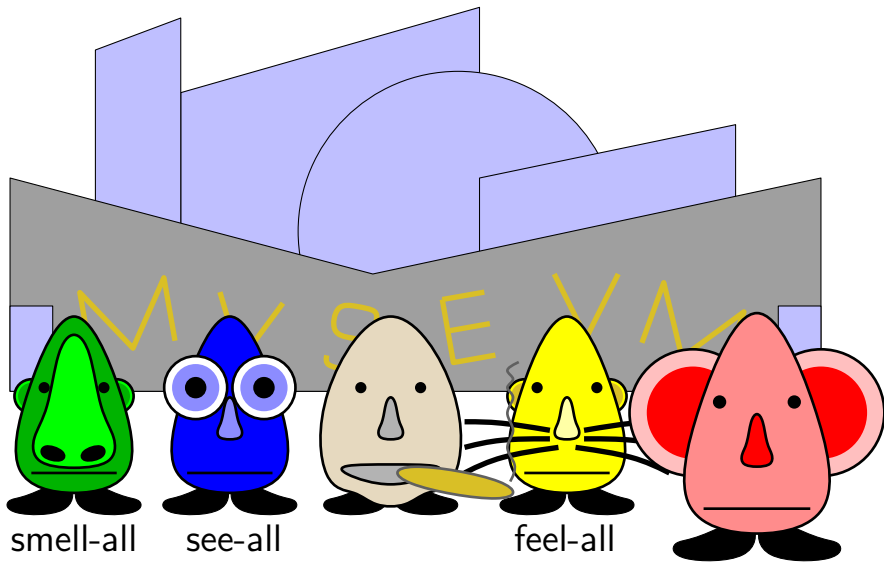
see-all



smell-all

see-all

feel-all



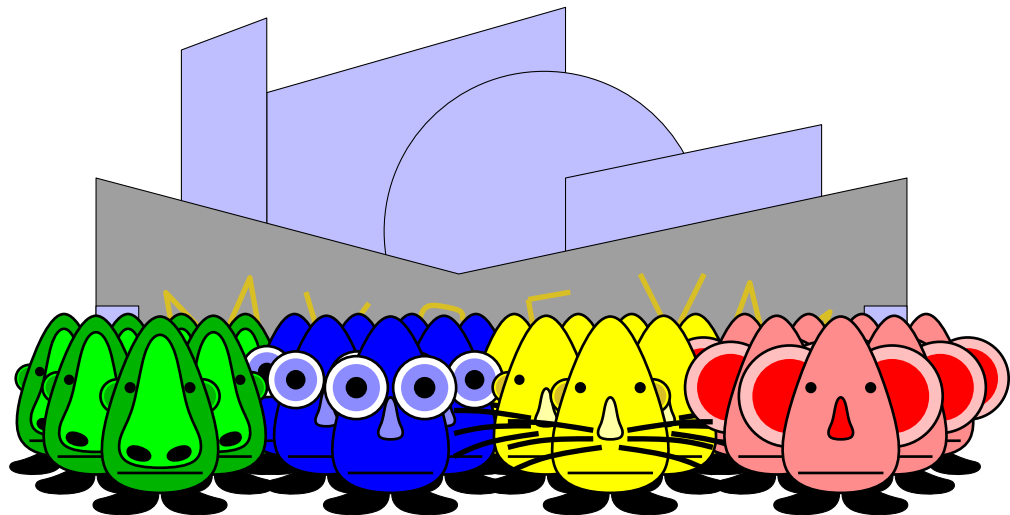
smell-all

see-all

feel-all

hear-all





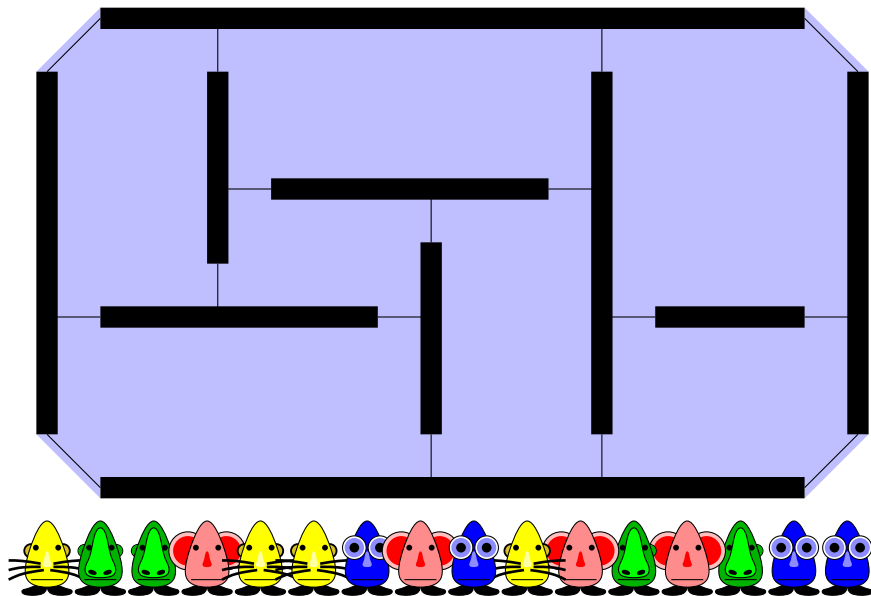
smell-all

see-all

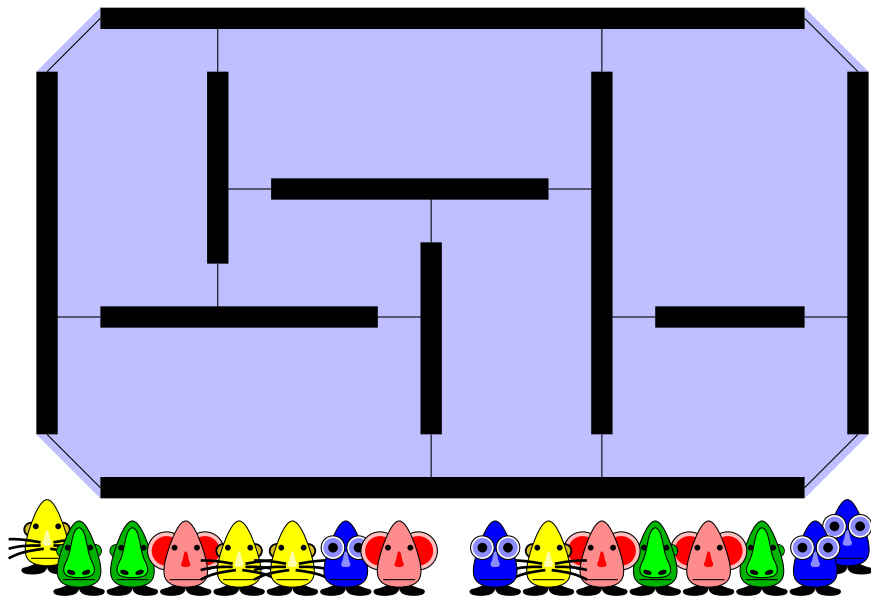
feel-all

hear-all

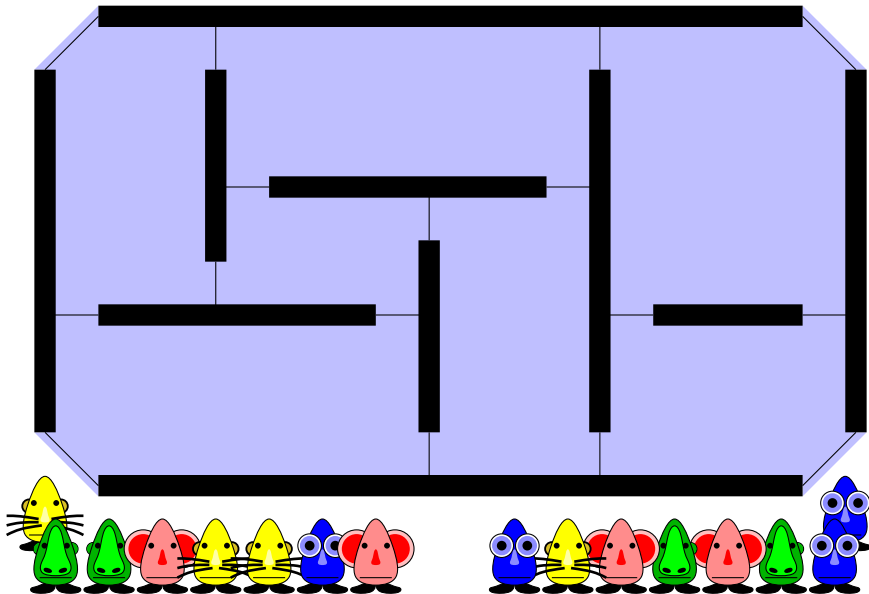
All rooms are rectangles with four doors, one in every corner



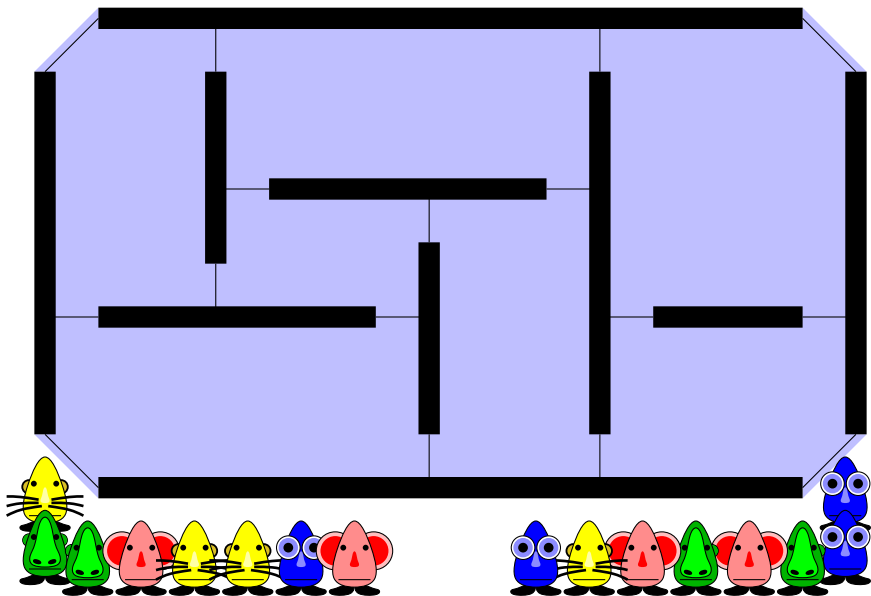
All rooms are rectangles with four doors, one in every corner



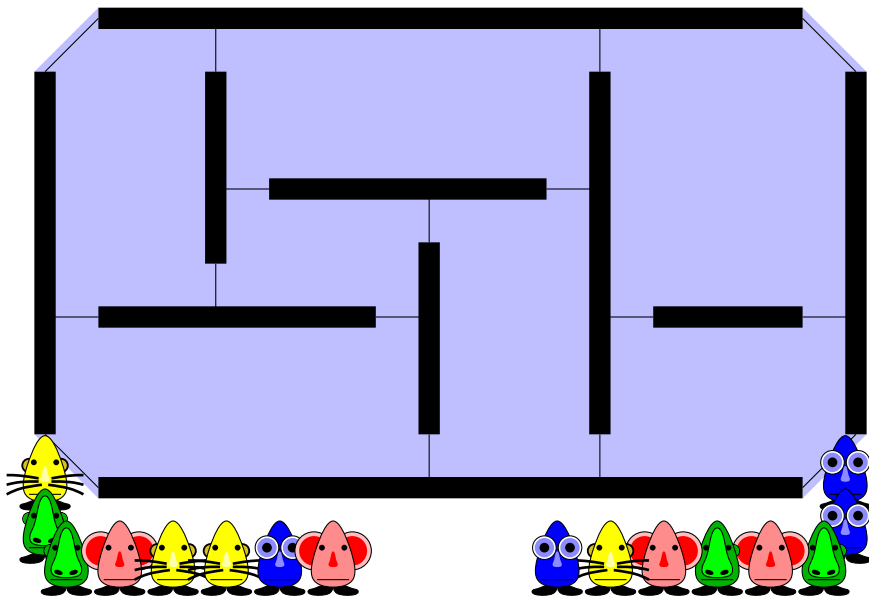
All rooms are rectangles with four doors, one in every corner



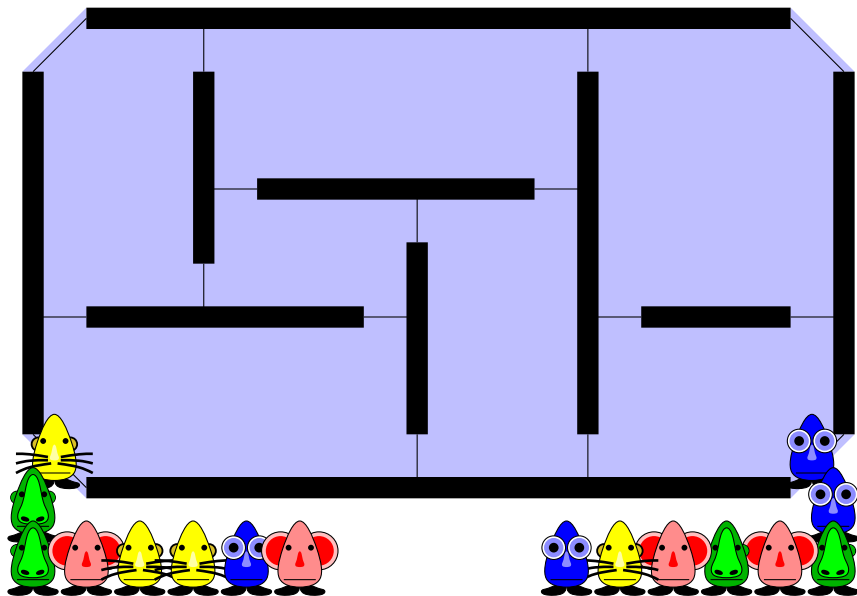
All rooms are rectangles with four doors, one in every corner



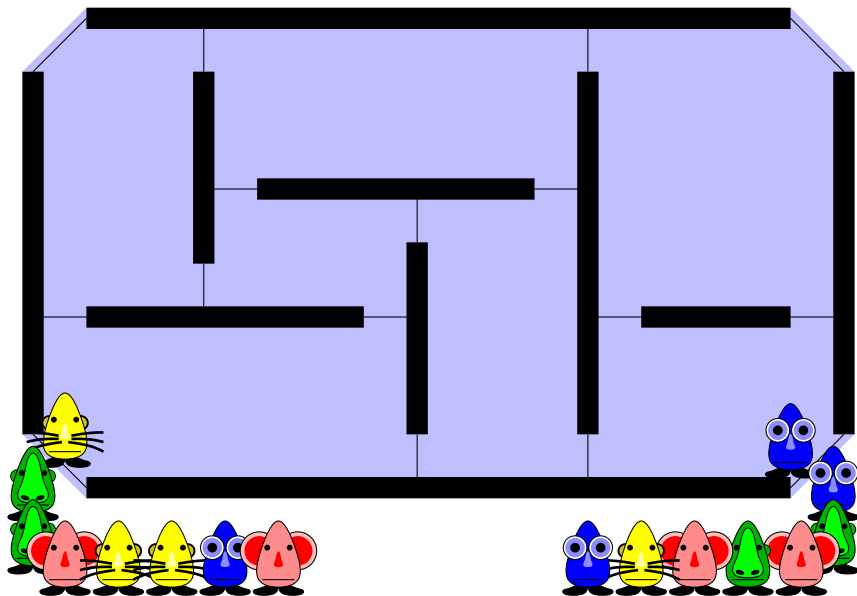
All rooms are rectangles with four doors, one in every corner



All rooms are rectangles with four doors, one in every corner

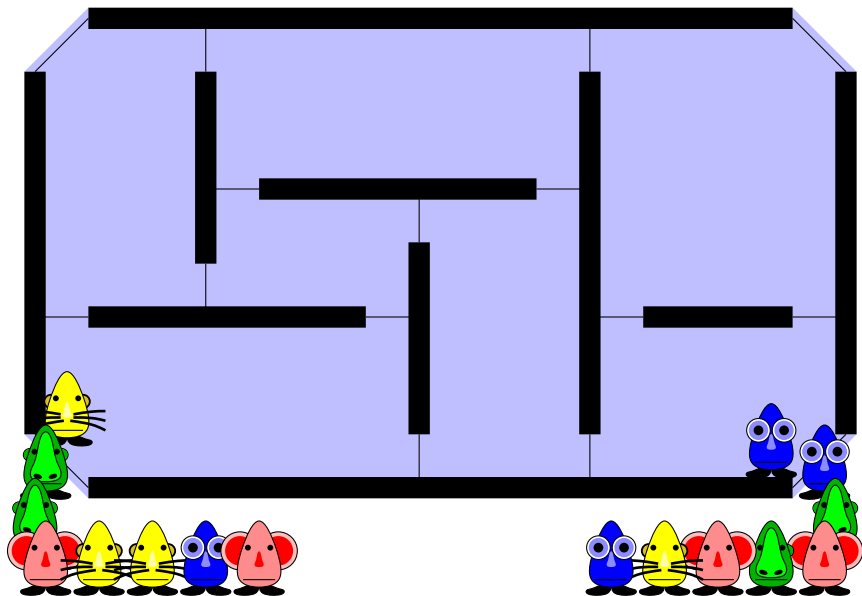


All rooms are rectangles with four doors, one in every corner

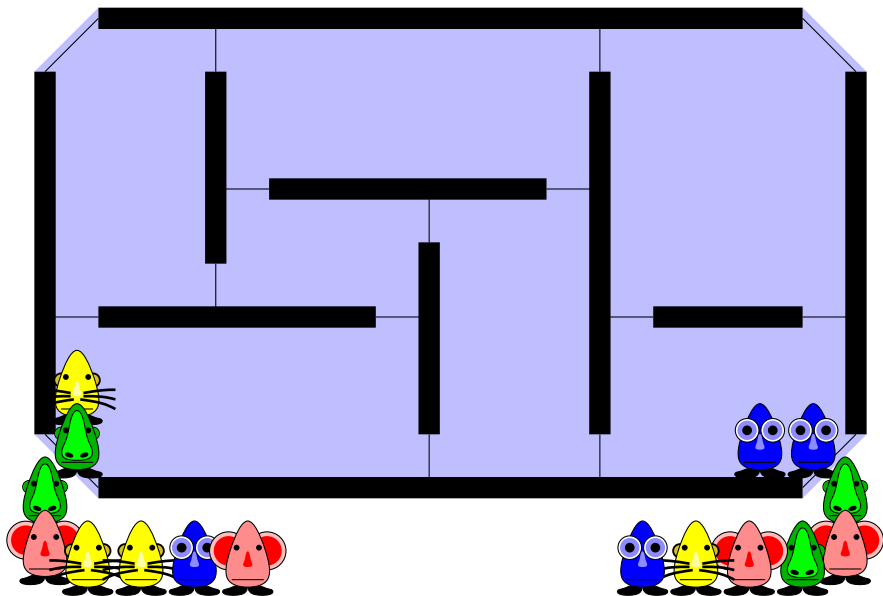




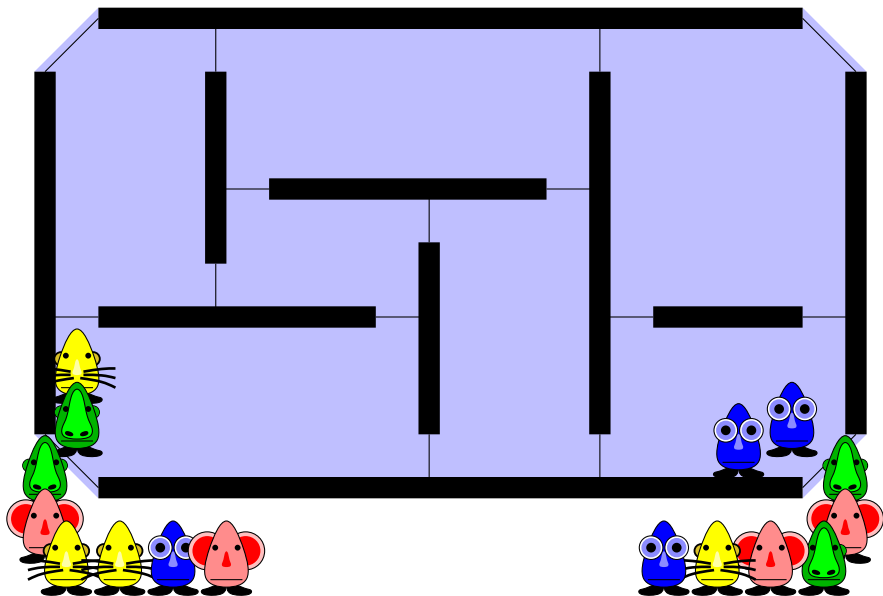
All rooms are rectangles with four doors, one in every corner



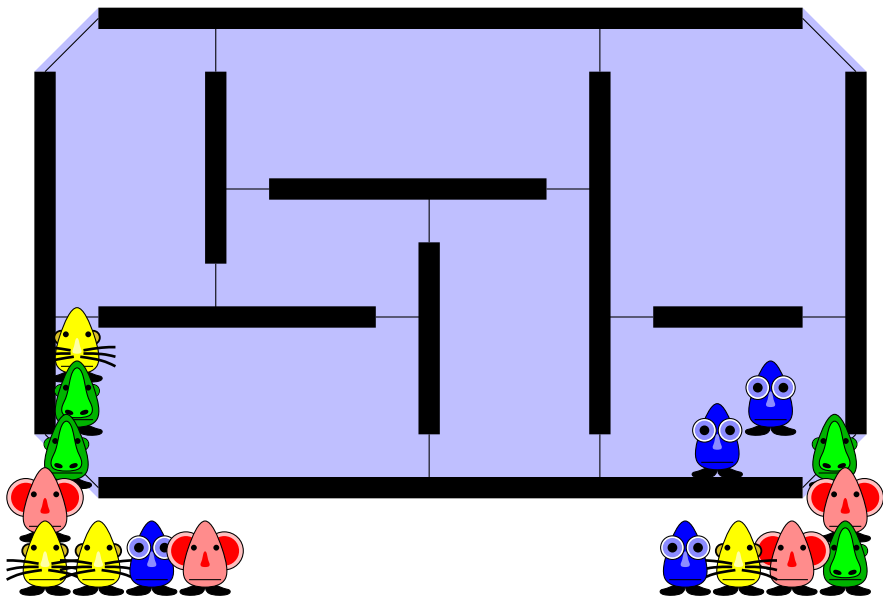
All rooms are rectangles with four doors, one in every corner



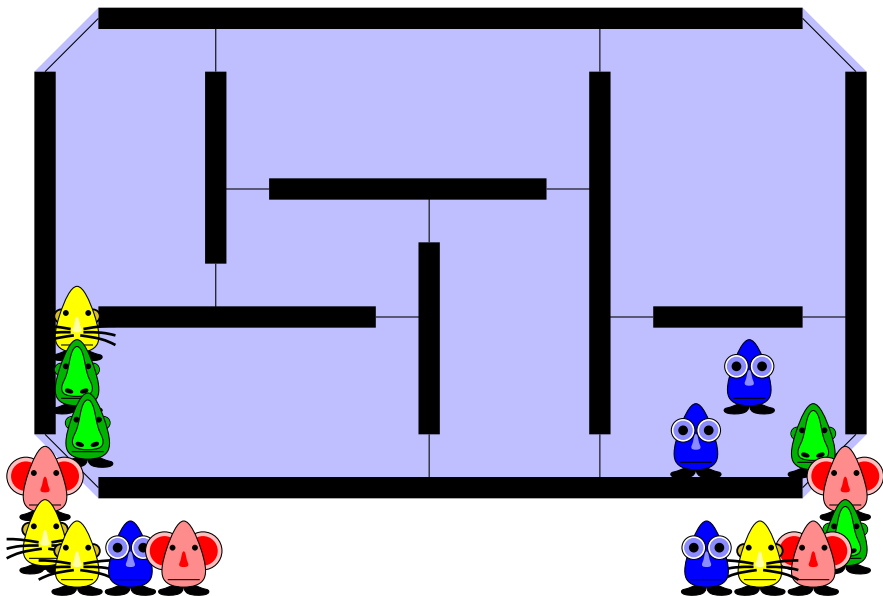
All rooms are rectangles with four doors, one in every corner



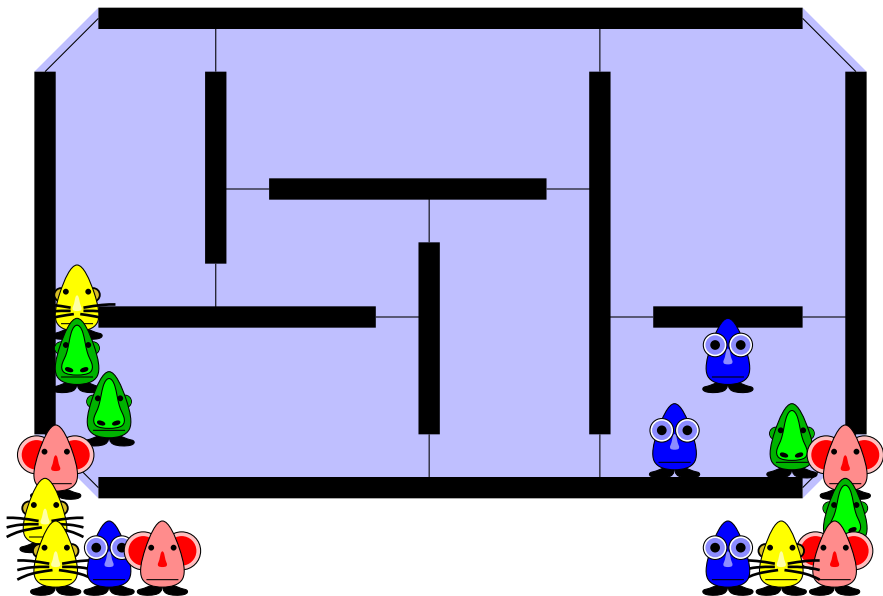
All rooms are rectangles with four doors, one in every corner



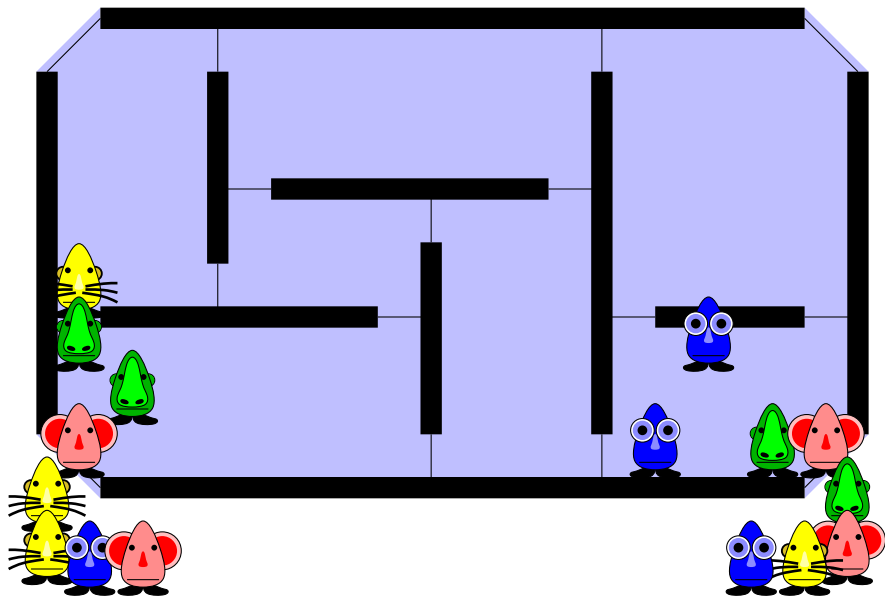
All rooms are rectangles with four doors, one in every corner



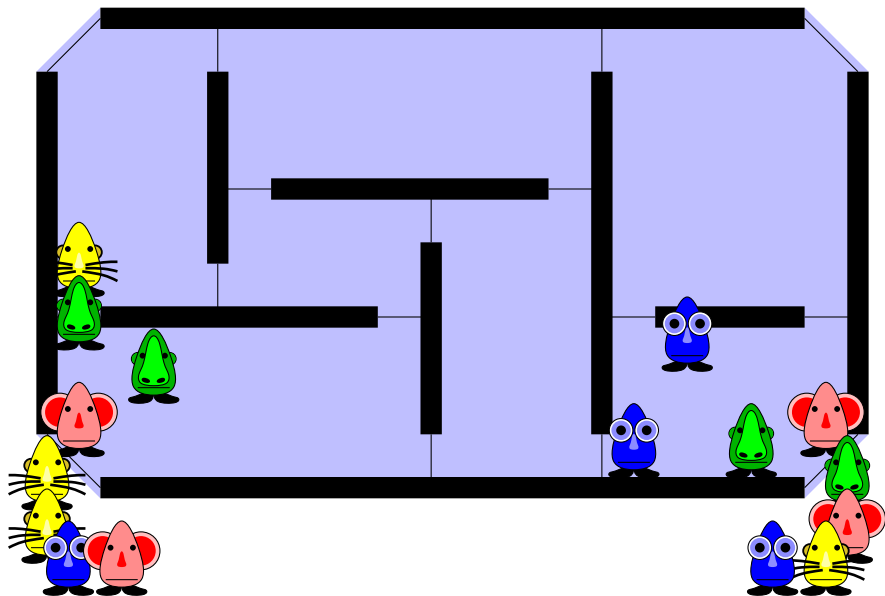
All rooms are rectangles with four doors, one in every corner



All rooms are rectangles with four doors, one in every corner

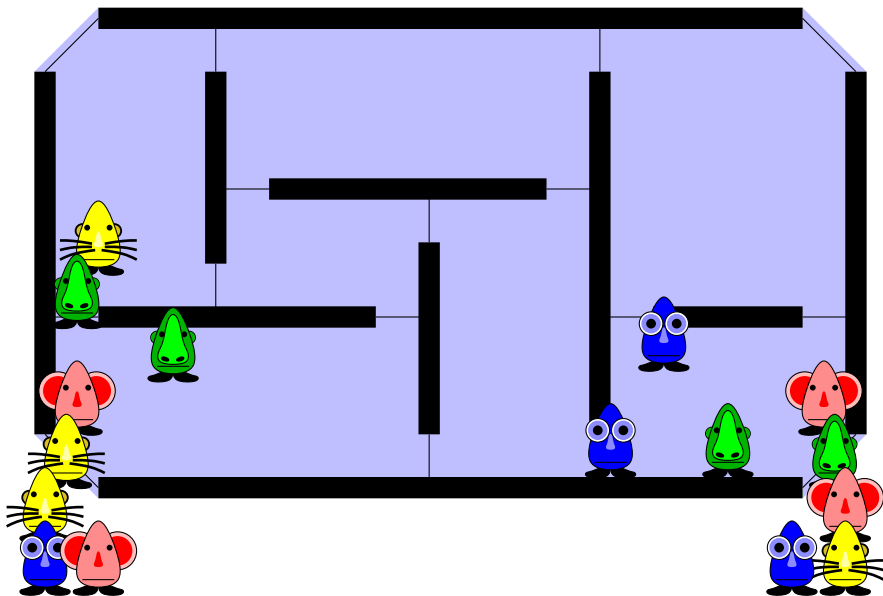


All rooms are rectangles with four doors, one in every corner

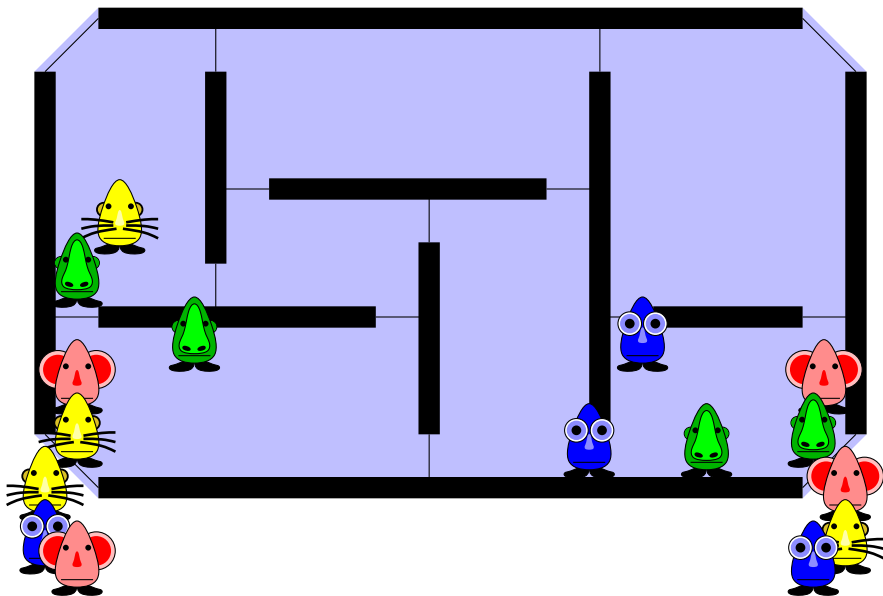




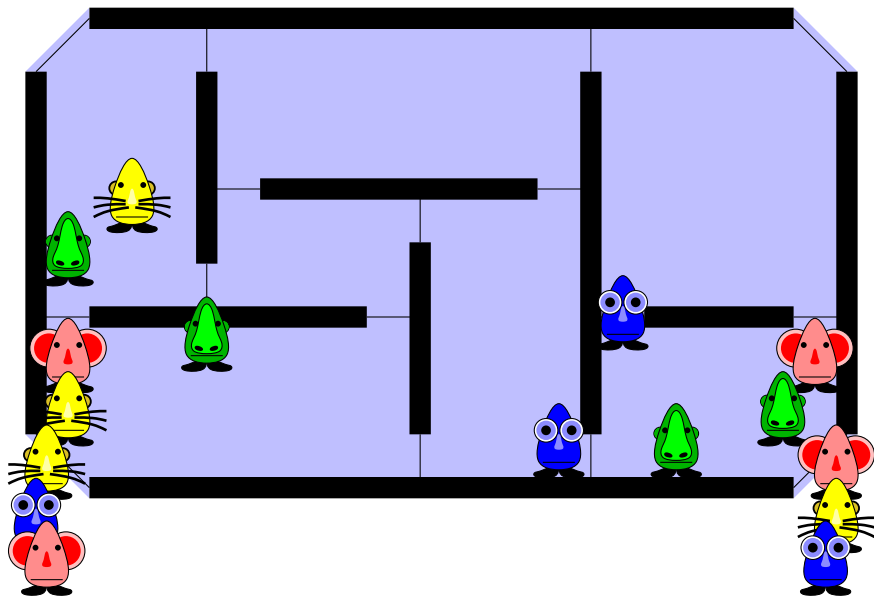
All rooms are rectangles with four doors, one in every corner



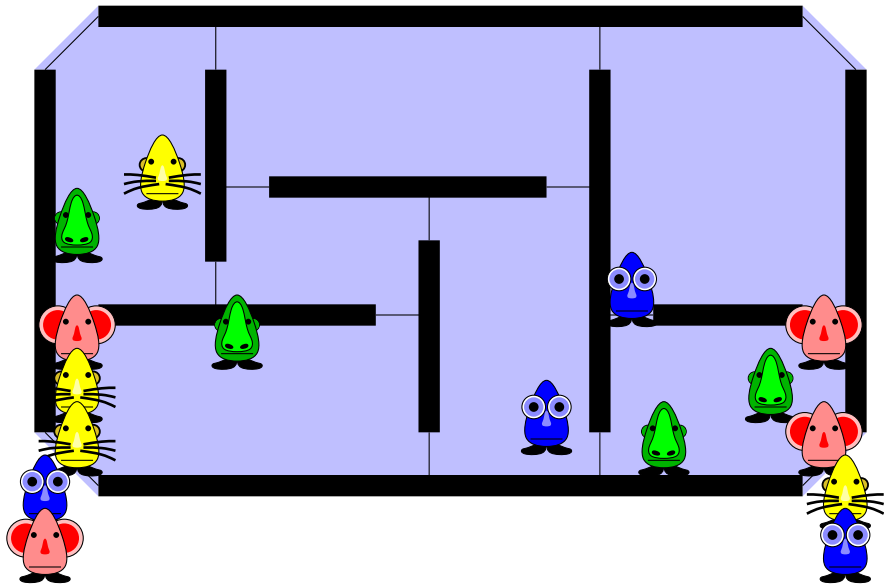
All rooms are rectangles with four doors, one in every corner



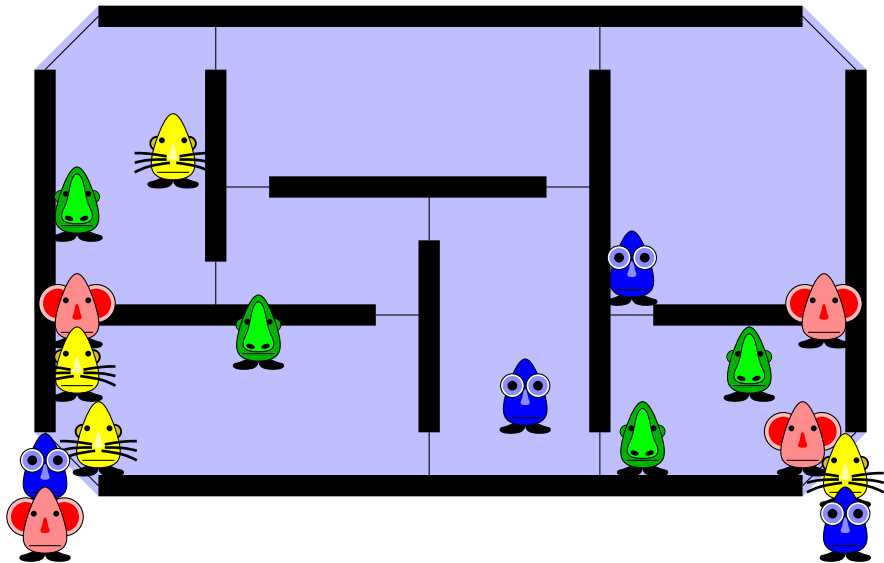
All rooms are rectangles with four doors, one in every corner



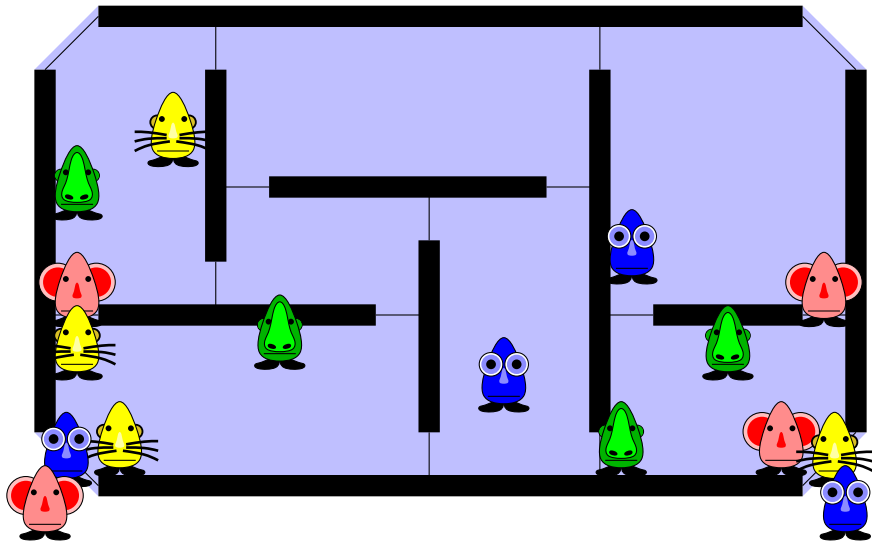
All rooms are rectangles with four doors, one in every corner



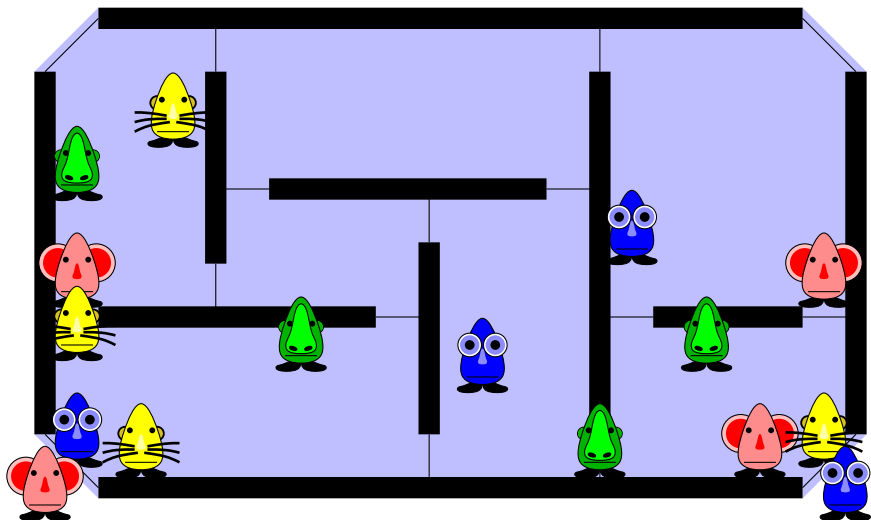
All rooms are rectangles with four doors, one in every corner



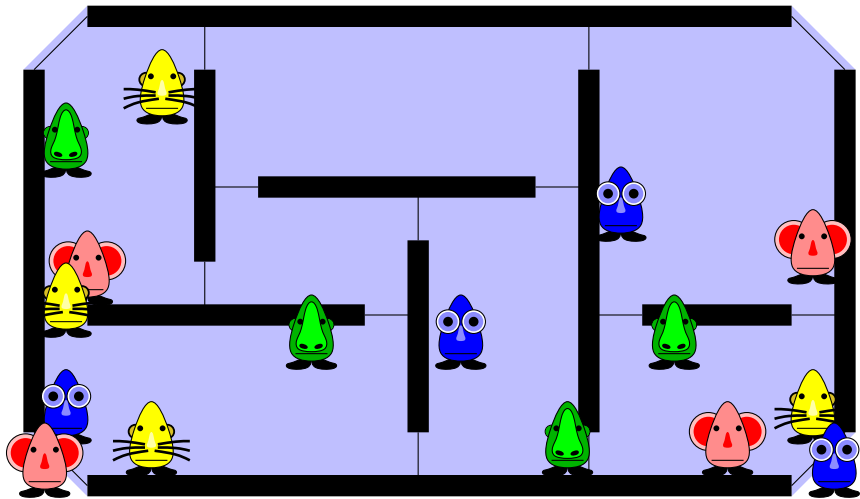
All rooms are rectangles with four doors, one in every corner



All rooms are rectangles with four doors, one in every corner

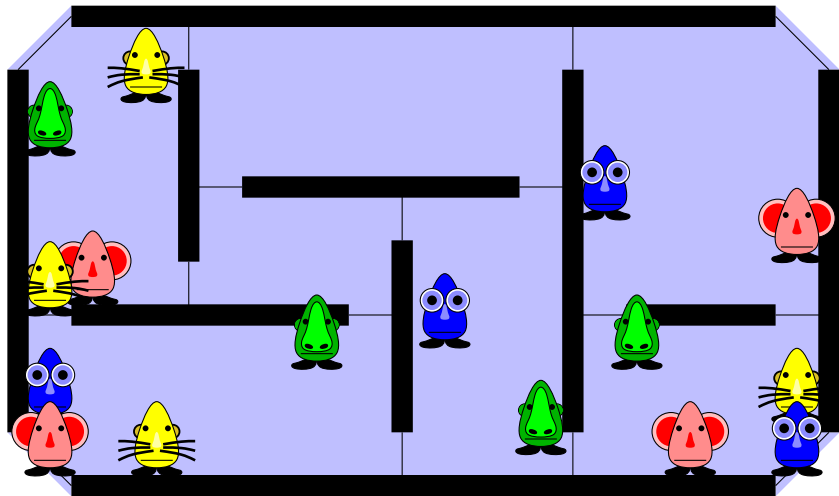


All rooms are rectangles with four doors, one in every corner

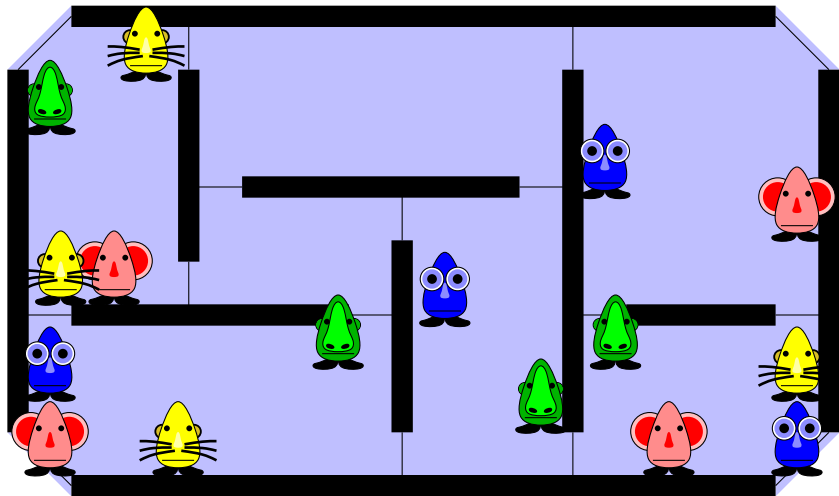




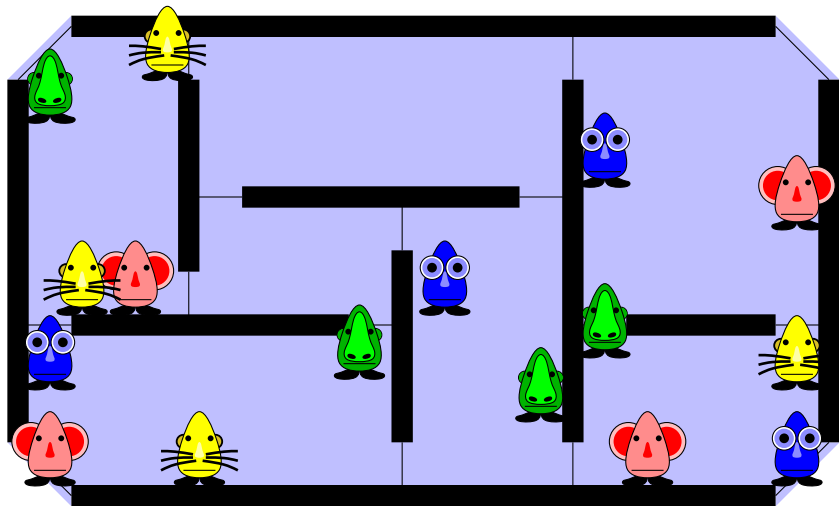
All rooms are rectangles with four doors, one in every corner



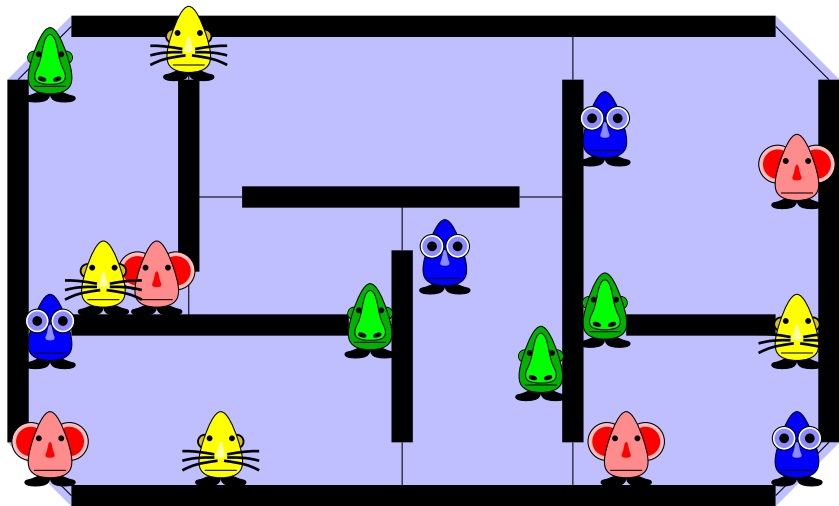
All rooms are rectangles with four doors, one in every corner



All rooms are rectangles with four doors, one in every corner

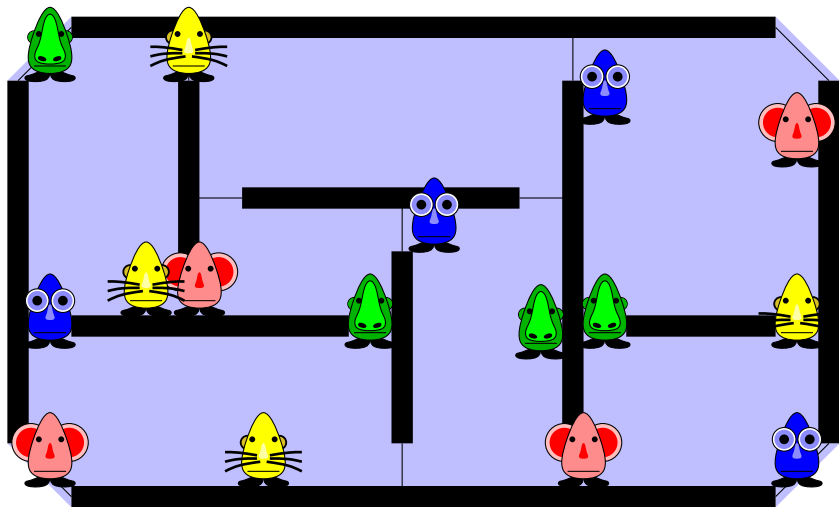


All rooms are rectangles with four doors, one in every corner

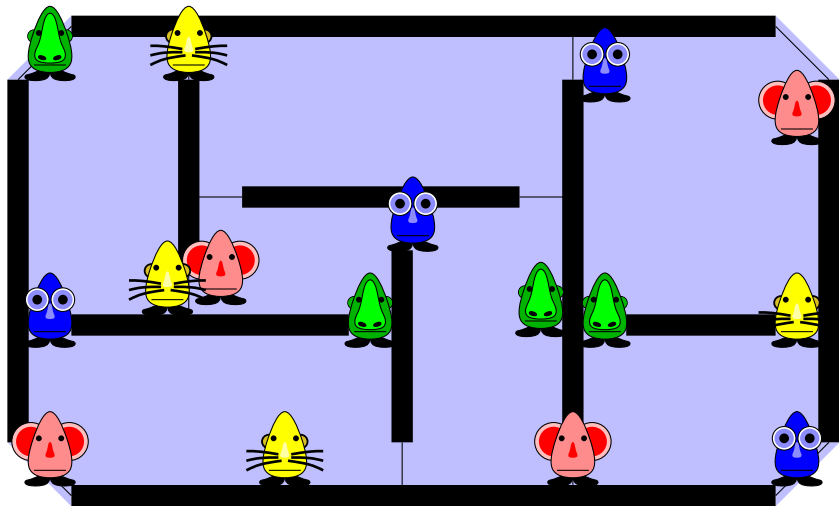




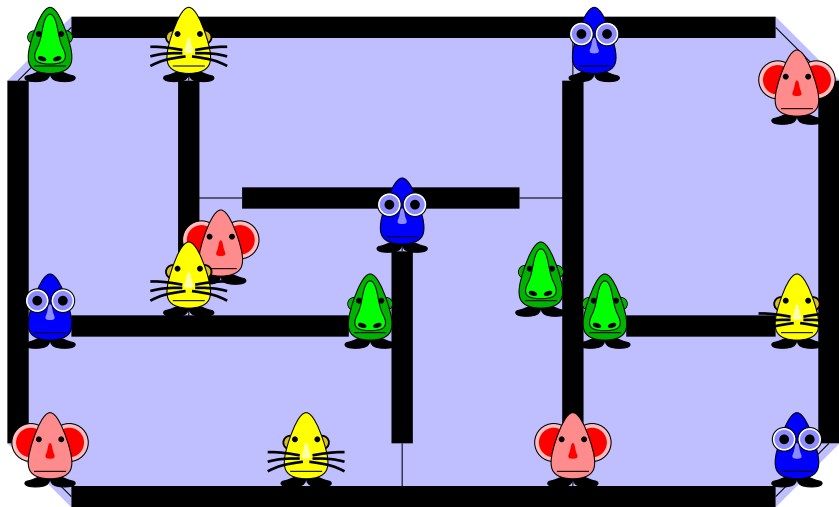
All rooms are rectangles with four doors, one in every corner



All rooms are rectangles with four doors, one in every corner

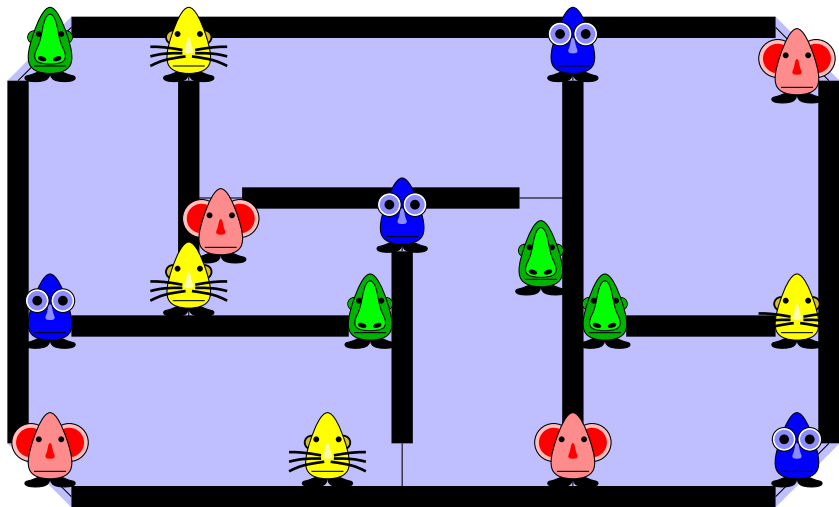


All rooms are rectangles with four doors, one in every corner

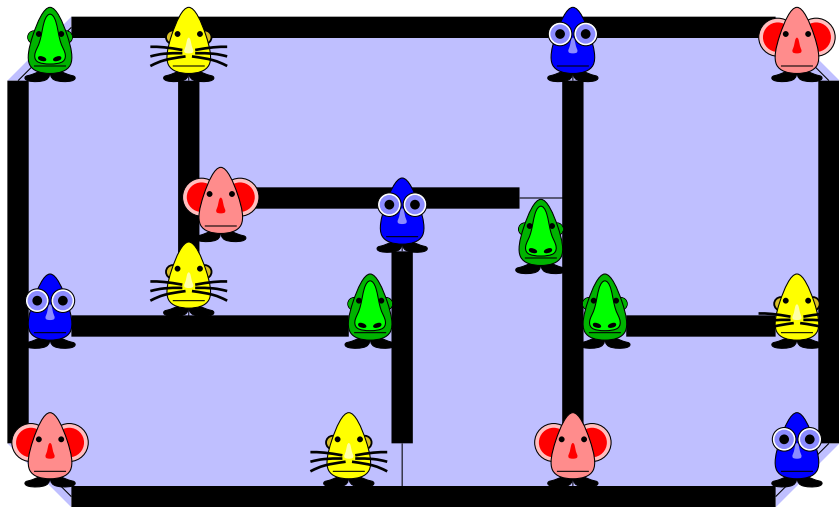




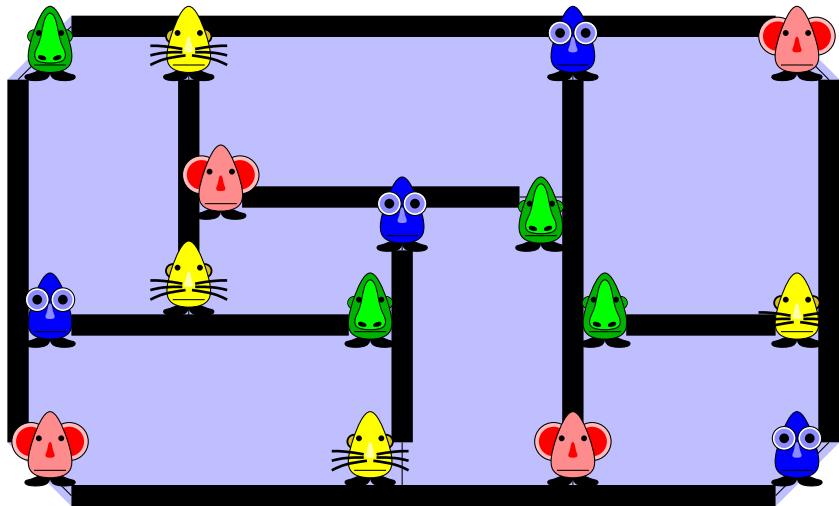
All rooms are rectangles with four doors, one in every corner



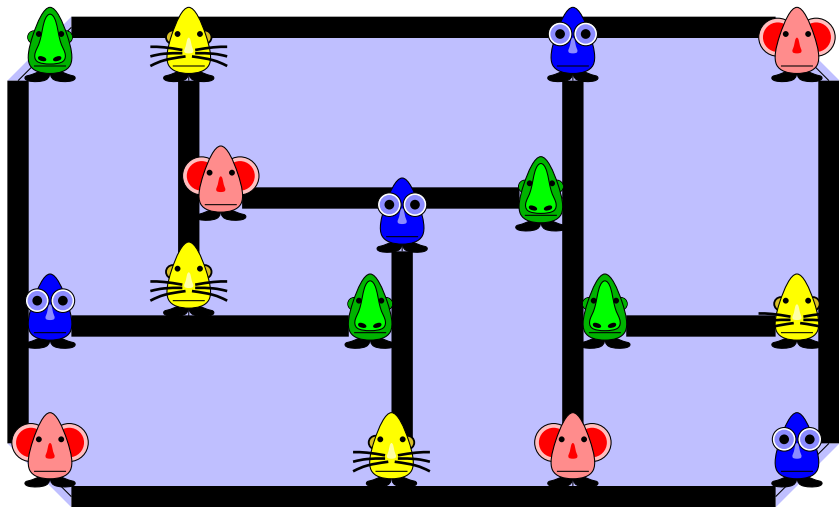
All rooms are rectangles with four doors, one in every corner



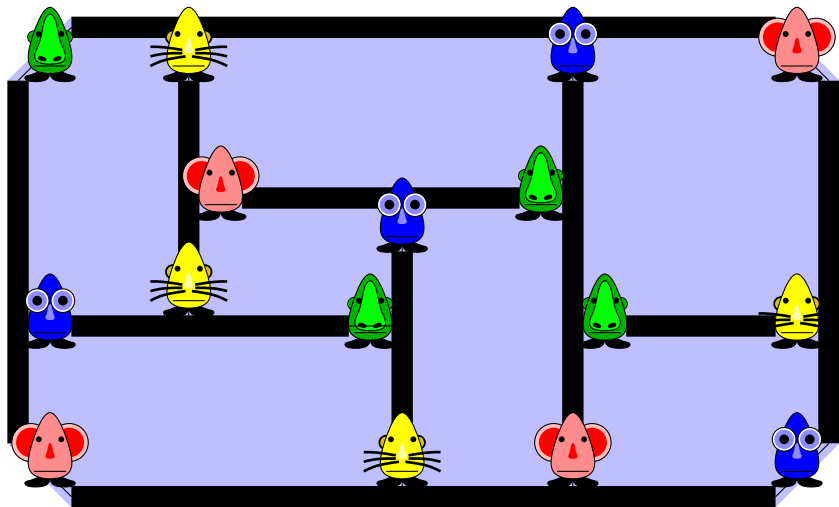
All rooms are rectangles with four doors, one in every corner



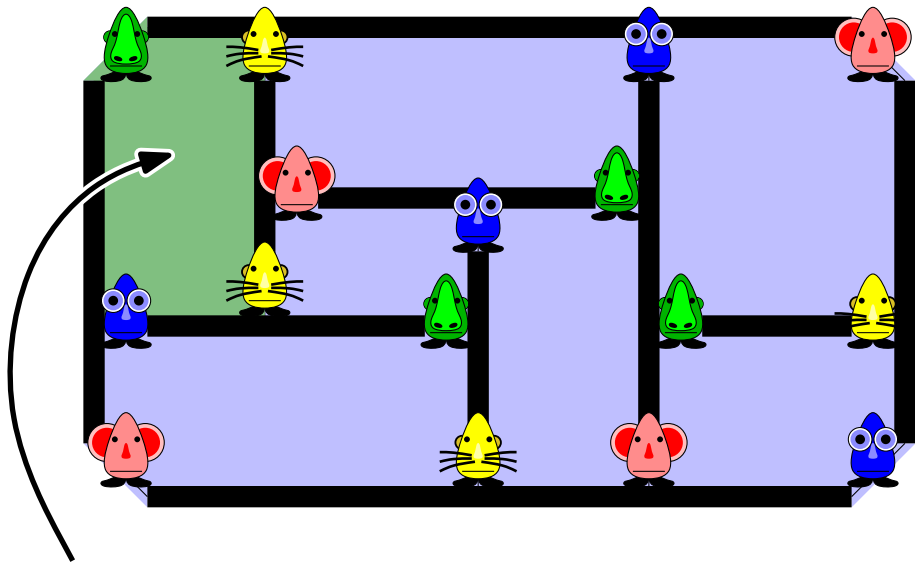
All rooms are rectangles with four doors, one in every corner



All rooms are rectangles with four doors, one in every corner

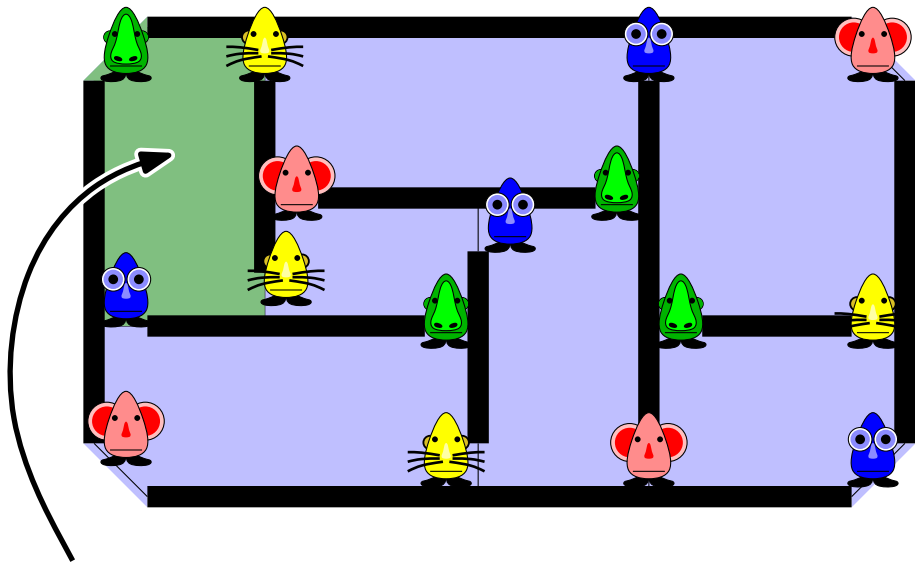


All rooms are rectangles with four doors, one in every corner



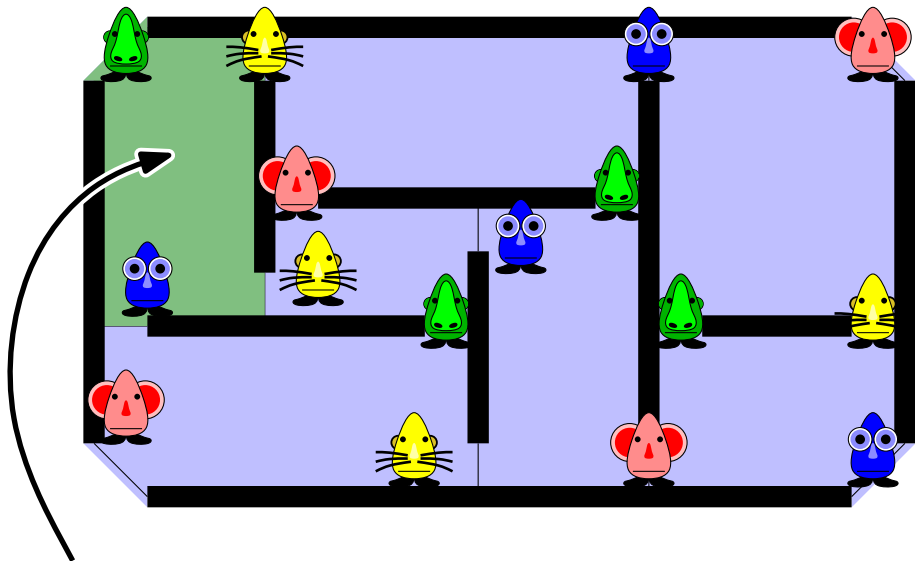
Oops... no red hear-all guard in this room!

All rooms are rectangles with four doors, one in every corner



Oops... no red hear-all guard in this room!

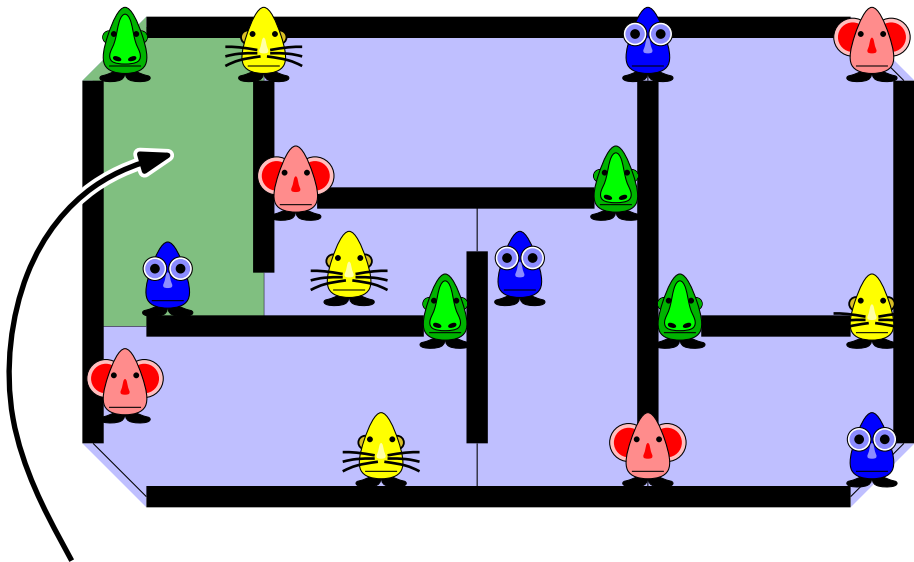
All rooms are rectangles with four doors, one in every corner



Oops... no red hear-all guard in this room!

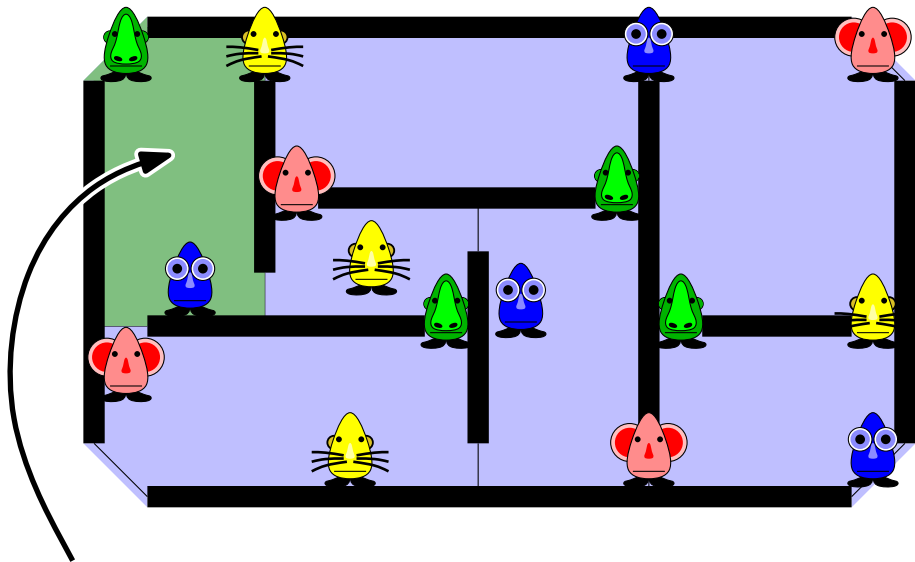


All rooms are rectangles with four doors, one in every corner



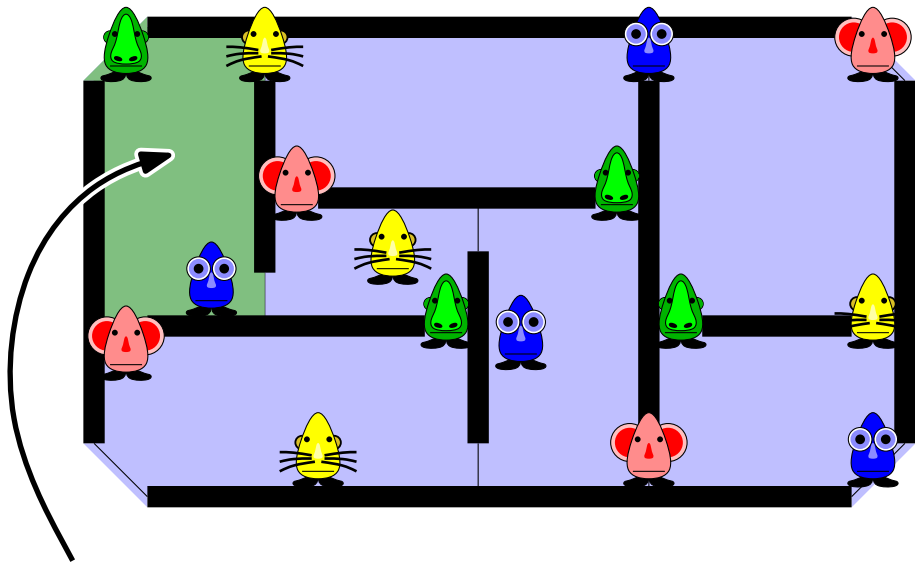
Oops... no red hear-all guard in this room!

All rooms are rectangles with four doors, one in every corner



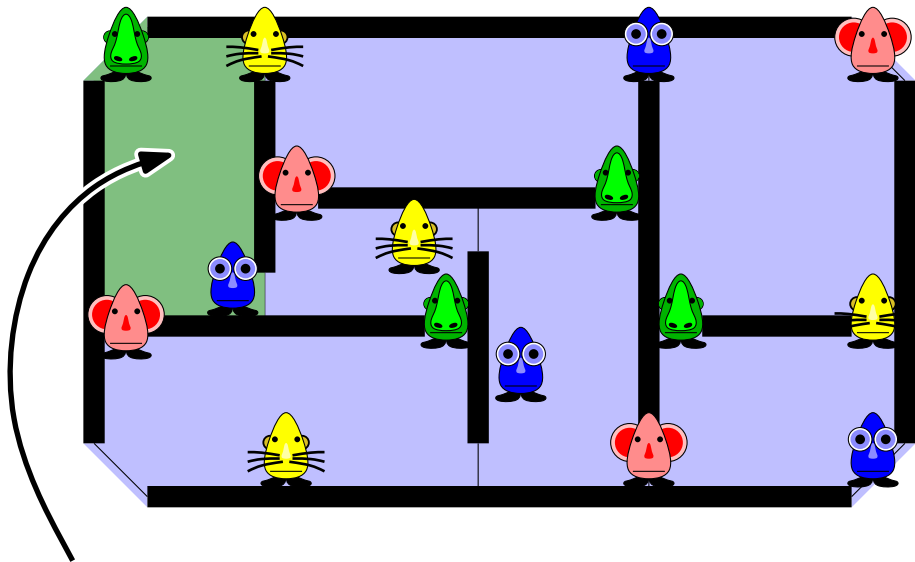
Oops... no red hear-all guard in this room!

All rooms are rectangles with four doors, one in every corner



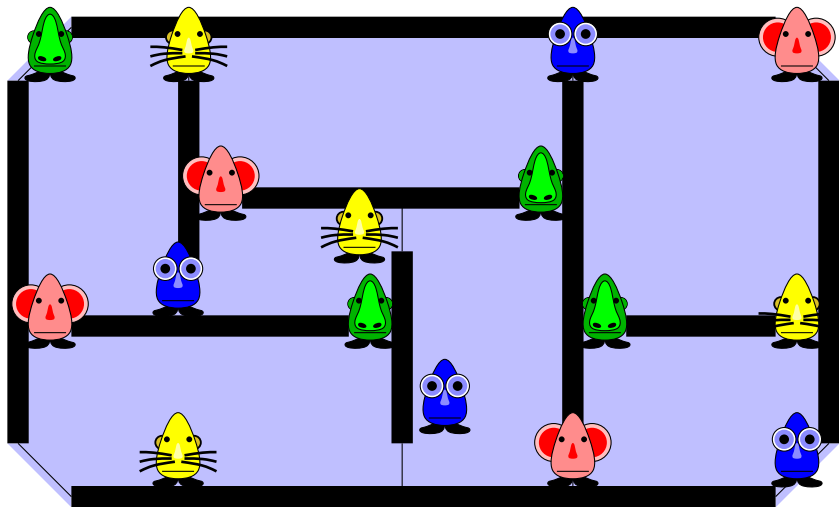
Oops... no red hear-all guard in this room!

All rooms are rectangles with four doors, one in every corner

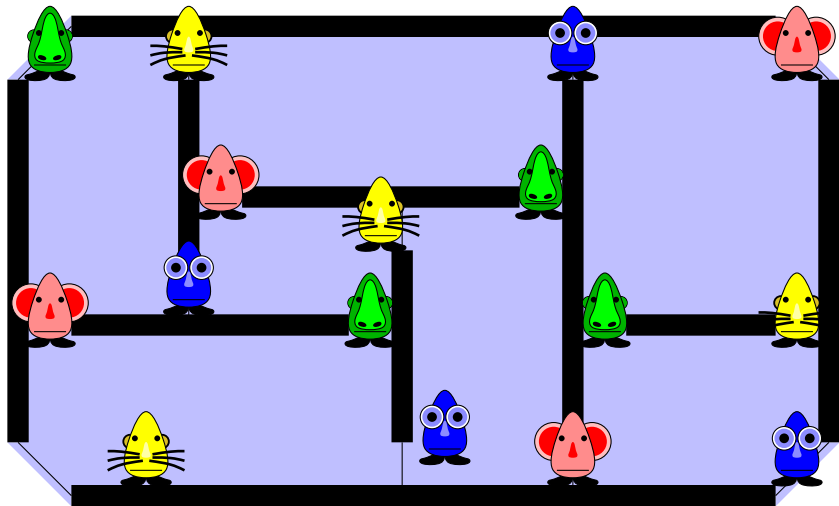


Oops... no red hear-all guard in this room!

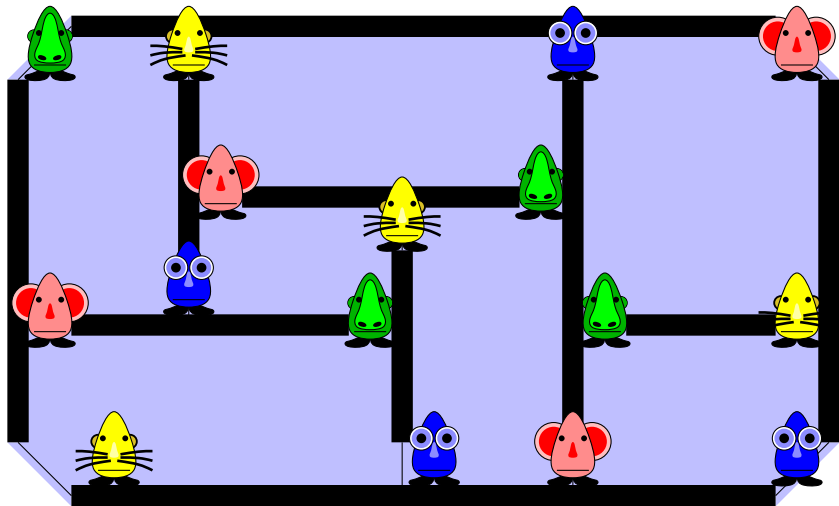
All rooms are rectangles with four doors, one in every corner



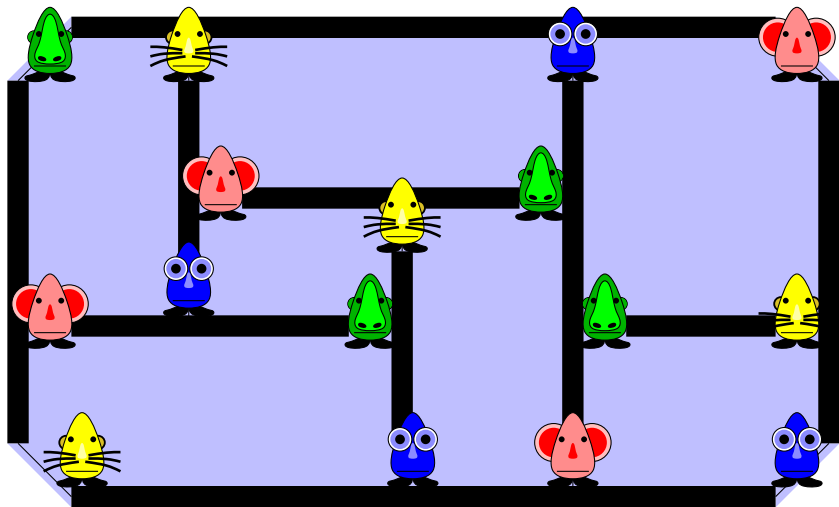
All rooms are rectangles with four doors, one in every corner



All rooms are rectangles with four doors, one in every corner

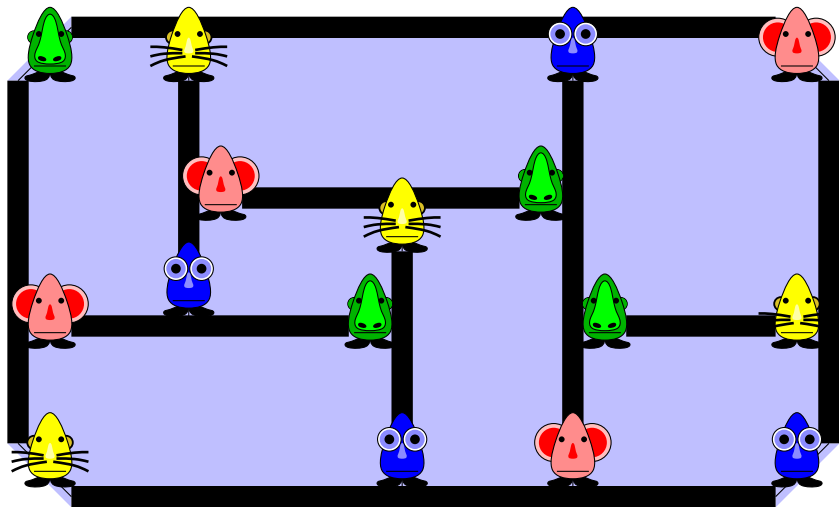


All rooms are rectangles with four doors, one in every corner



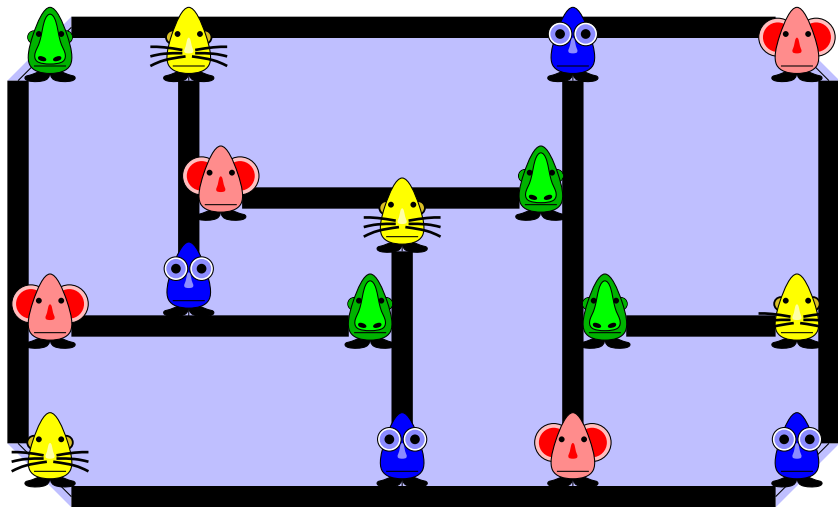


All rooms are rectangles with four doors, one in every corner



All rooms have exactly one guard of each type

All rooms are rectangles with four doors, one in every corner



All rooms have exactly one guard of each type

Can guards always be selected and placed so that this is true?

Given: • a rectangular building subdivided into rectangular rooms  
(all rooms have four doors, one in each corner)  
• guards in  $k$  colours.

Can we put one guard at each door, such that each room's four guards have  $k$  different colours?

Given: • a rectangular building subdivided into rectangular rooms  
(all rooms have four doors, one in each corner)  
• guards in  $k$  colours.

Can we put one guard at each door, such that each room's four guards have  $k$  different colours?

$k = 3$ ? Yes (Dinitz, Katz, and Krakovski, EuroCG 2007)

Given: • a rectangular building subdivided into rectangular rooms  
(all rooms have four doors, one in each corner)  
• guards in  $k$  colours.

Can we put one guard at each door, such that each room's four guards have  $k$  different colours?

$k = 3$ ? Yes (Dinitz, Katz, and Krakovski, EuroCG 2007)

$k = 5$ ? No, obviously

- Given:
- a rectangular building subdivided into rectangular rooms (all rooms have four doors, one in each corner)
  - guards in  $k$  colours.

Can we put one guard at each door, such that each room's four guards have  $k$  different colours?

$k = 3$ ? Yes (Dinitz, Katz, and Krakovski, EuroCG 2007)

$k = 4$ ? That's the question (Dinitz, Katz, and Krakovski, EuroCG 2007)

$k = 5$ ? No, obviously

- Given:
- a rectangular building subdivided into rectangular rooms (all rooms have four doors, one in each corner)
  - guards in  $k$  colours.

Can we put one guard at each door, such that each room's four guards have  $k$  different colours?

$k = 3$ ? Yes (Dinitz, Katz, and Krakovski, EuroCG 2007)

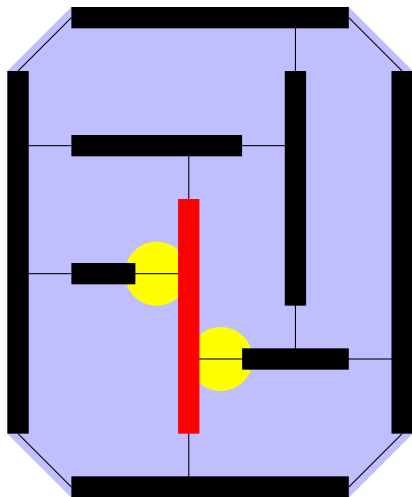
$k = 4$ ? Yes (Dimitrov, Horev, and Krakovski, next talk, based on Guenin)

Our results:

- simple, efficient algorithm for sliceable subdivisions
- simple, efficient algorithm for one-sided subdivisions

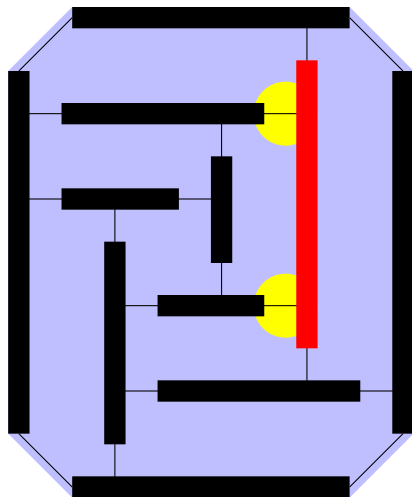
$k = 5$ ? No, obviously

two-sided museum



wall has doors on both sides

one-sided museum

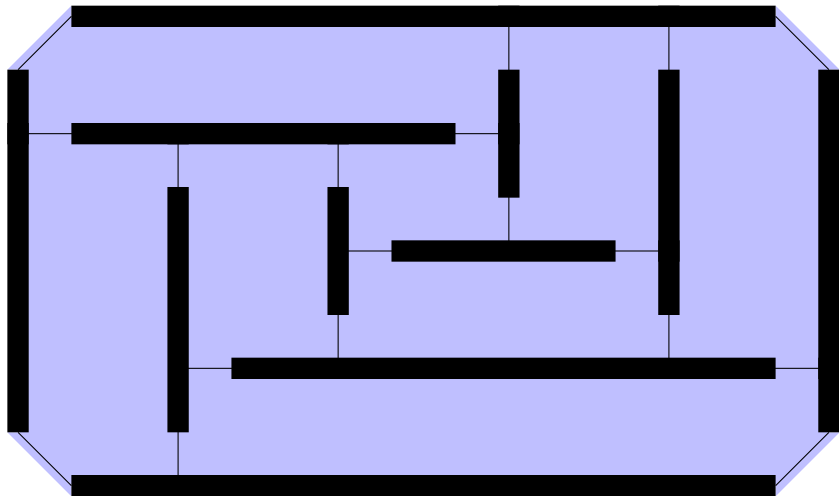


each wall has doors on at  
most one side



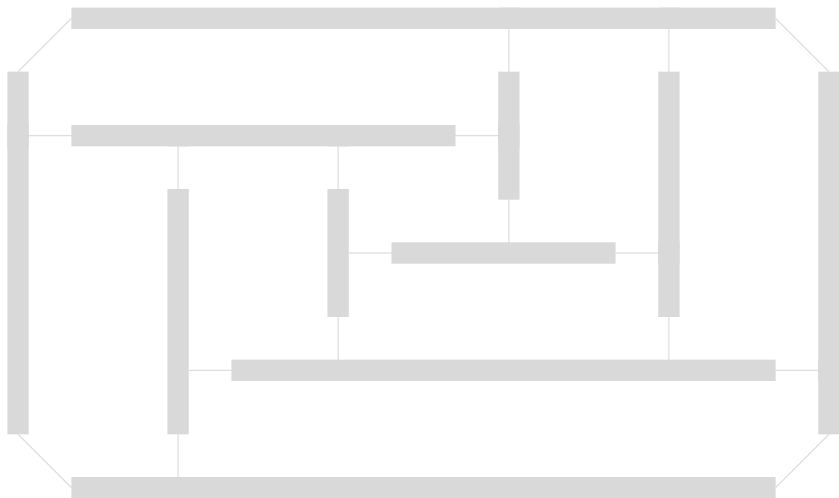
## guarding one-sided museums

---



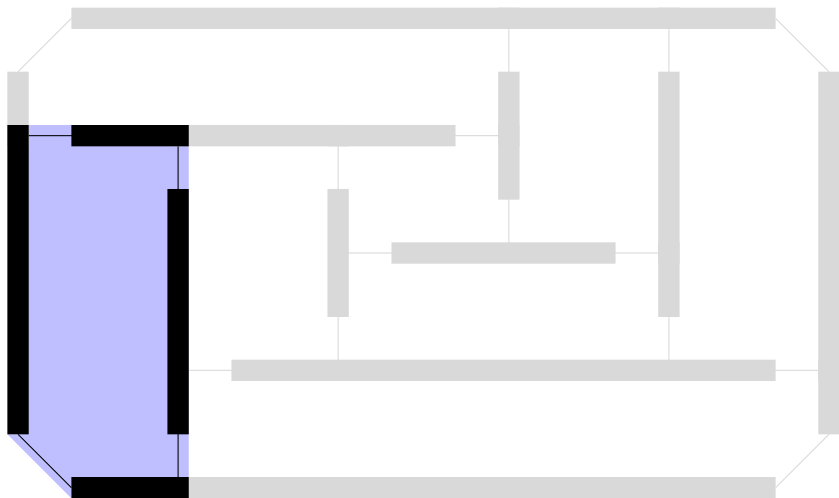
## guarding one-sided museums

---



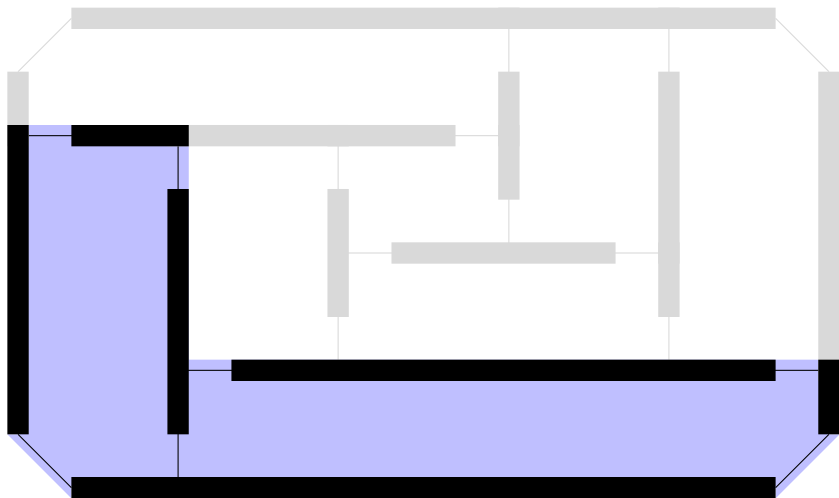
## guarding one-sided museums

---



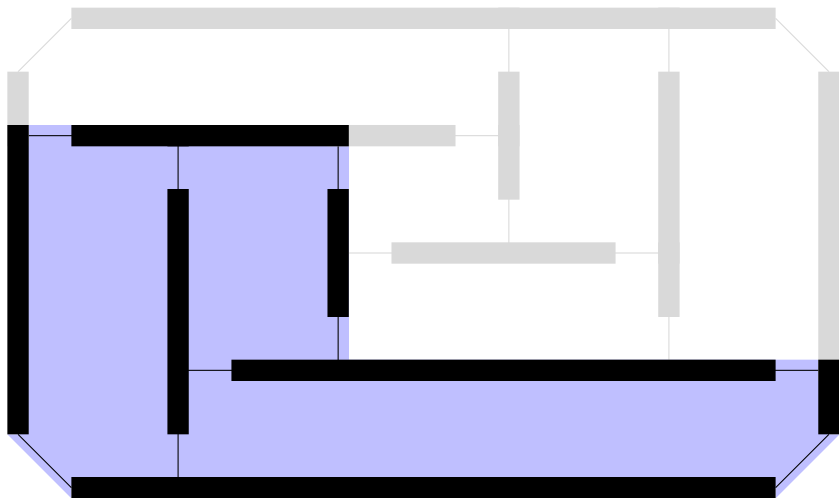
## guarding one-sided museums

---



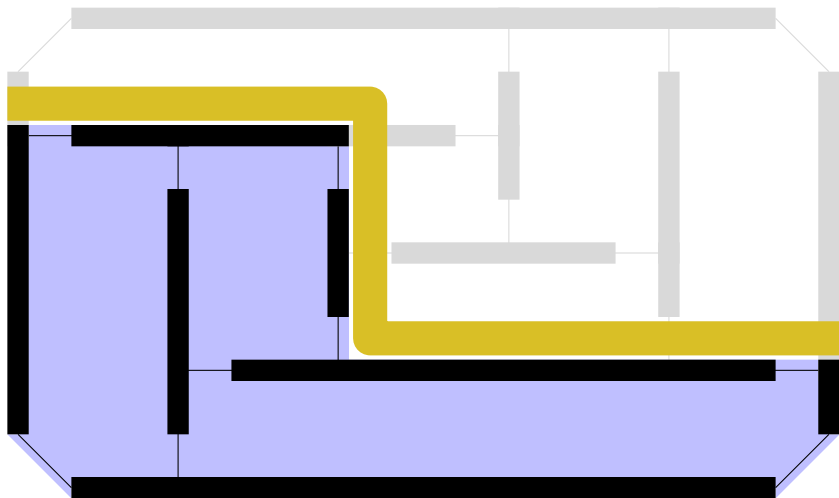
## guarding one-sided museums

---



## guarding one-sided museums

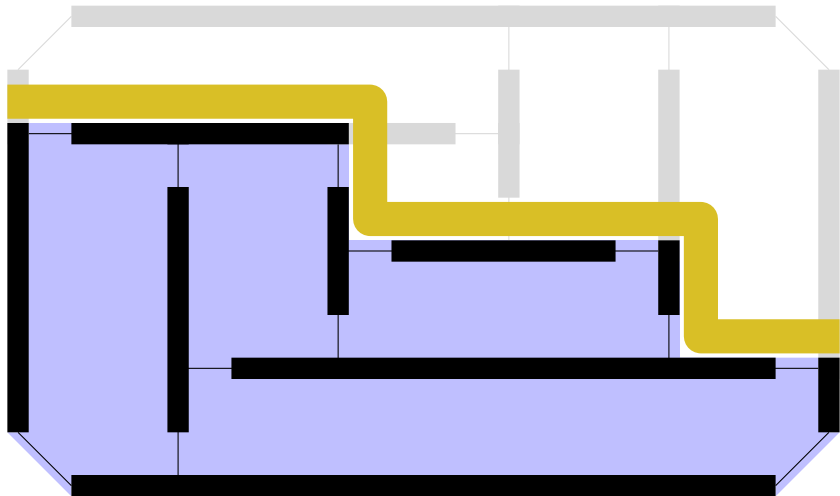
---



boundary is descending staircase

## guarding one-sided museums

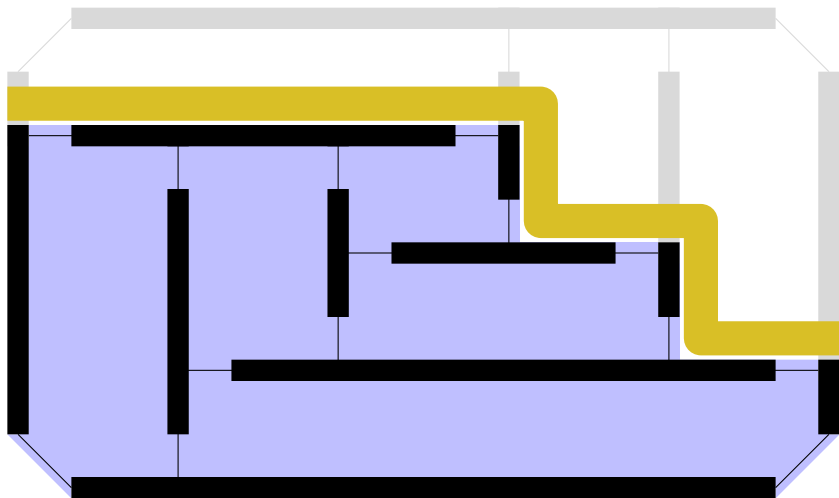
---



boundary is descending staircase

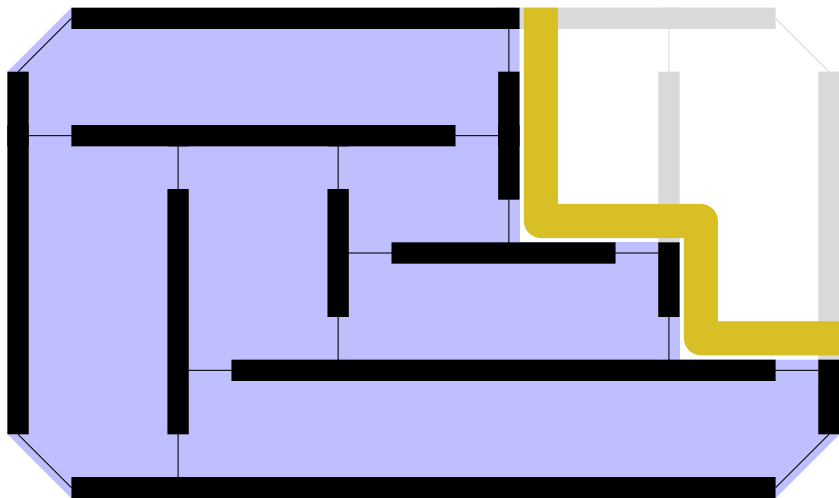
## guarding one-sided museums

---



boundary is descending staircase

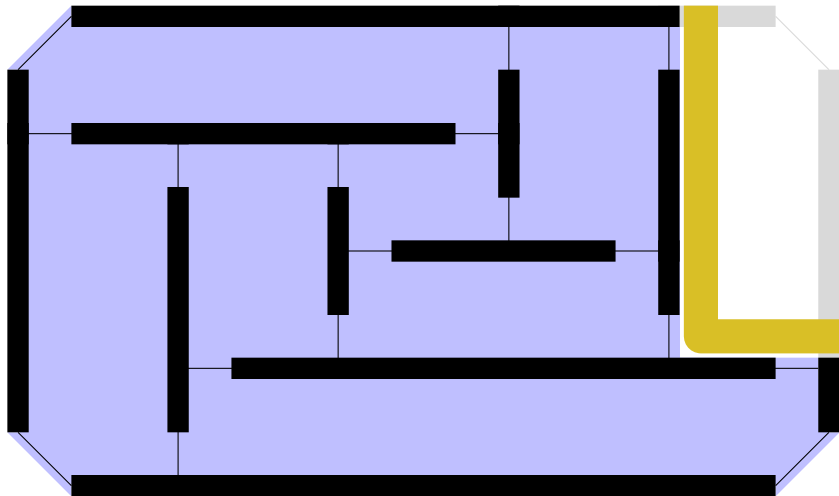




boundary is descending staircase

## guarding one-sided museums

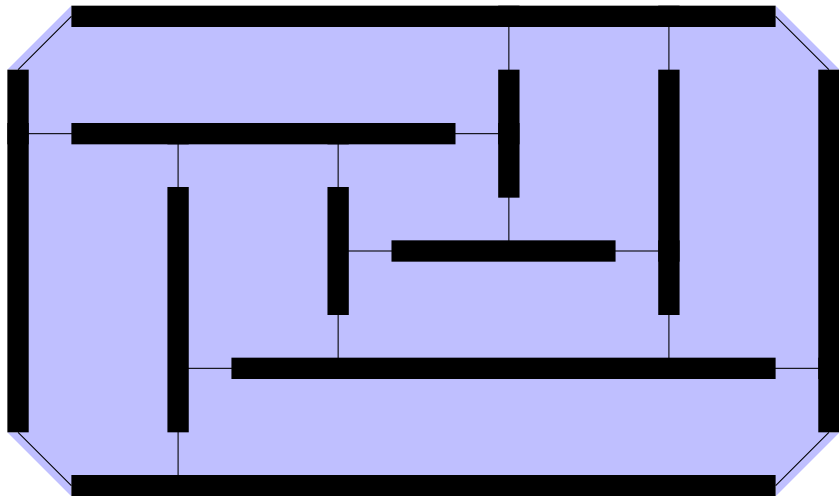
---



boundary is descending staircase

## guarding one-sided museums

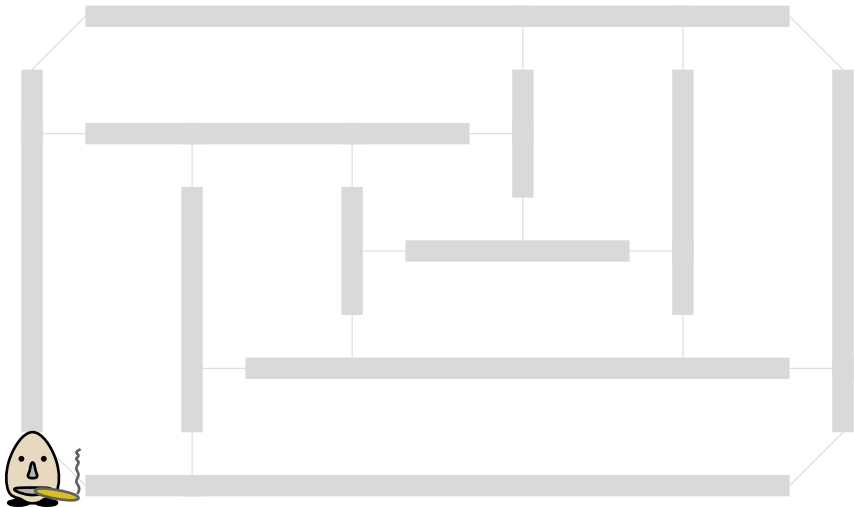
---



boundary is descending staircase

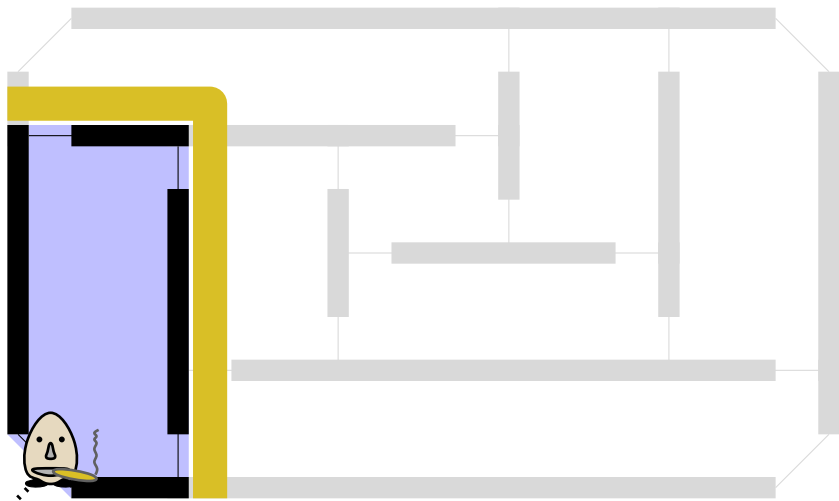
## guarding one-sided museums

---



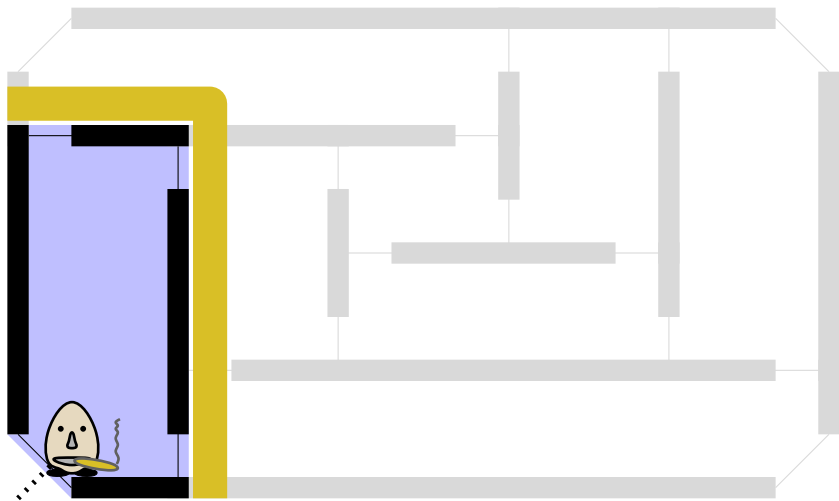
## guarding one-sided museums

---



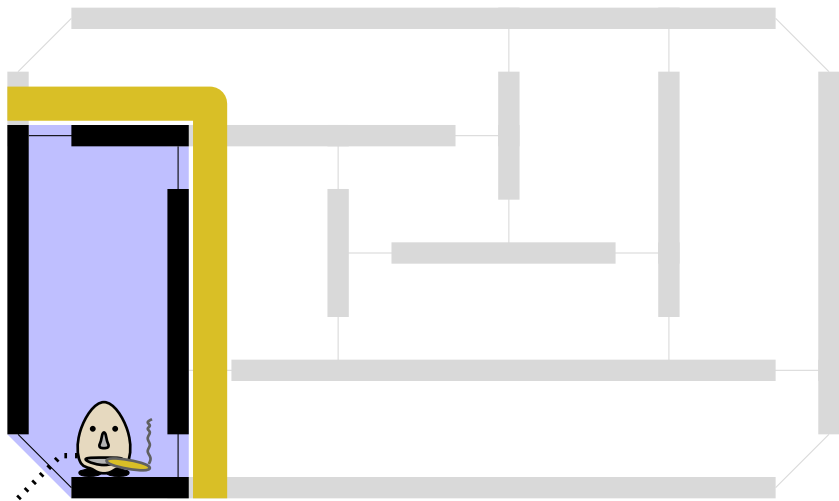
## guarding one-sided museums

---



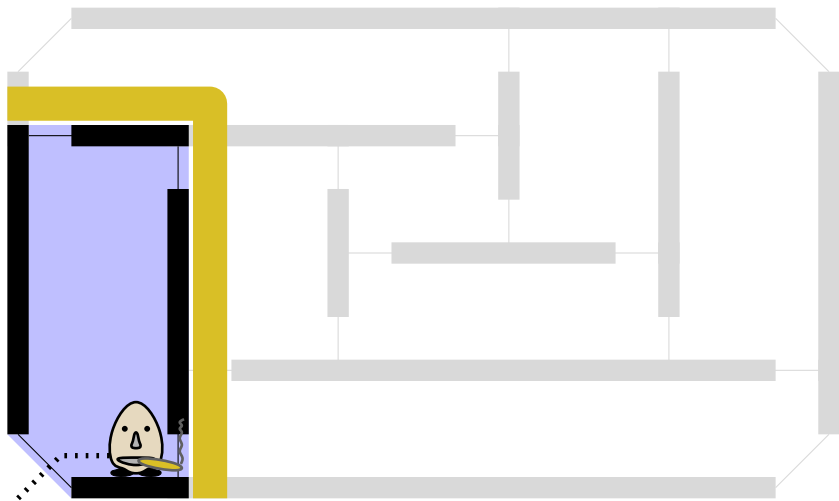
## guarding one-sided museums

---



## guarding one-sided museums

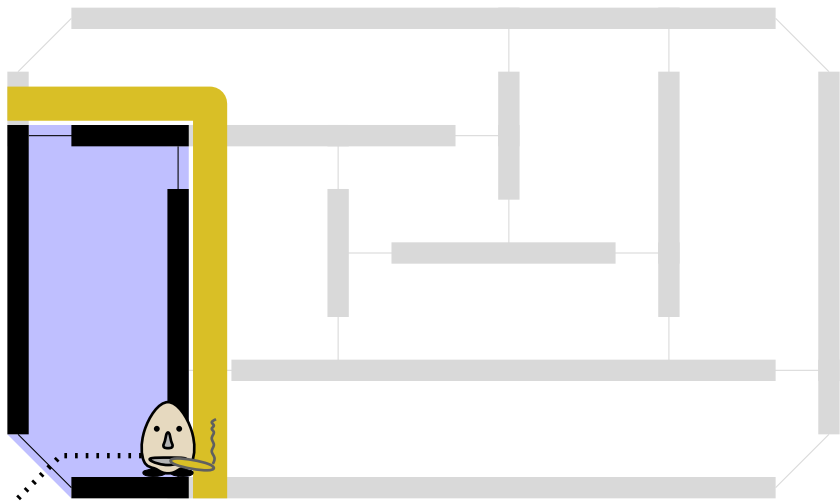
---





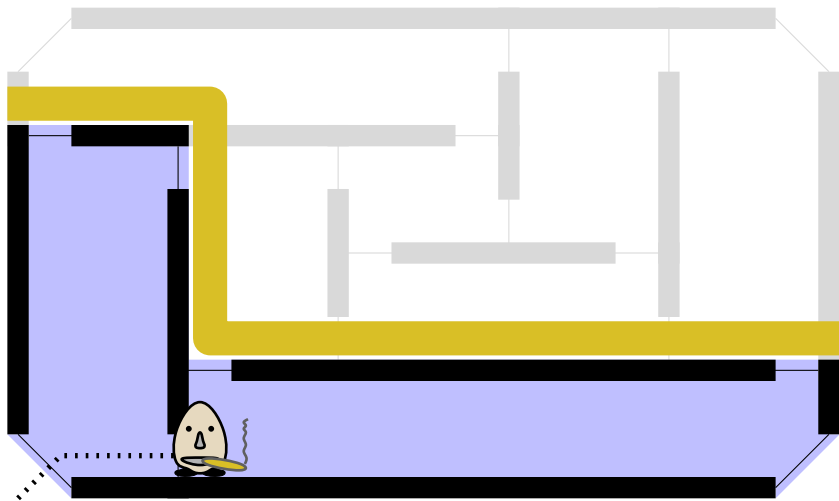
## guarding one-sided museums

---



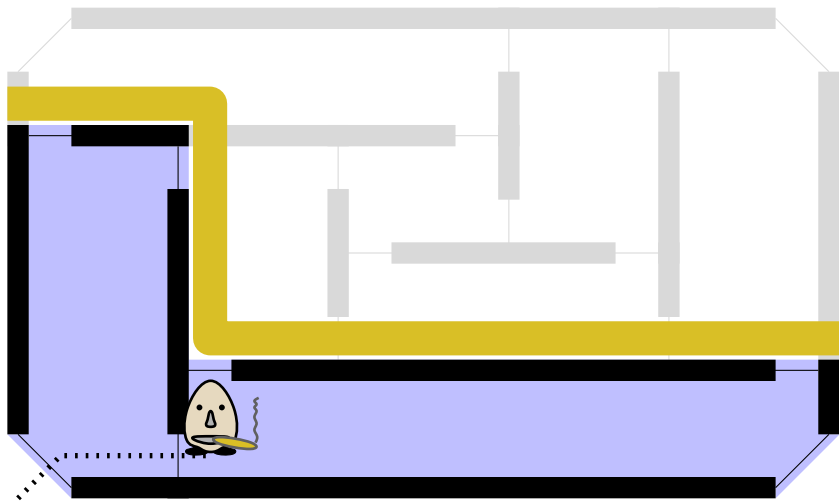
## guarding one-sided museums

---



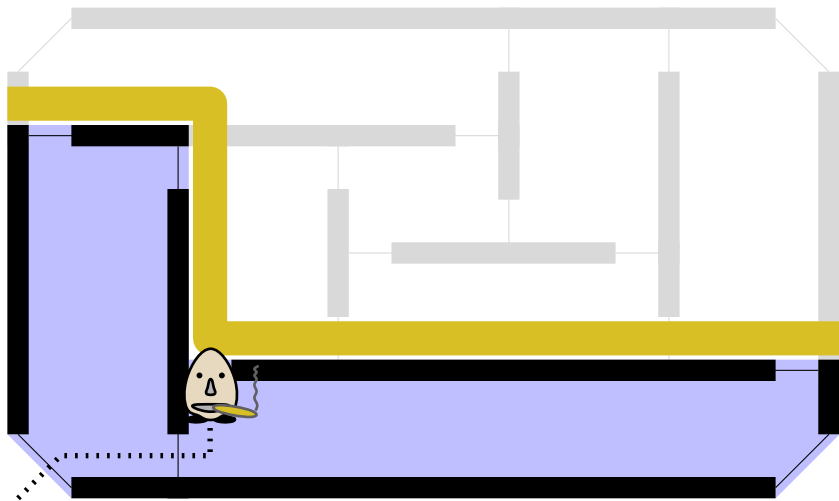
## guarding one-sided museums

---



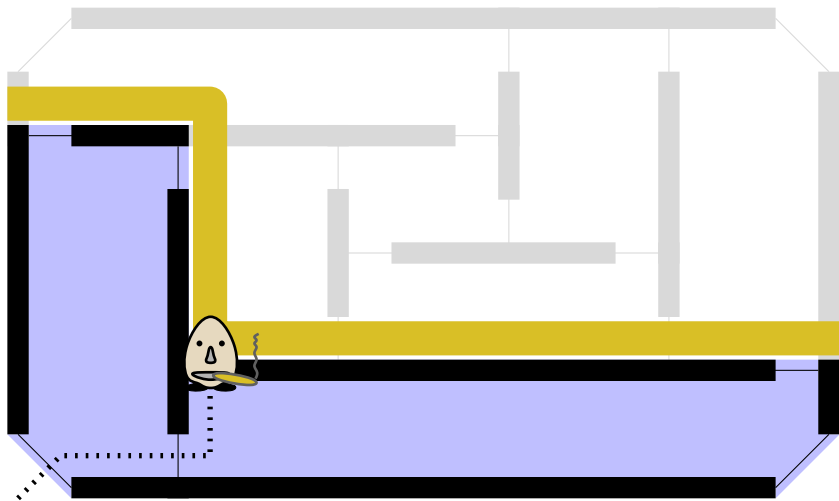
## guarding one-sided museums

---



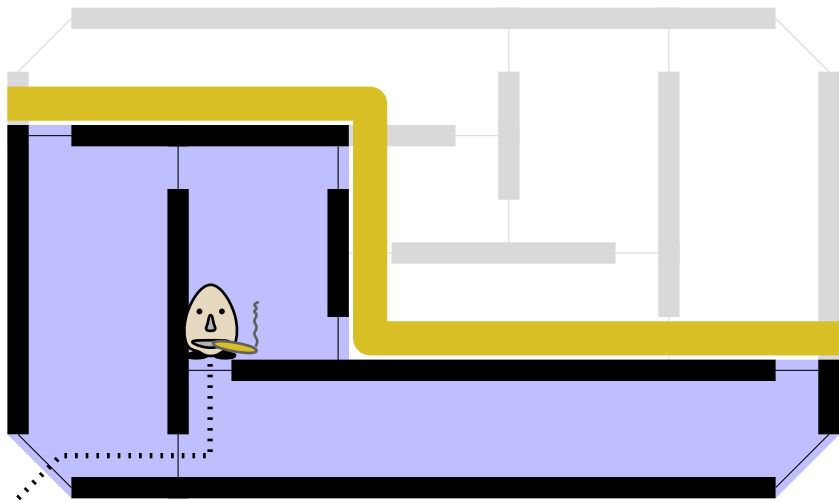
## guarding one-sided museums

---



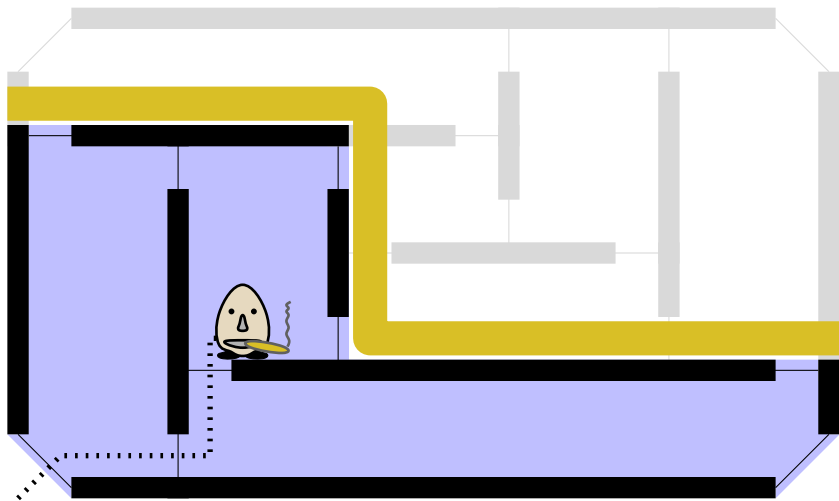
## guarding one-sided museums

---



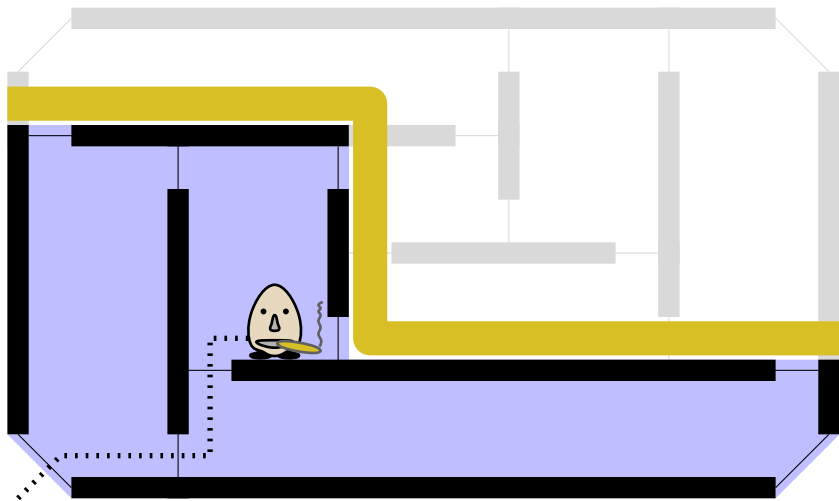
## guarding one-sided museums

---



## guarding one-sided museums

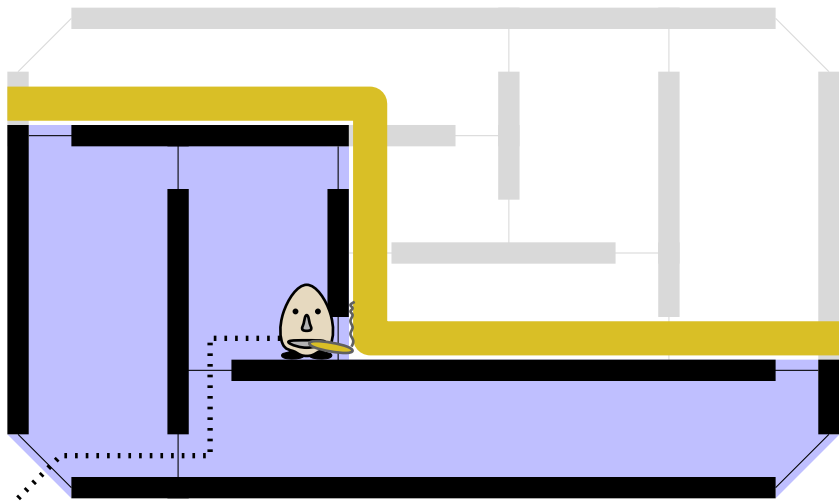
---





## guarding one-sided museums

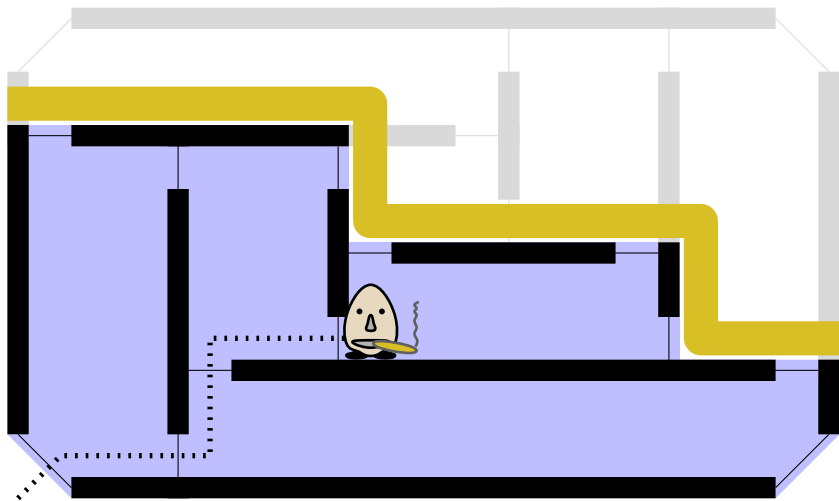
---



guarding one-sided museums

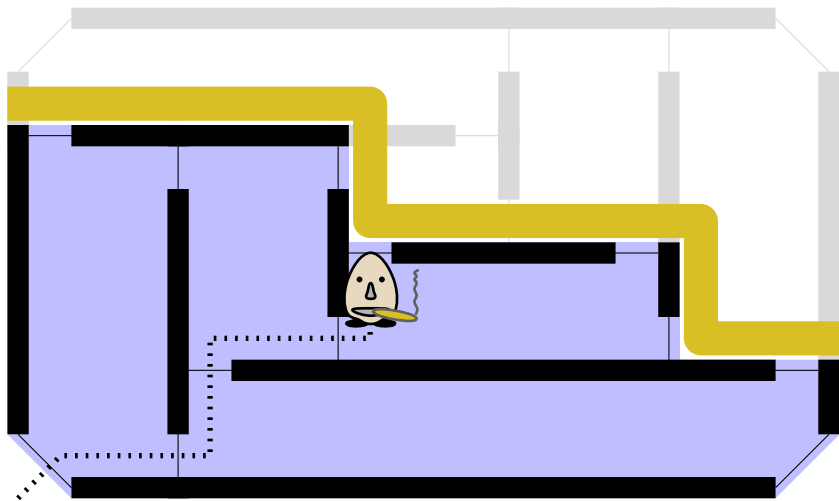
## guarding one-sided museums

---



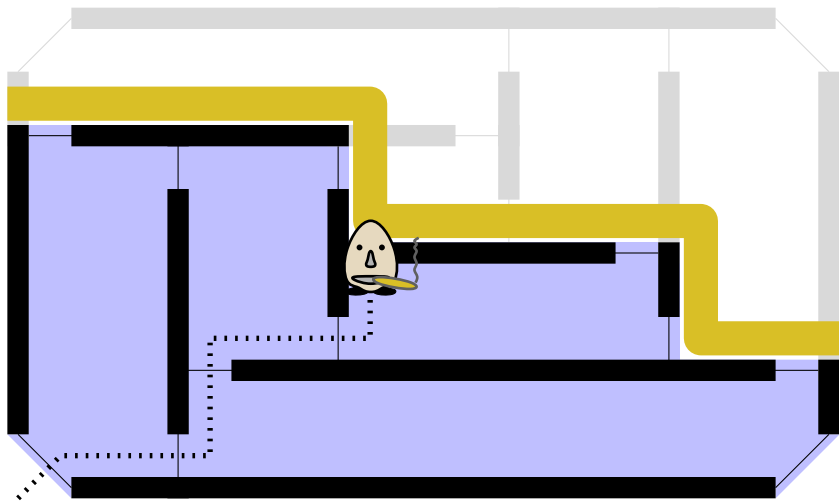
## guarding one-sided museums

---



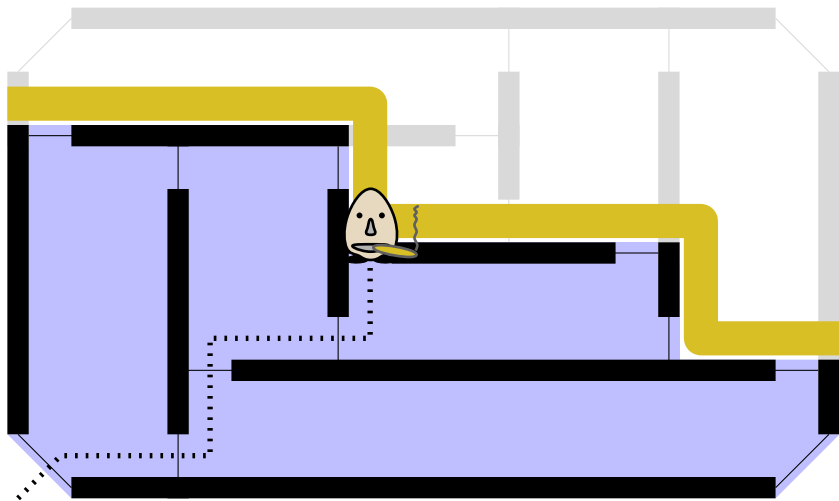
## guarding one-sided museums

---



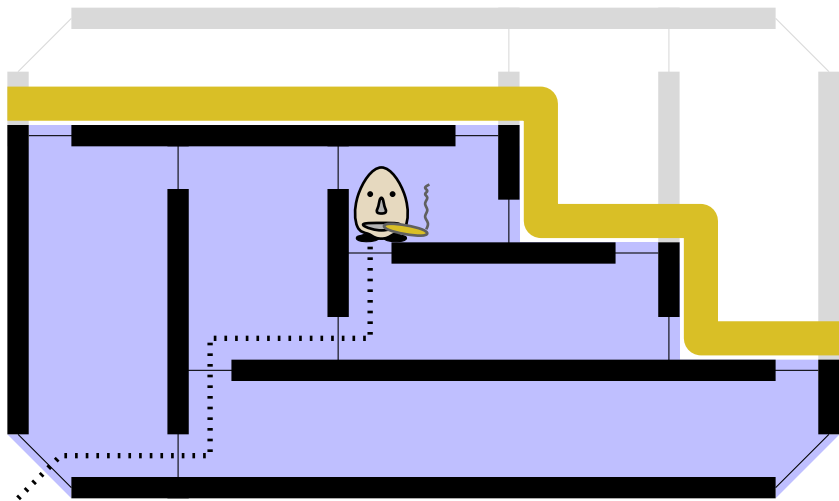
## guarding one-sided museums

---



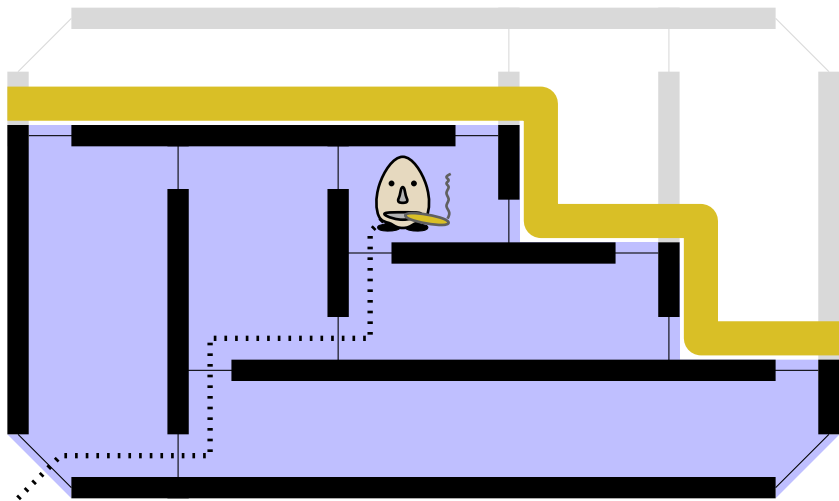
## guarding one-sided museums

---



## guarding one-sided museums

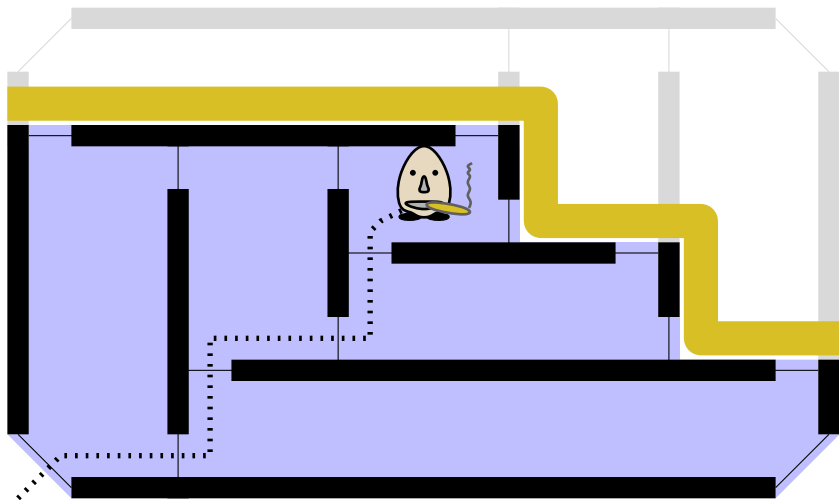
---





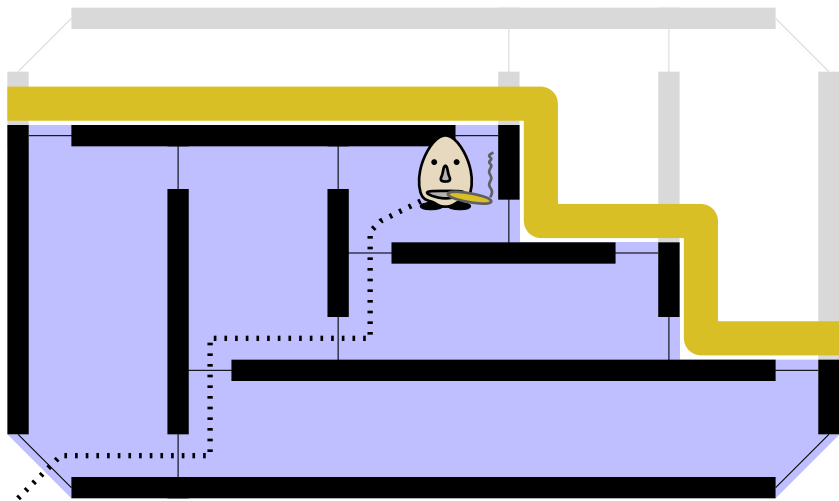
## guarding one-sided museums

---



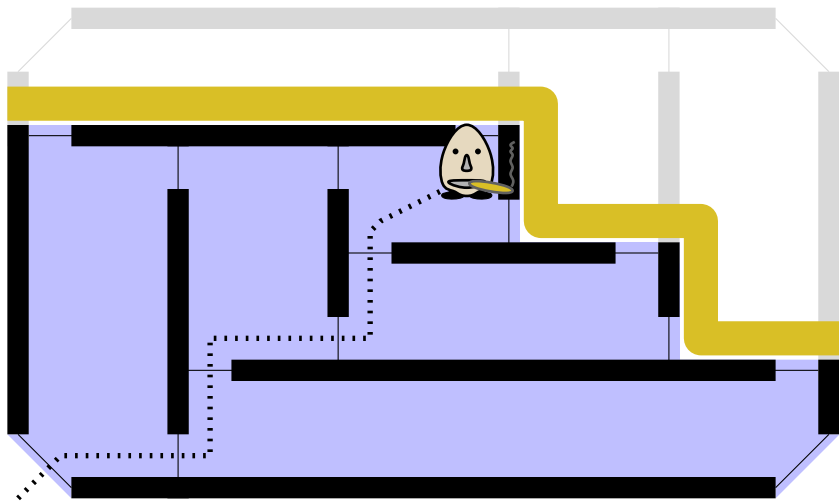
## guarding one-sided museums

---



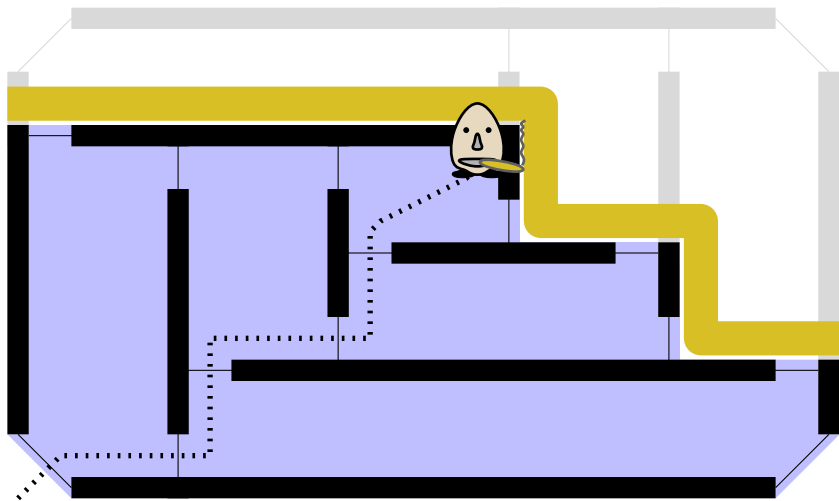
## guarding one-sided museums

---



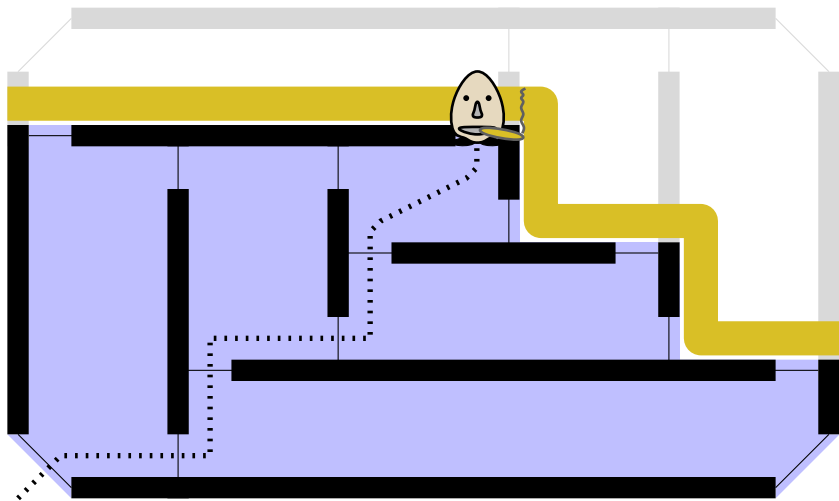
## guarding one-sided museums

---

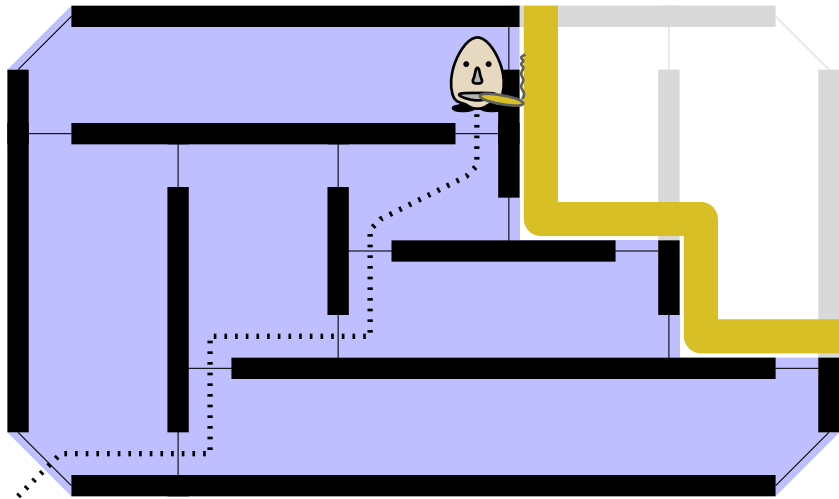


## guarding one-sided museums

---

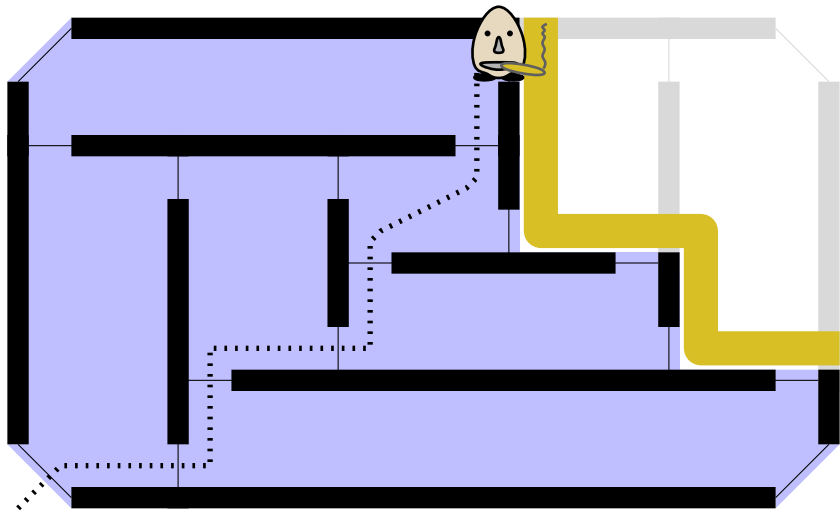


guarding one-sided museums



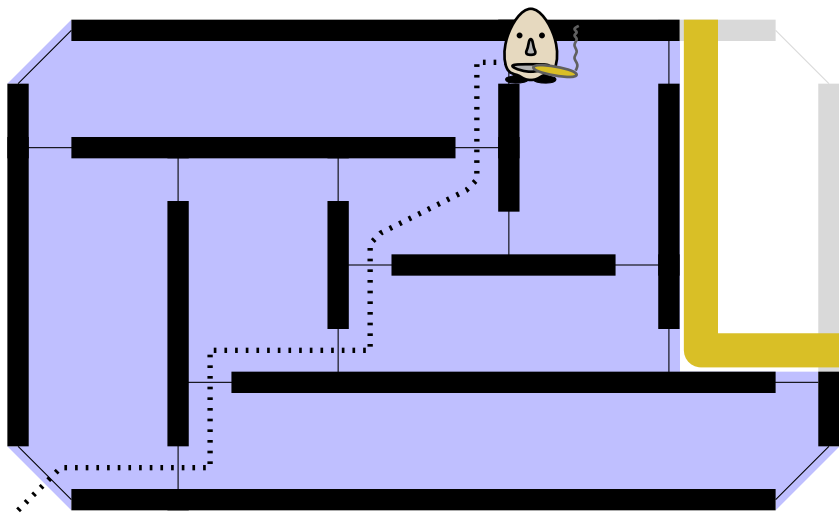
guarding one-sided museums

guarding one-sided museums

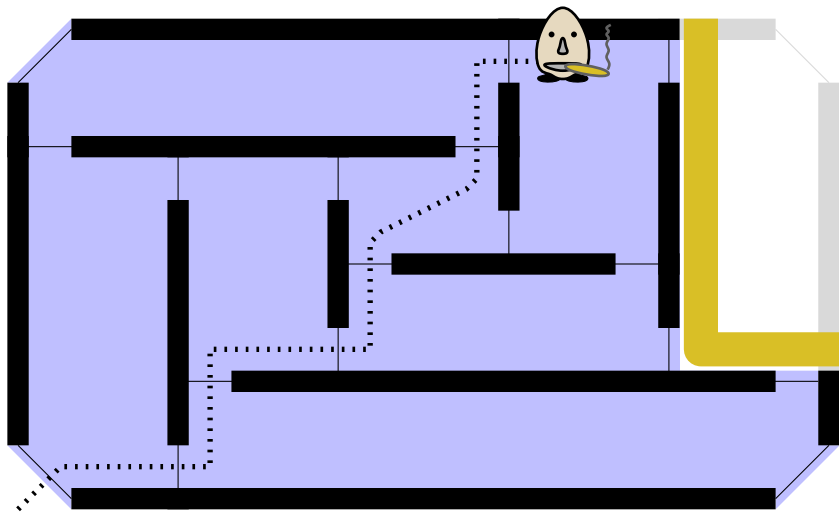




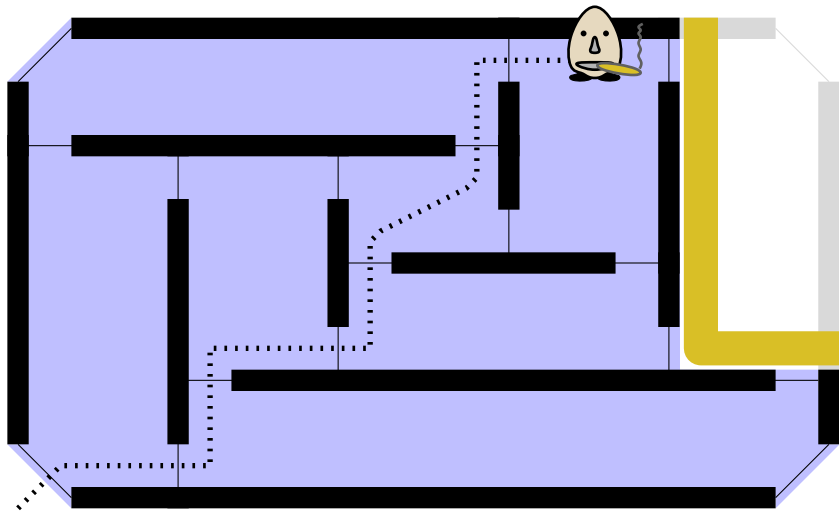
guarding one-sided museums



guarding one-sided museums

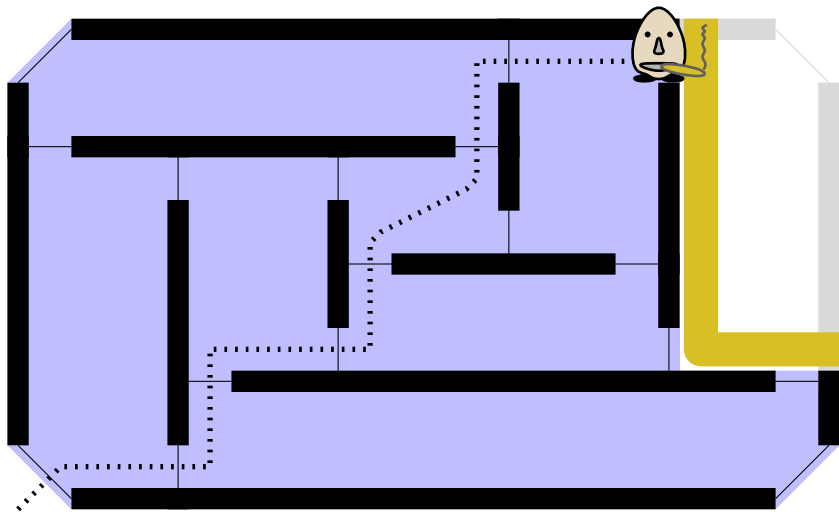


guarding one-sided museums

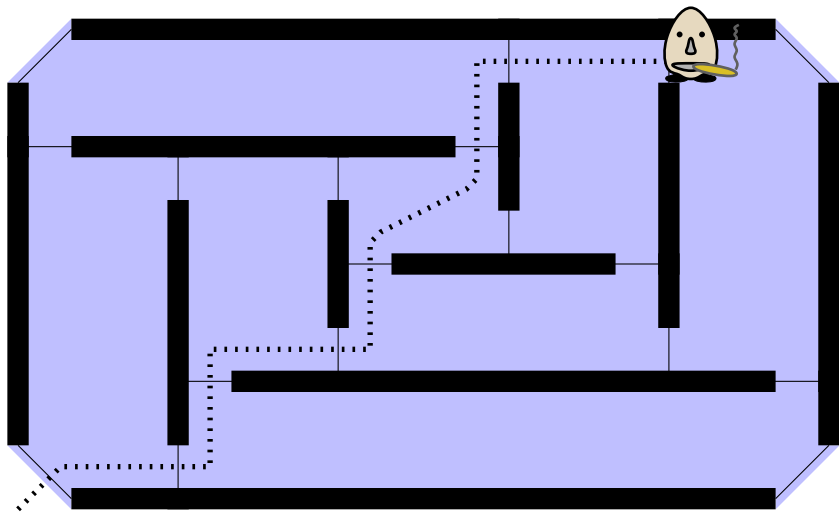




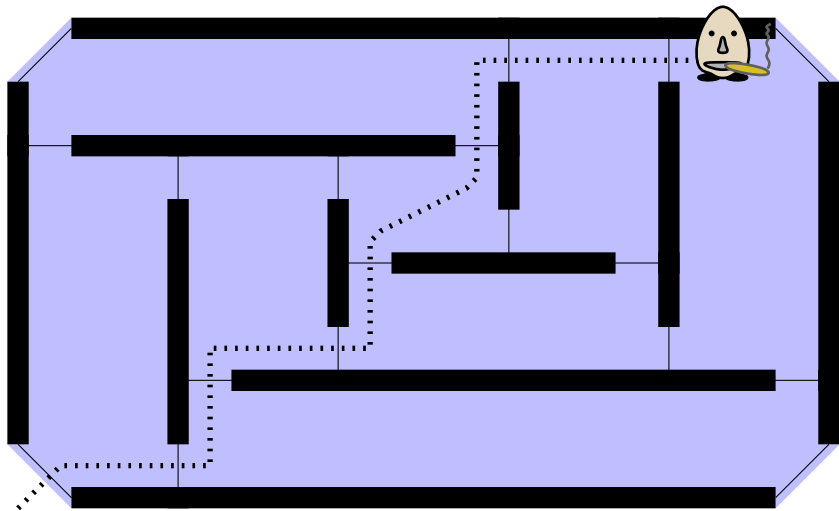
guarding one-sided museums



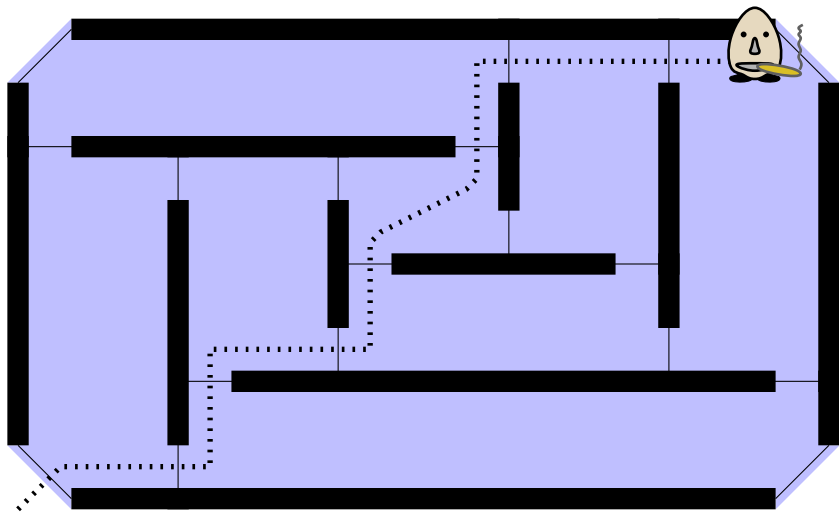
guarding one-sided museums



guarding one-sided museums

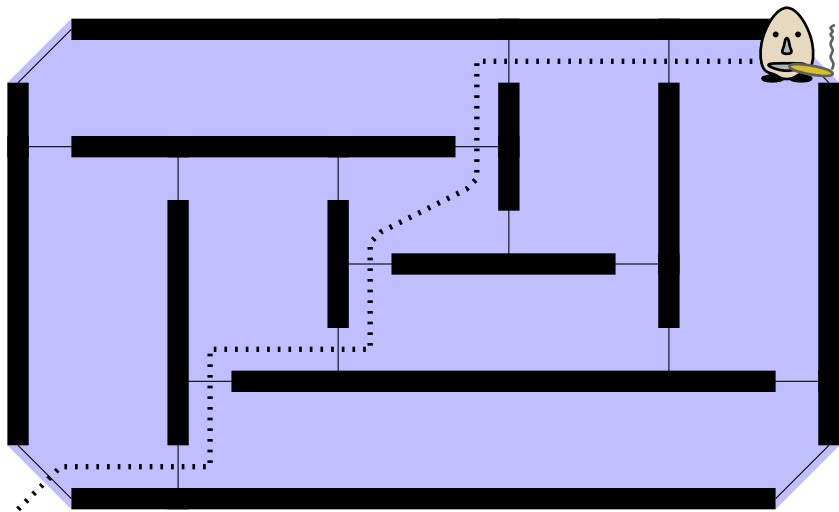


guarding one-sided museums

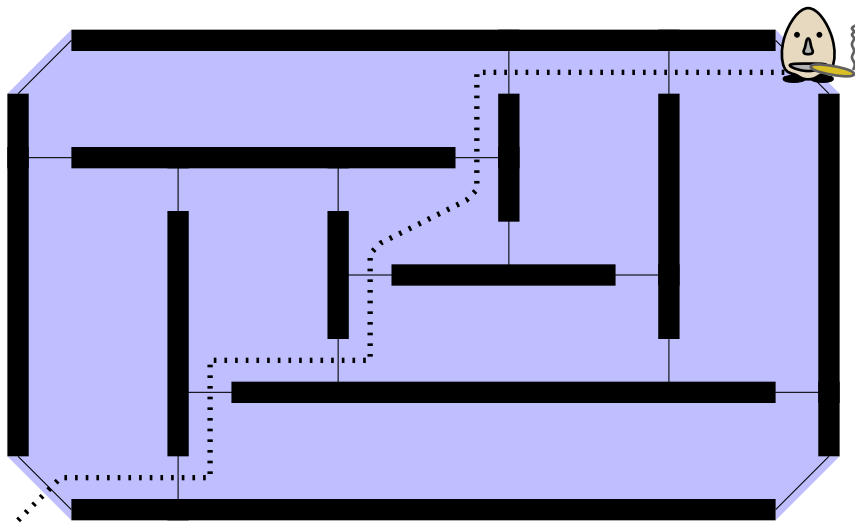




guarding one-sided museums

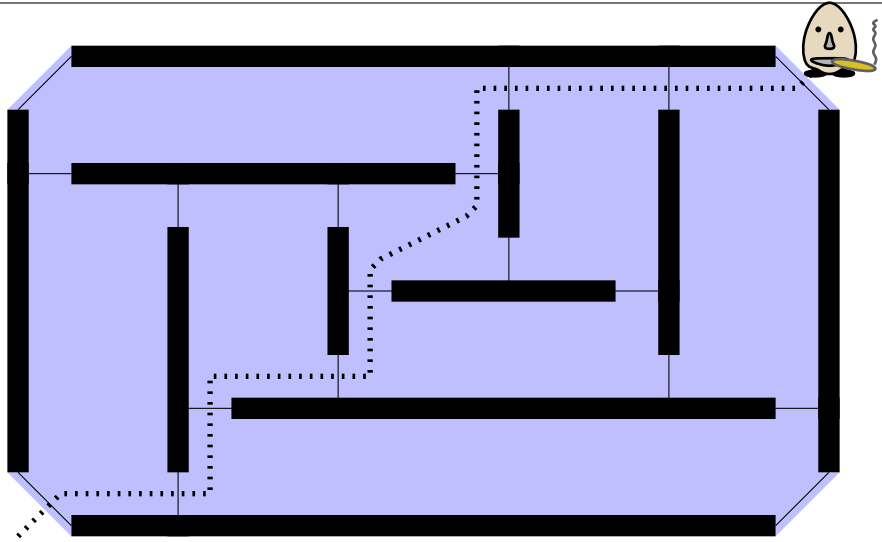


guarding one-sided museums





## guarding one-sided museums



Visit rooms in order s.t.

- inner walls on boundary of any prefix form desc. staircase;
- every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

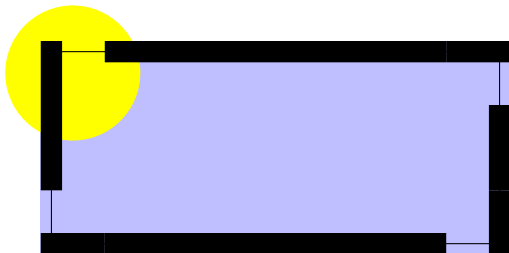


## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

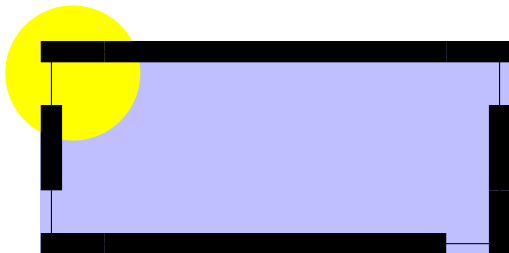


## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room



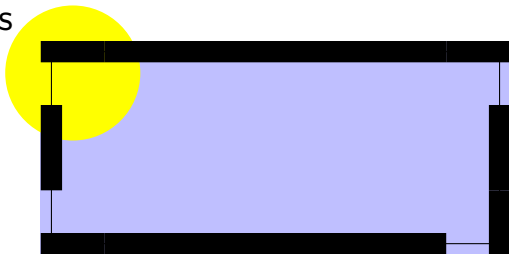
## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

2 possibilities





## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

2 possibilities



## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

2 possibilities



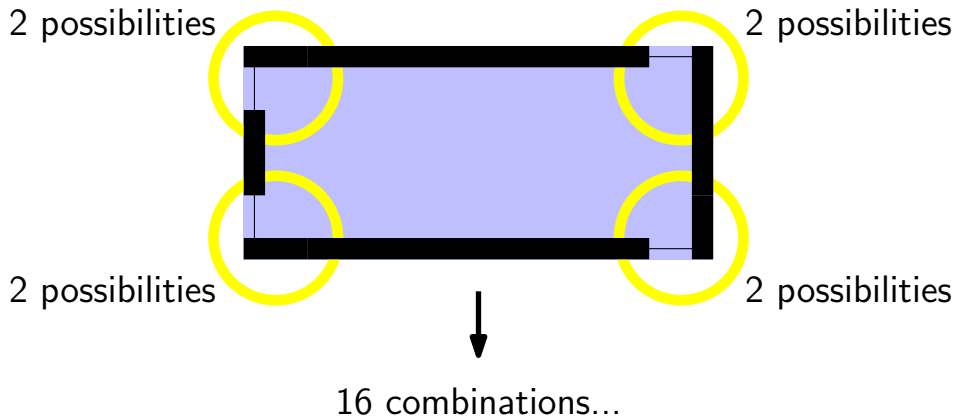
2 possibilities

## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room



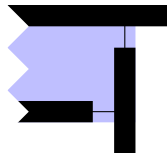
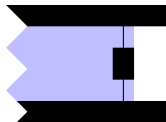
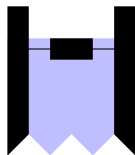
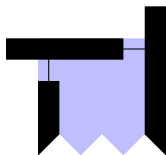
## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:



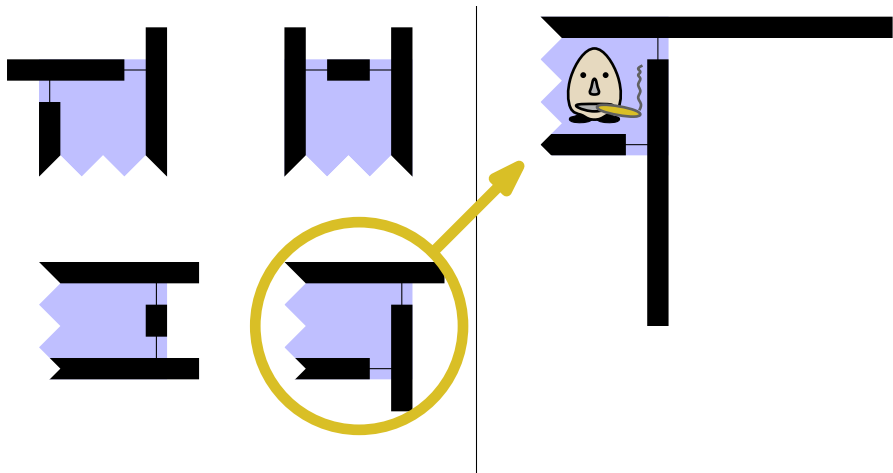
## guarding one-sided museums

---

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:

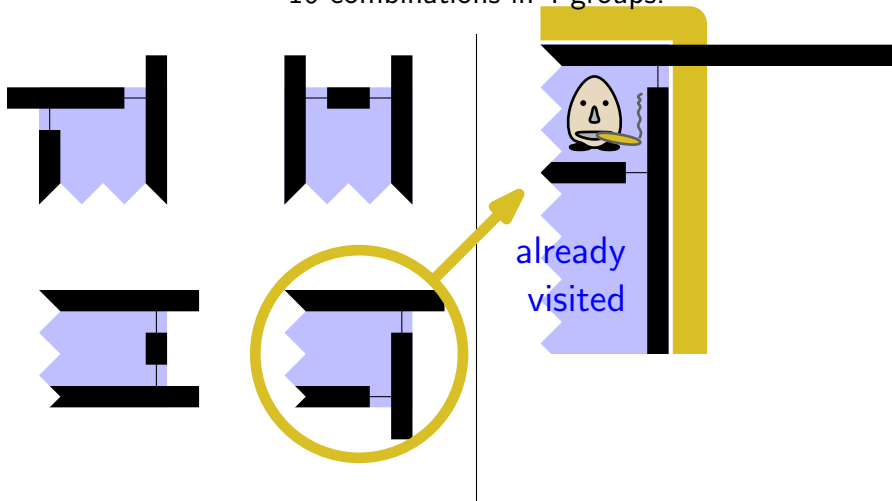


## guarding one-sided museums

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:

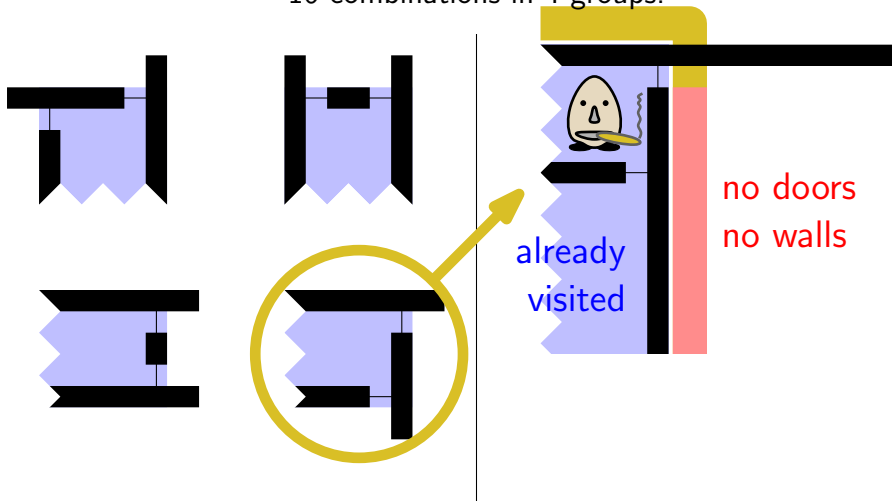


## guarding one-sided museums

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:

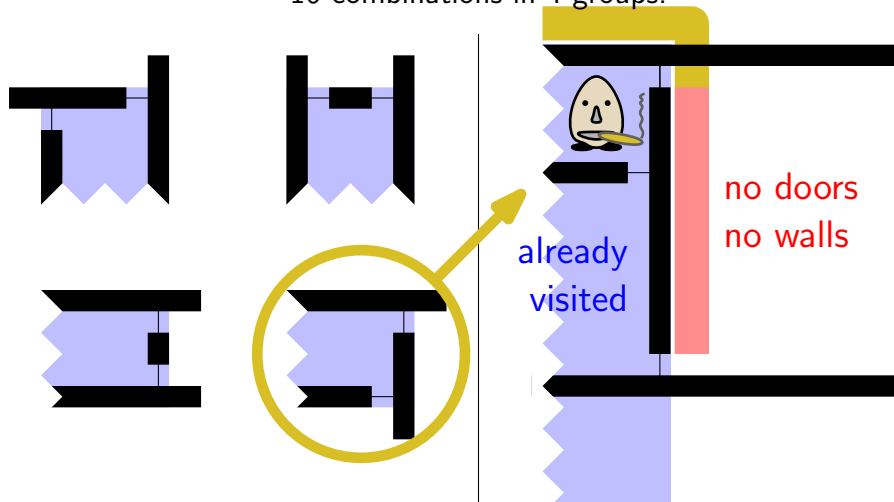


## guarding one-sided museums

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:



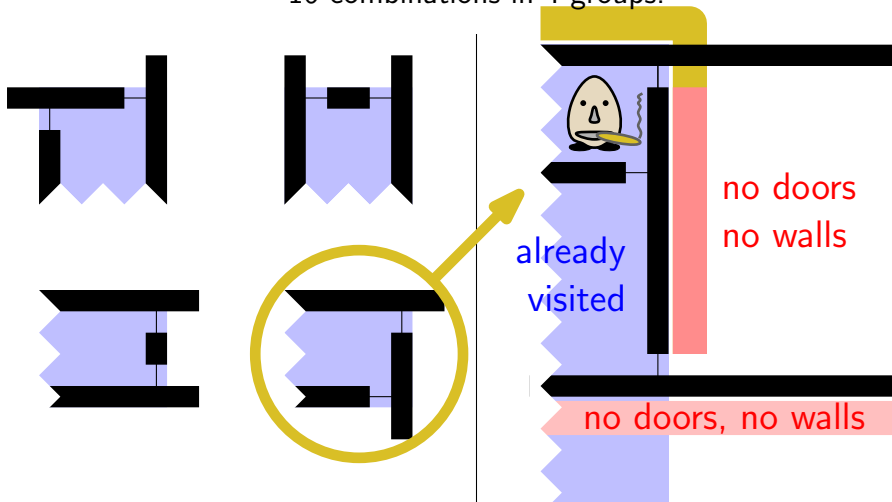


## guarding one-sided museums

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:

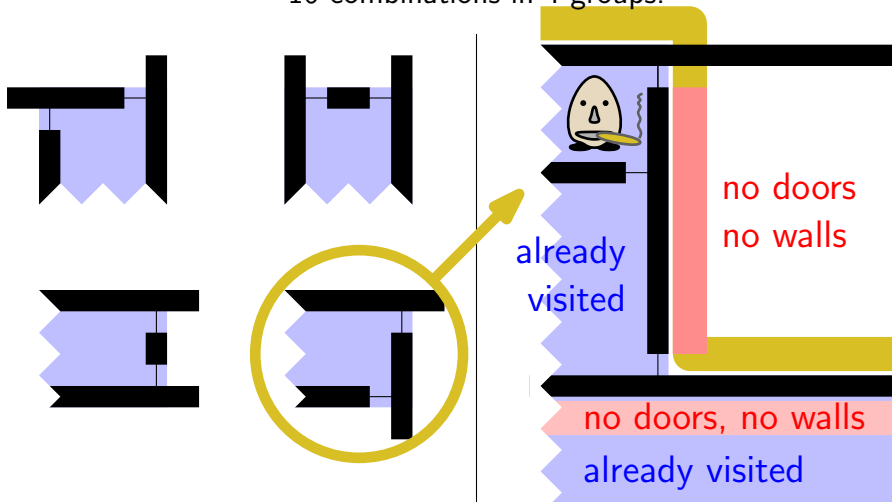


## guarding one-sided museums

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:

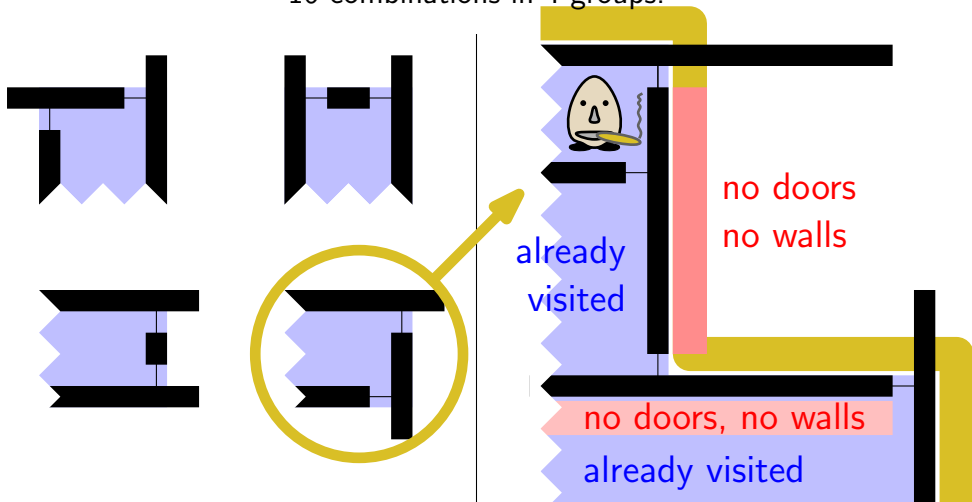


## guarding one-sided museums

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:

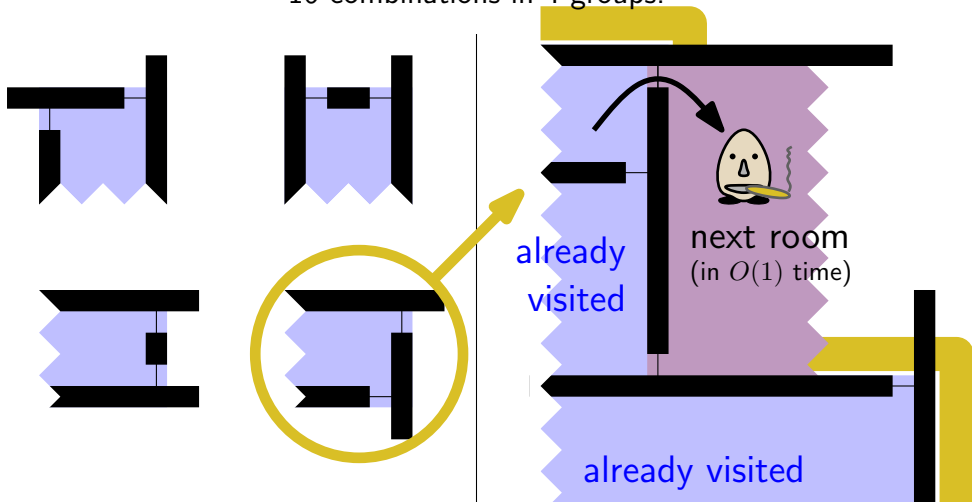


## guarding one-sided museums

Visit rooms in order s.t. • inner walls on boundary of any prefix form desc. staircase;  
• every room has a door to the next room

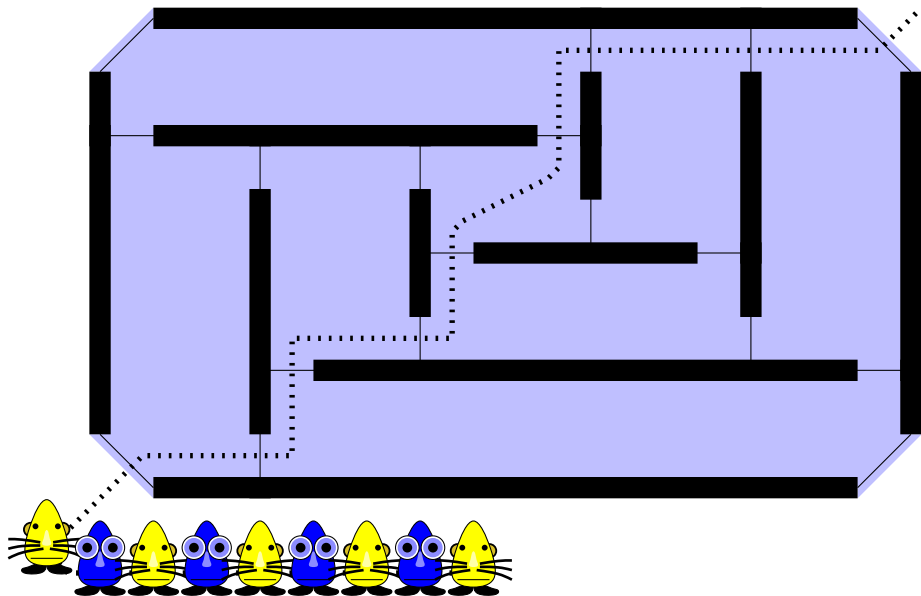
Proof: there is always a next room to go to, until reaching last = right back room

16 combinations in 4 groups:



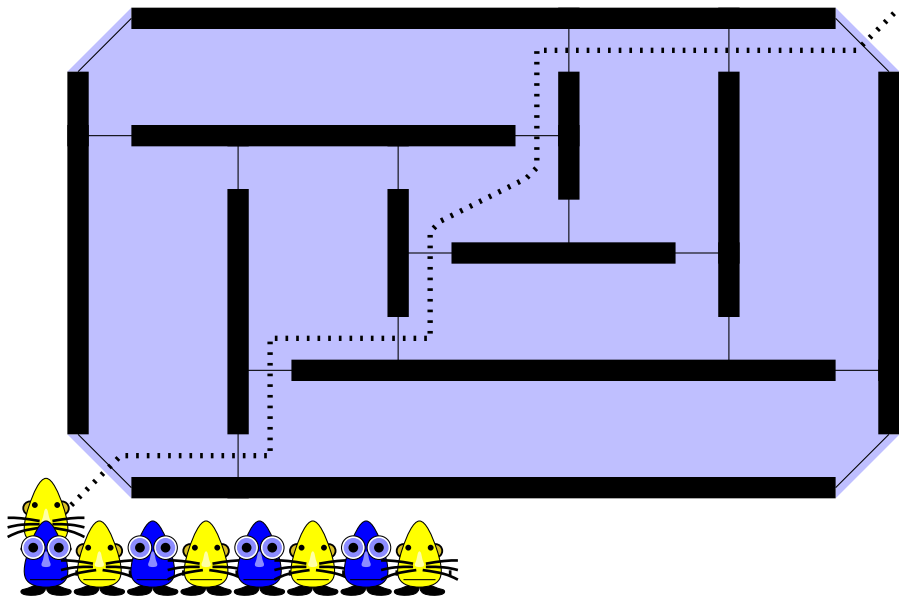
## guarding one-sided museums

---



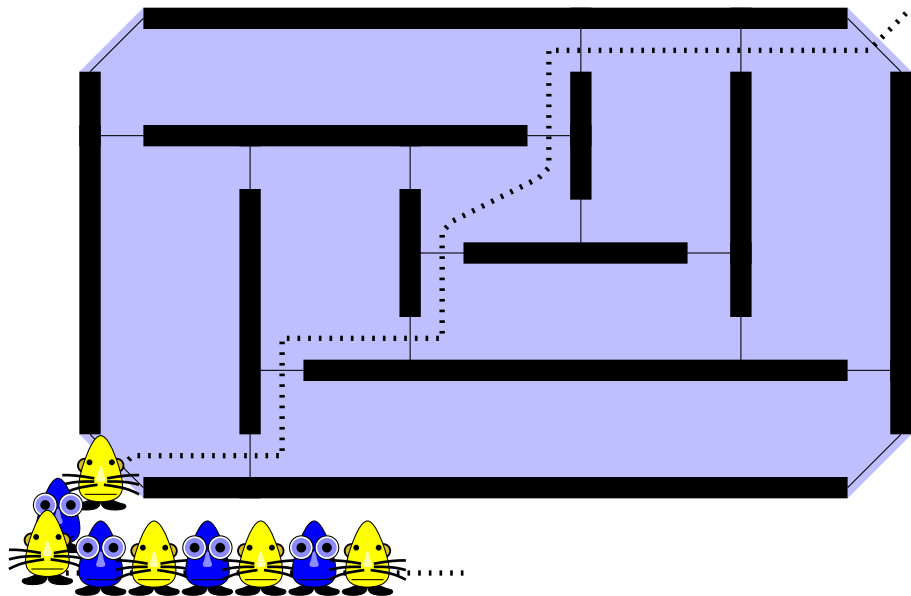
## guarding one-sided museums

---



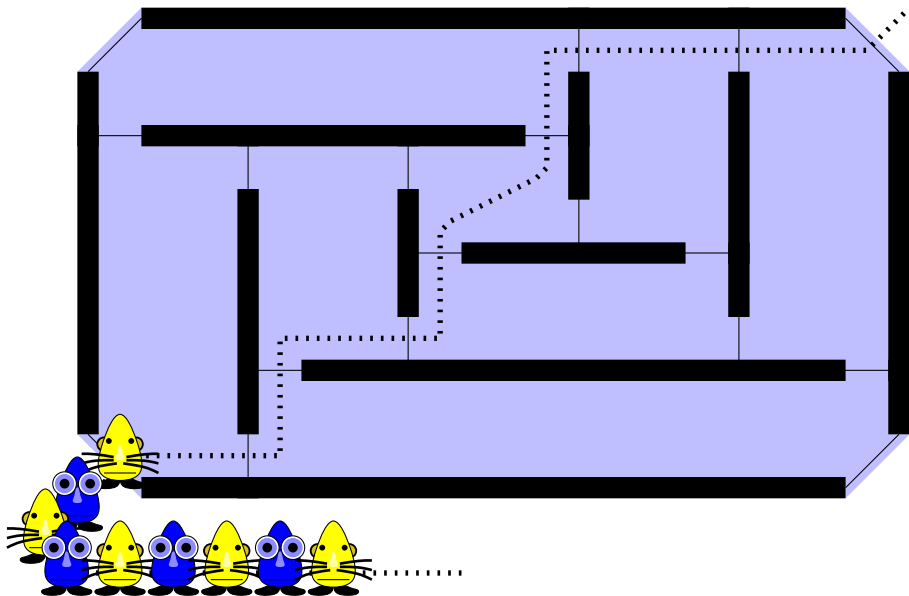


## guarding one-sided museums



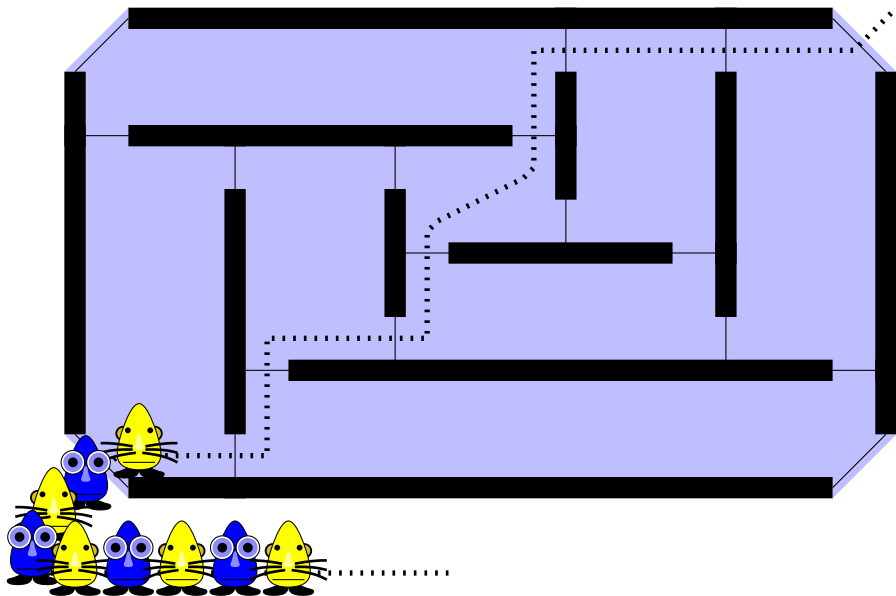


guarding one-sided museums



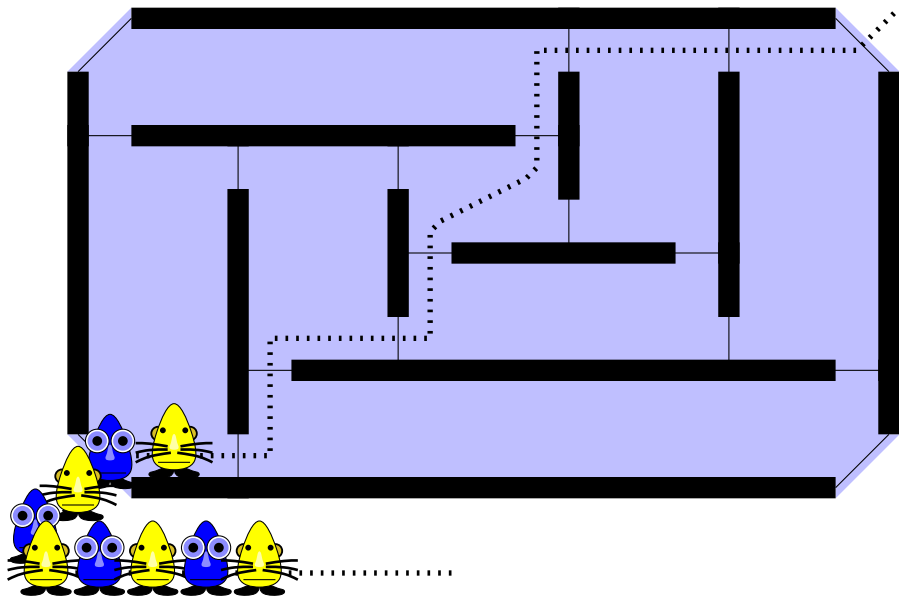
## guarding one-sided museums

---



## guarding one-sided museums

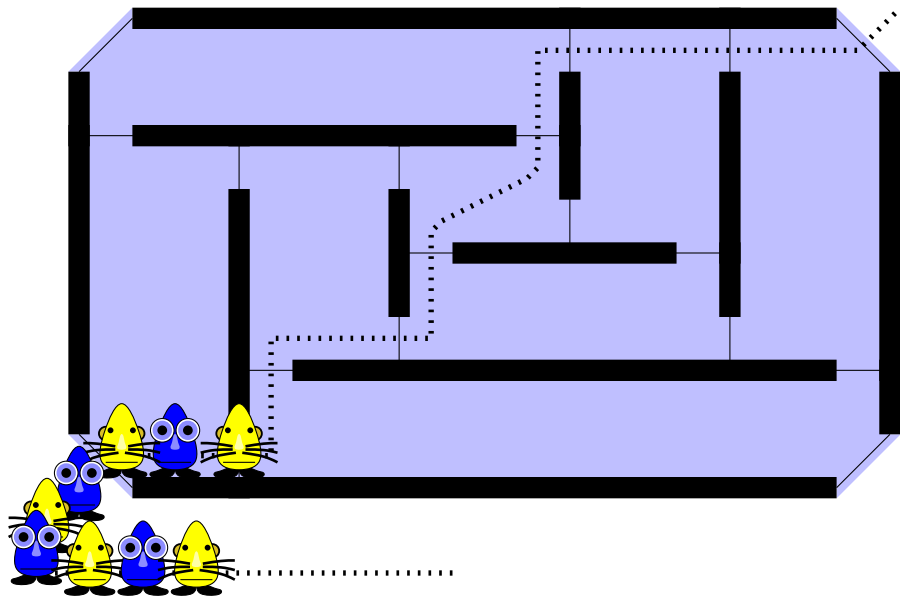
---





## guarding one-sided museums

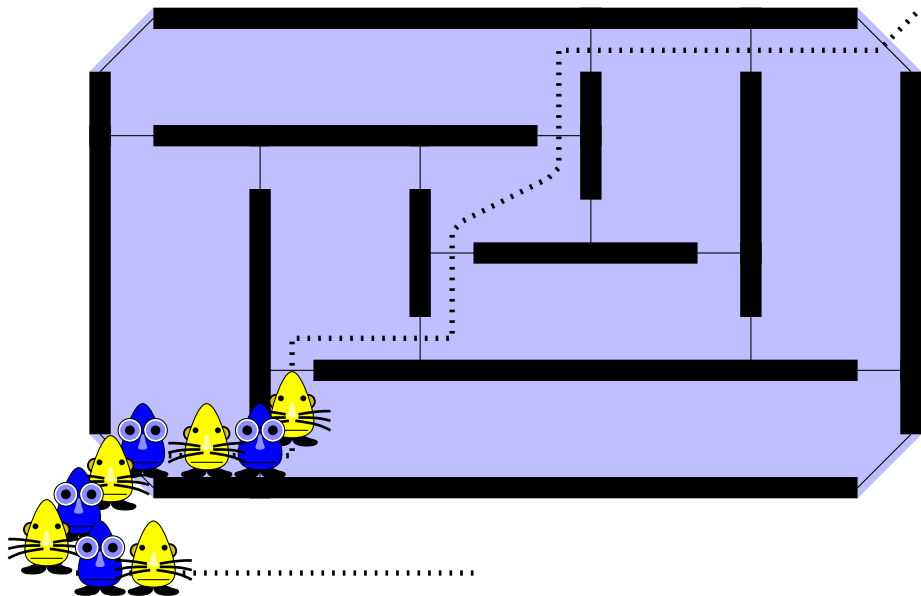
---





## guarding one-sided museums

---

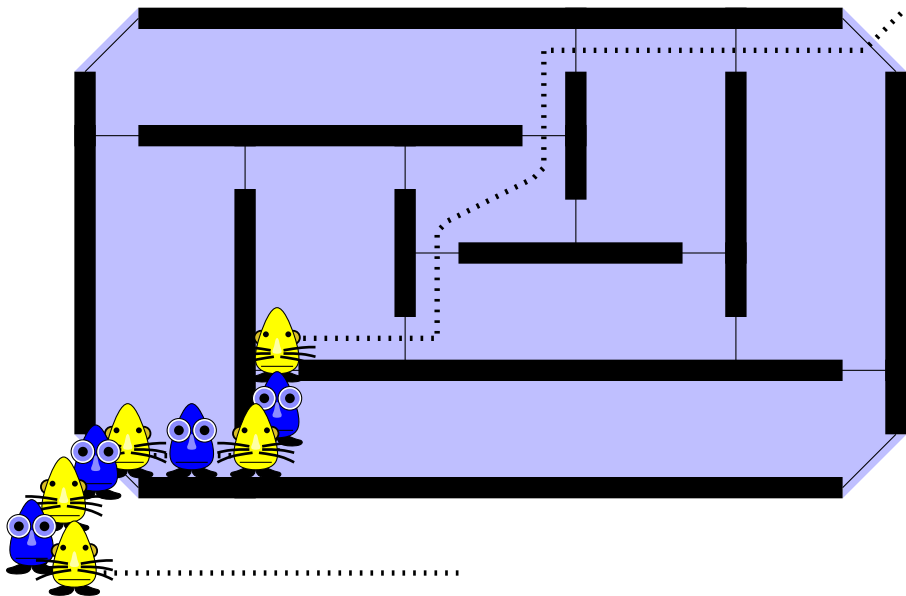






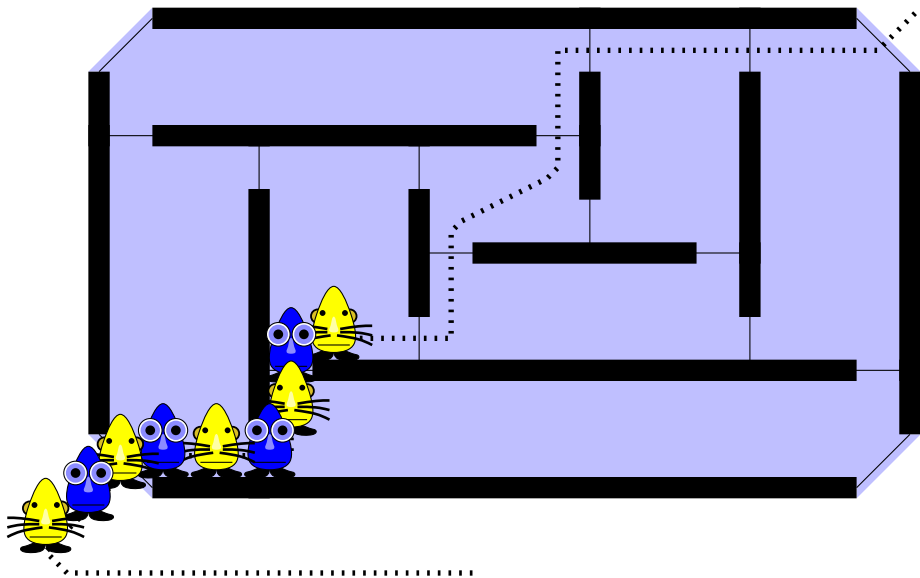
## guarding one-sided museums

---

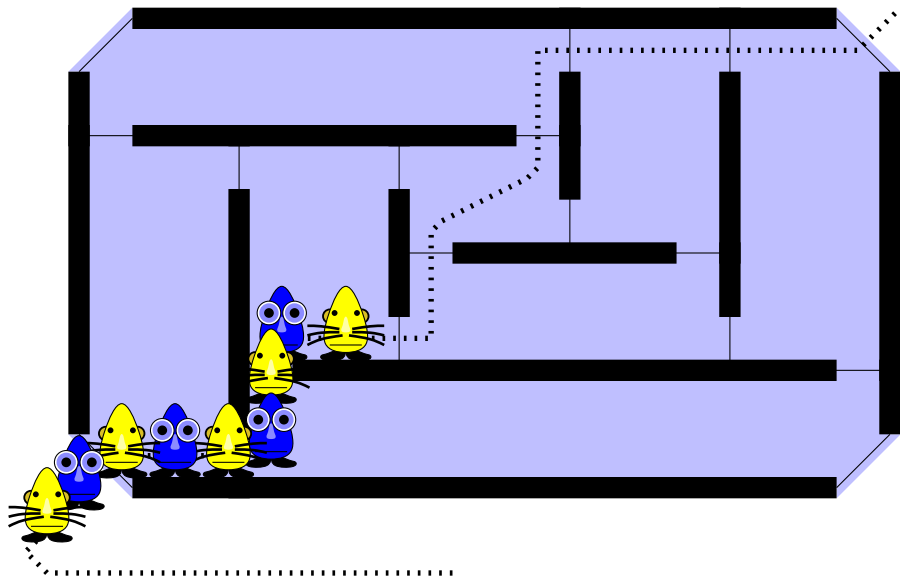




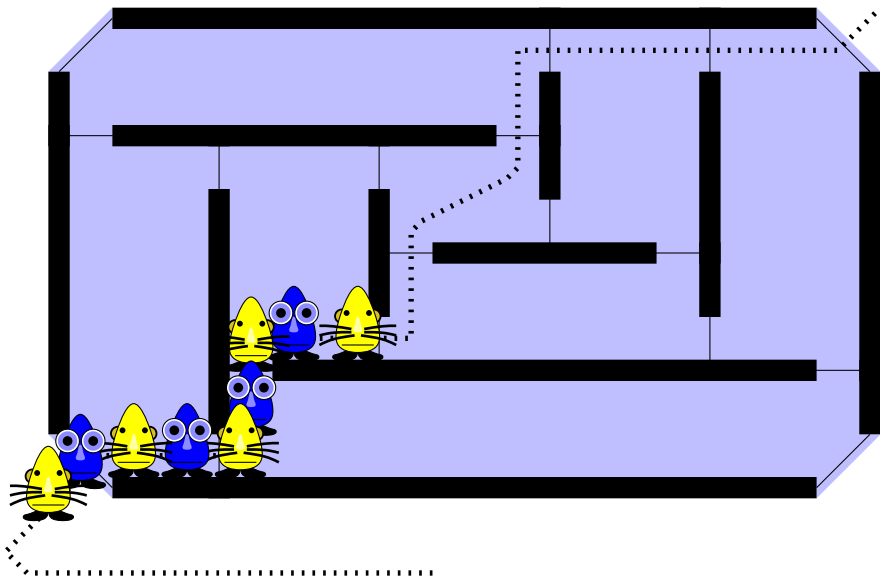
## guarding one-sided museums



## guarding one-sided museums



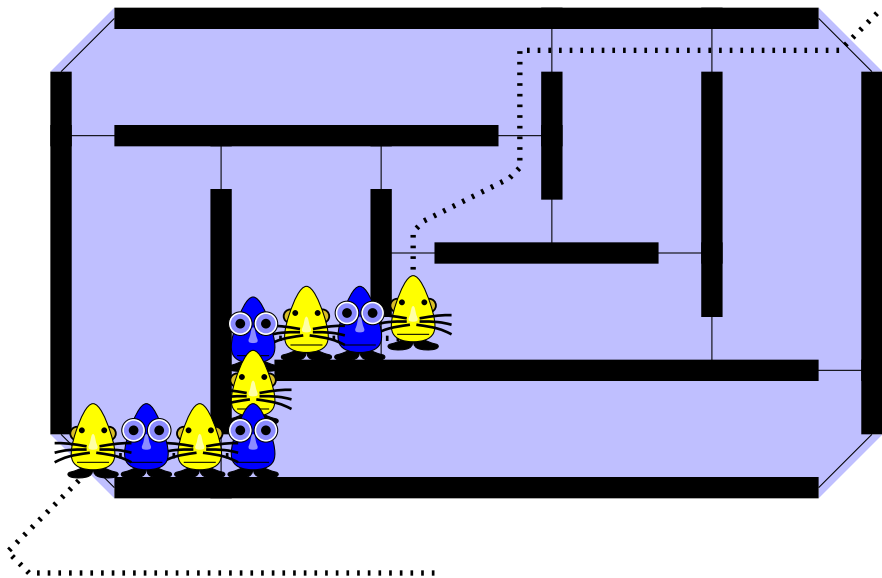
## guarding one-sided museums



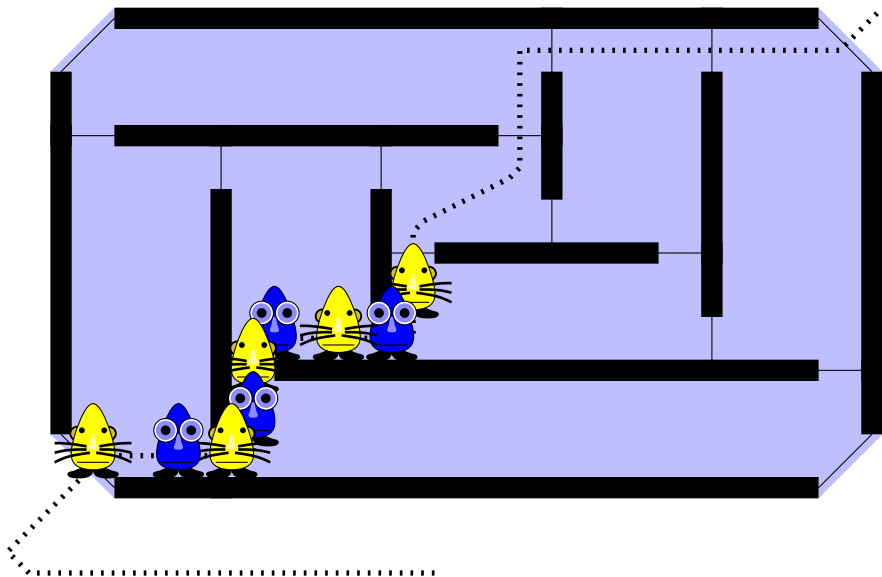


## guarding one-sided museums

---

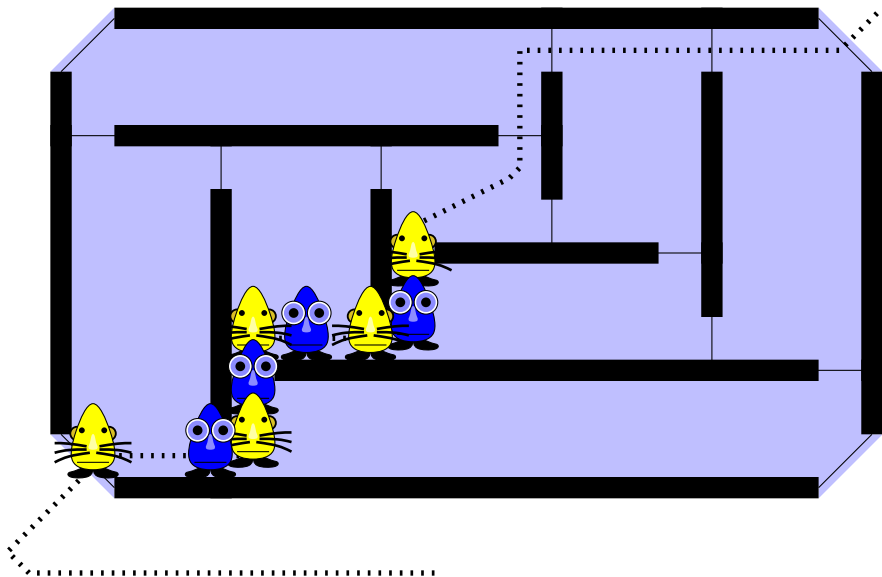


guarding one-sided museums

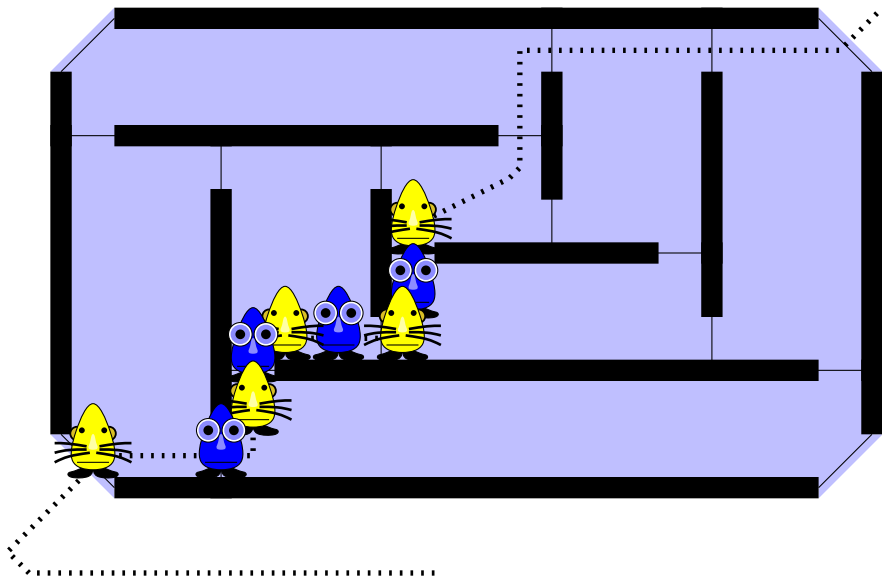




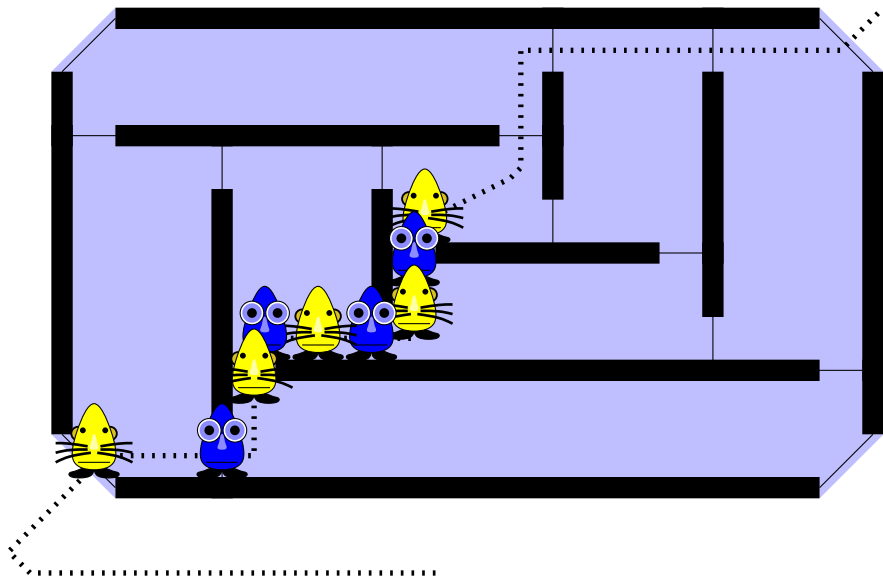
## guarding one-sided museums



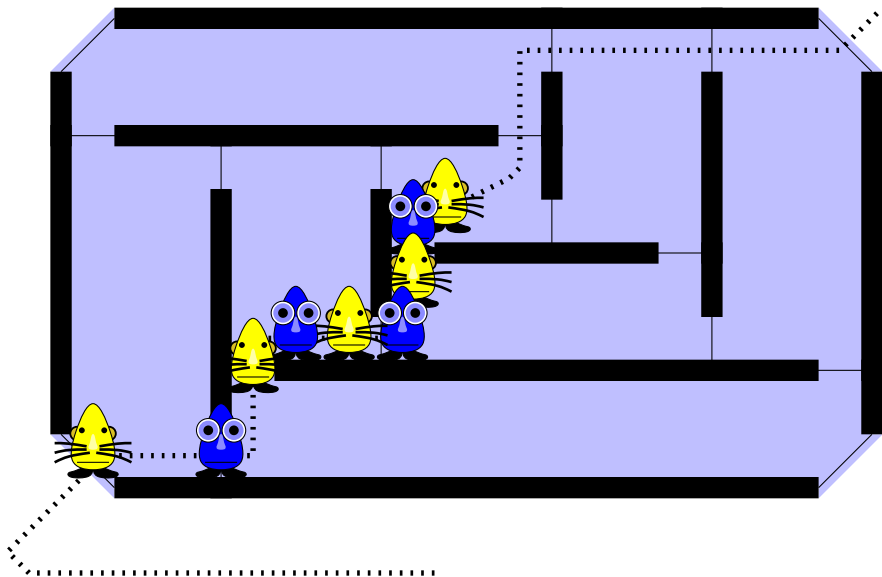
## guarding one-sided museums



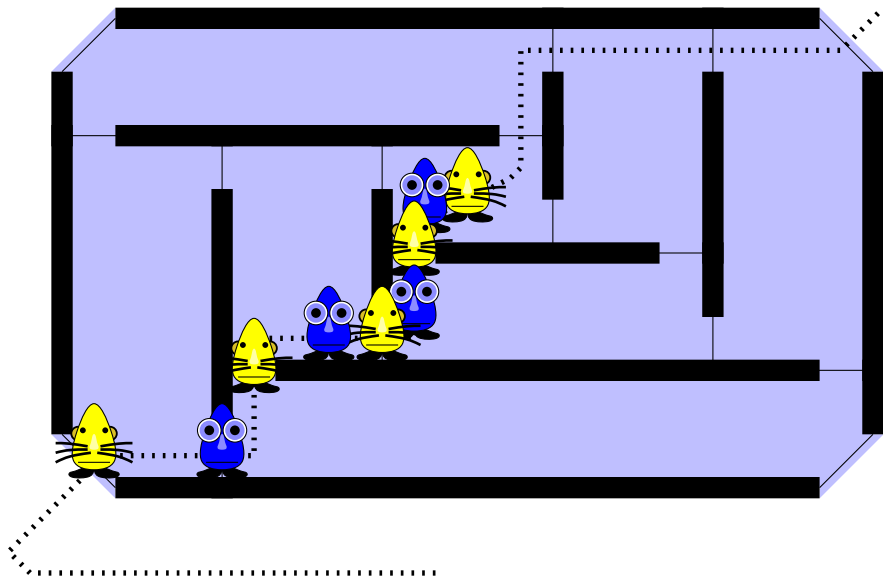
## guarding one-sided museums



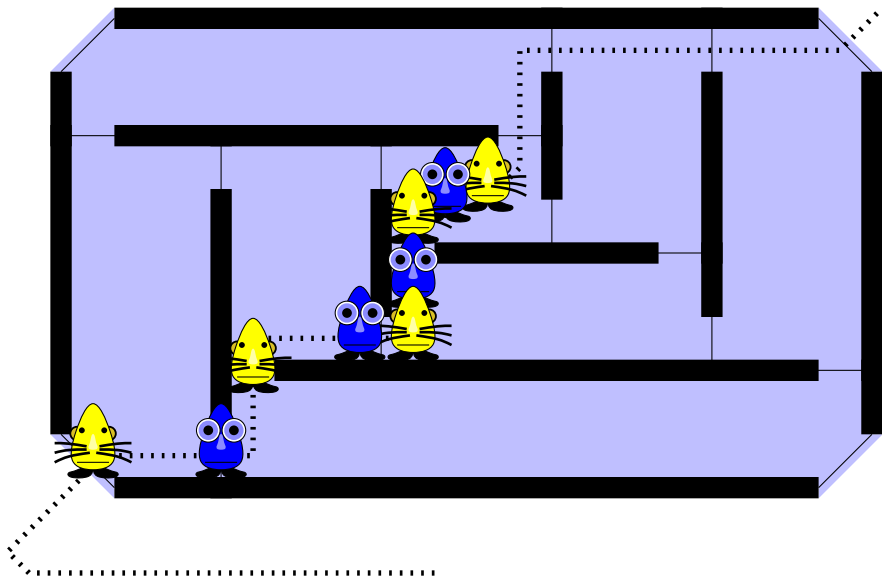
## guarding one-sided museums



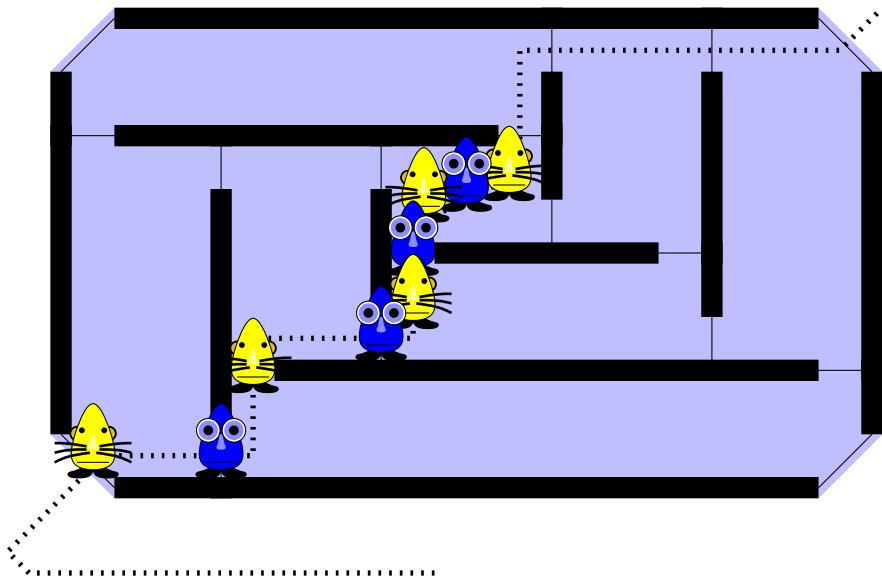
## guarding one-sided museums



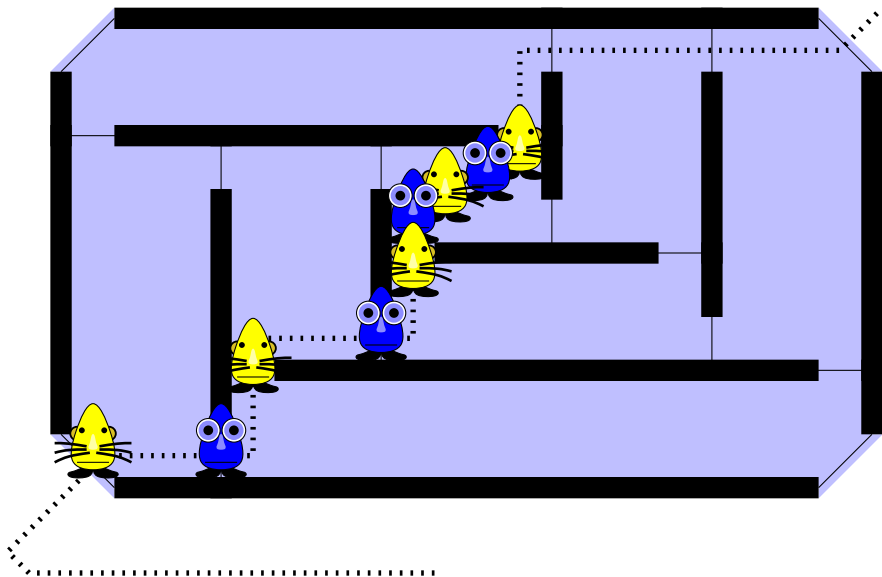
guarding one-sided museums



## guarding one-sided museums

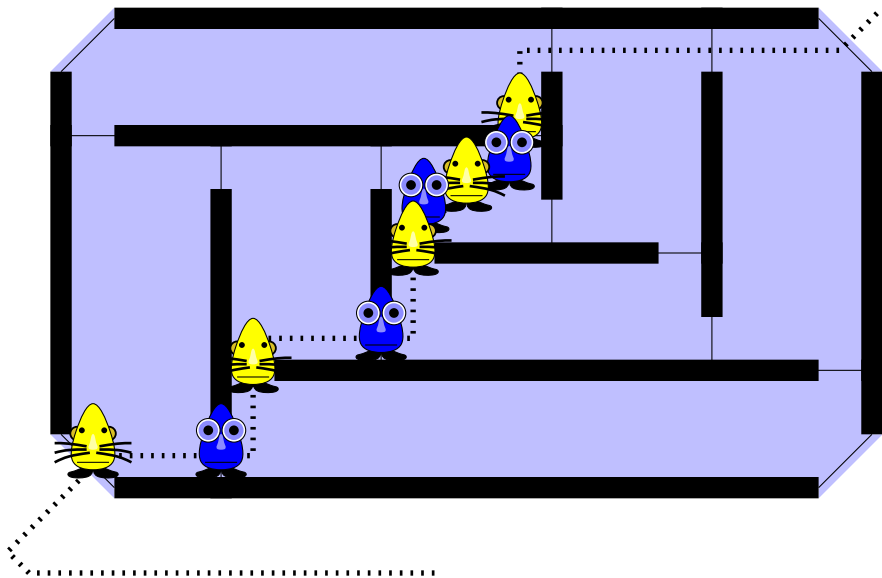


## guarding one-sided museums



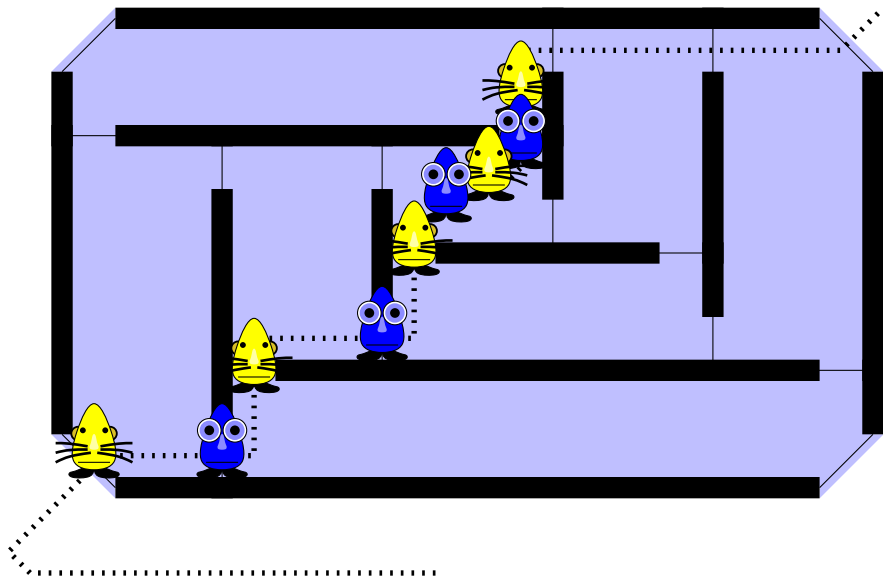


## guarding one-sided museums

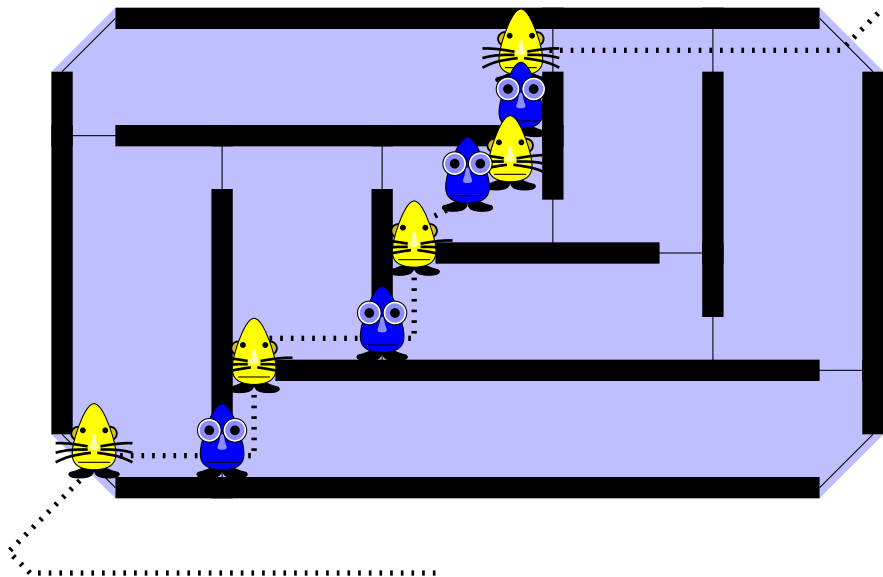


## guarding one-sided museums

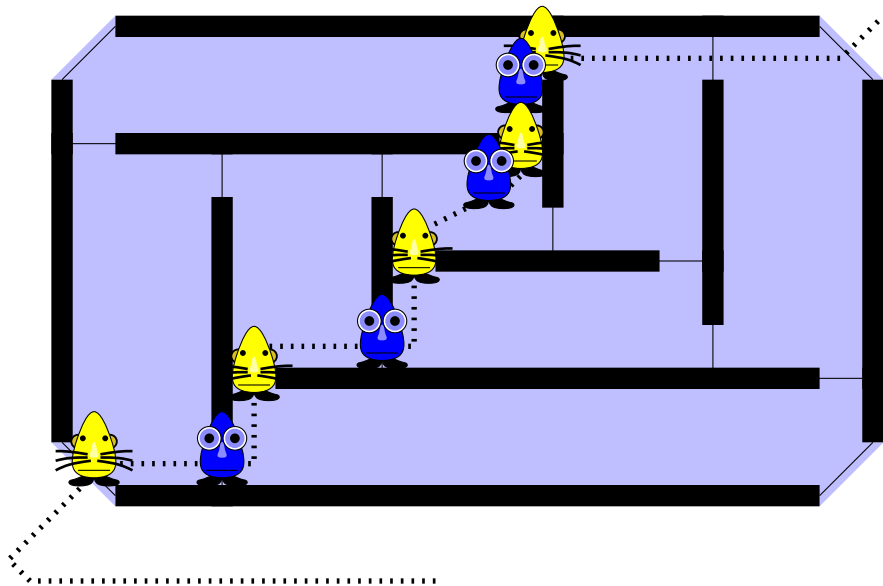
---



## guarding one-sided museums

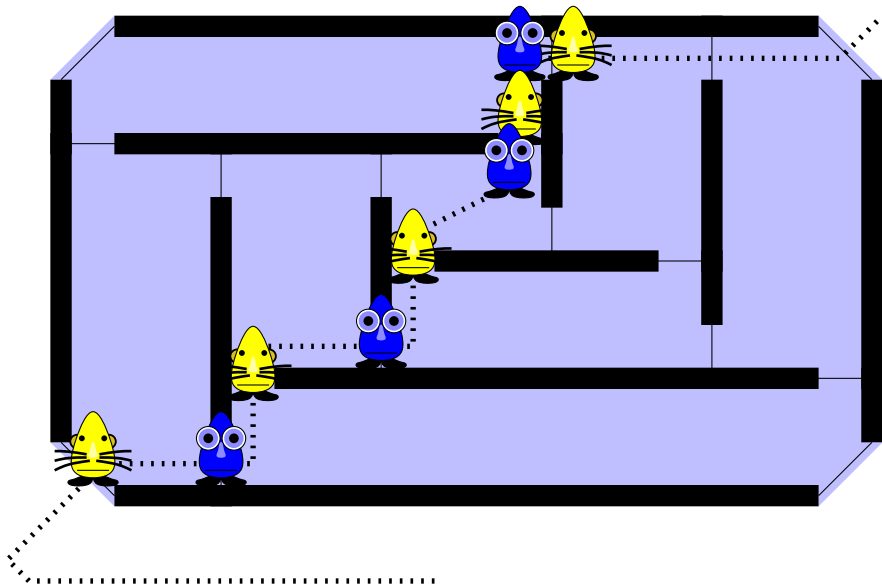


## guarding one-sided museums

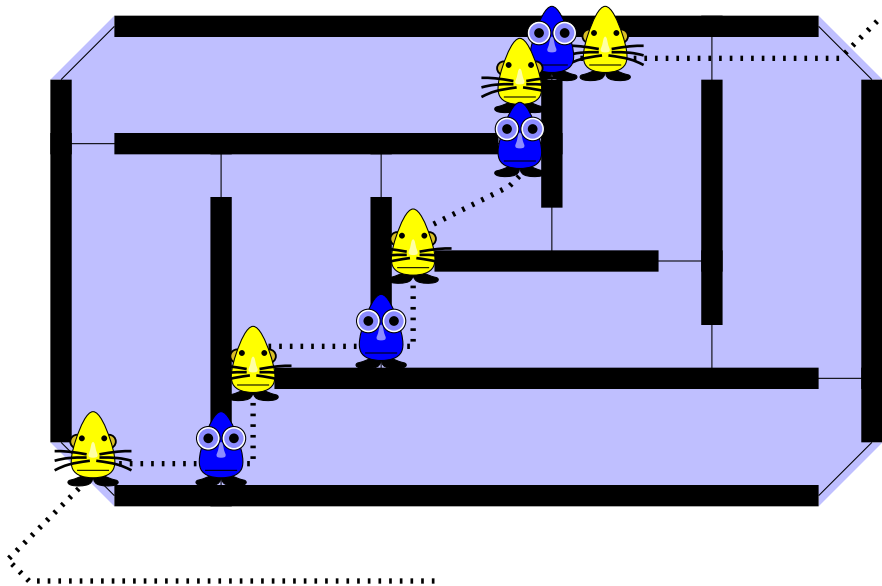


## guarding one-sided museums

---

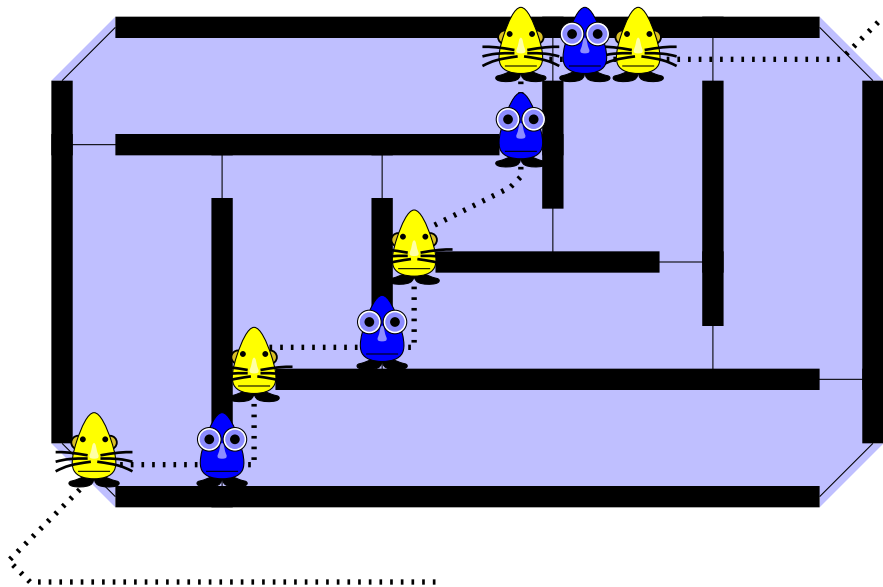


## guarding one-sided museums

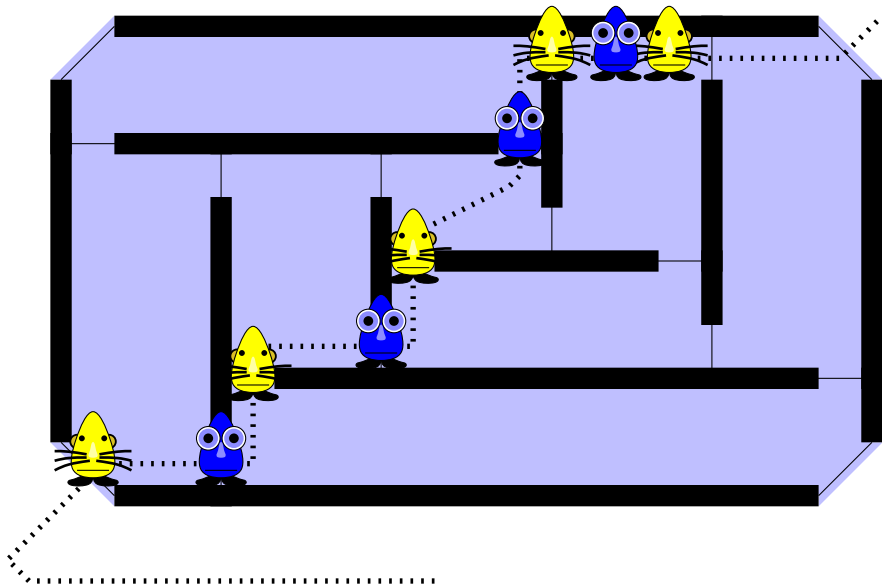


## guarding one-sided museums

---

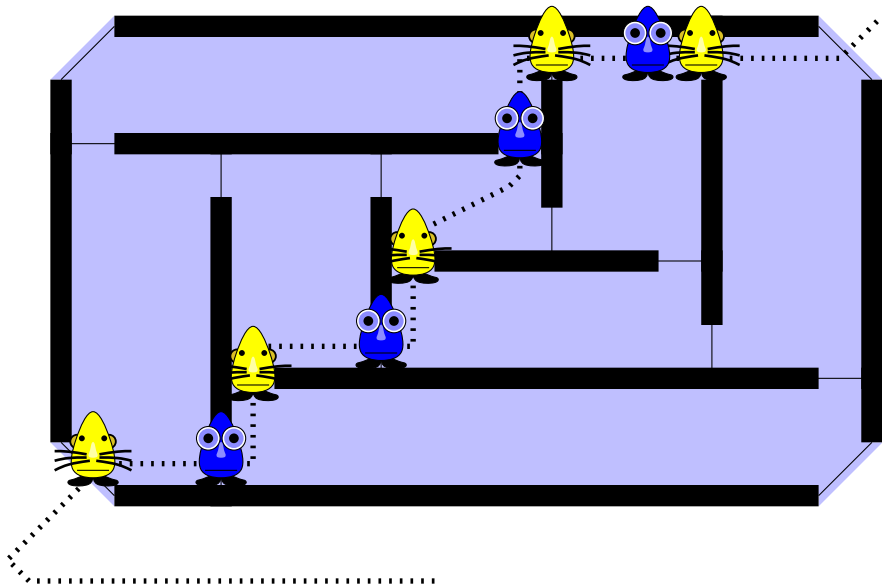


## guarding one-sided museums

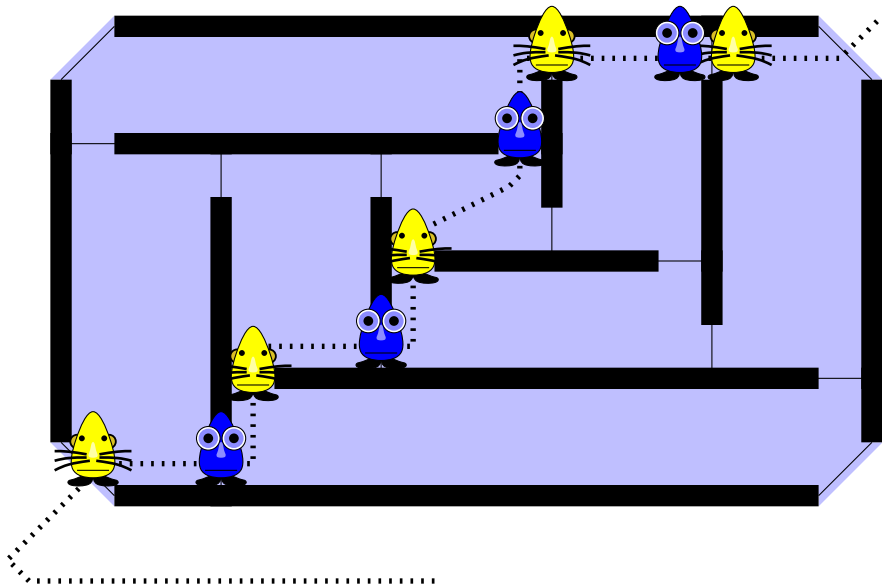




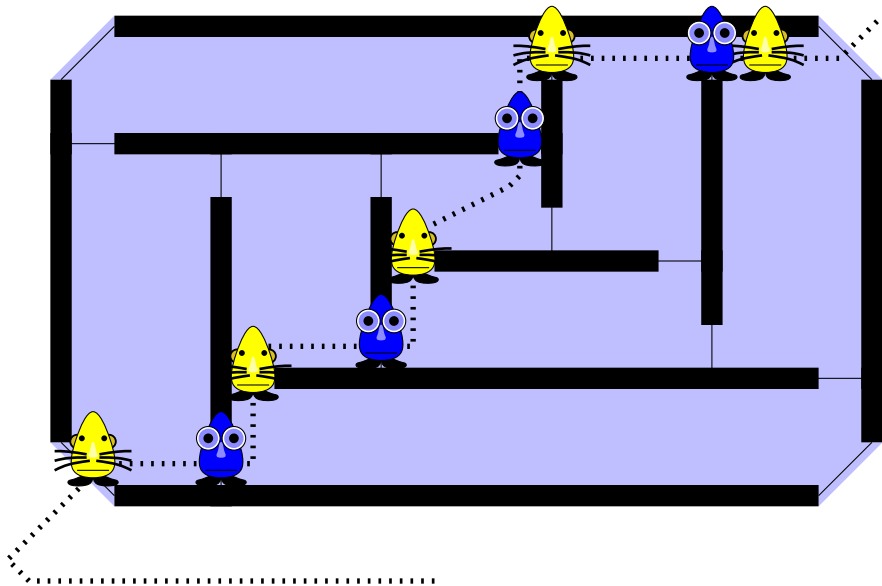
guarding one-sided museums



## guarding one-sided museums

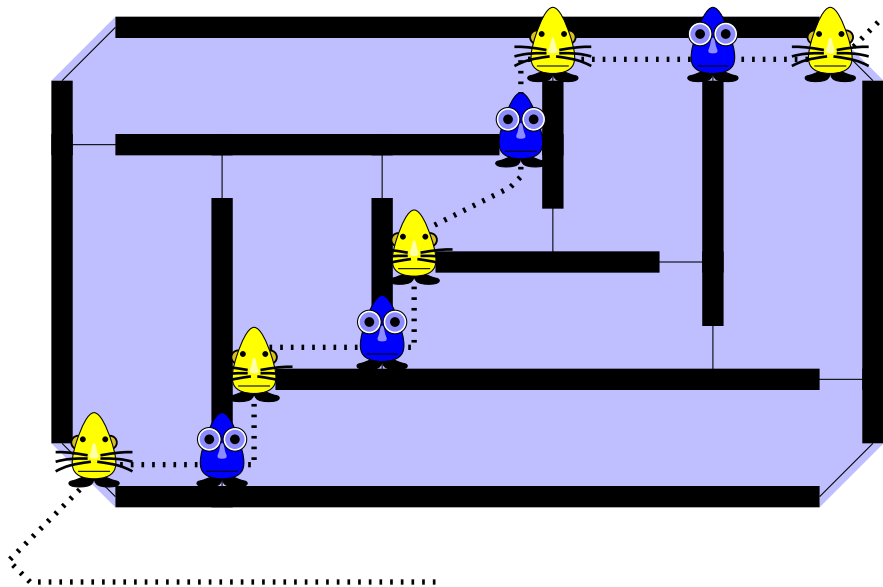


## guarding one-sided museums

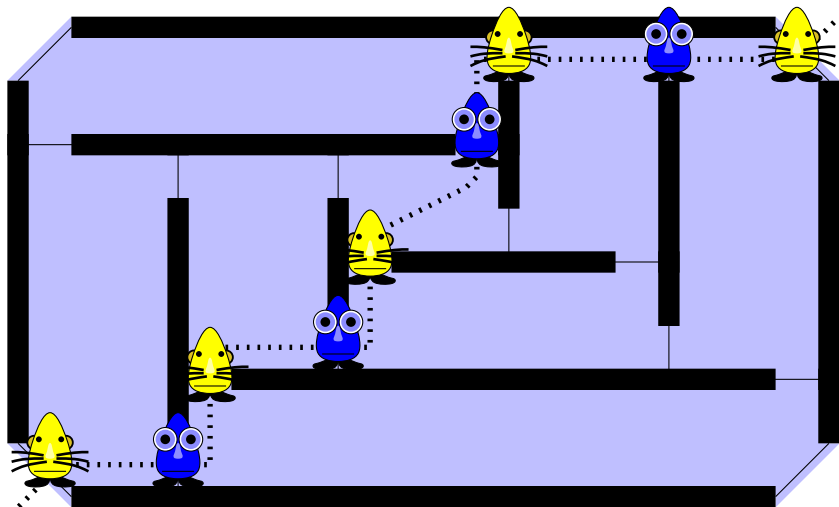




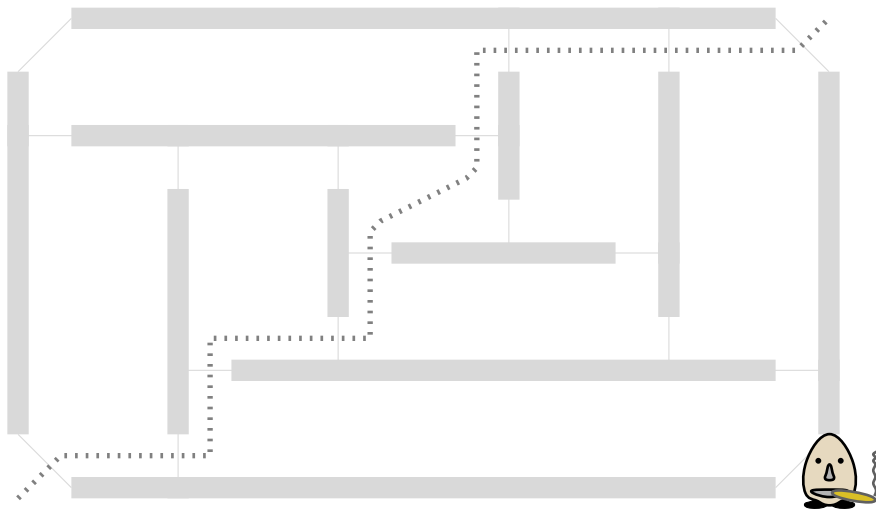
## guarding one-sided museums



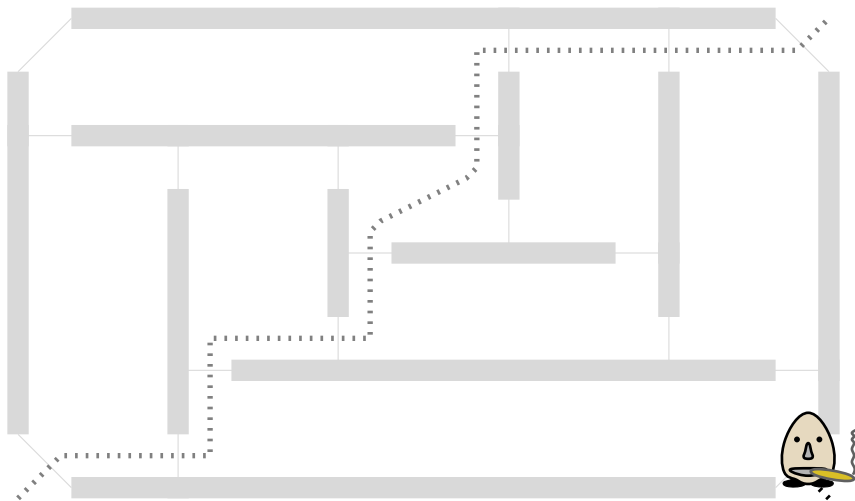
## guarding one-sided museums



each room has feel-all at entrance and see-all at exit,  
or the other way around

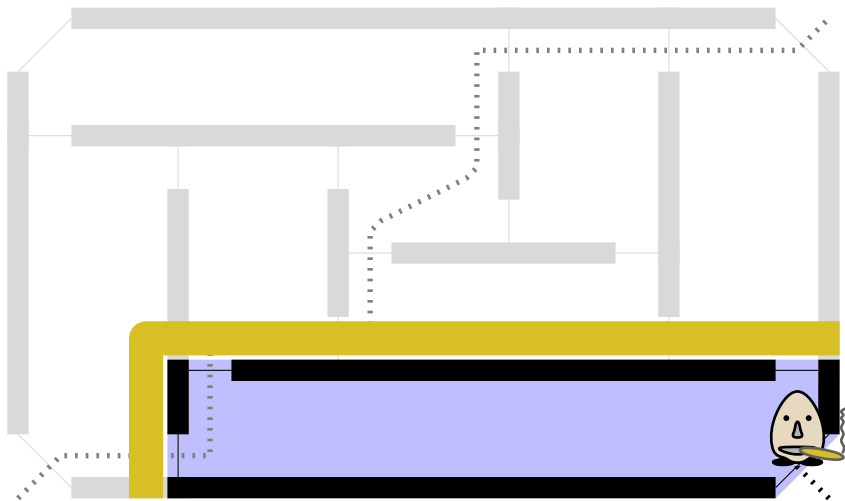


Repeat from right front to left back corner  
(boundary is *ascending* staircase)

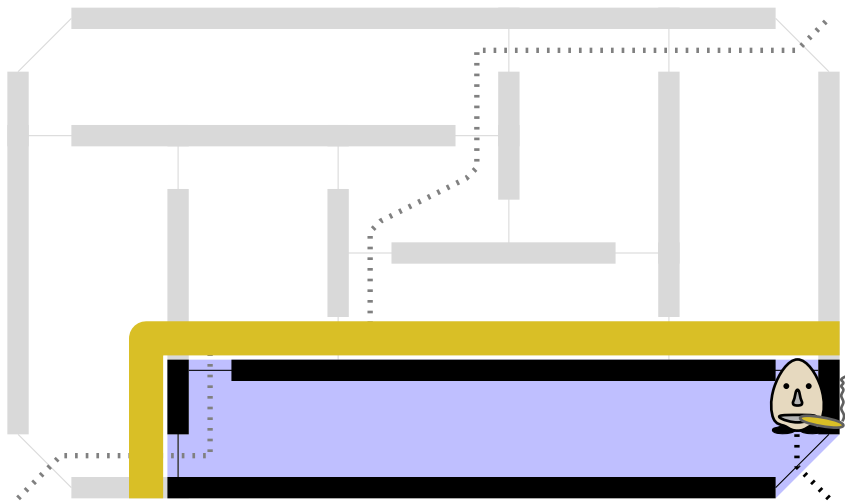


Repeat from right front to left back corner  
(boundary is *ascending* staircase)

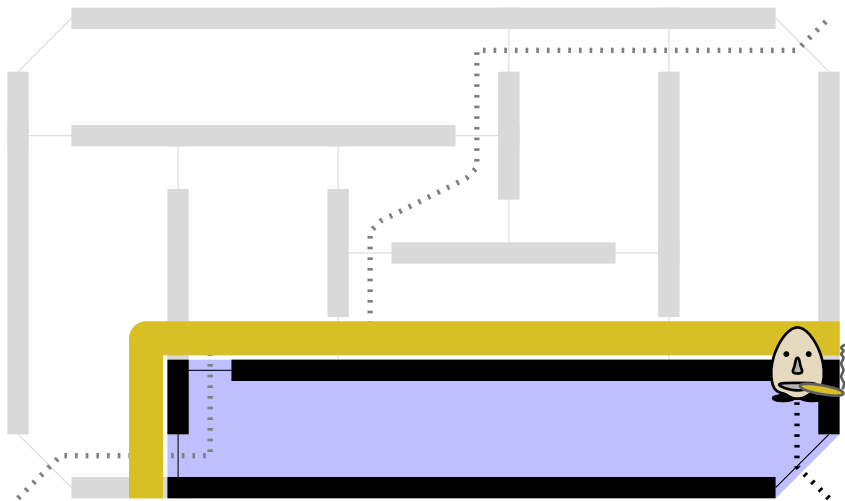




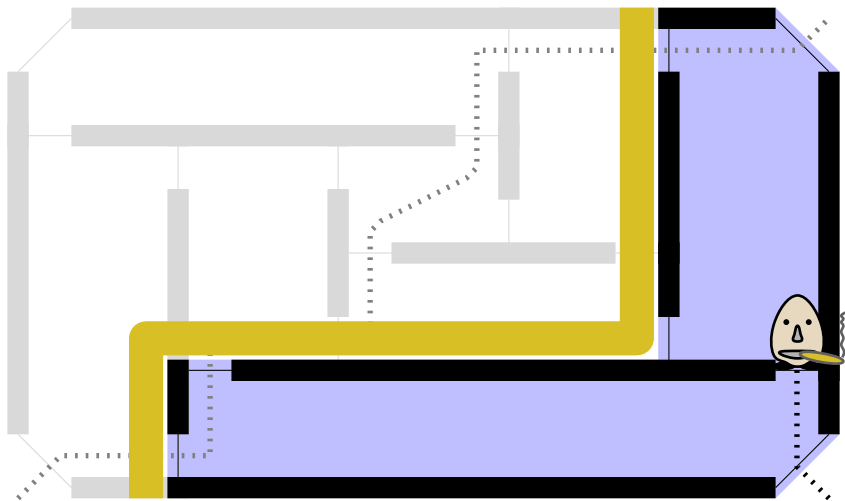
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



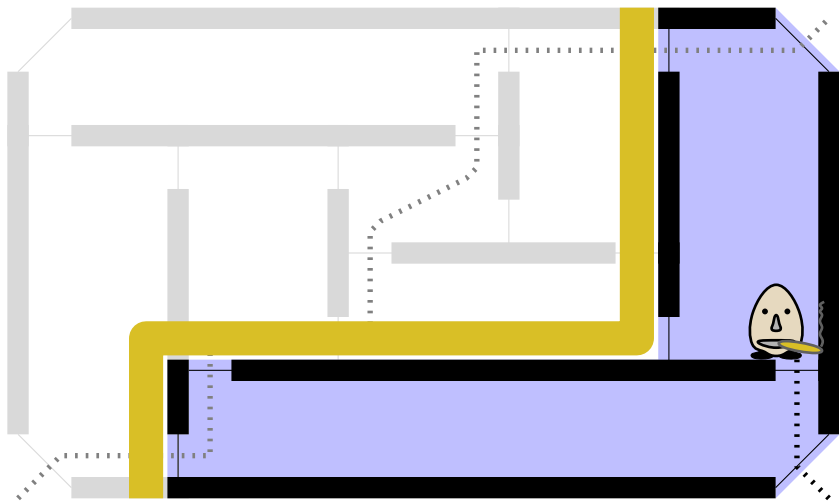
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



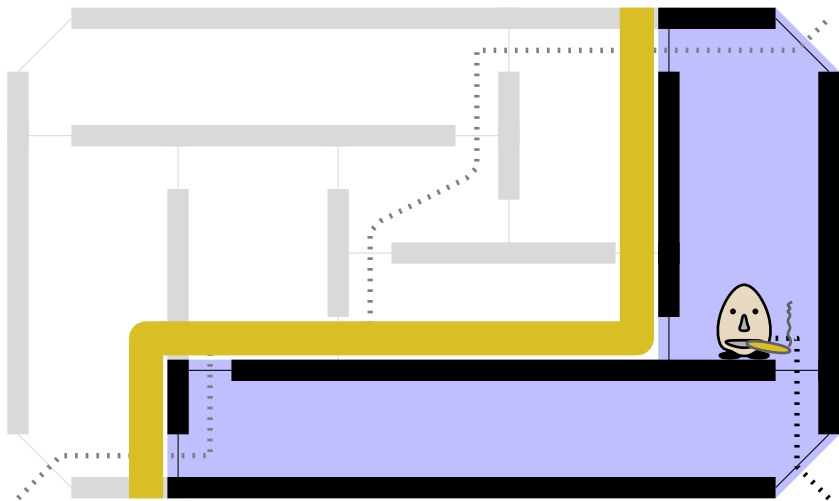
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



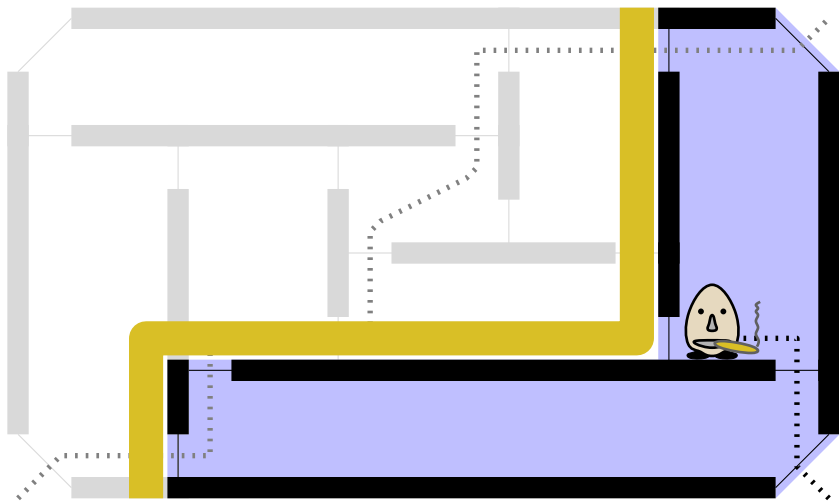
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



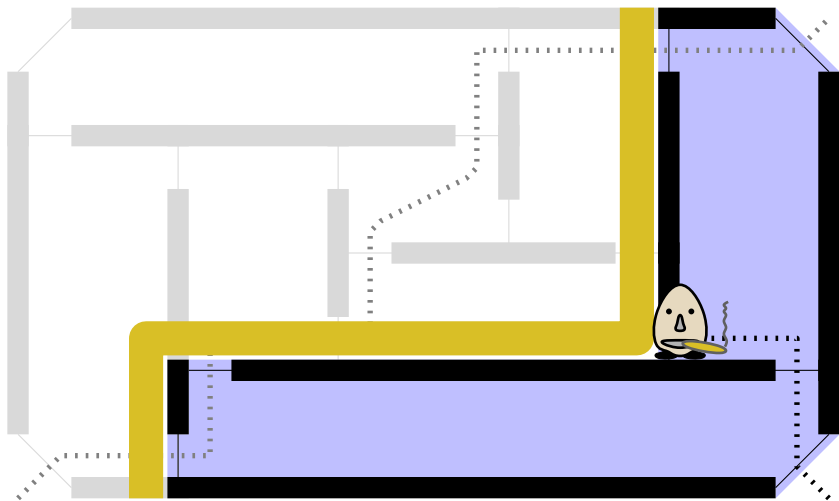
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



Repeat from right front to left back corner  
(boundary is *ascending* staircase)

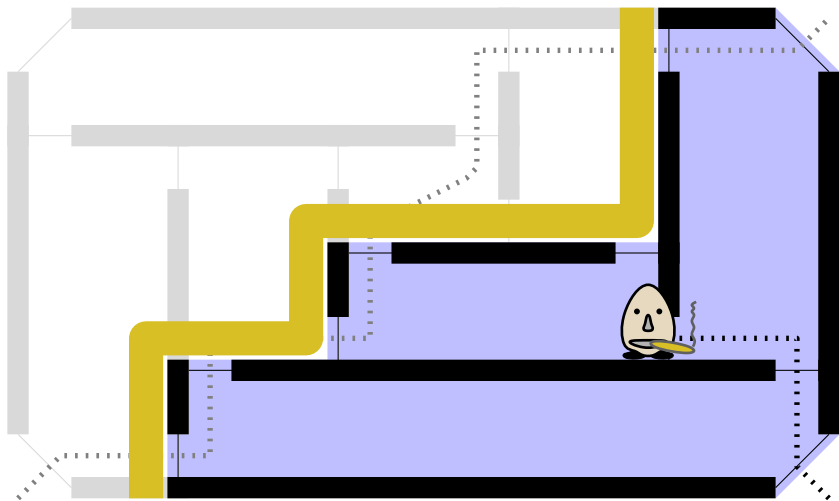


Repeat from right front to left back corner  
(boundary is *ascending* staircase)

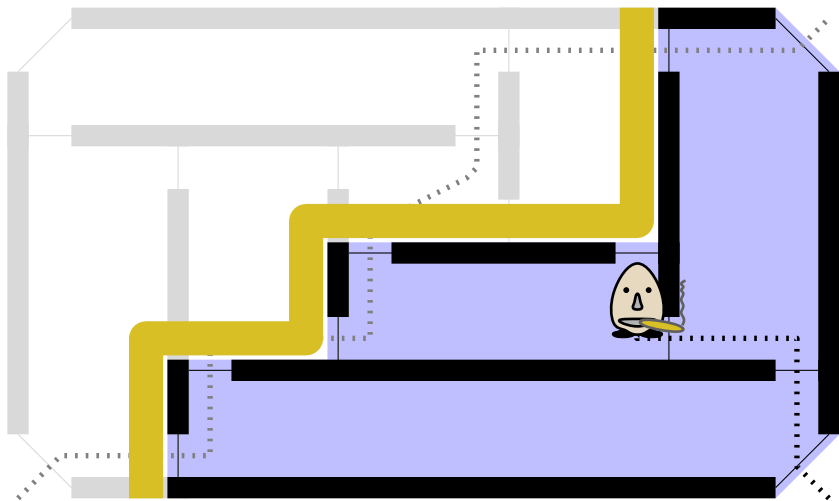


Repeat from right front to left back corner  
(boundary is *ascending* staircase)

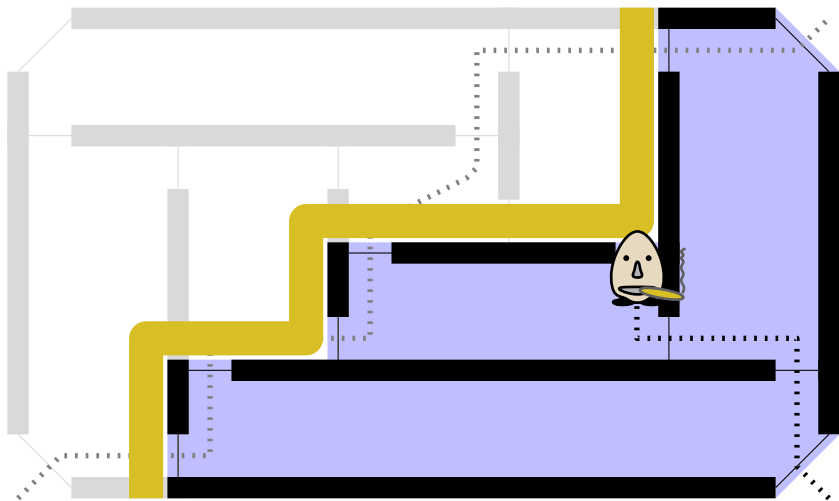




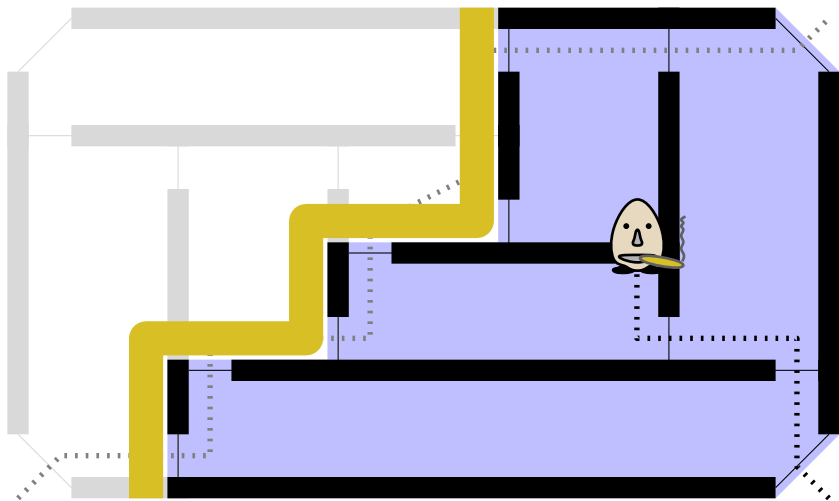
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



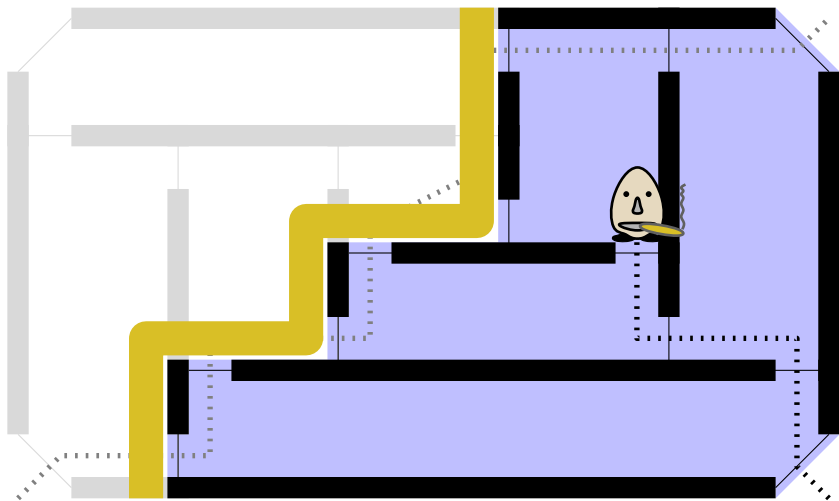
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



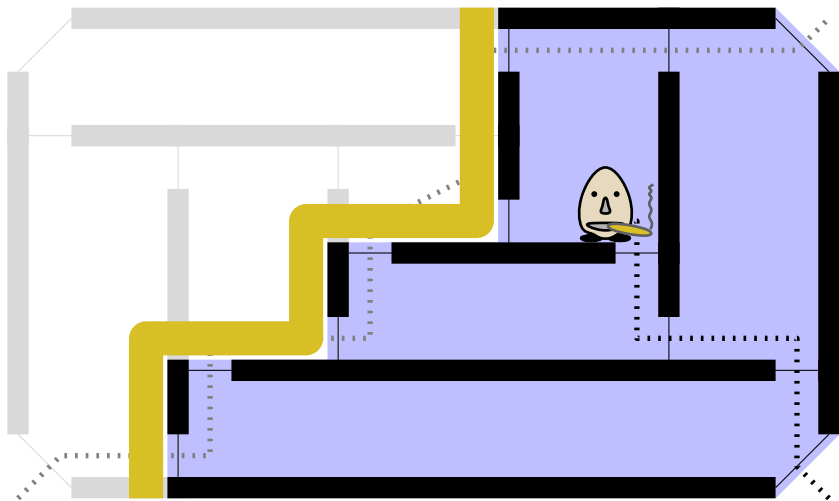
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



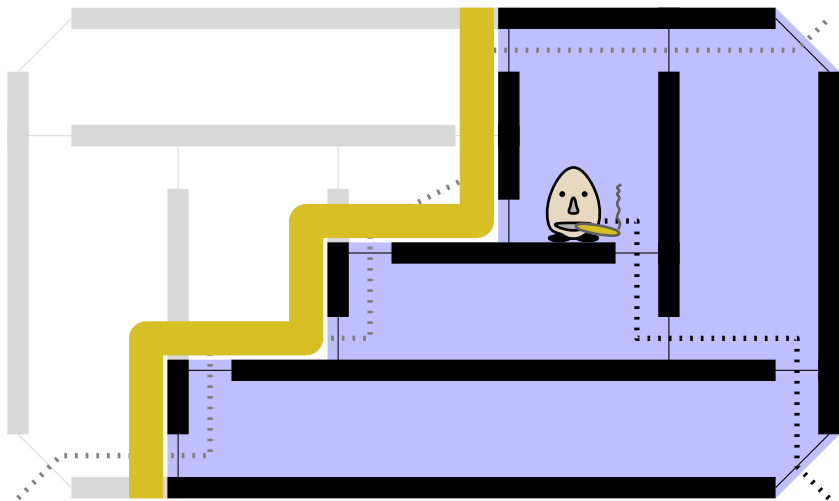
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



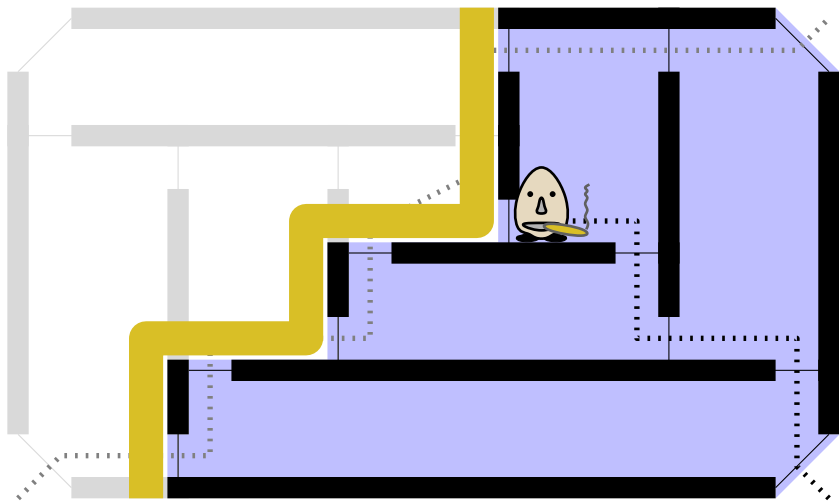
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



Repeat from right front to left back corner  
(boundary is *ascending* staircase)

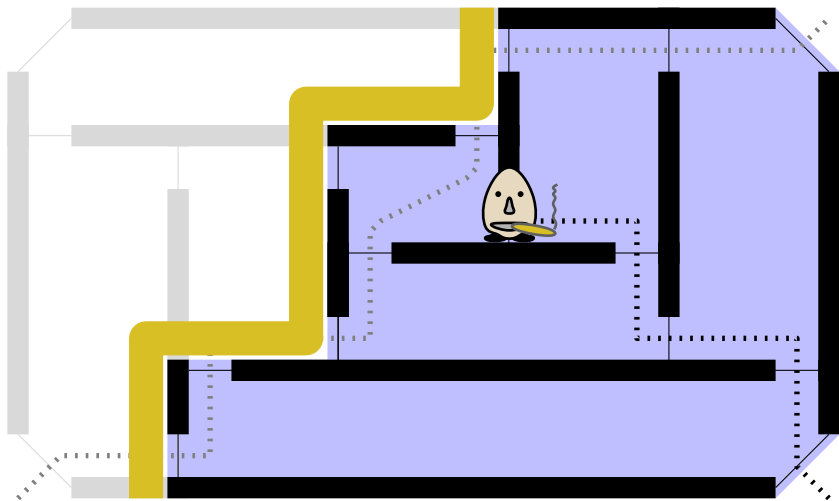


Repeat from right front to left back corner  
(boundary is *ascending* staircase)

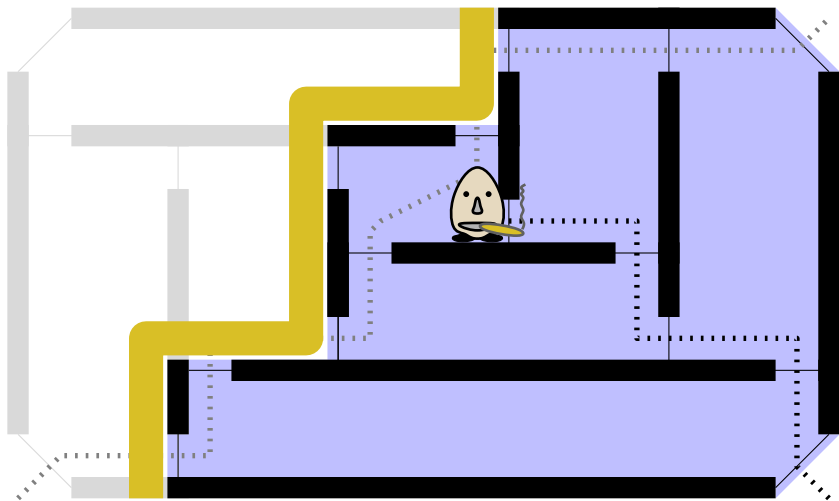


Repeat from right front to left back corner  
(boundary is *ascending* staircase)

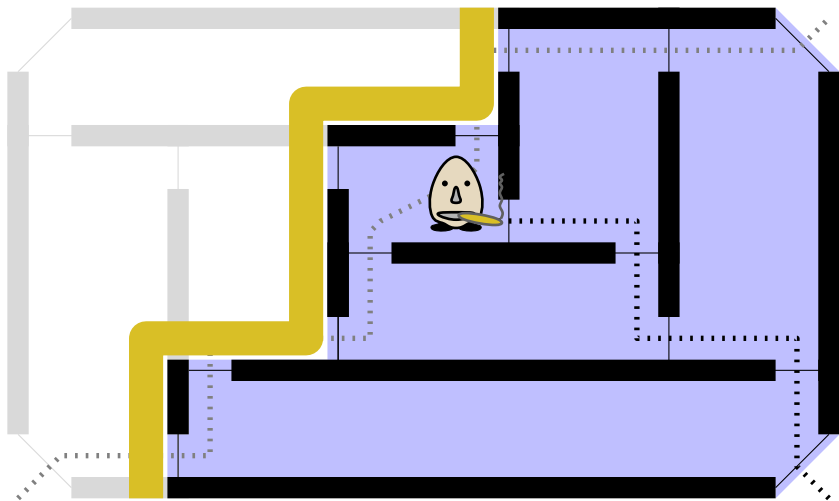




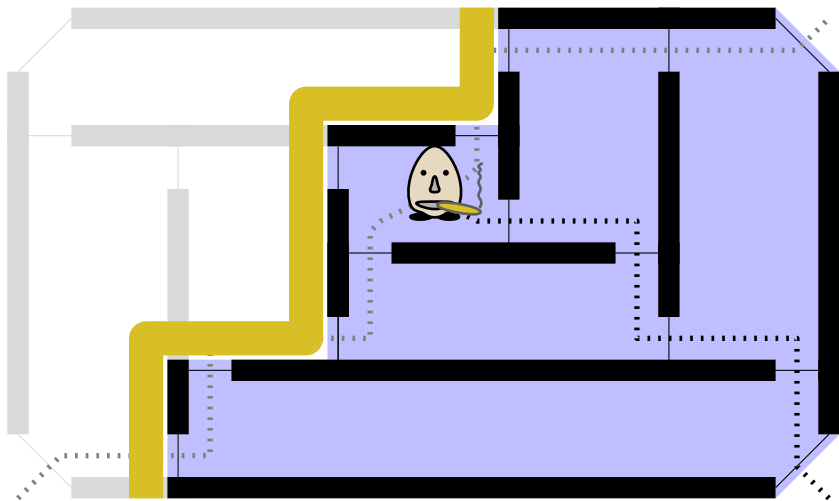
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



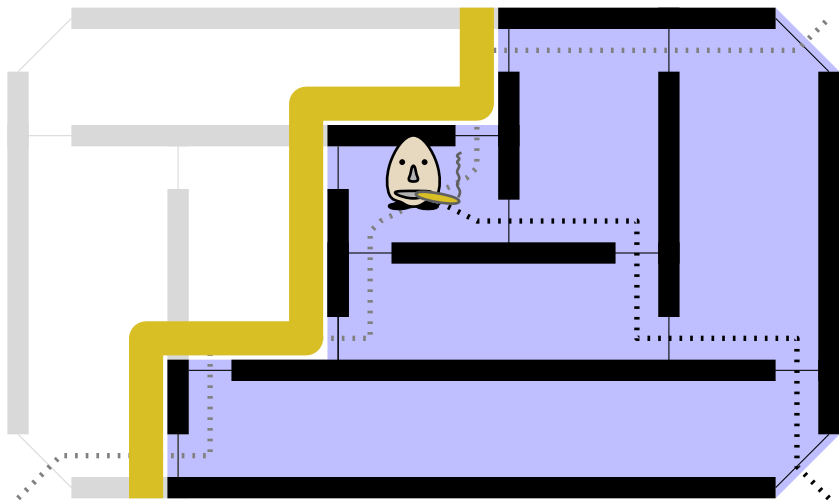
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



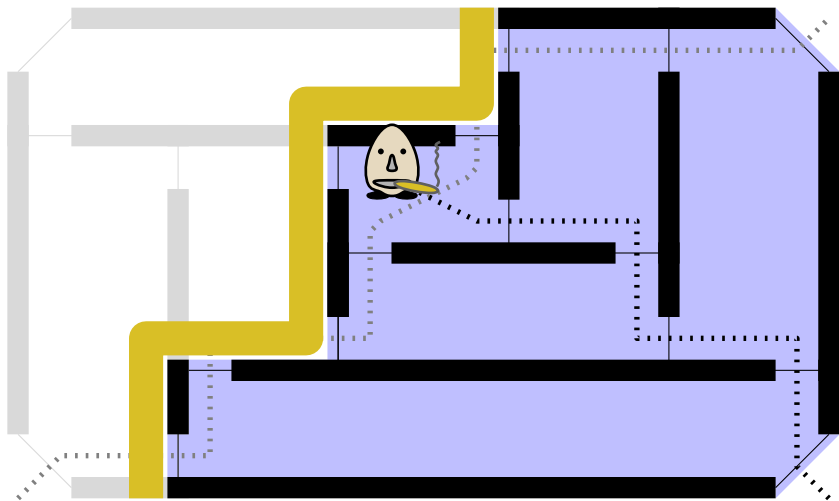
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



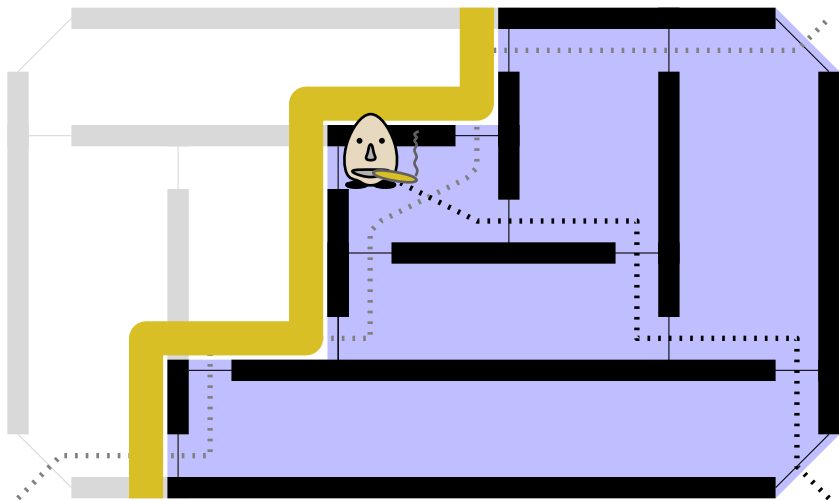
Repeat from right front to left back corner  
(boundary is *ascending* staircase)



Repeat from right front to left back corner  
(boundary is *ascending* staircase)



Repeat from right front to left back corner  
(boundary is *ascending* staircase)



Repeat from right front to left back corner  
(boundary is *ascending* staircase)





















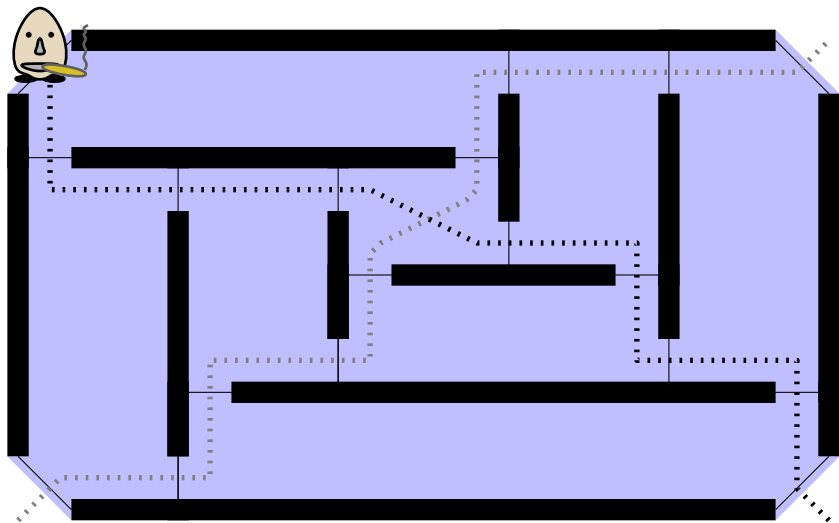




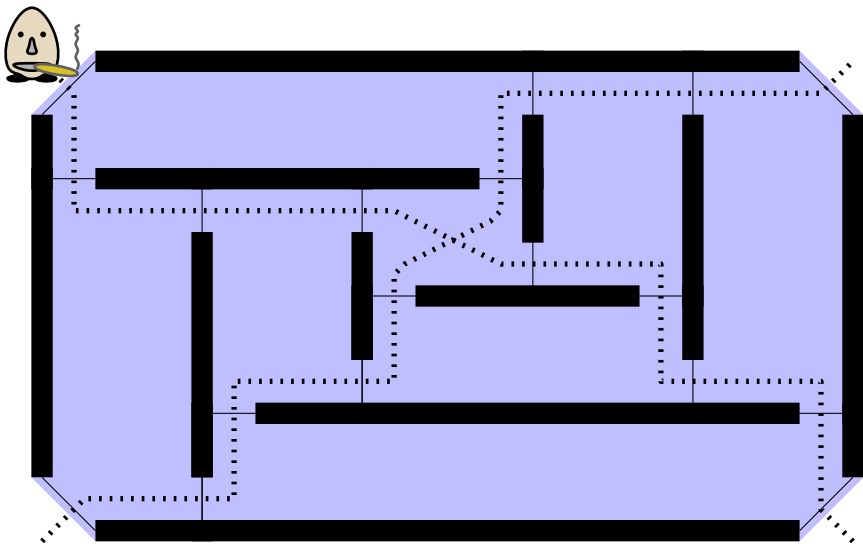






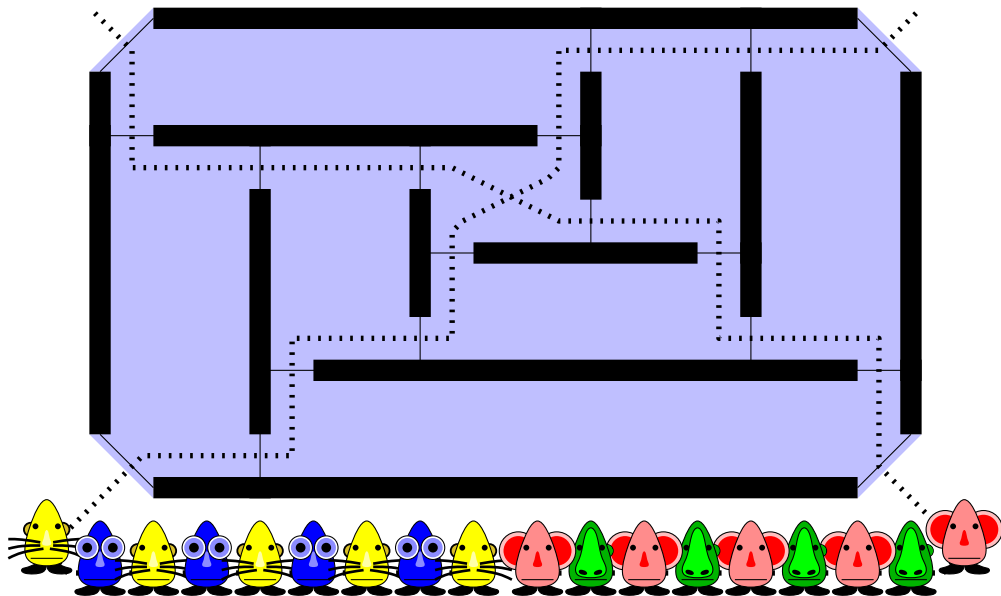


Repeat from right front to left back corner  
(boundary is *ascending* staircase)



These paths do not share doors  
→ every room has four doors: two on each path

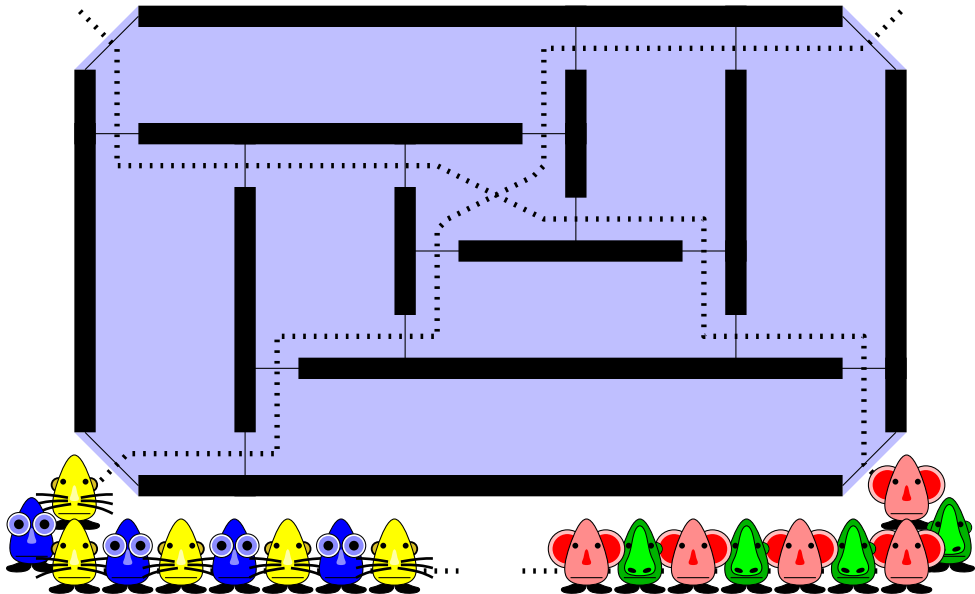
## guarding one-sided museums







guarding one-sided museums





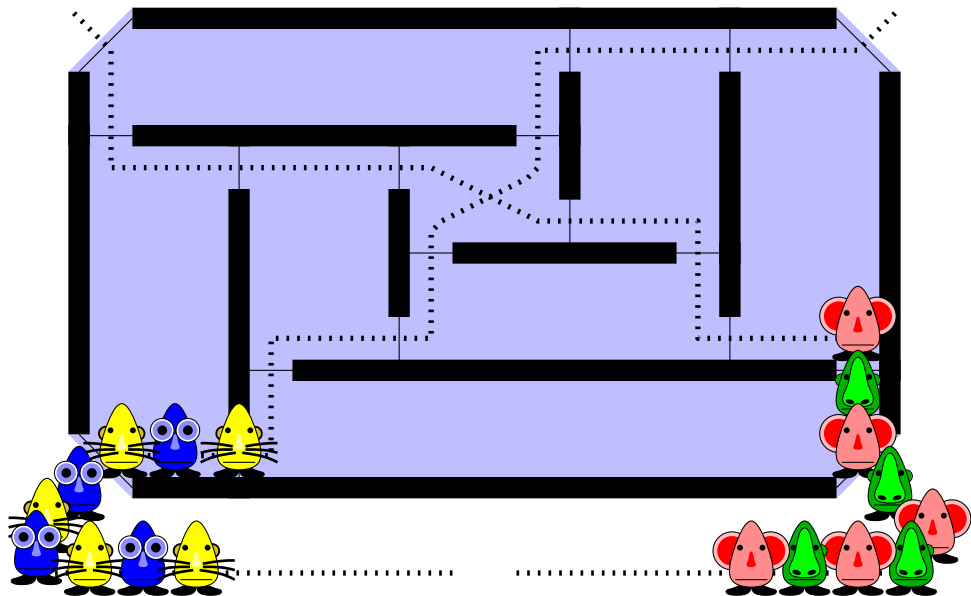








## guarding one-sided museums

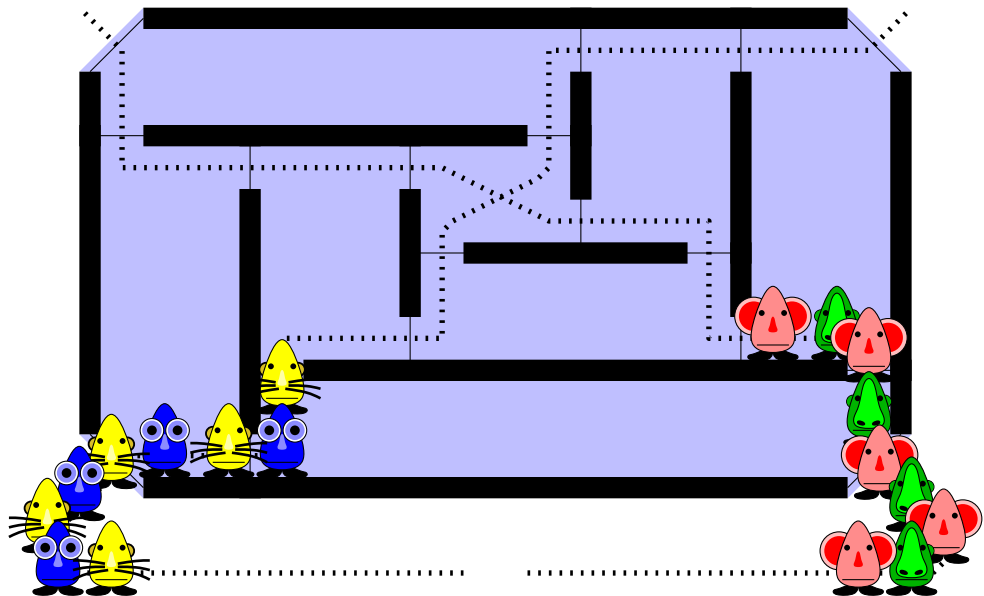






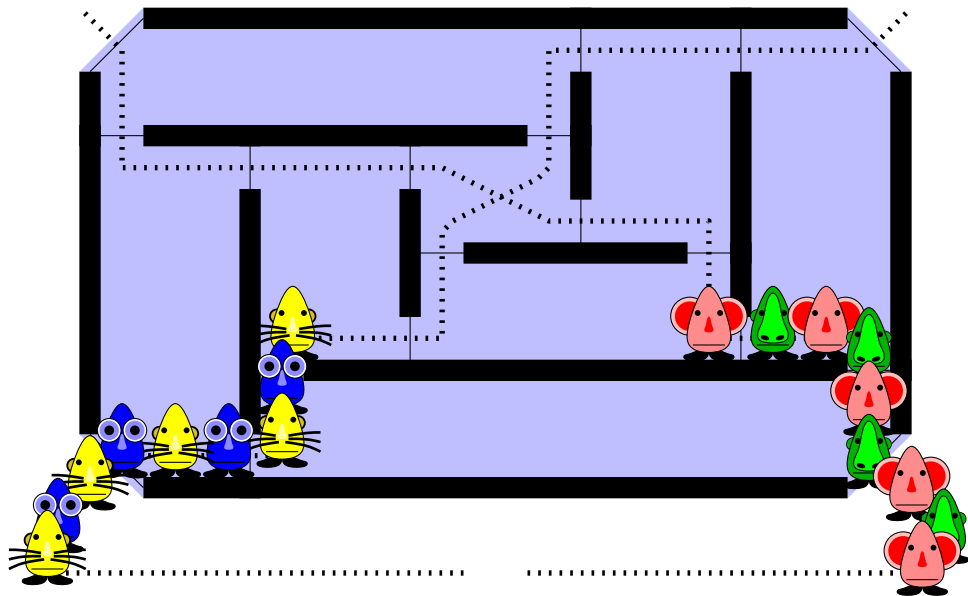


## guarding one-sided museums

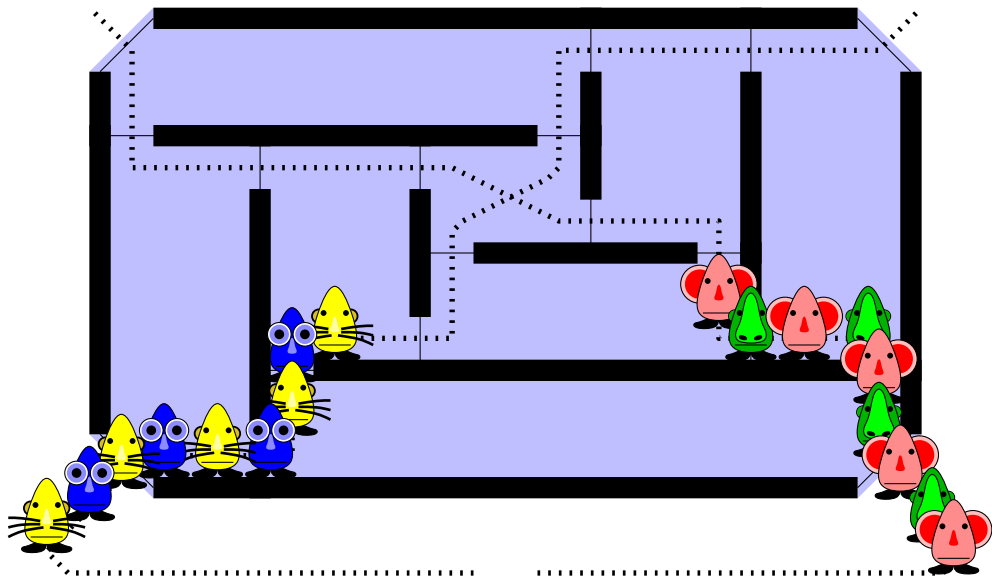




## guarding one-sided museums

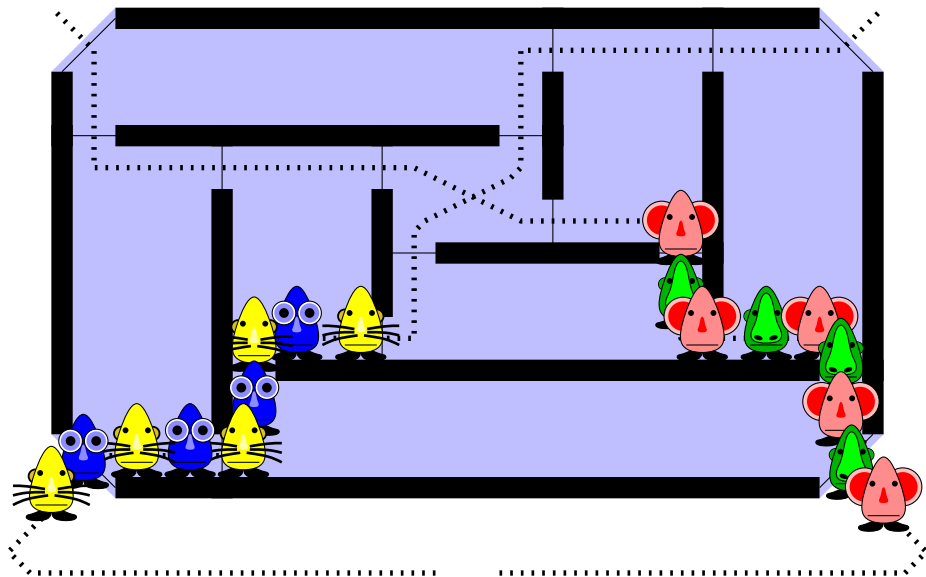


guarding one-sided museums

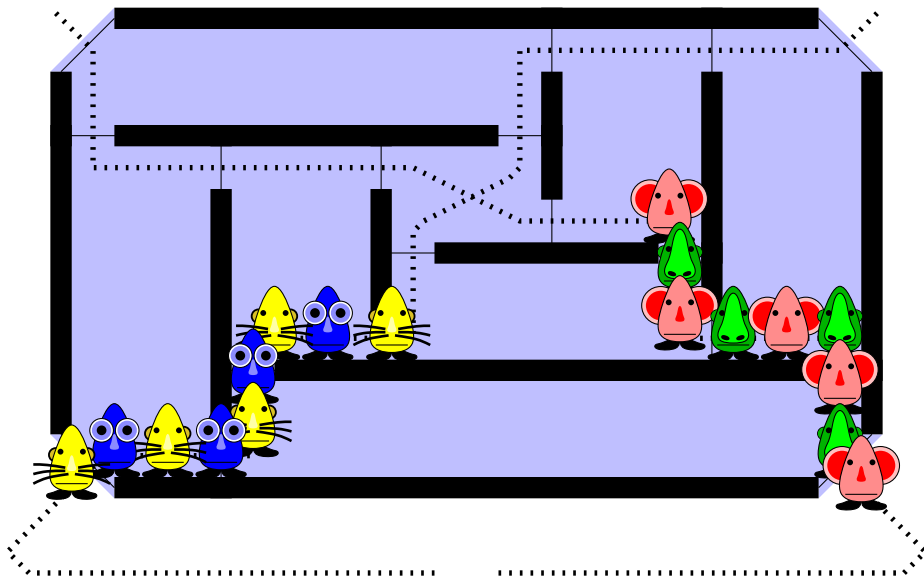




guarding one-sided museums



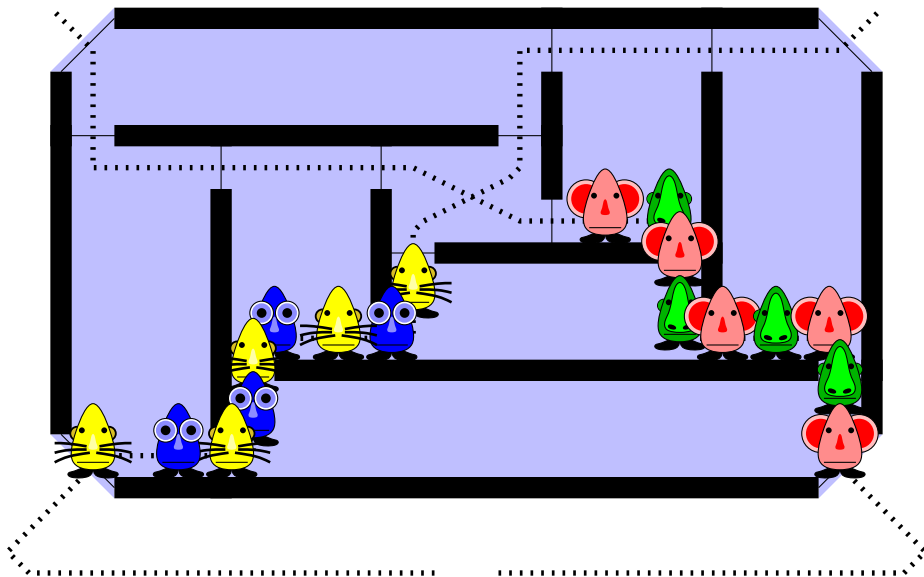
## guarding one-sided museums



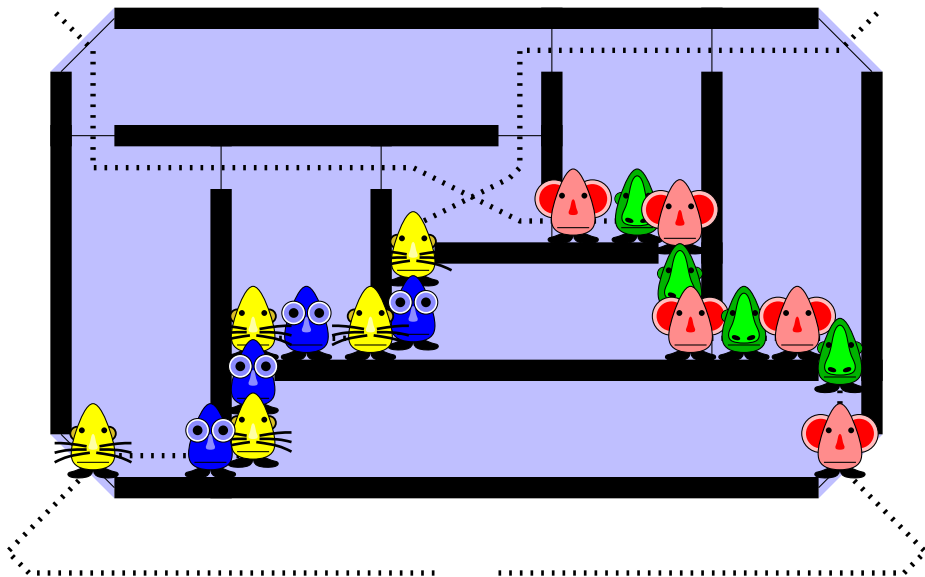




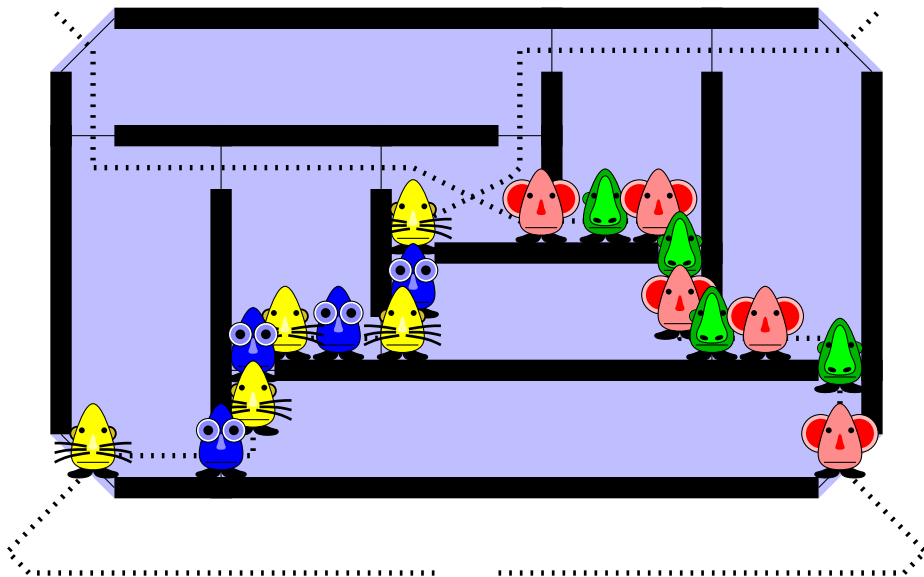
## guarding one-sided museums



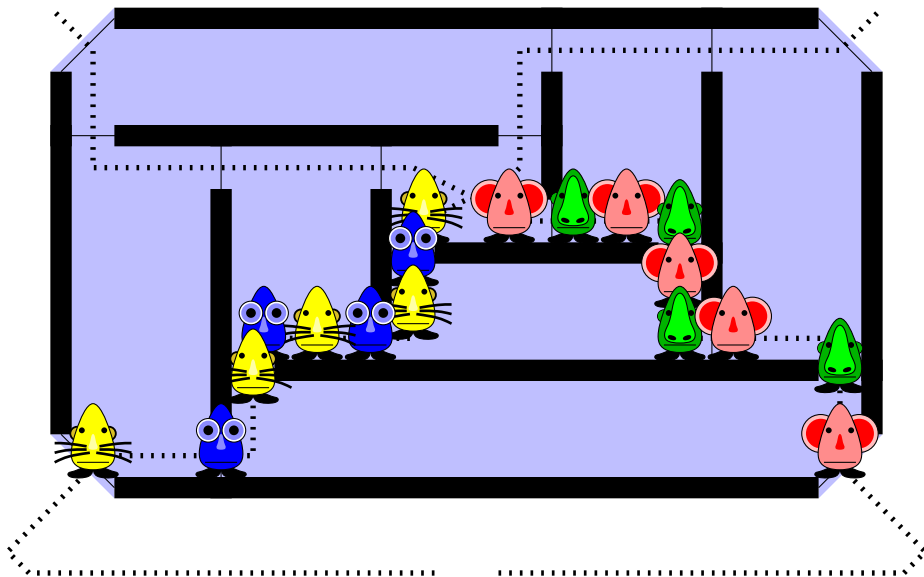
## guarding one-sided museums



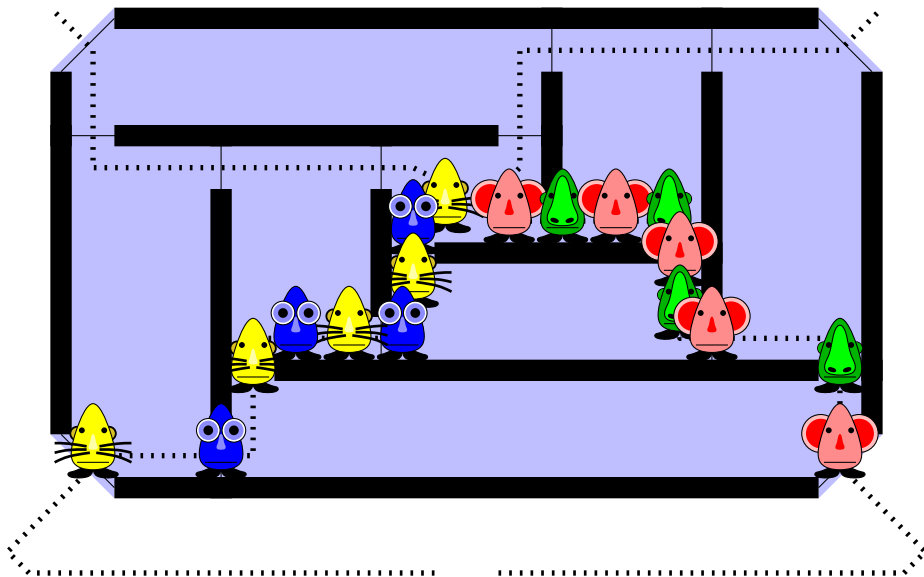
## guarding one-sided museums



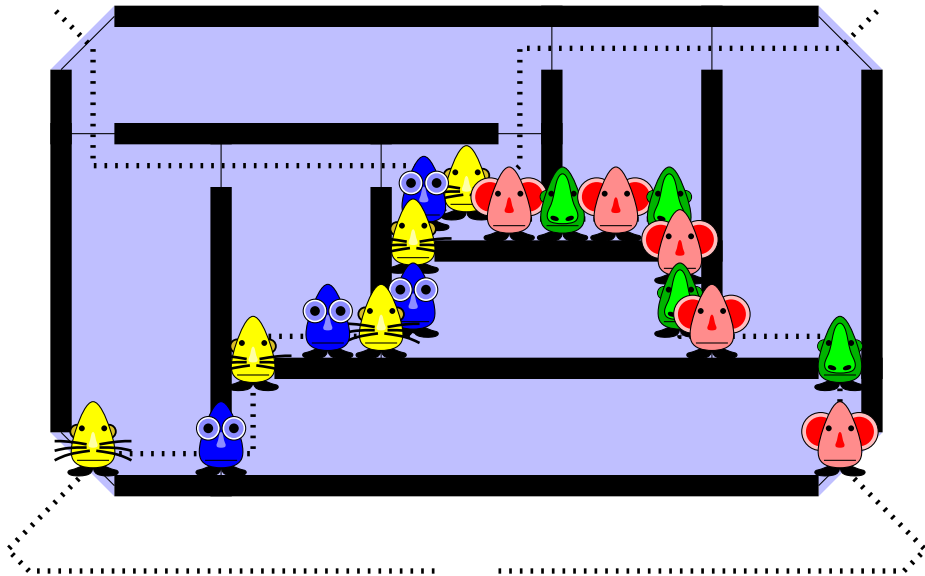
## guarding one-sided museums



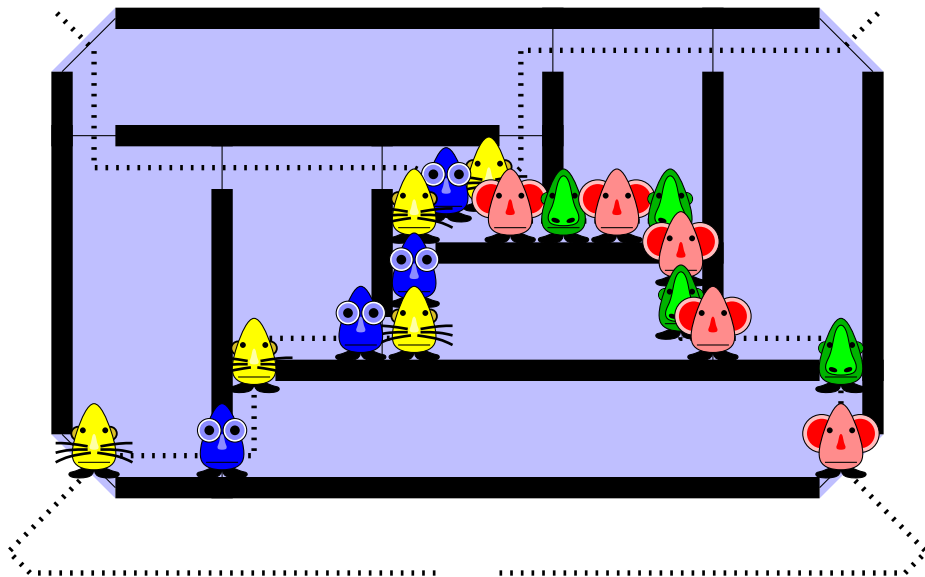
## guarding one-sided museums



## guarding one-sided museums

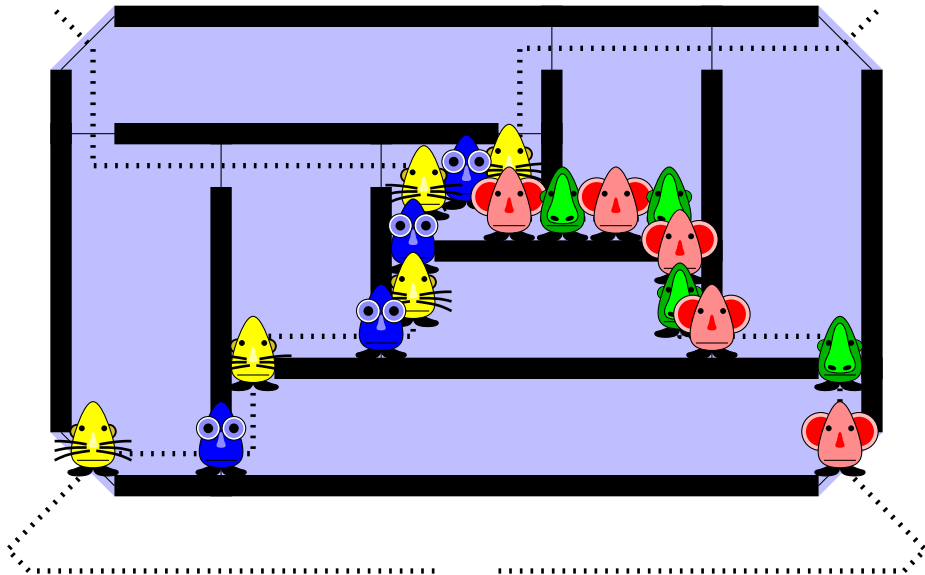


## guarding one-sided museums

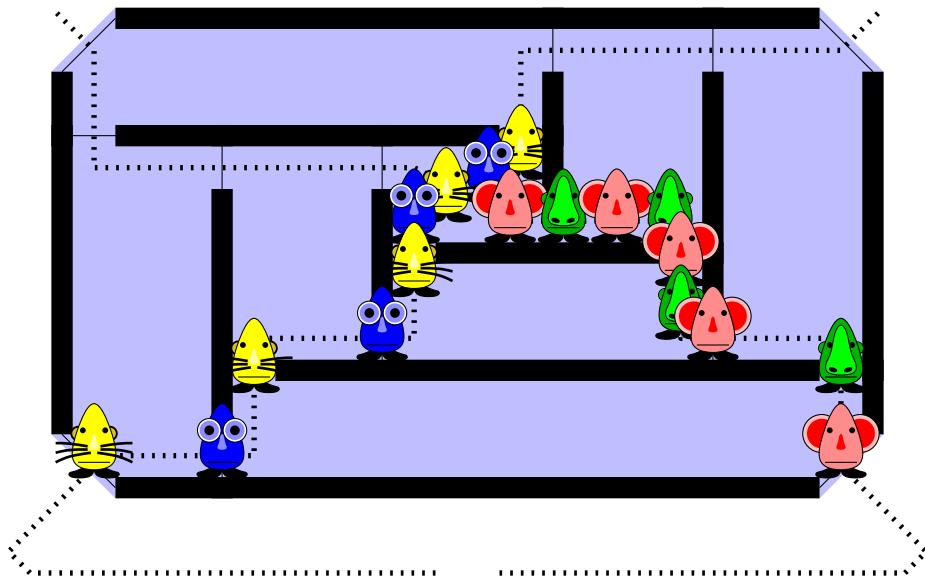




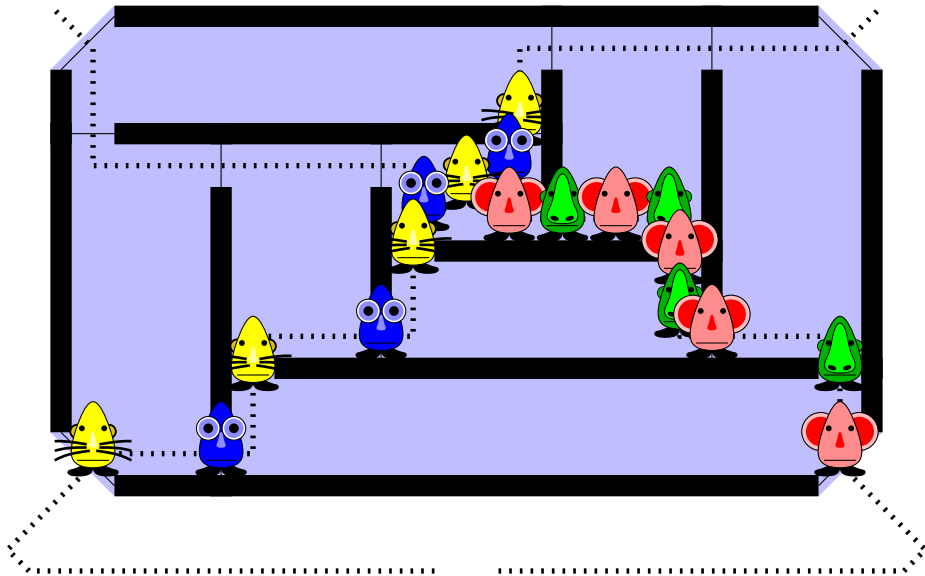
## guarding one-sided museums



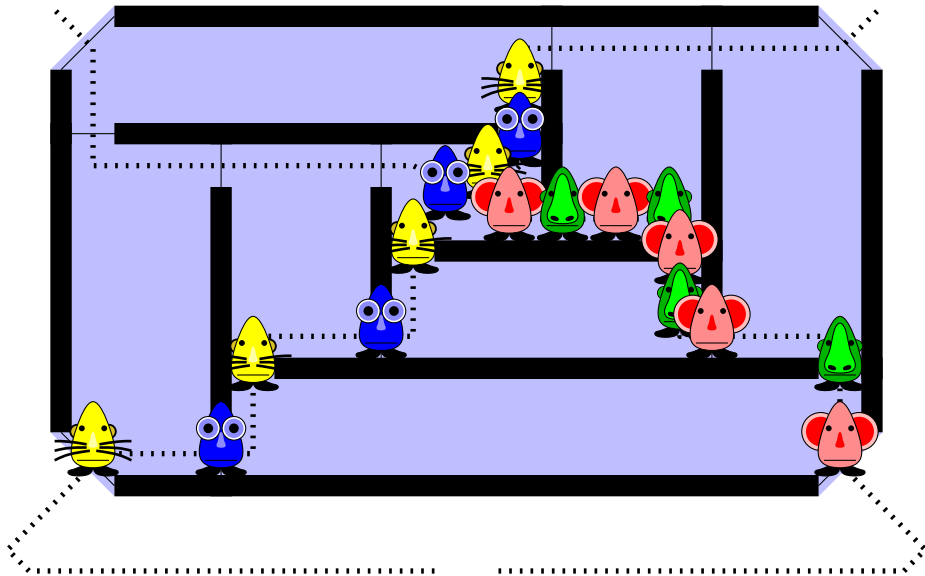
## guarding one-sided museums



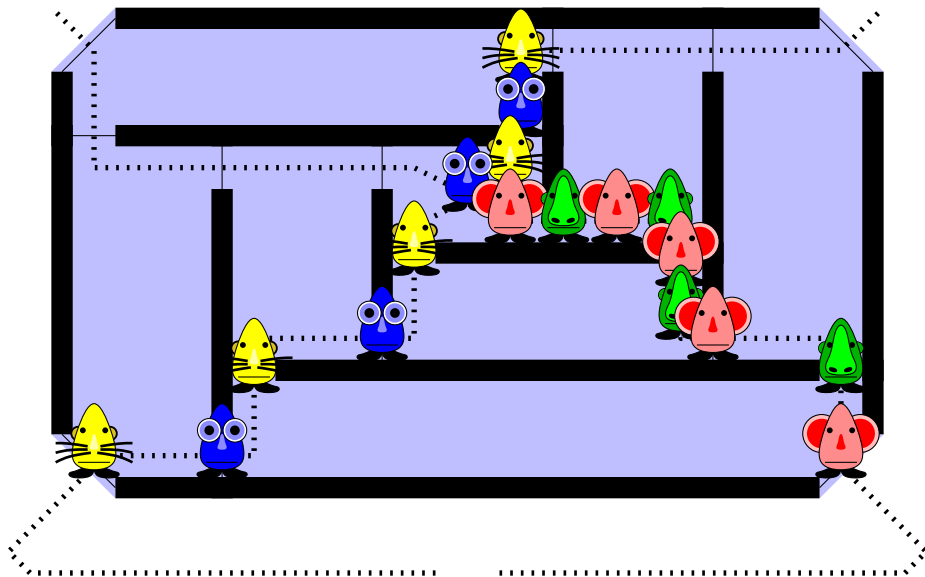
## guarding one-sided museums



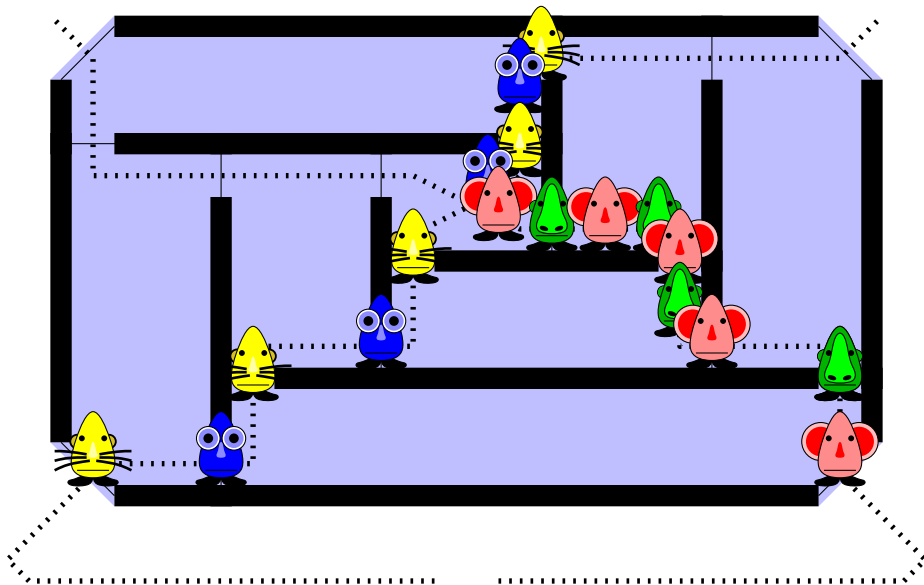
## guarding one-sided museums



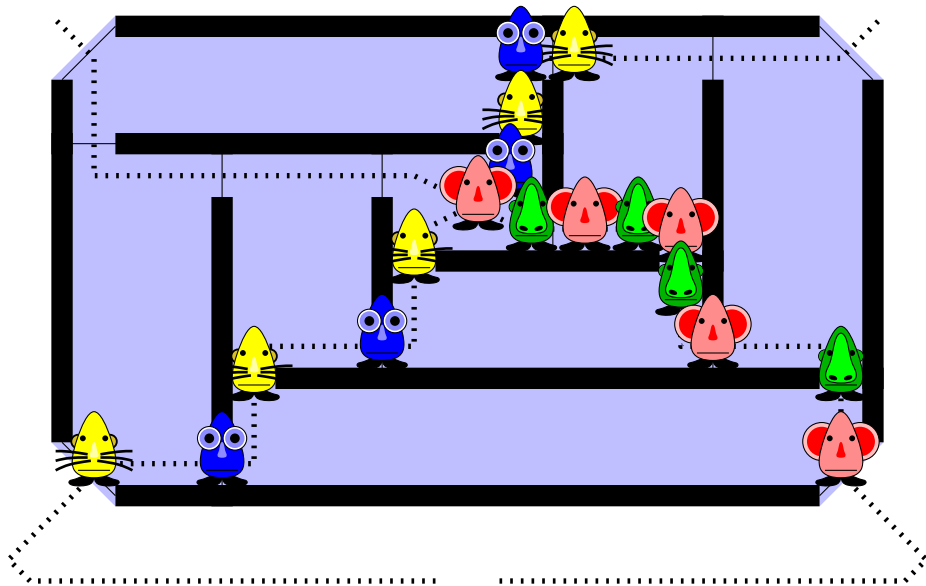
## guarding one-sided museums



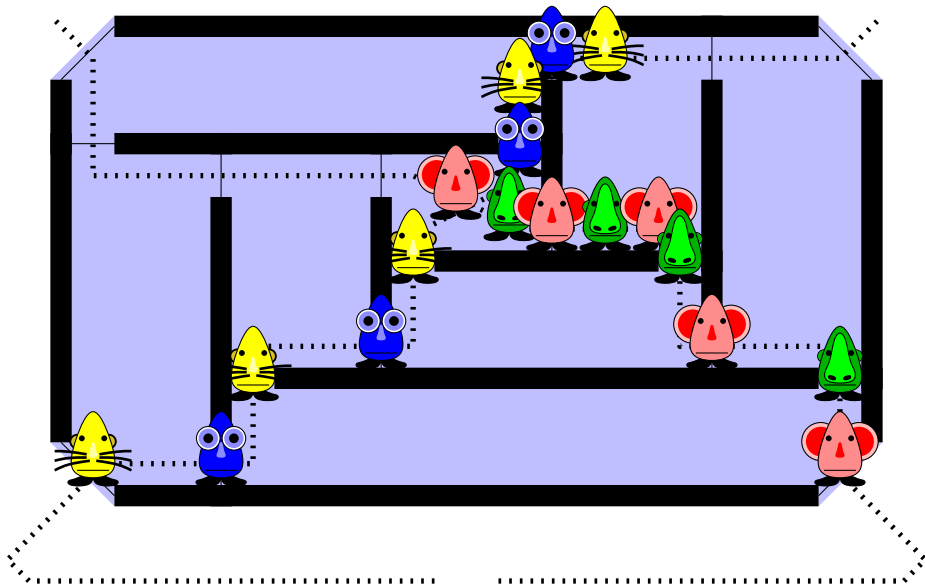
## guarding one-sided museums



## guarding one-sided museums

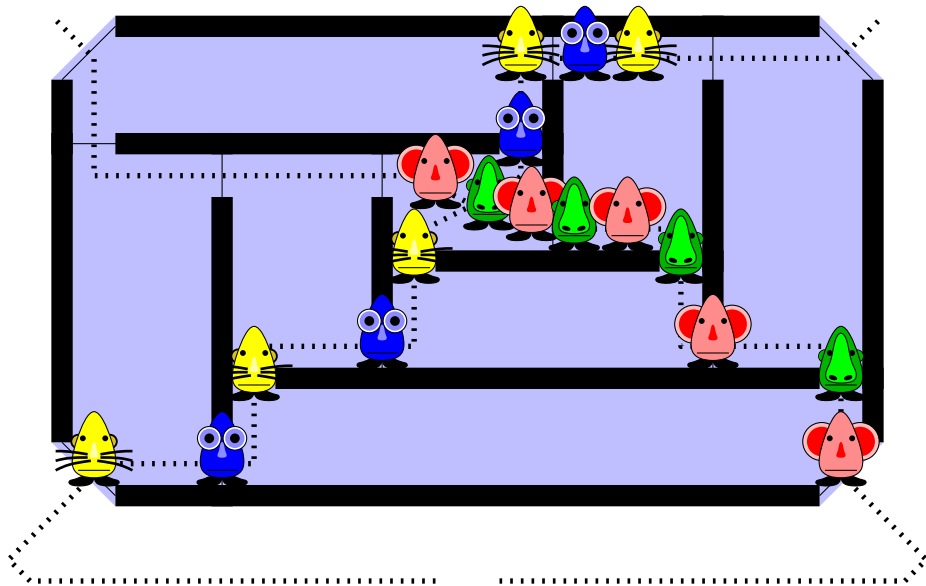


## guarding one-sided museums

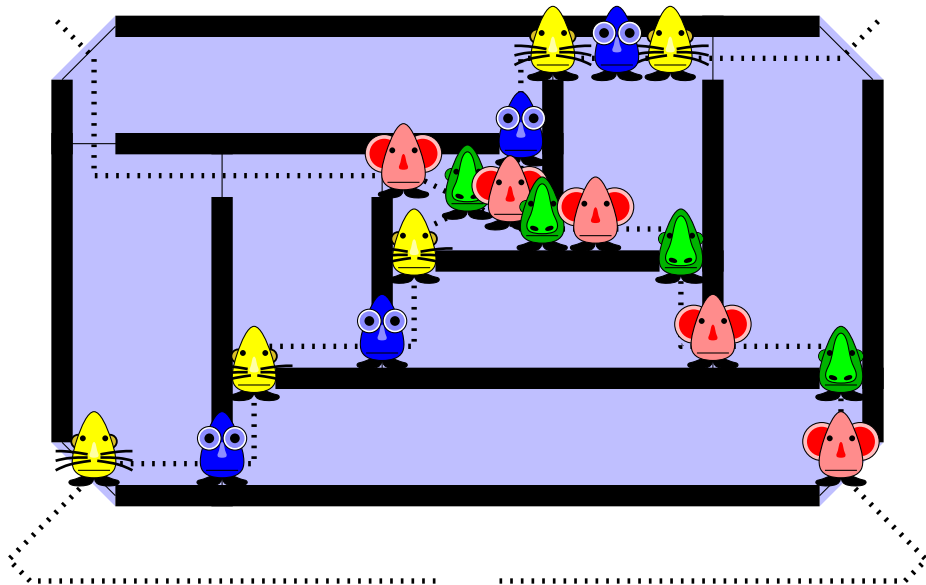




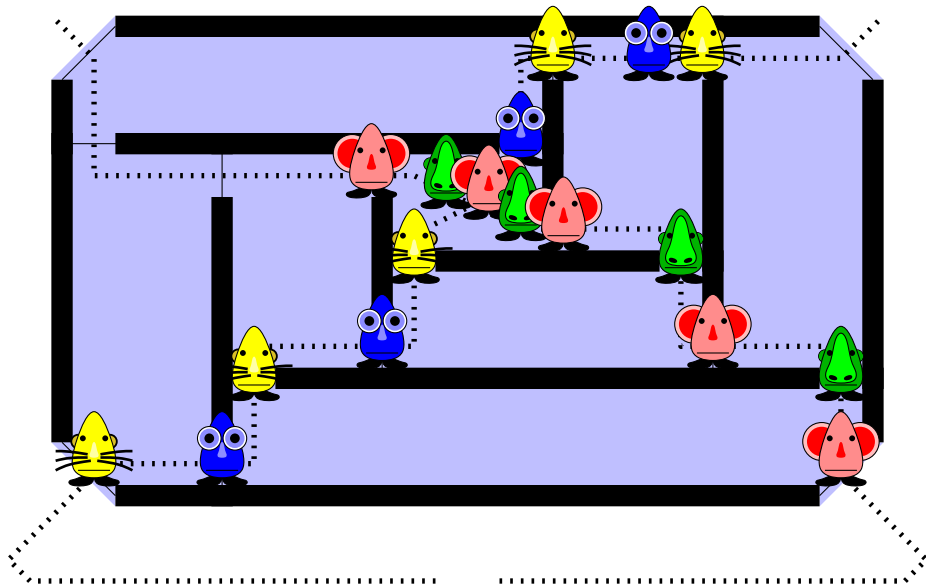
## guarding one-sided museums



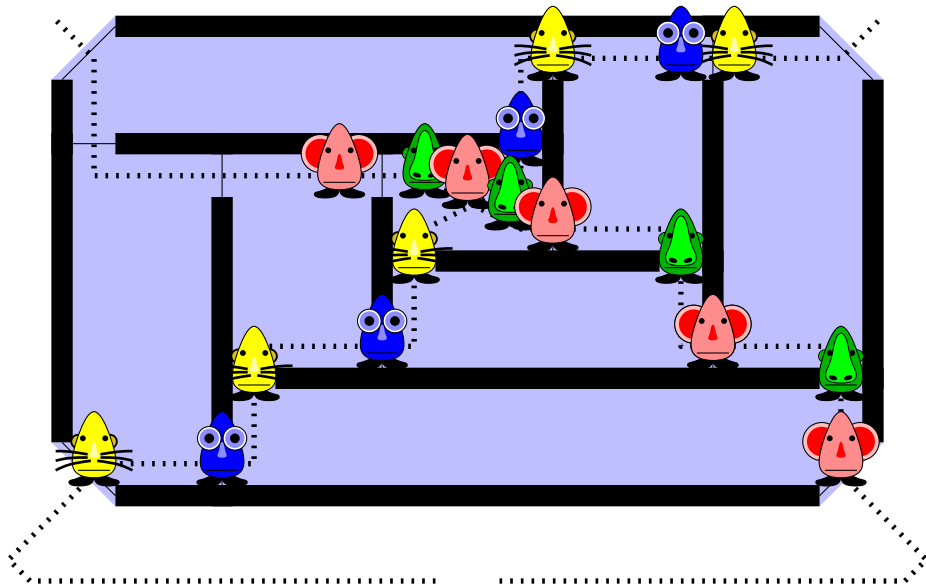
## guarding one-sided museums



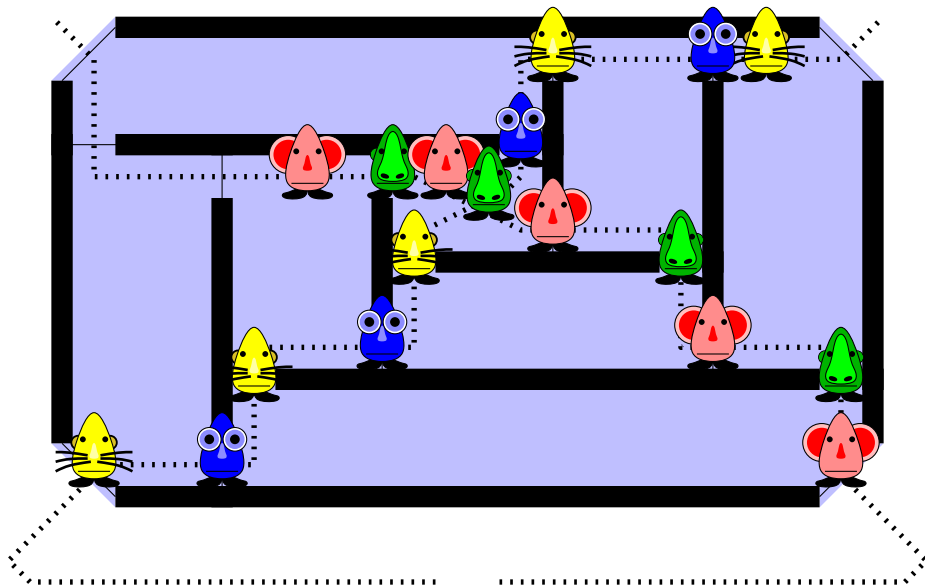
## guarding one-sided museums



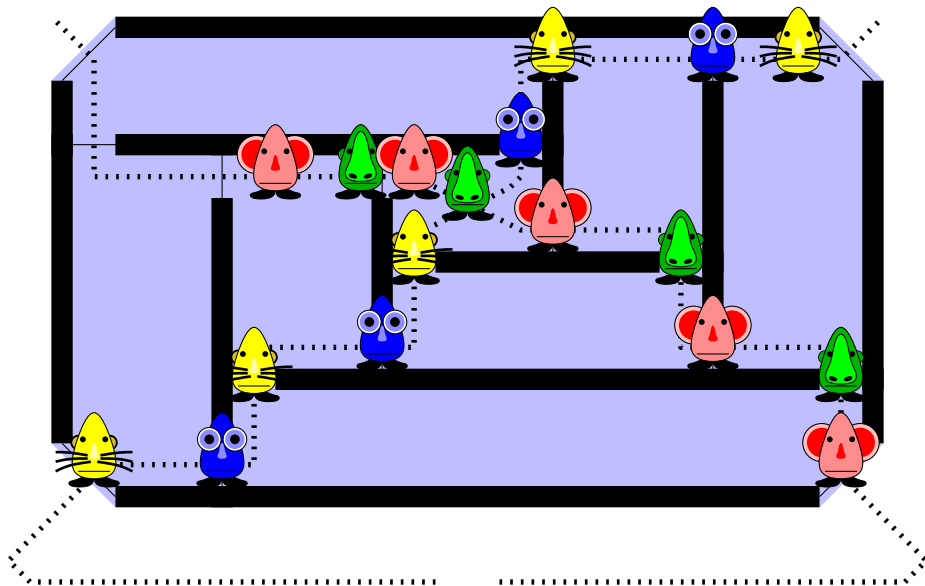
## guarding one-sided museums



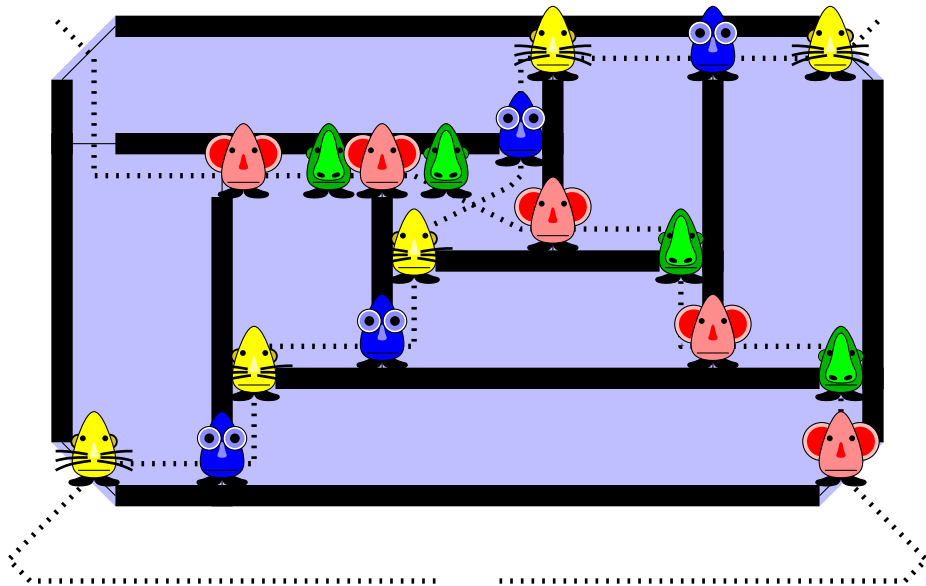
## guarding one-sided museums



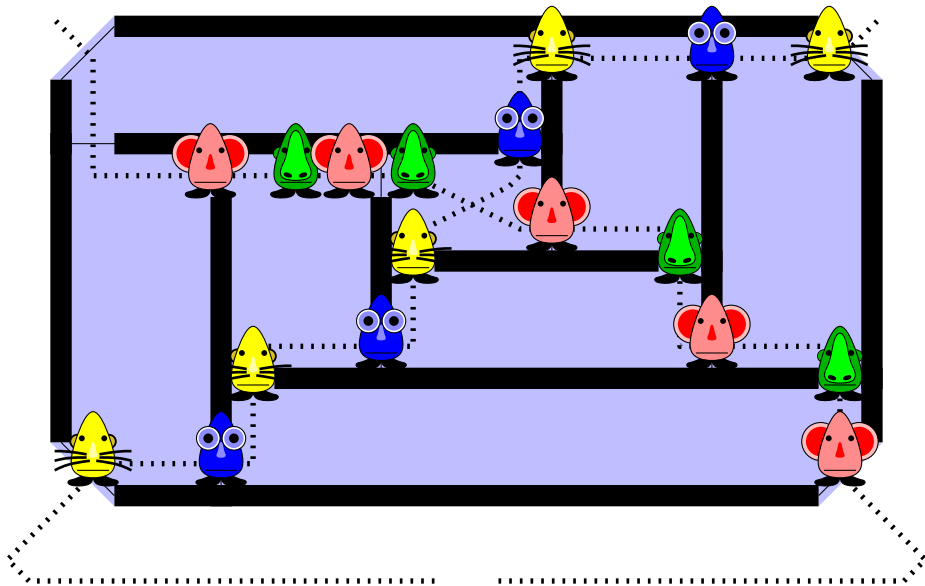
## guarding one-sided museums



## guarding one-sided museums

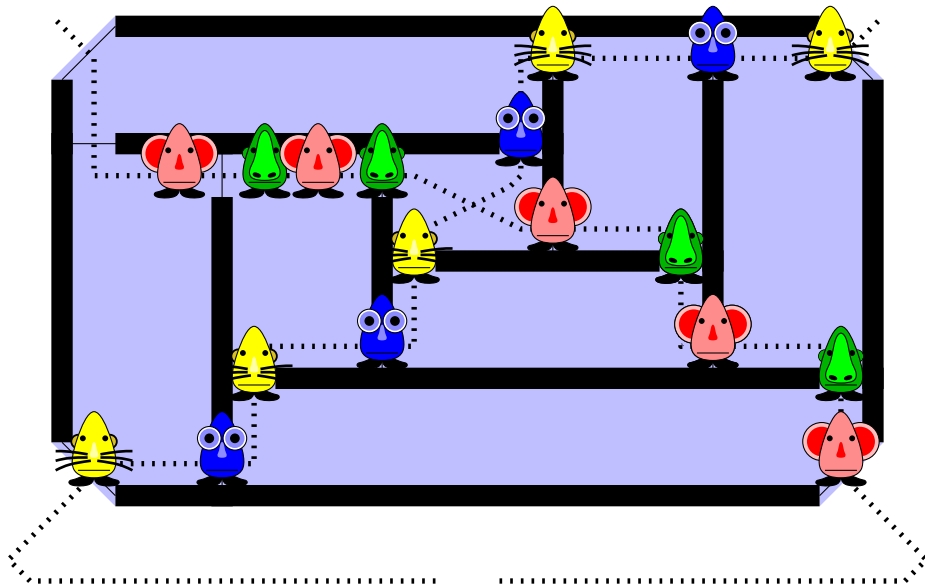


guarding one-sided museums

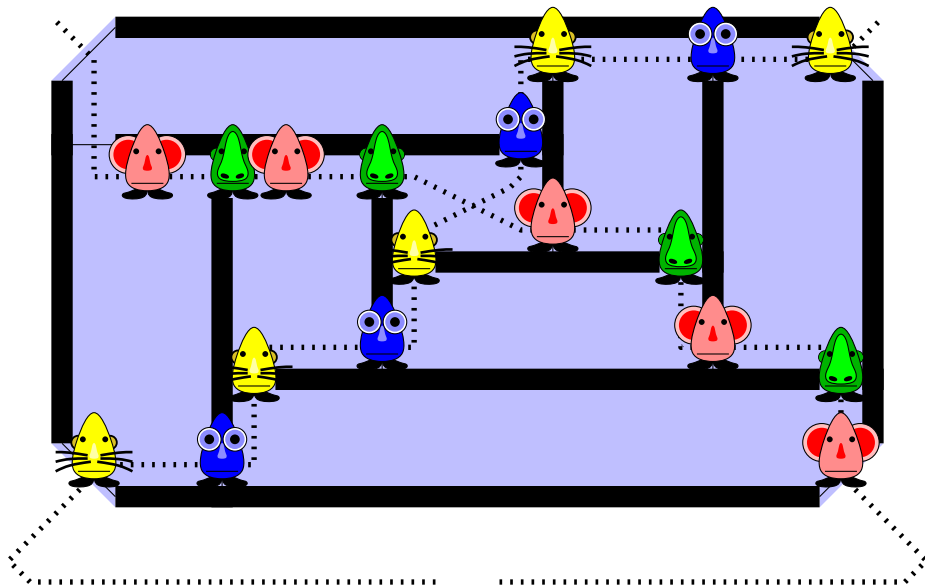




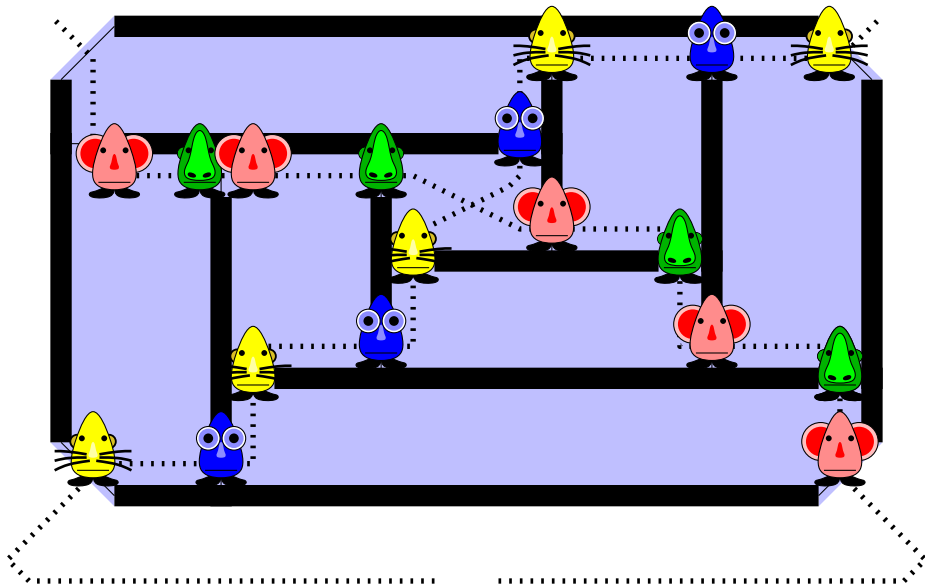
## guarding one-sided museums



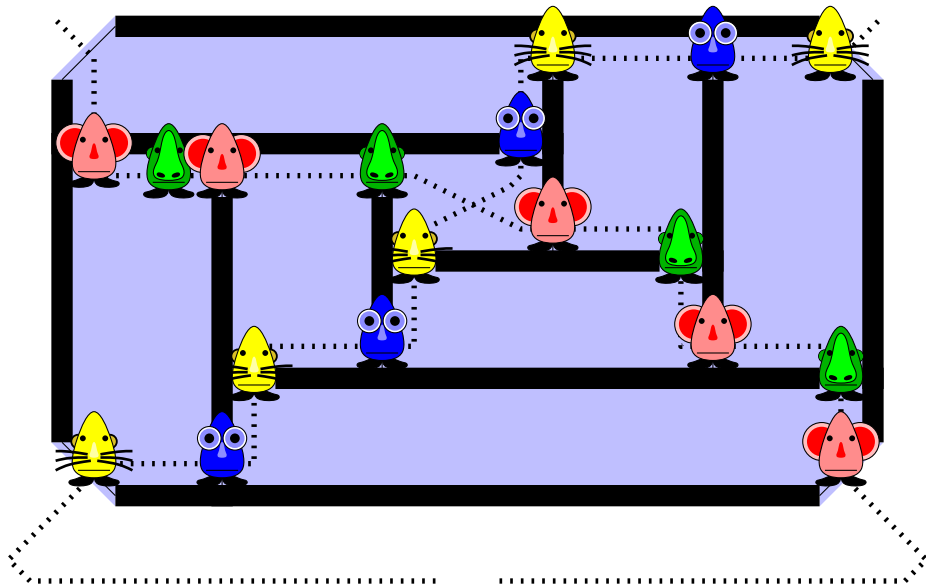
## guarding one-sided museums



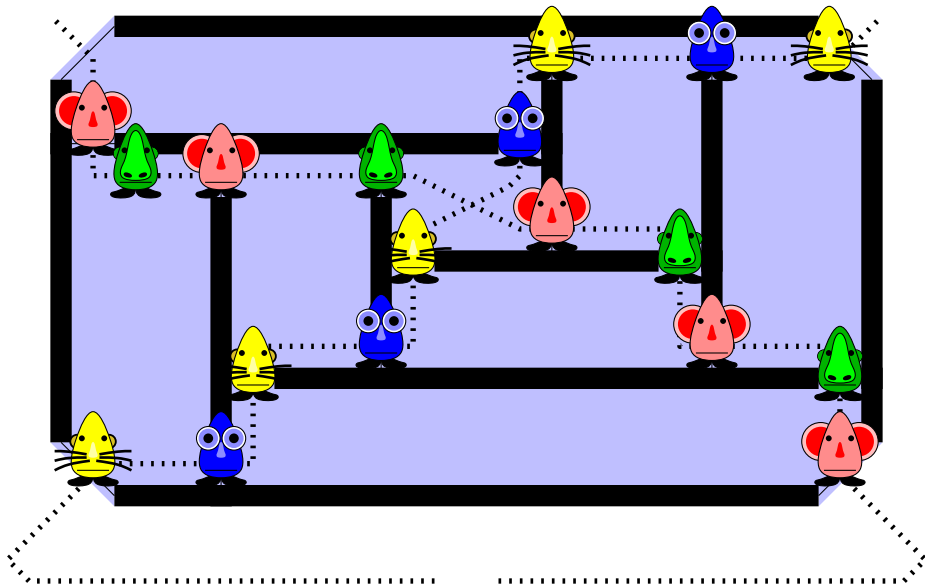
## guarding one-sided museums



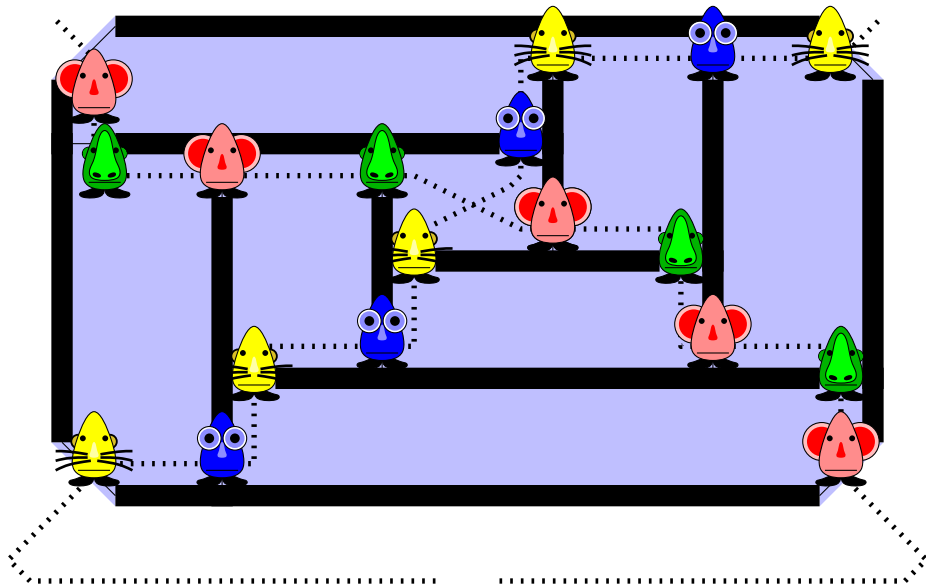
## guarding one-sided museums



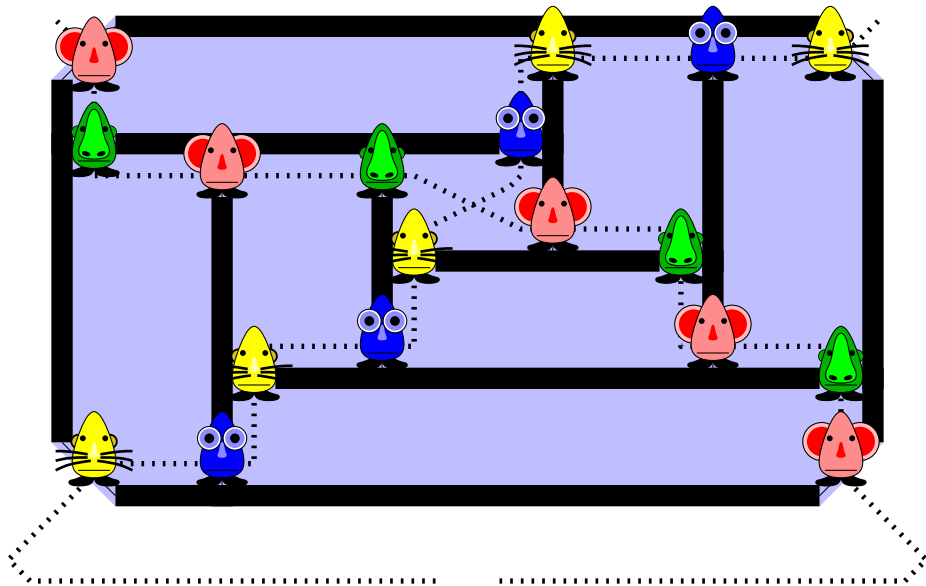
## guarding one-sided museums



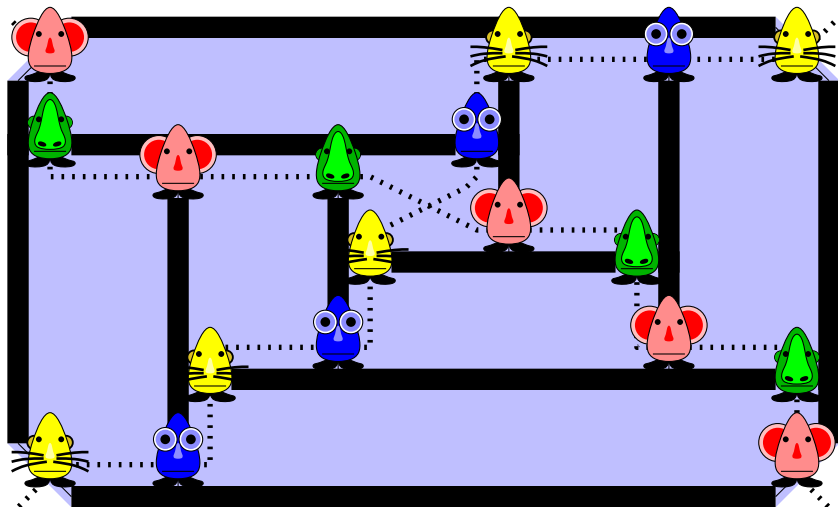
## guarding one-sided museums



## guarding one-sided museums



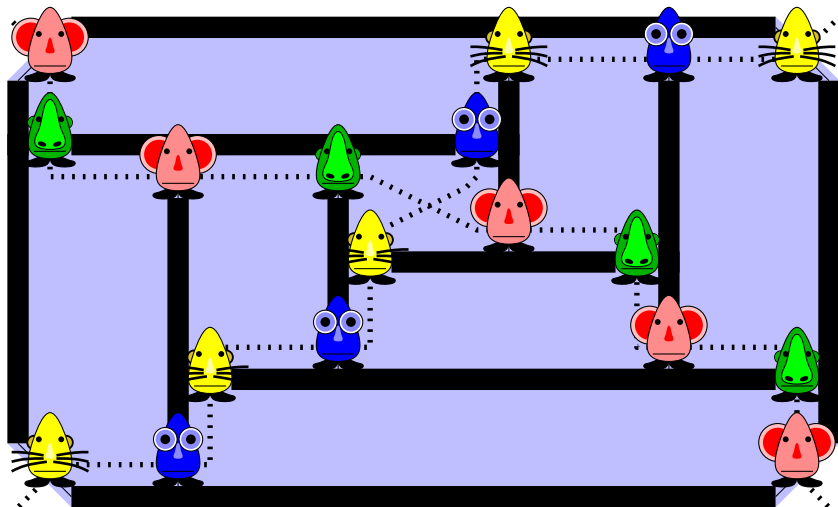
## guarding one-sided museums



each room has feel-all and see-all on one path,  
and hear-all and smell-all on the other path



## guarding one-sided museums



each room has feel-all and see-all on one path,  
and hear-all and smell-all on the other path

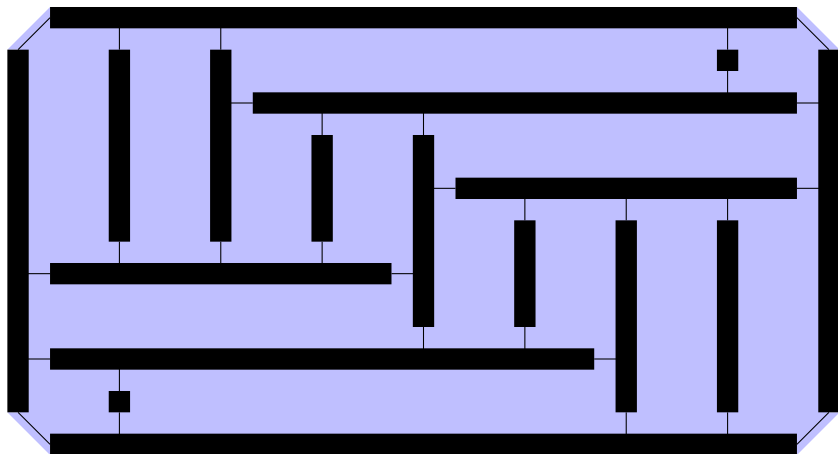
(building of  $n$  rectangles guarded in  $O(n)$  time)

## guarding two-sided museums?

---

Is there a path through all rectangles, such that:

- we can alternate two colours along that path, *and*
- guard the rooms with the remaining two colours at the remaining doors?

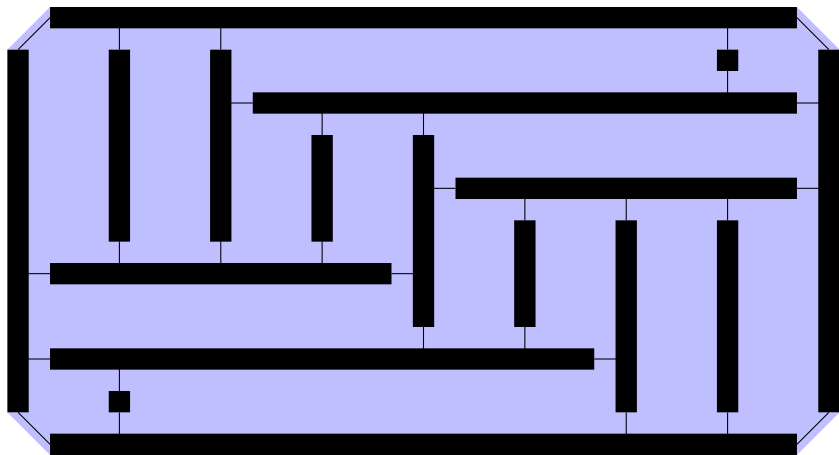


## guarding two-sided museums?

---

Is there a path through all rectangles, such that:

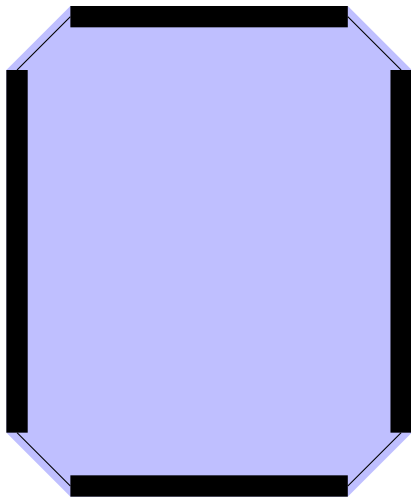
- we can alternate two colours along that path, *and*
- guard the rooms with the remaining two colours at the remaining doors?



No, such a path does not always exist

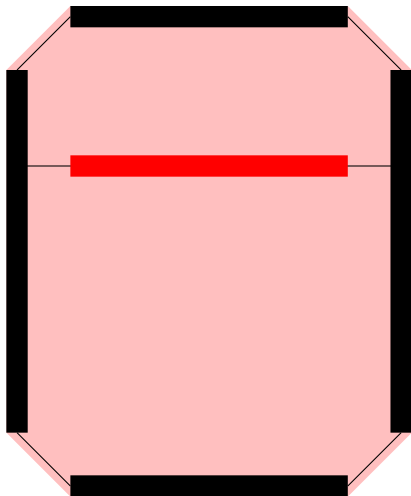


sliceable museum



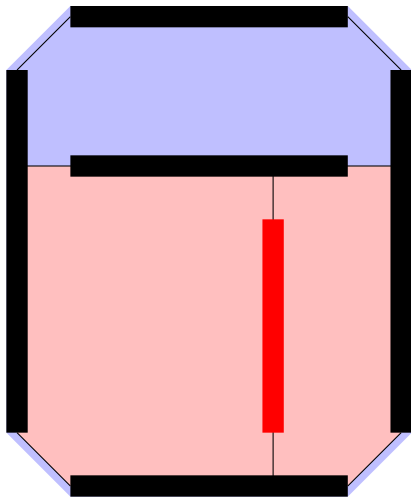
floor plan obtained by recursively  
subdividing rooms by parallel wall

sliceable museum



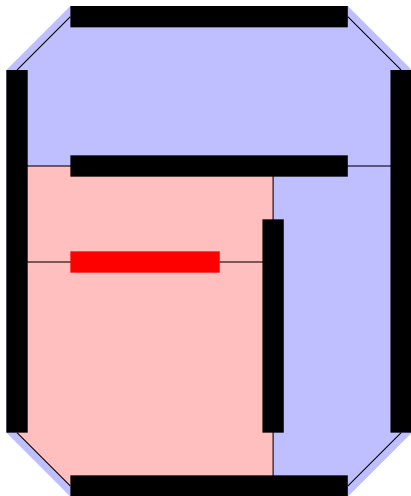
floor plan obtained by recursively  
subdividing rooms by parallel wall

sliceable museum



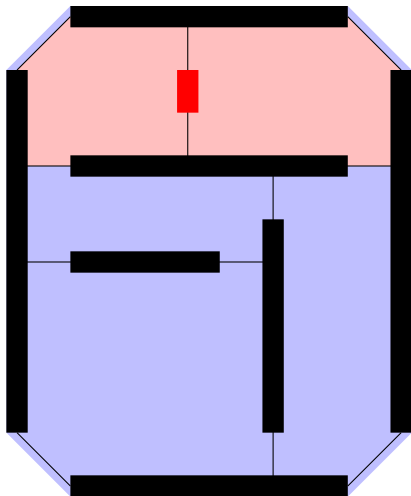
floor plan obtained by recursively  
subdividing rooms by parallel wall

sliceable museum



floor plan obtained by recursively  
subdividing rooms by parallel wall

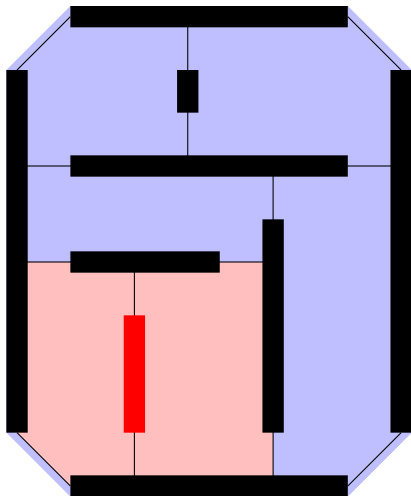
## sliceable museum



floor plan obtained by recursively  
subdividing rooms by parallel wall

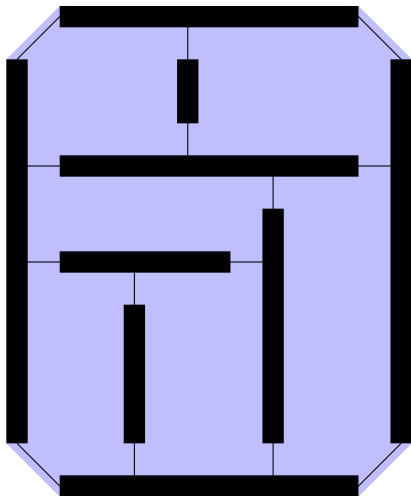


sliceable museum



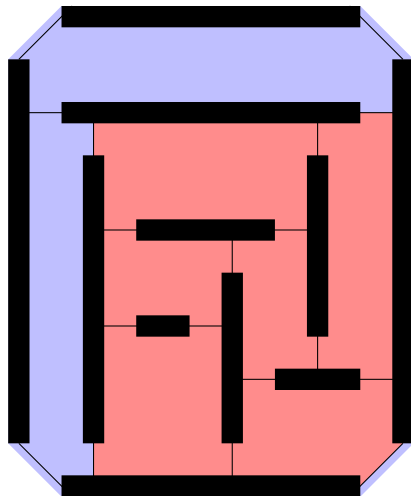
floor plan obtained by recursively  
subdividing rooms by parallel wall

sliceable museum



floor plan obtained by recursively subdividing rooms by parallel wall

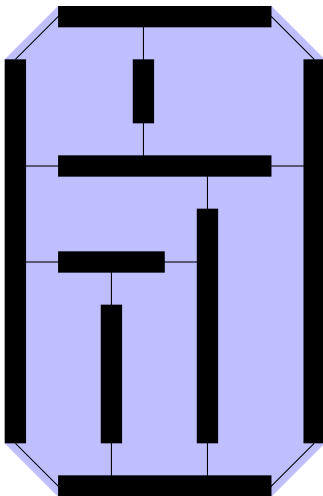
non-sliceable museum



subdivision of red area cannot be obtained by slicing

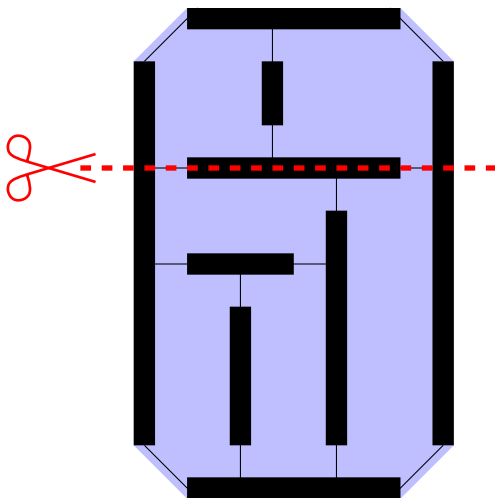
## sliceable museums

---



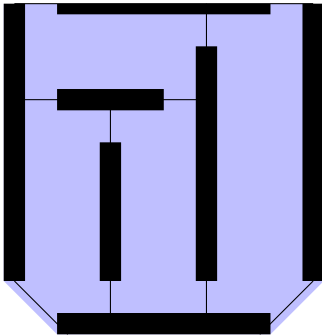
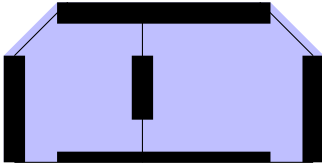
## sliceable museums

---



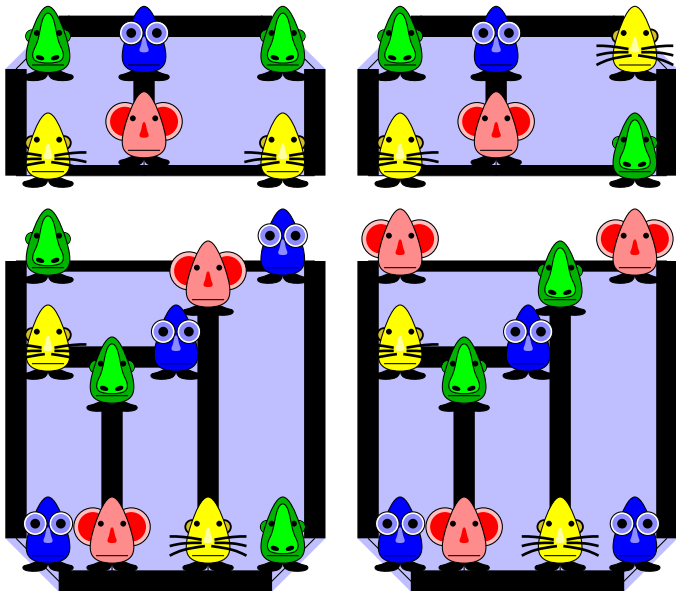
## sliceable museums

---



## sliceable museums

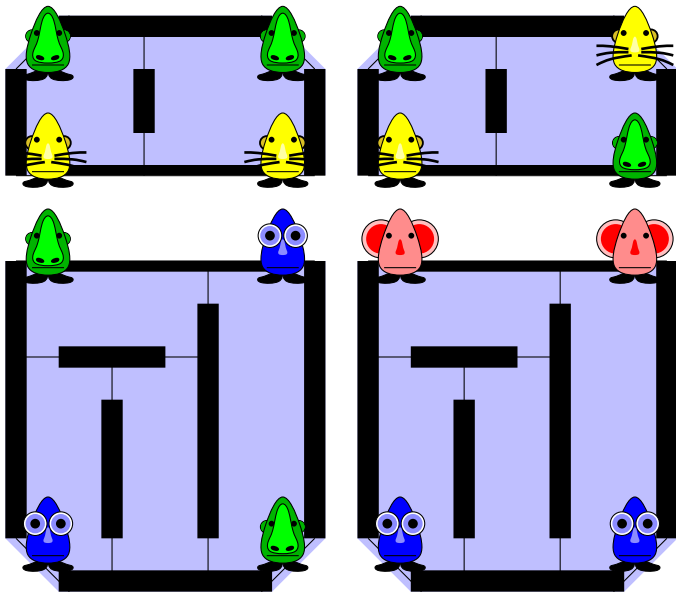
---



Two different solutions for each part

## sliceable museums

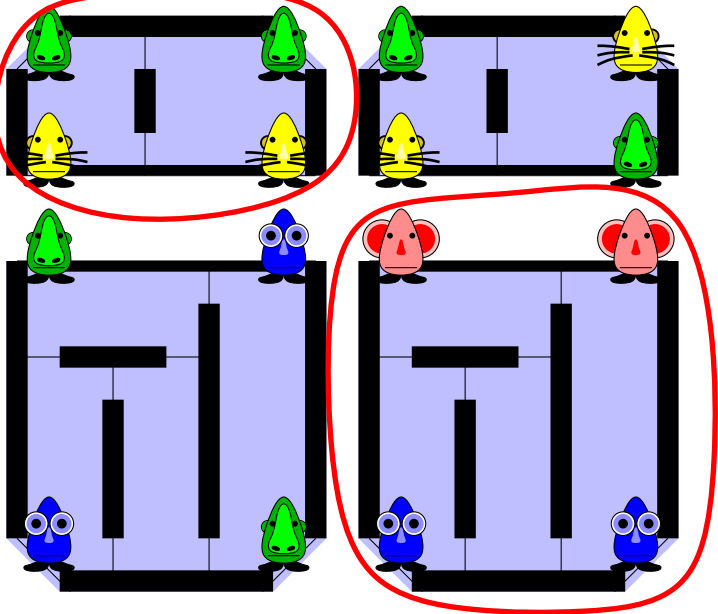
---



Two different solutions for each part

## sliceable museums

---

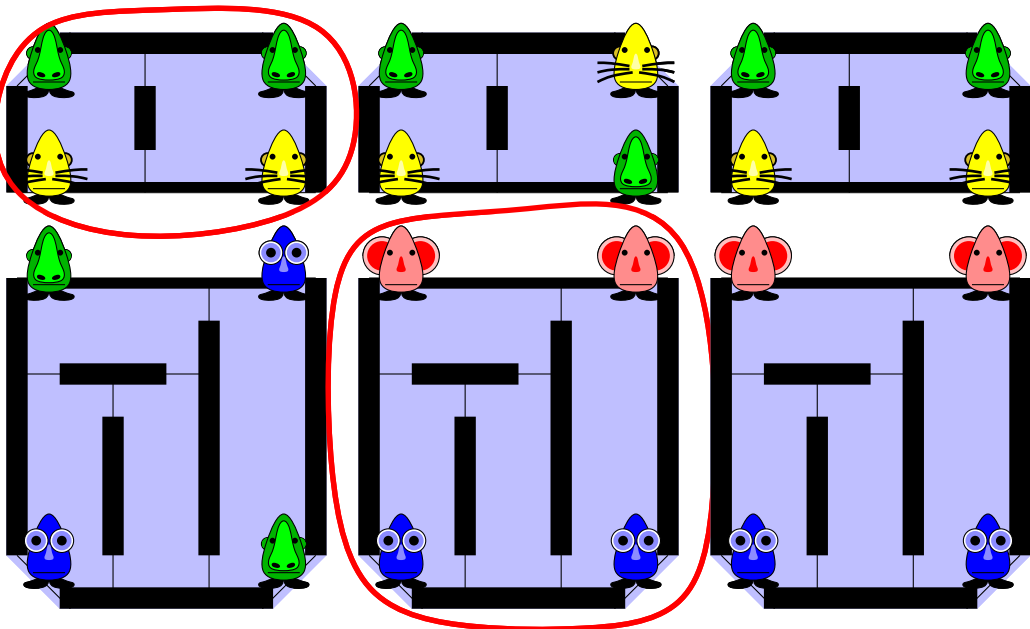


Two different solutions for each part

Some combinations can  
be matched



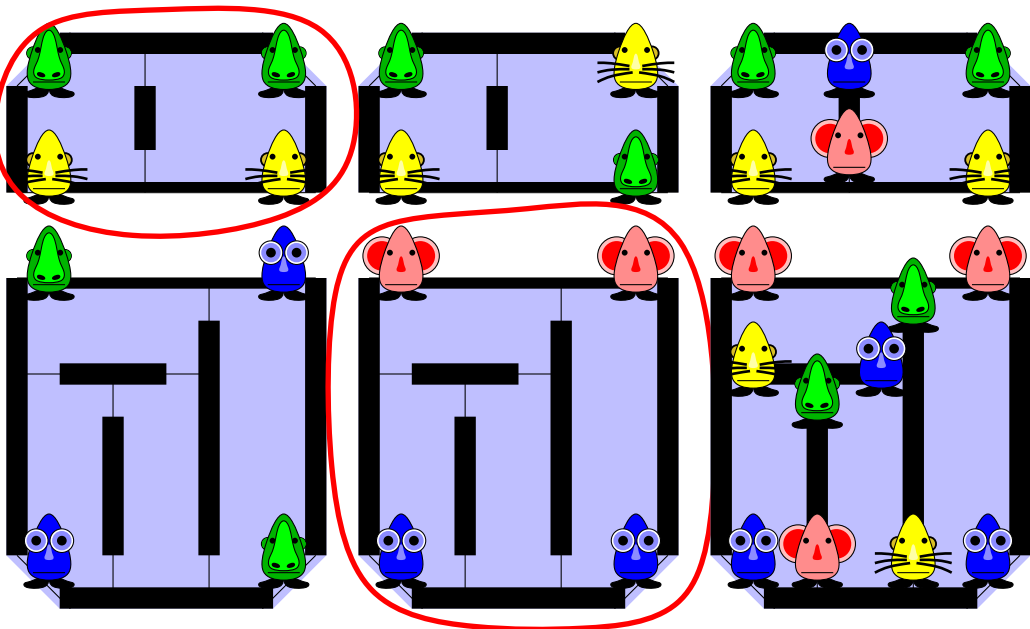
## sliceable museums



Two different solutions for each part

Some combinations can  
be matched

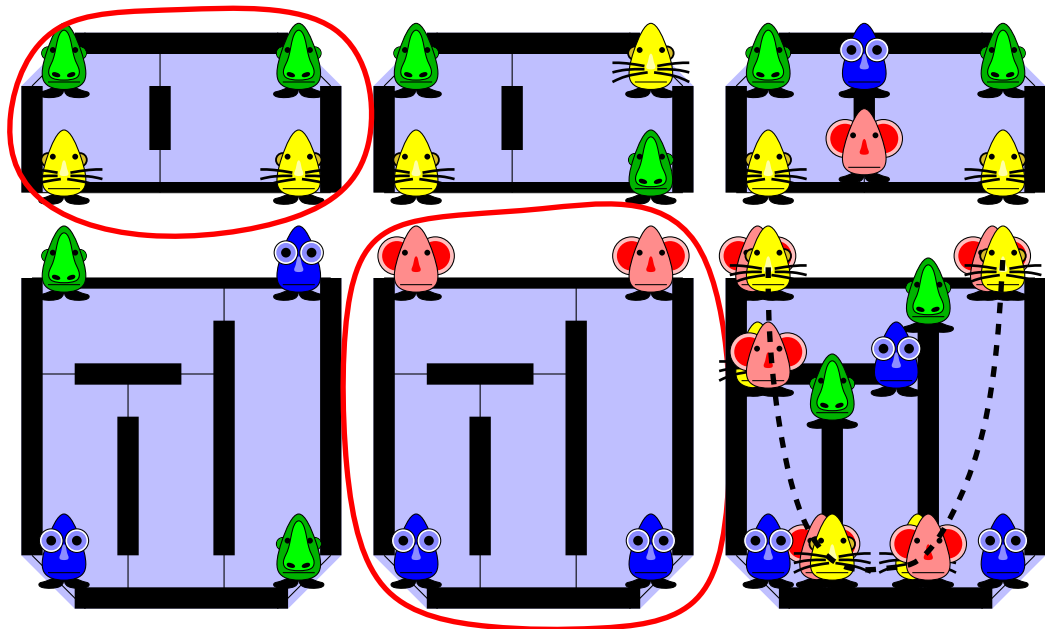
## sliceable museums



Two different solutions for each part

Some combinations can  
be matched

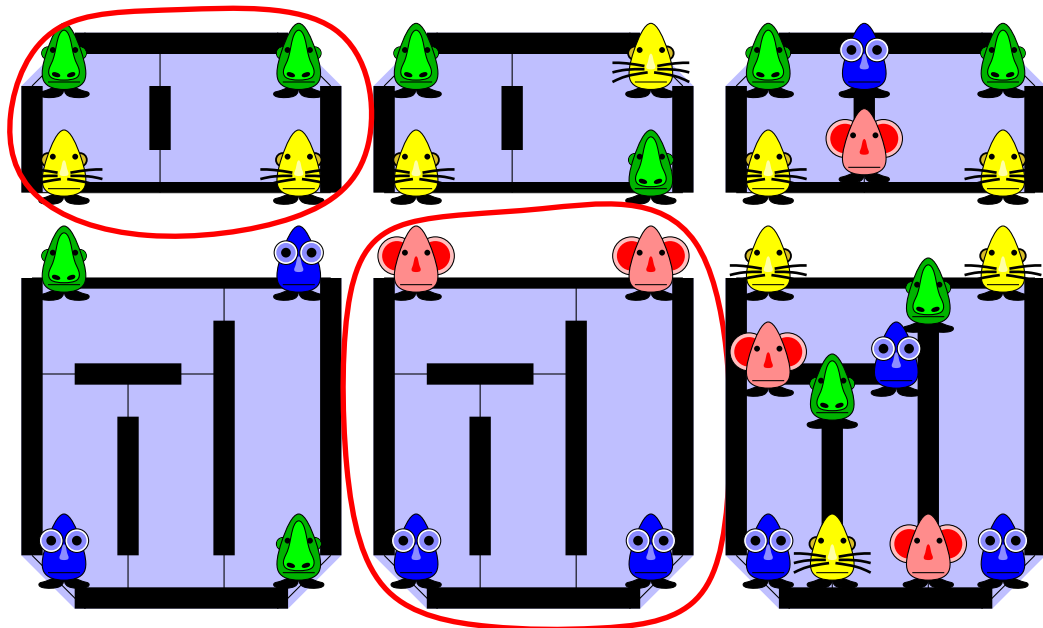
## sliceable museums



Two different solutions for each part

Some combinations can  
be matched

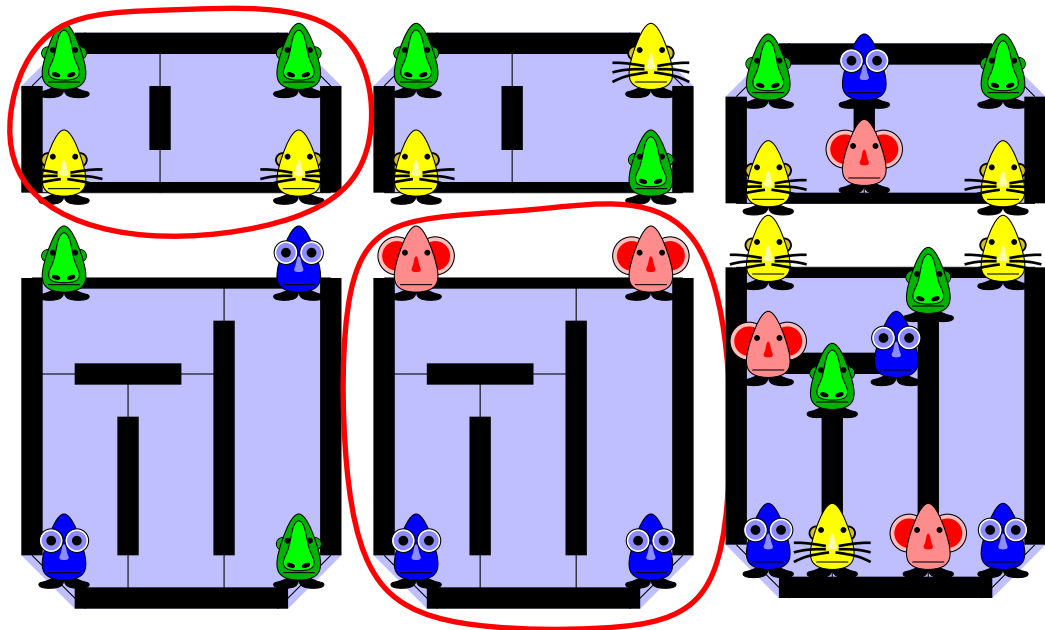
## sliceable museums



Two different solutions for each part

Some combinations can  
be matched

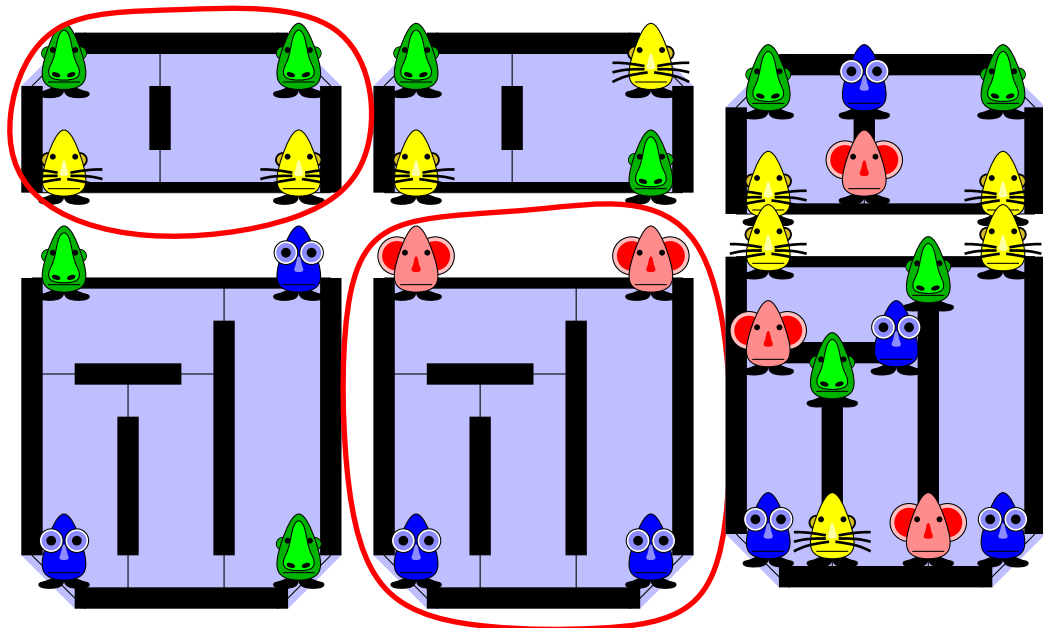
## sliceable museums



Two different solutions for each part

Some combinations can  
be matched

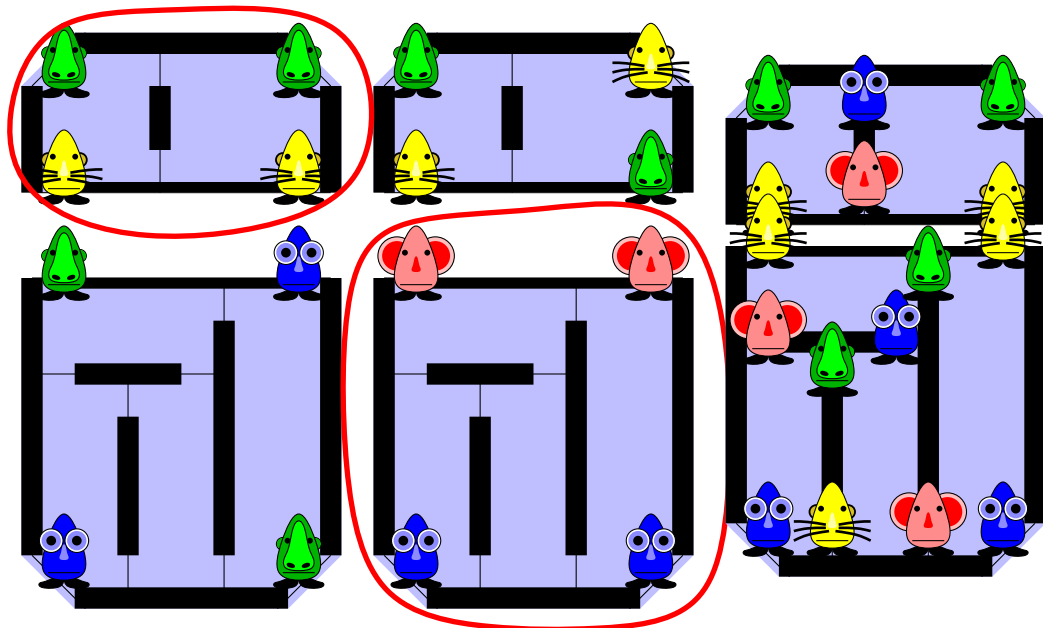
## sliceable museums



Two different solutions for each part

Some combinations can  
be matched

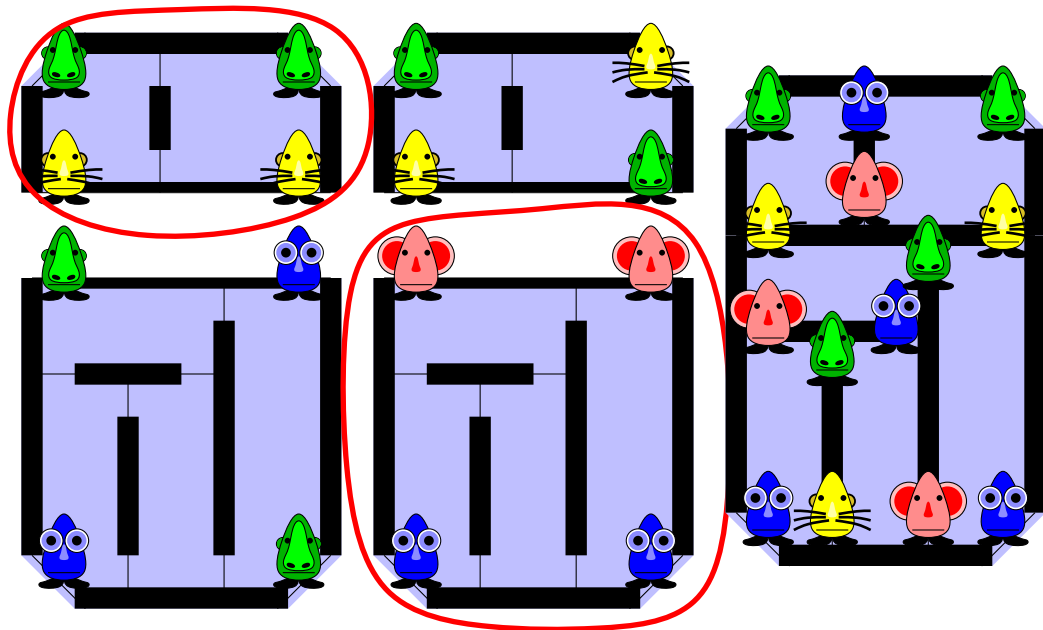
## sliceable museums



Two different solutions for each part

Some combinations can be matched

## sliceable museums

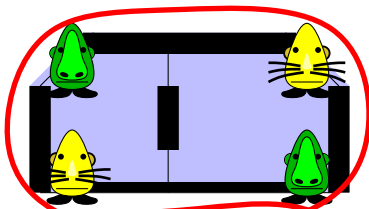
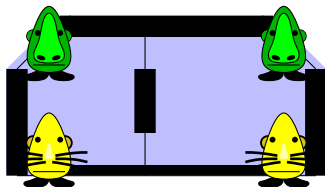


Two different solutions for each part

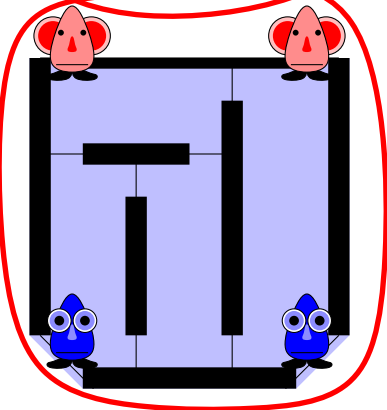
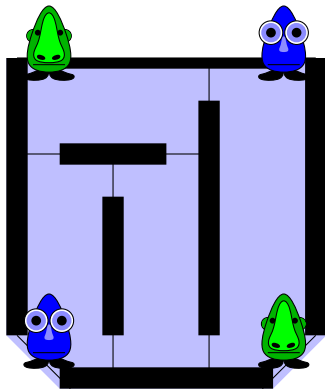
Some combinations can be matched



## sliceable museums

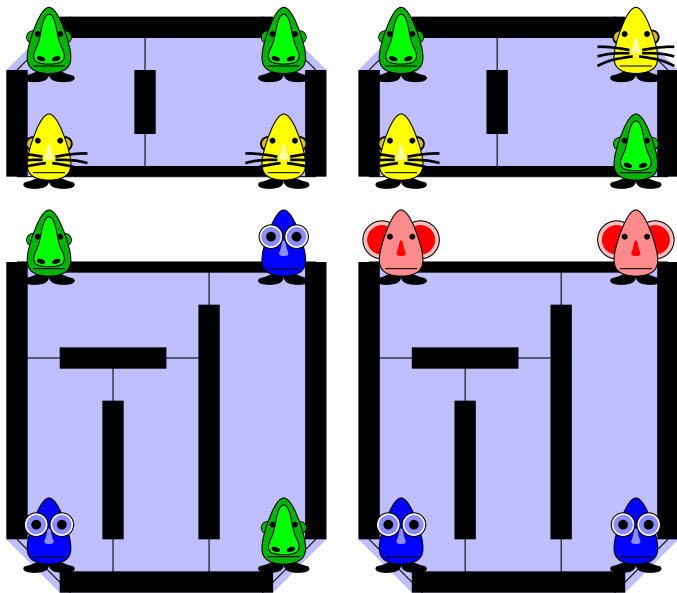


If matching is possible depends on pattern on corners.



Two different solutions for each part

## sliceable museums



Two different solutions for each part

If matching is possible depends on pattern on corners.

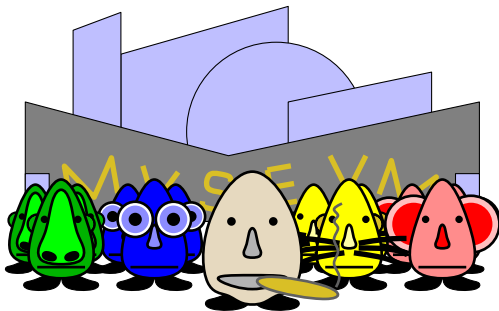
Algorithm:

1. compute bottom-up which partial solutions can be matched;
2. choose the solution for every subproblem top-down.

$O(n)$  time, given the slicing decomposition

Got linear-time algorithms for guarding sliceable or one-sided museums

Guarding non-sliceable two-sided museums seems difficult  
(but possible: stay for the next talk!)



Herman Haverkort ◦ Maarten Löffler ◦ Elena Mumford  
Matthew O'Meara ◦ Jack Snoeyink ◦ Bettina Speckmann  
SMELL-ALL-SEE-ALL-FEEL-ALL-HEAR-ALL SECURITY LTD