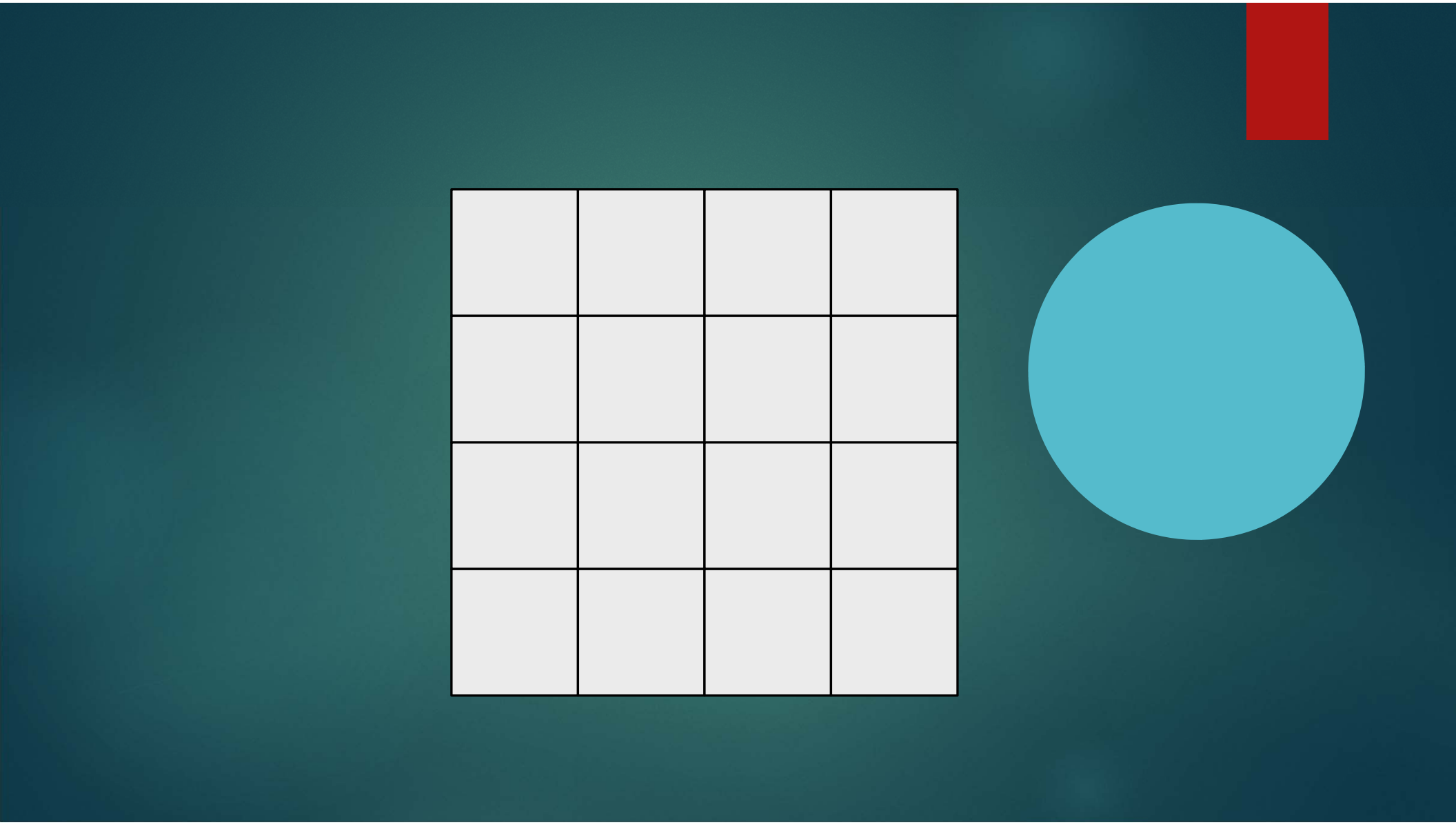
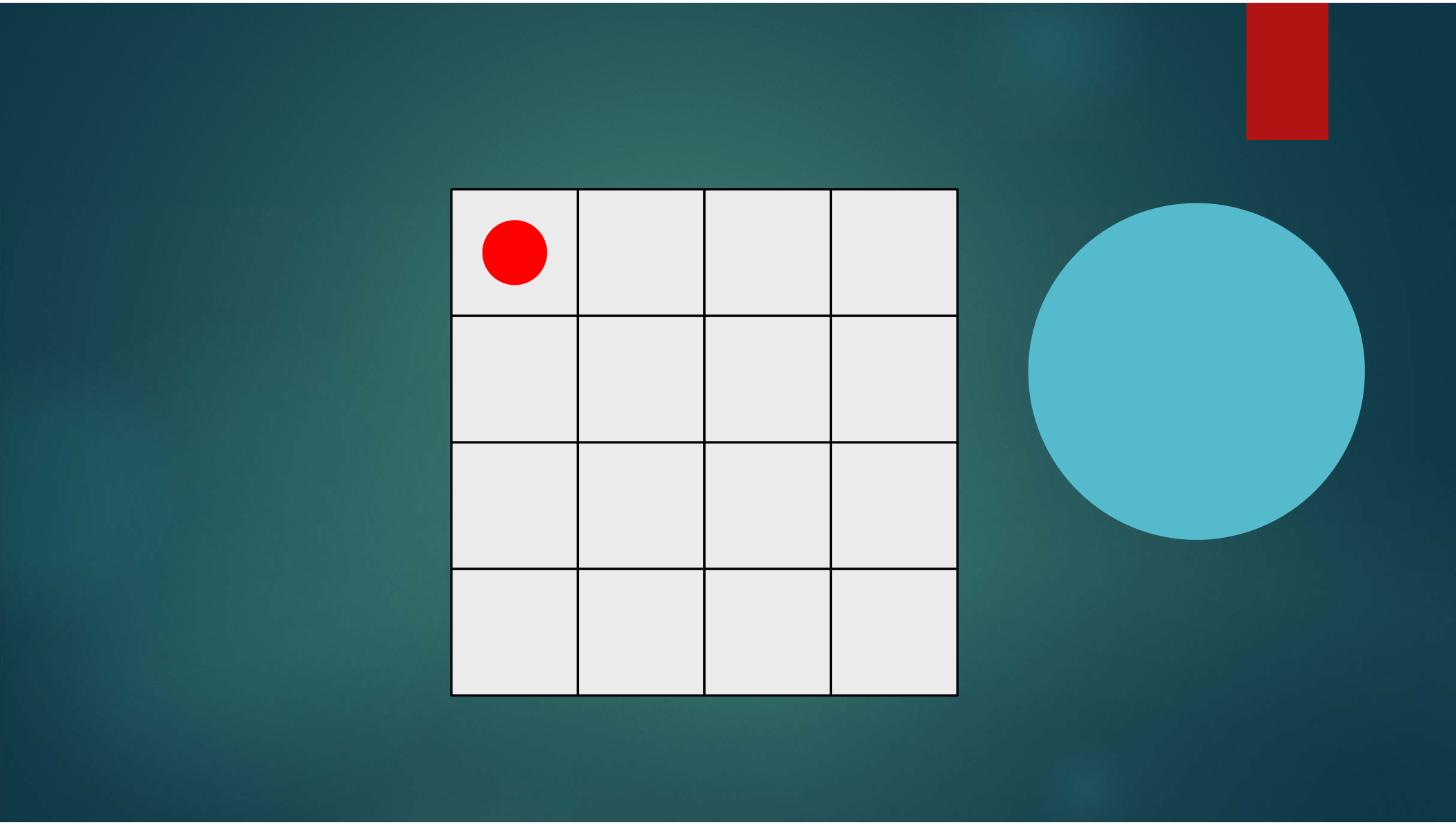
N-QUEENS PROBLEM

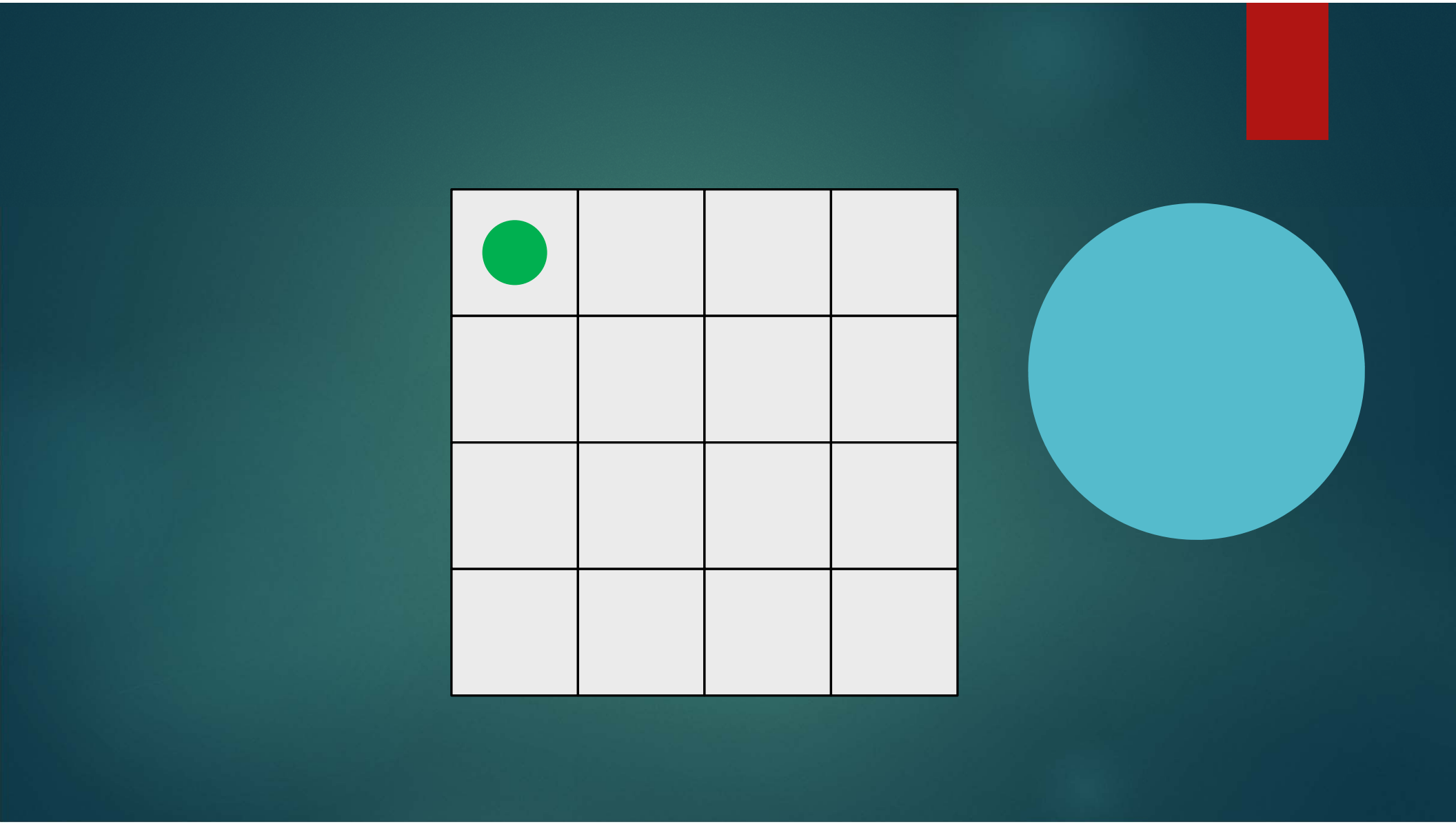
BACKTRACKING

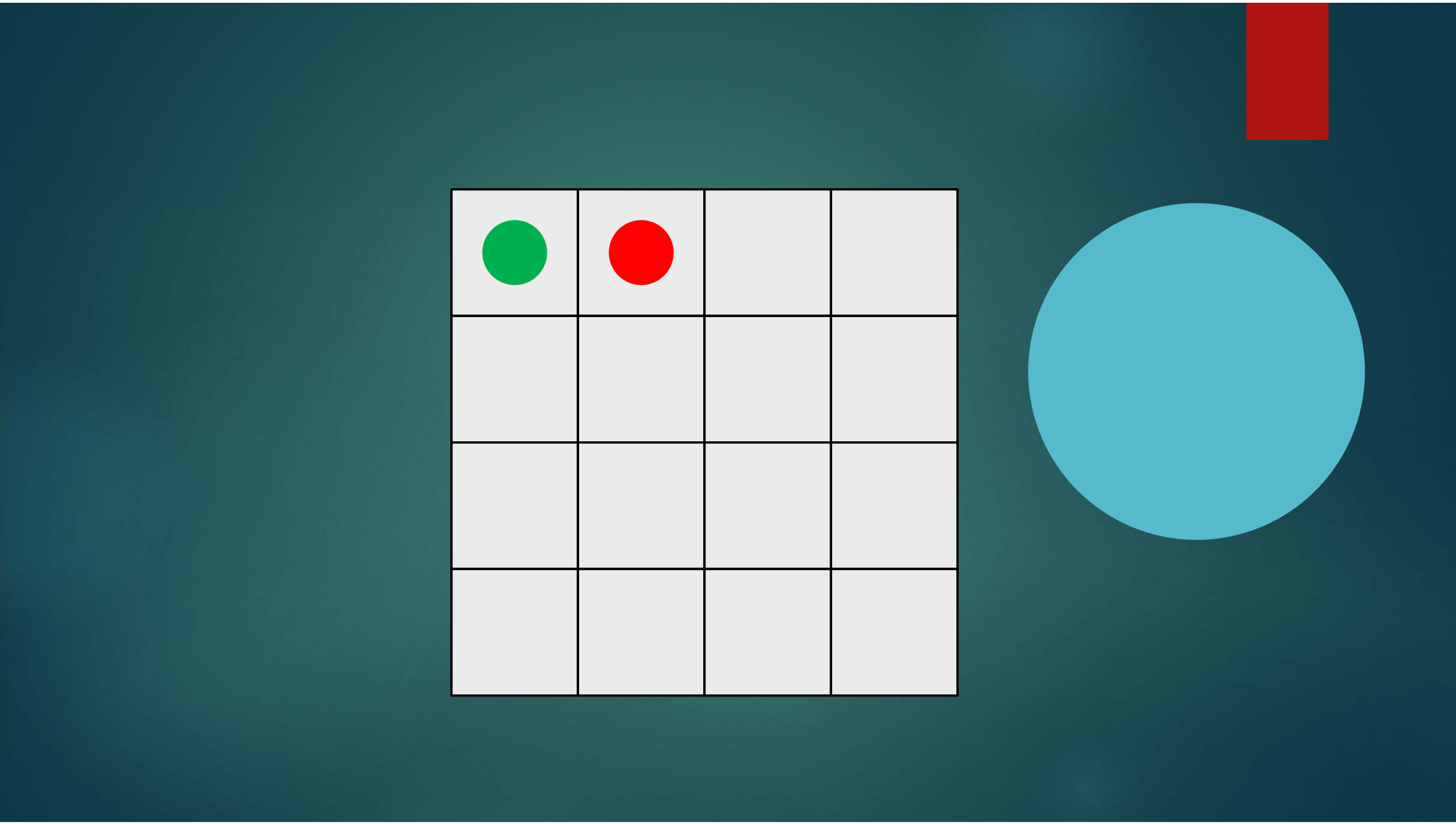
N-queens problem

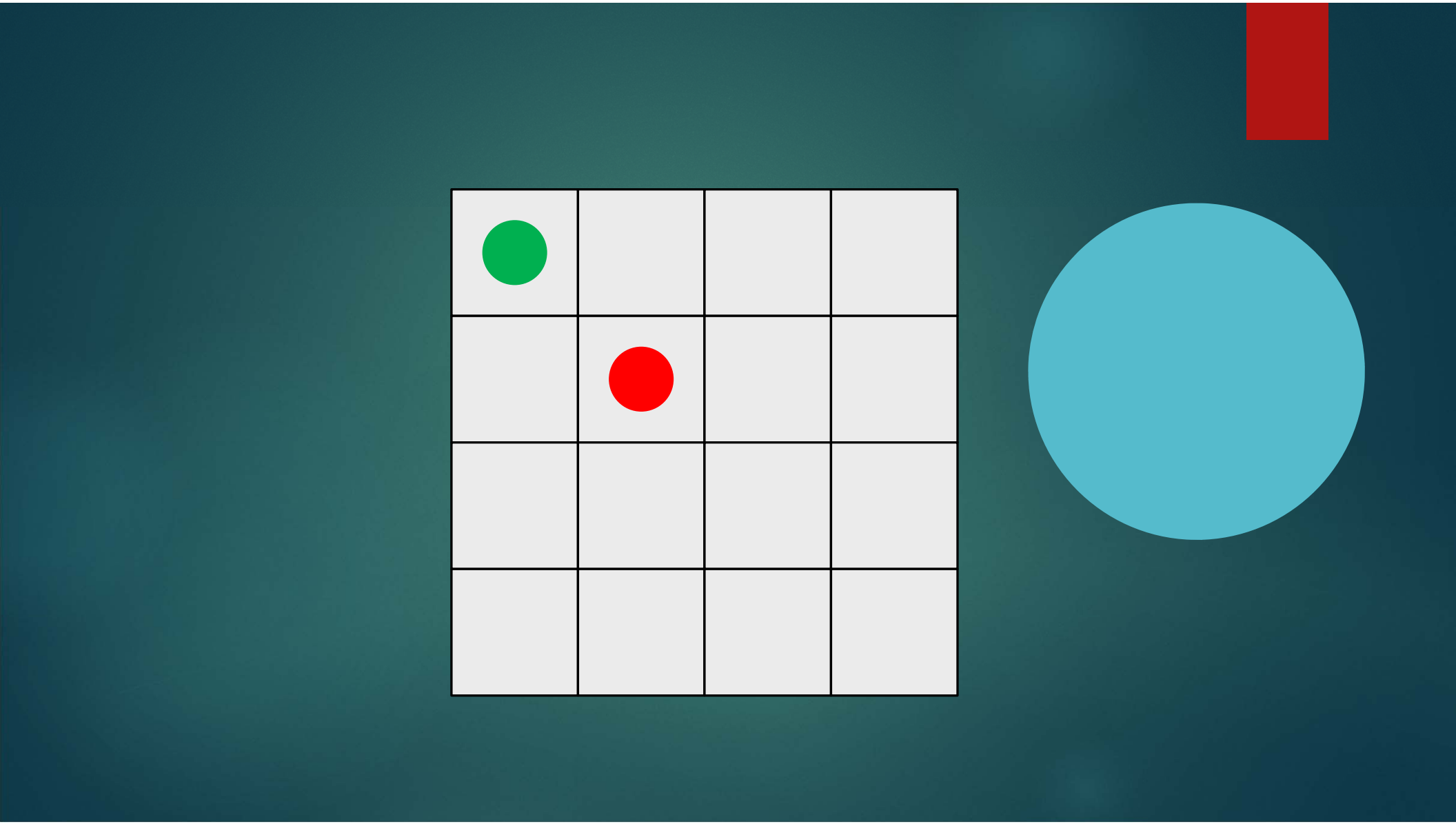
- The problem of placing **N** chess queens on an **N×N** chessboard so that no two queens threaten each other (they will not be able to attack each other)
- We have to consider: queens can move diagonal directions too ...
- The original problem was designed for **8** queens ... the general form is about **N** queens
- Gauss worked on this problem
- Dijkstra used this problem to illustrate the power of what he called structured programming

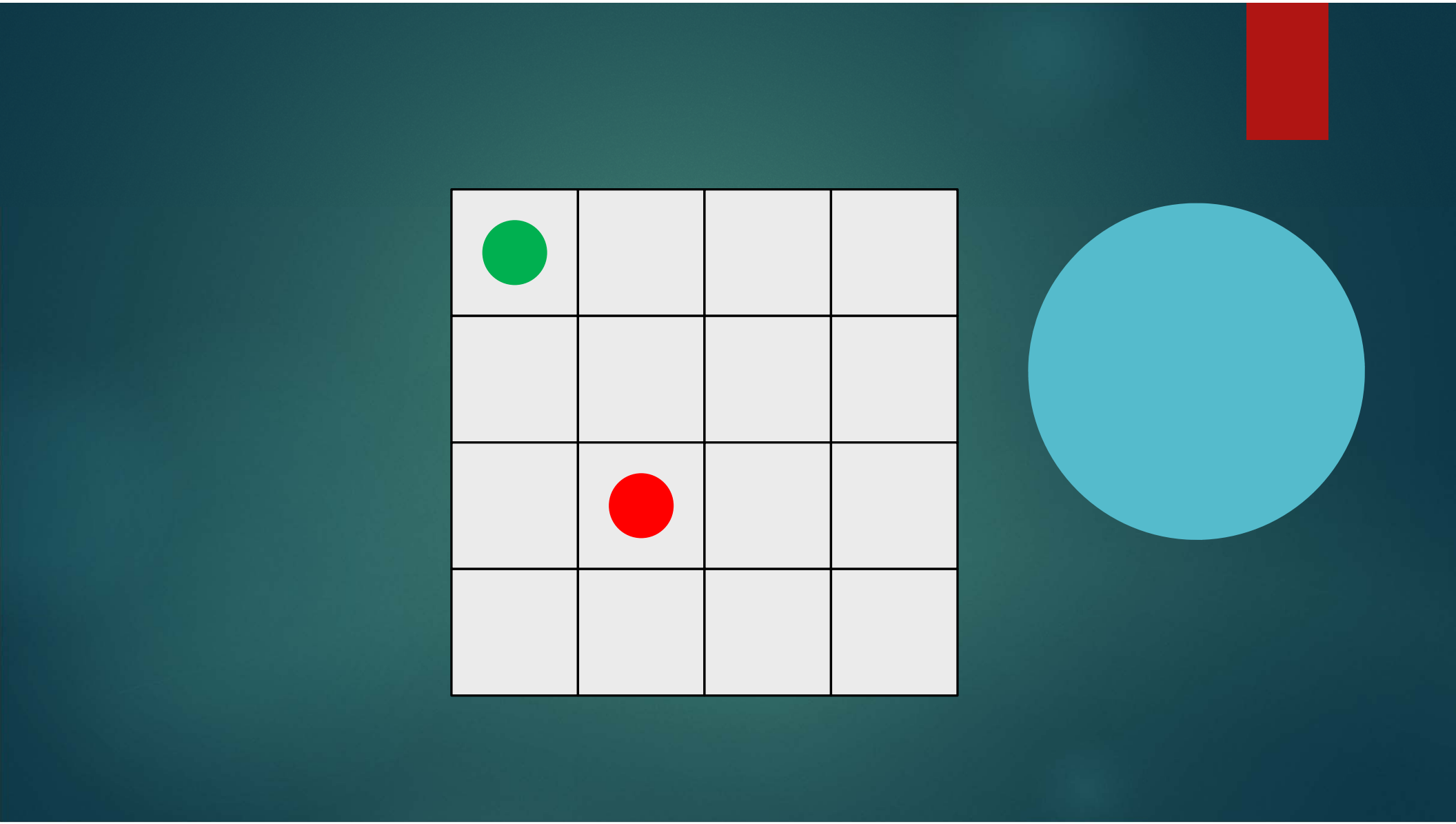


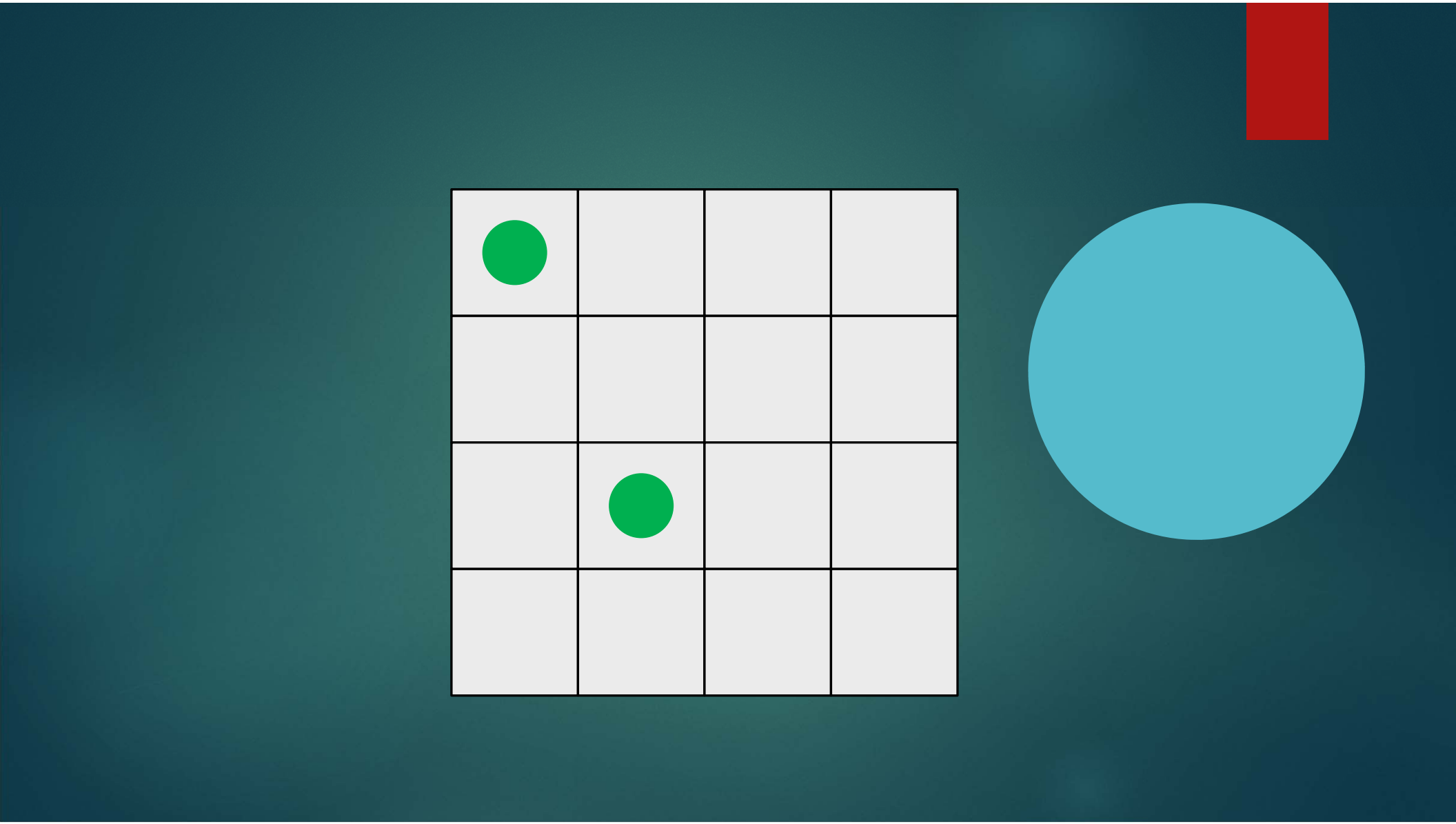


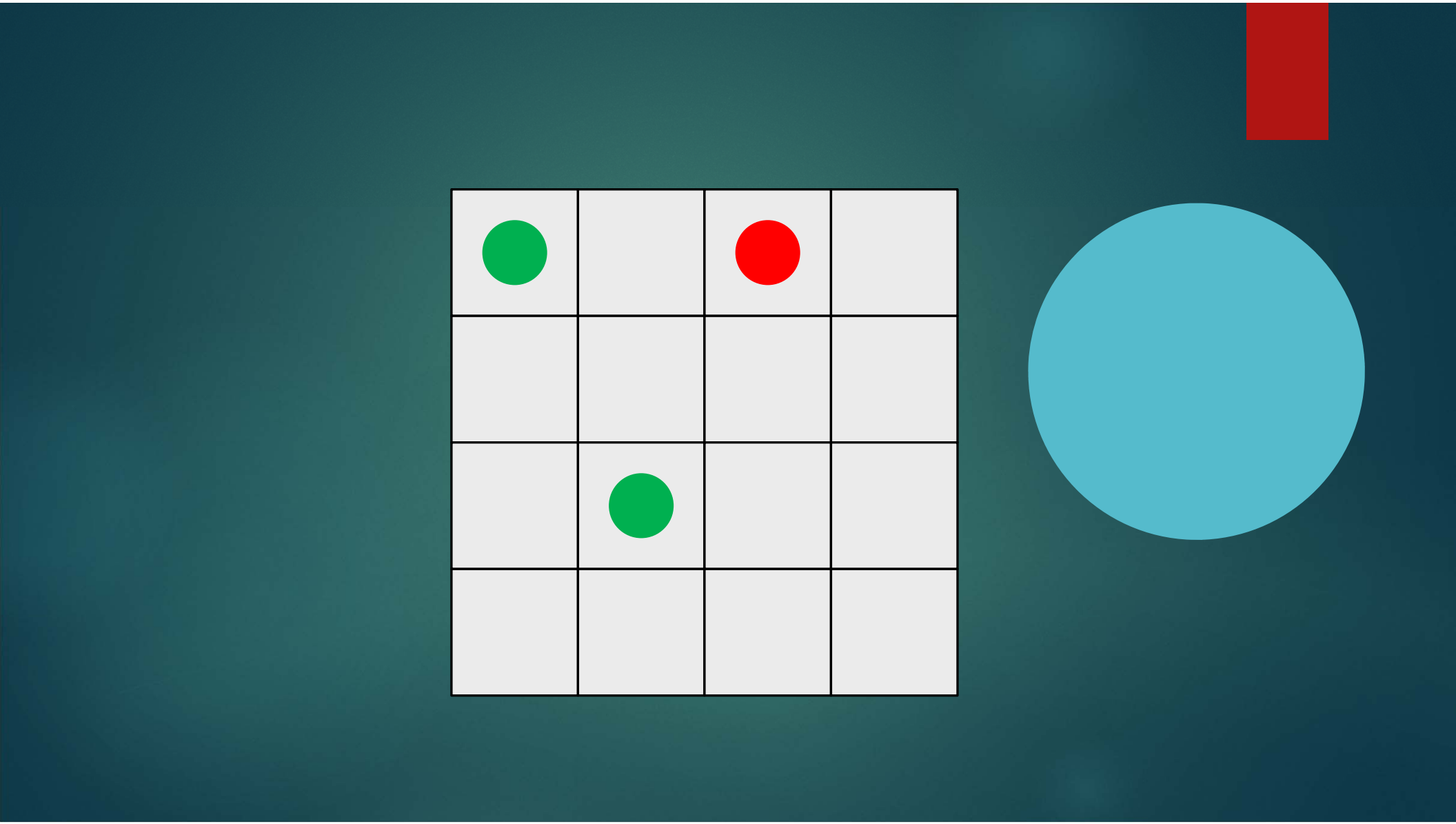


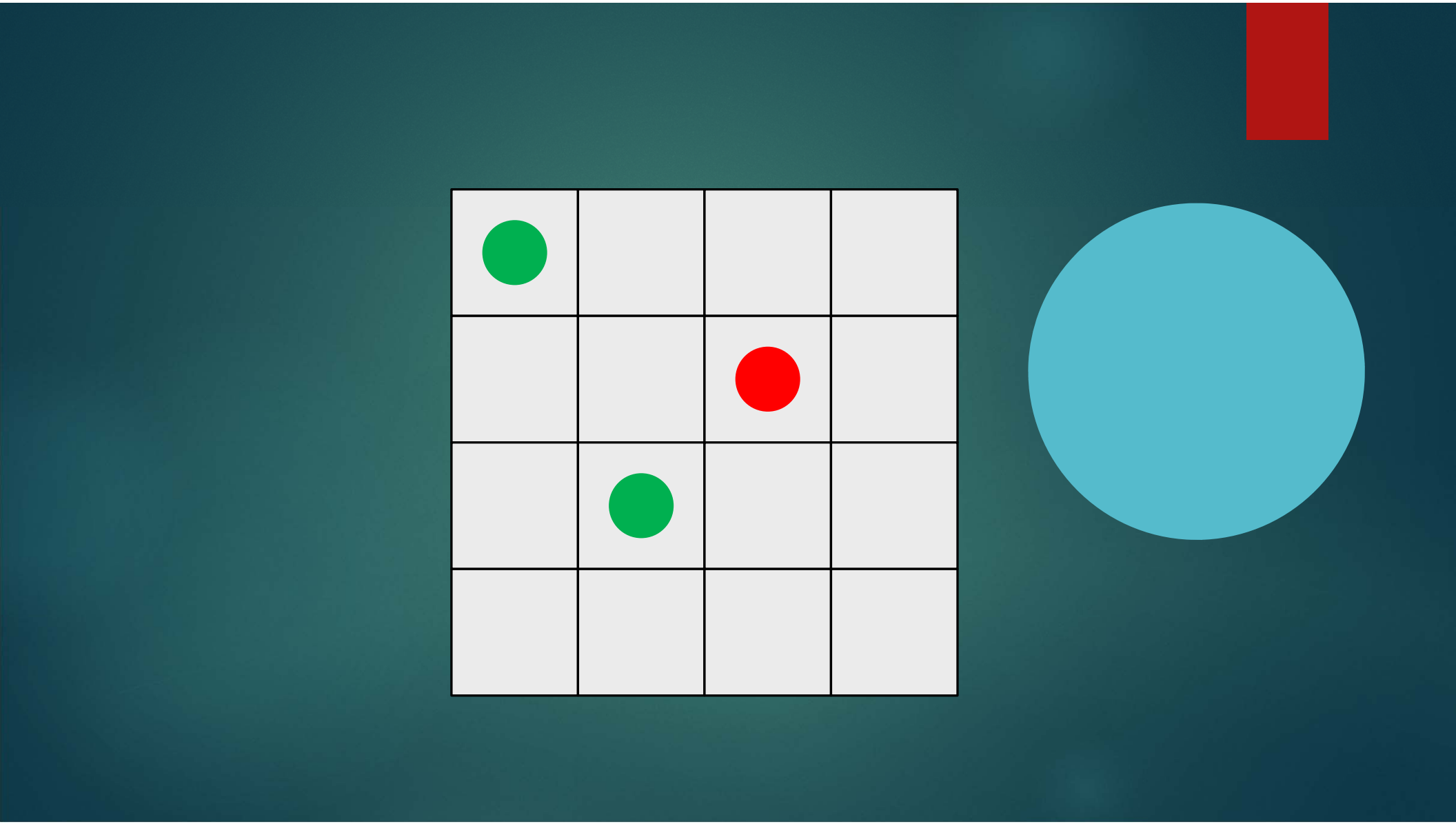


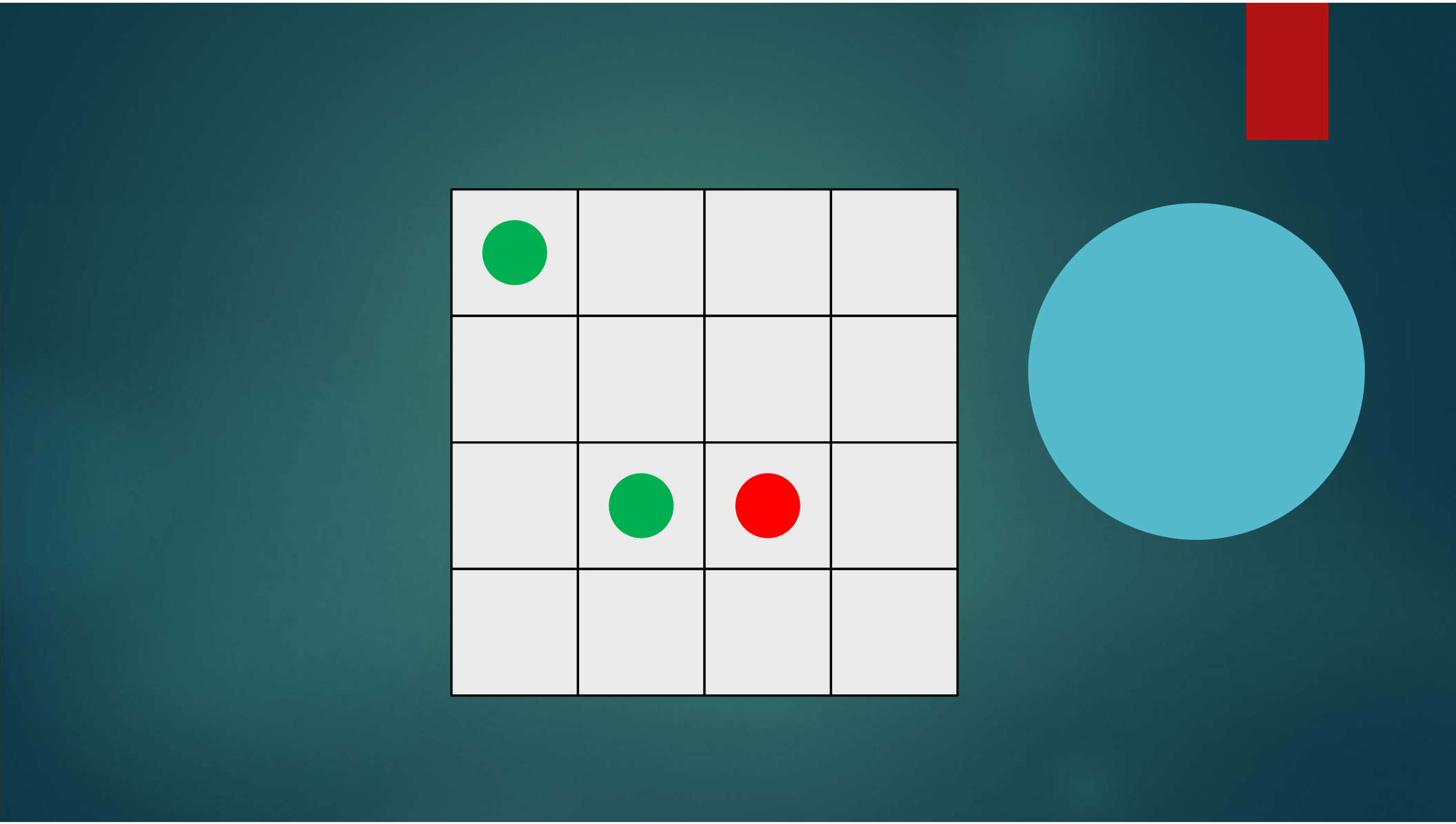




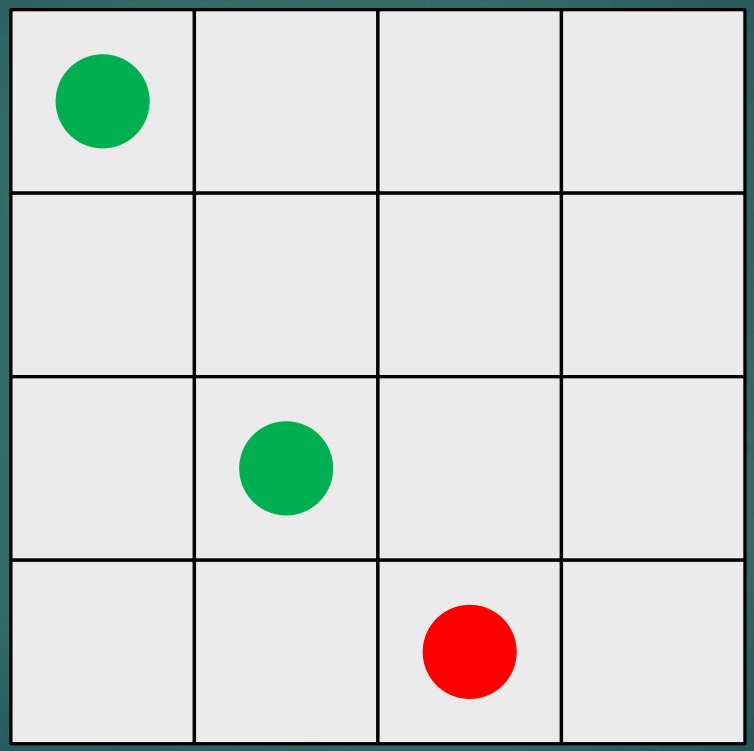


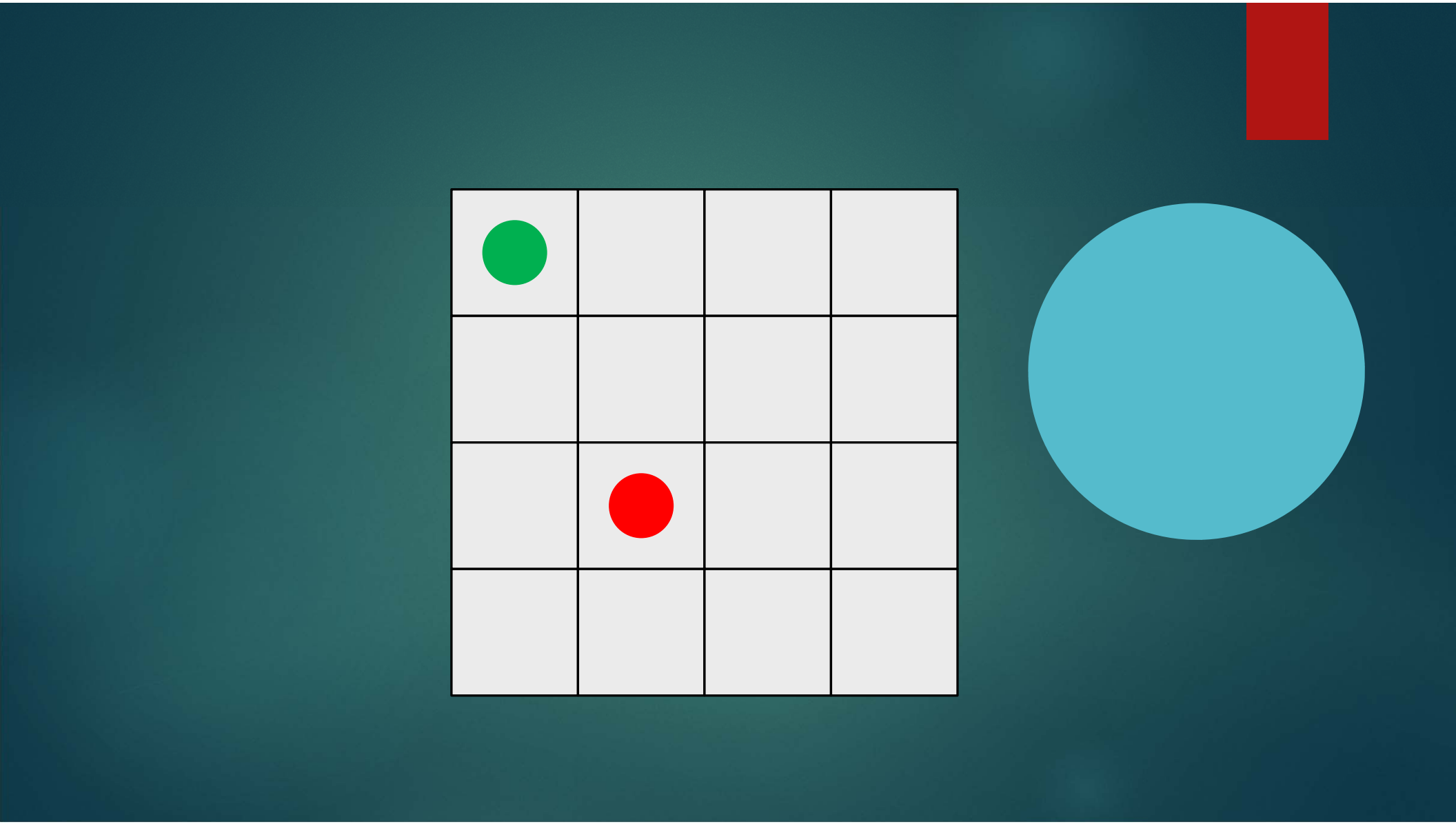


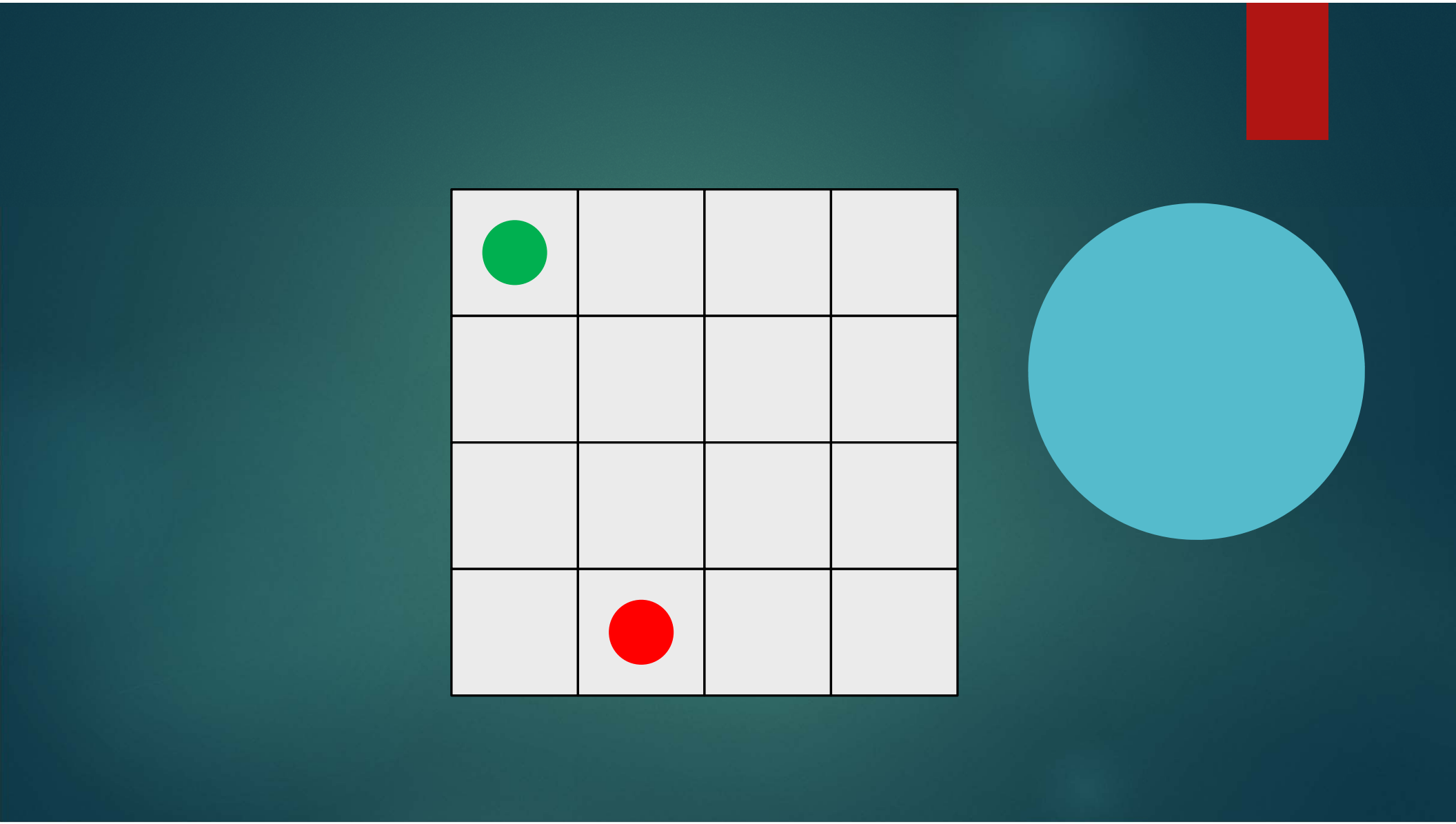


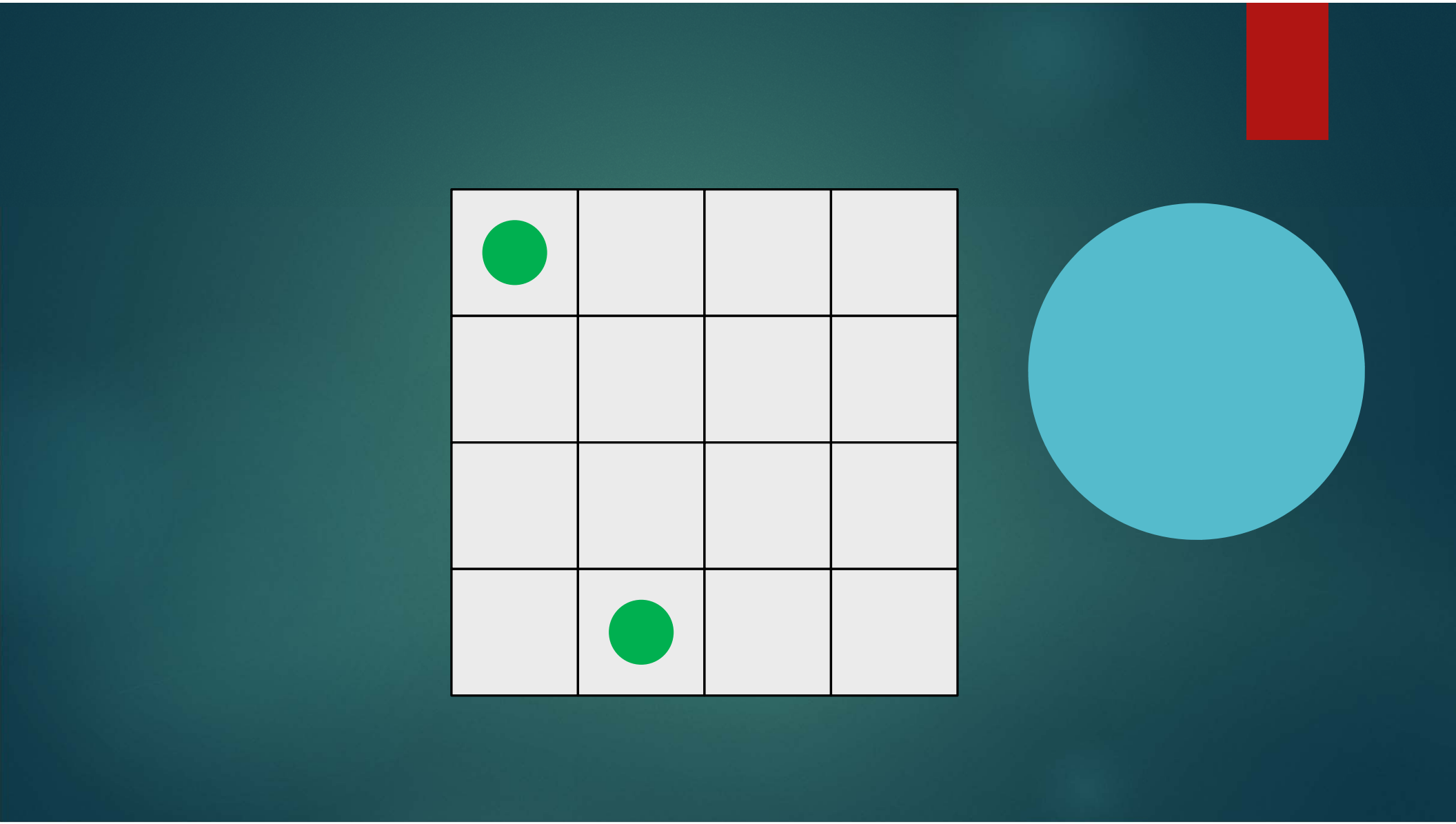


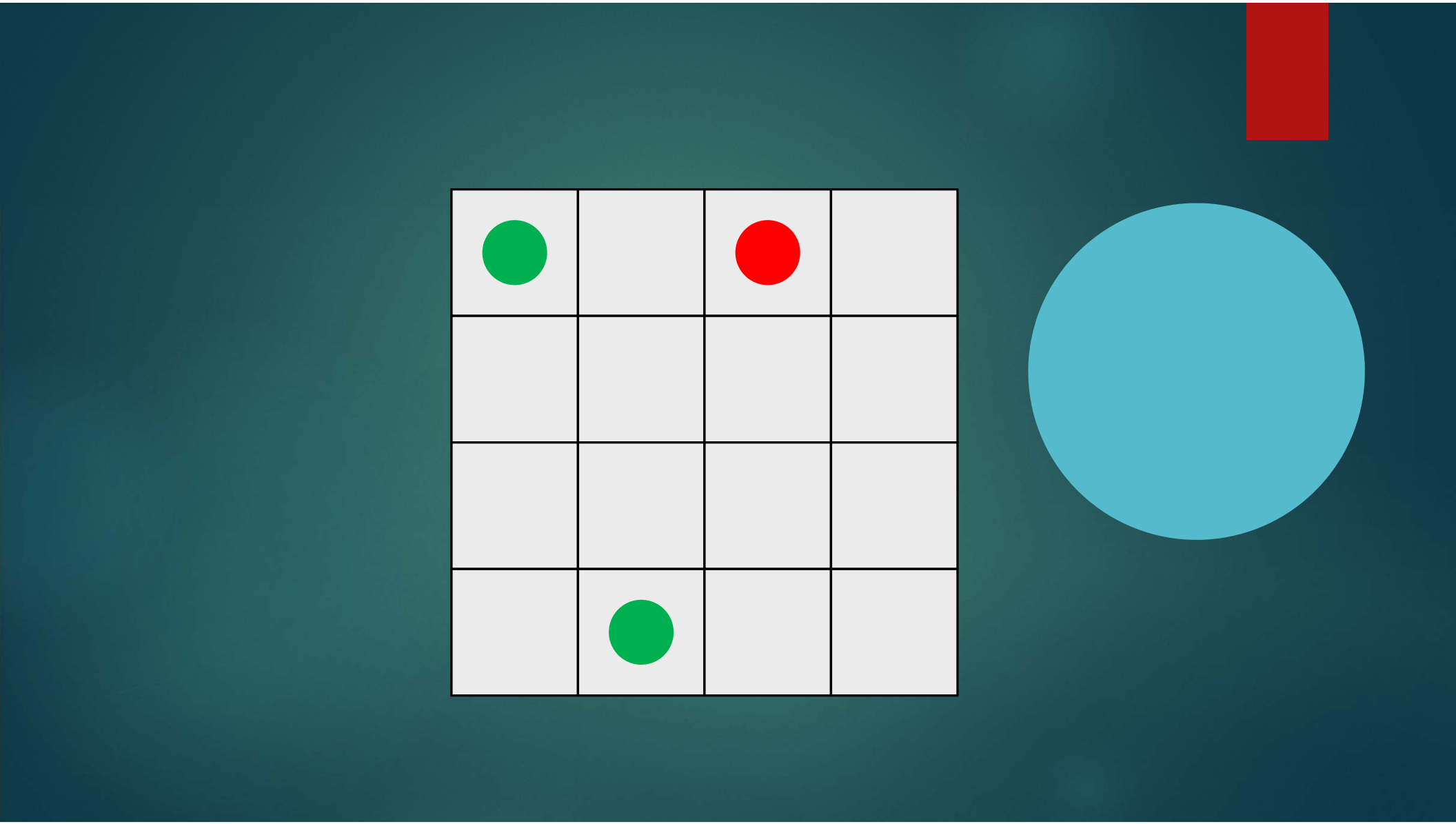
No feasible solution: we have to step back to the previous column and increment the position of the queen there ~ **BACKTRACK** !!!

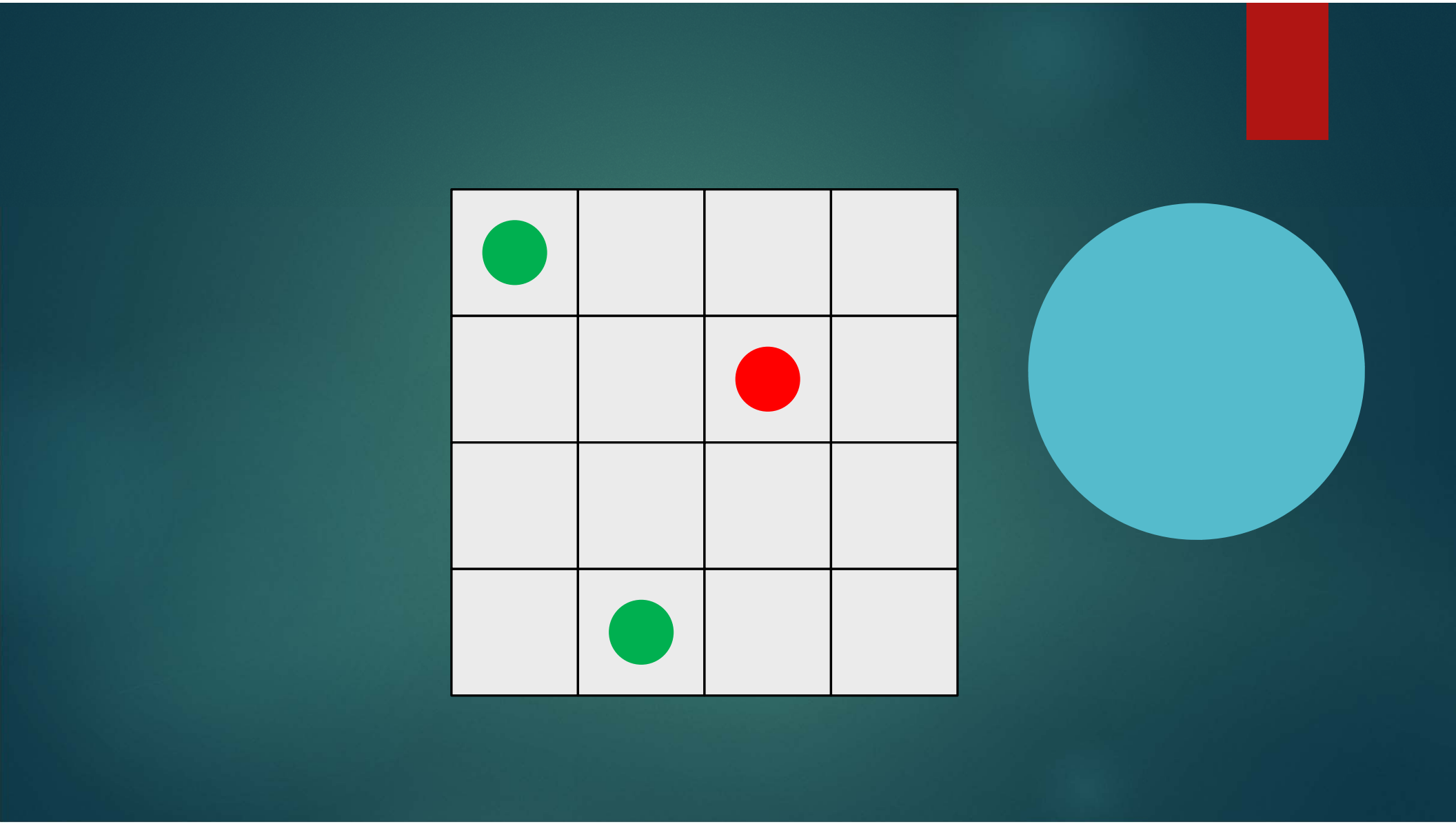


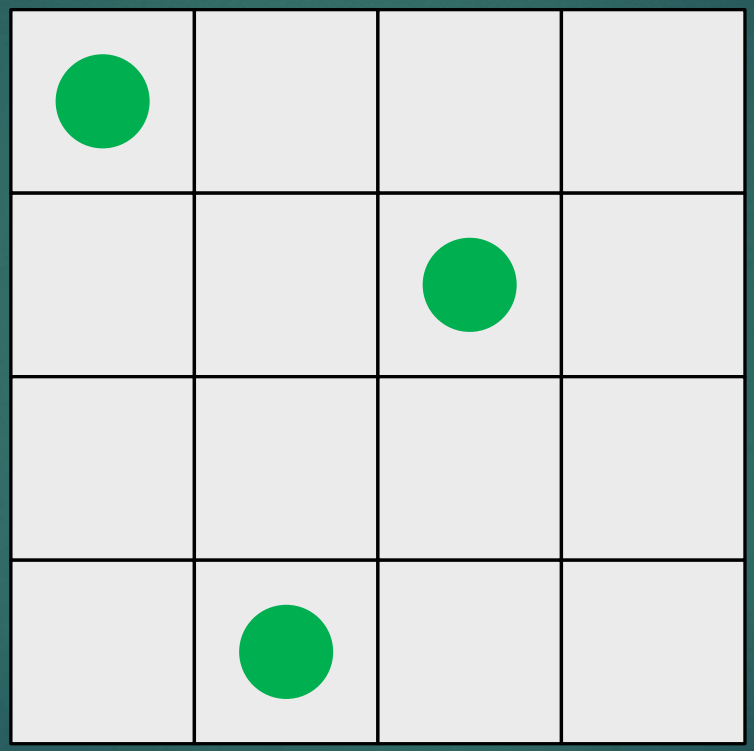


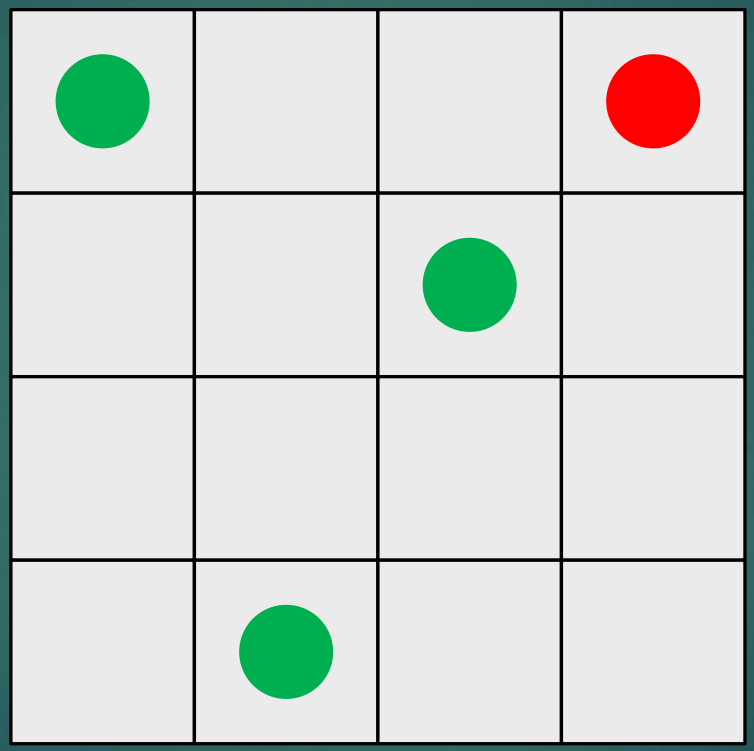


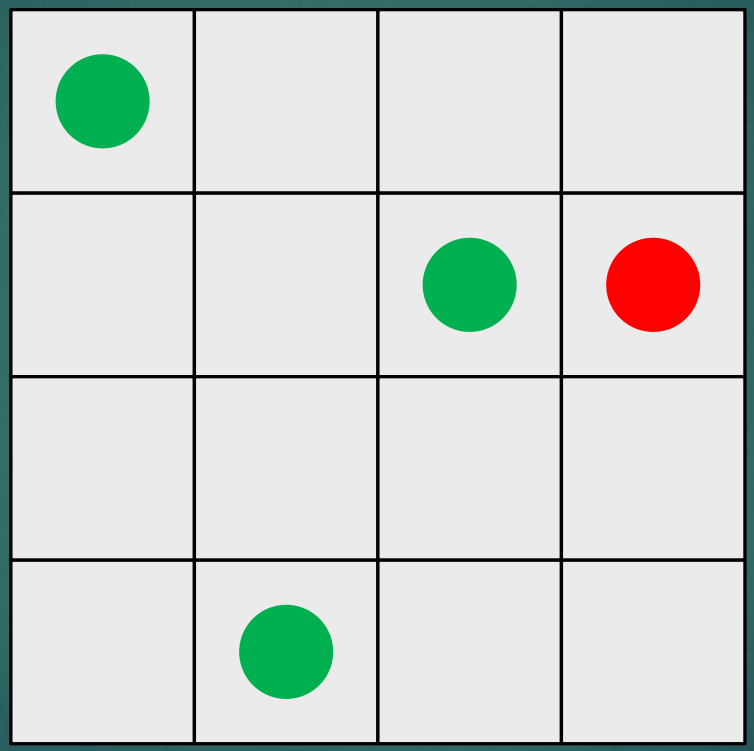


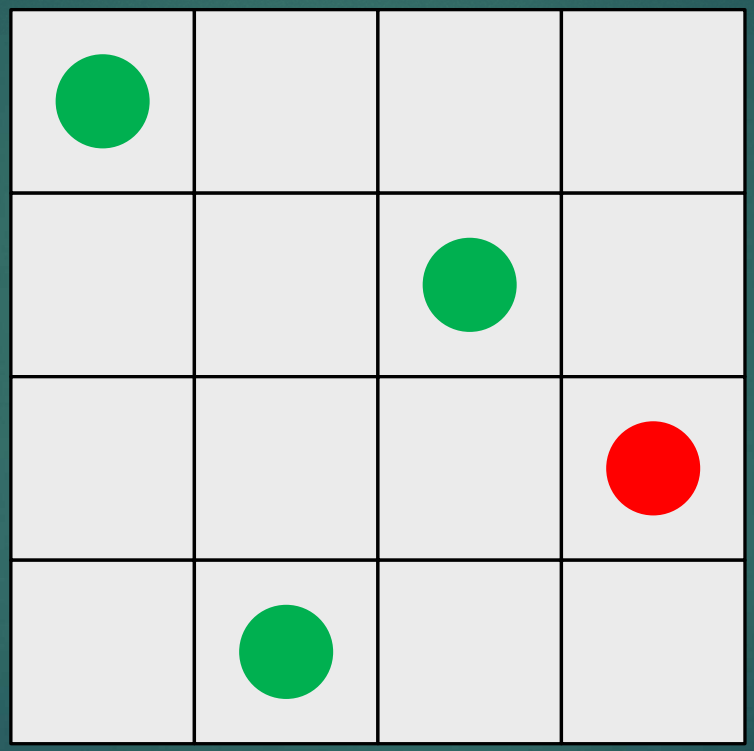




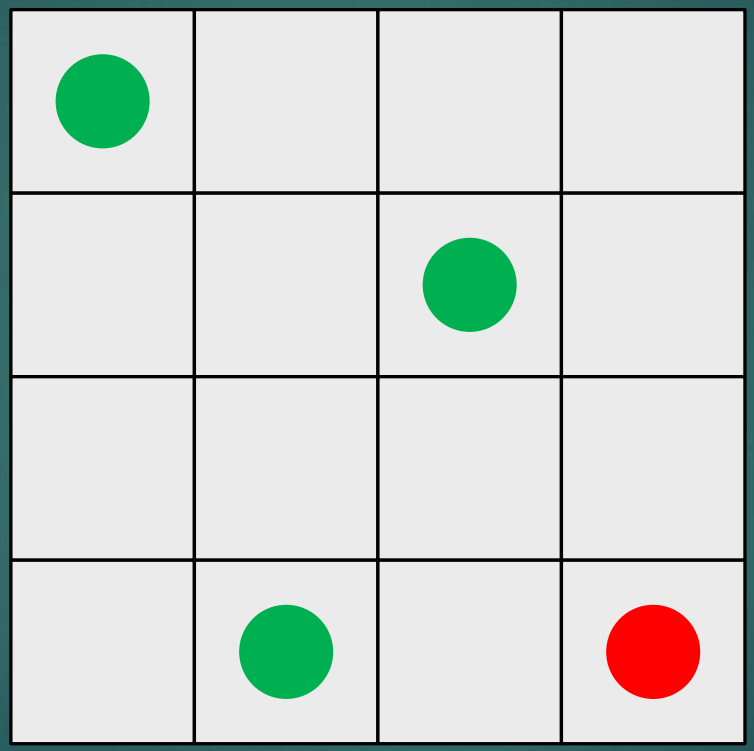


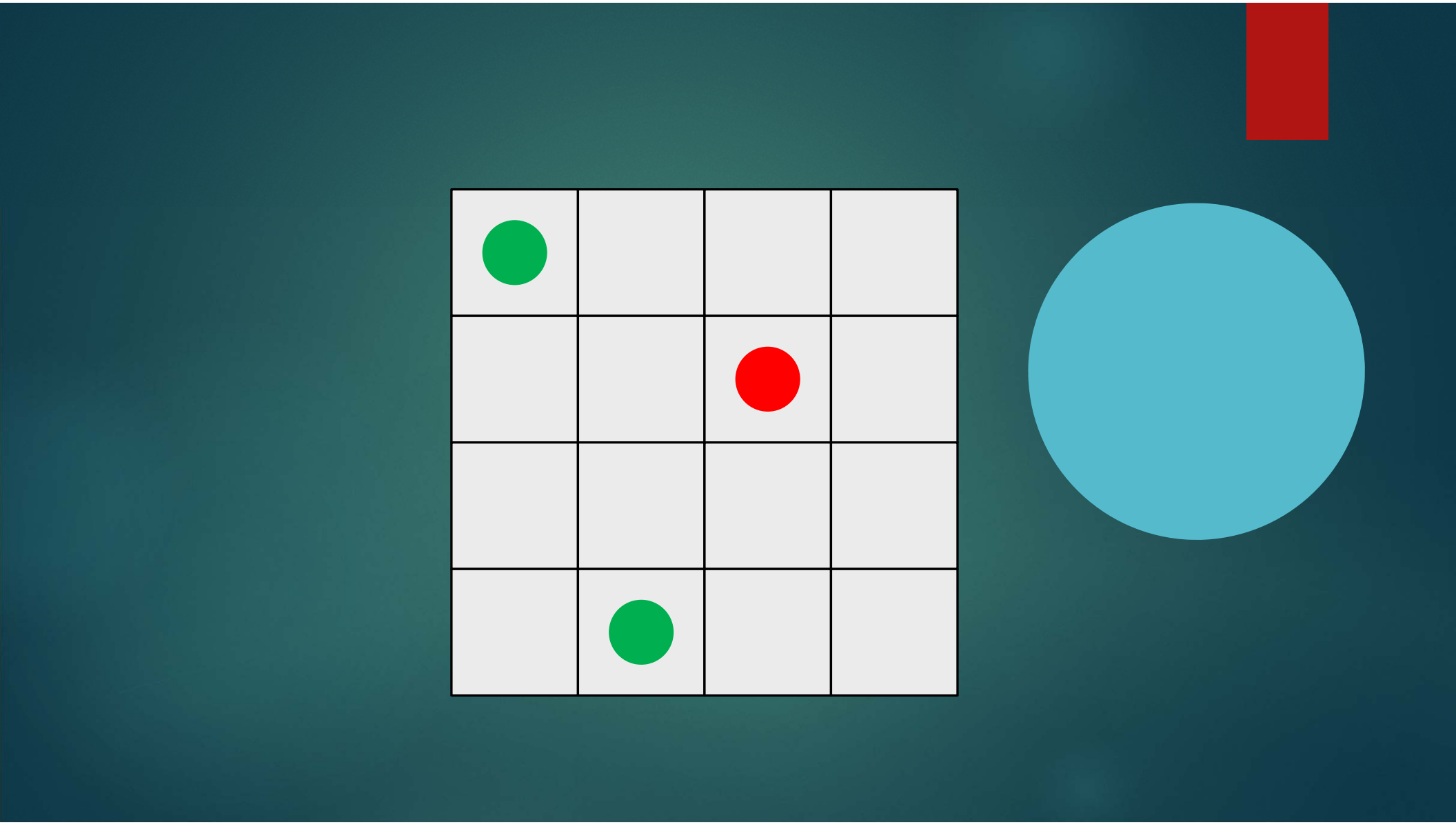


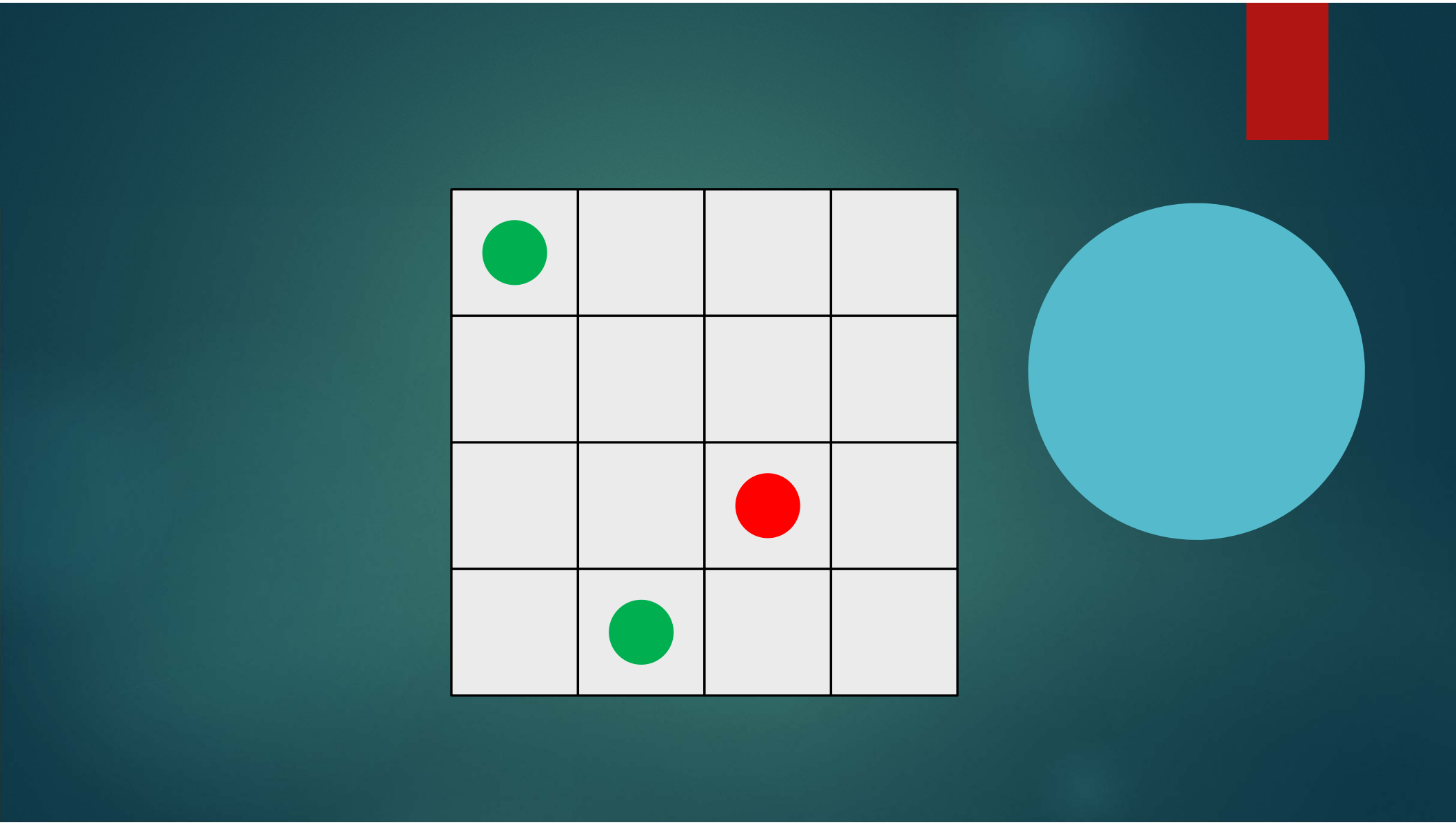


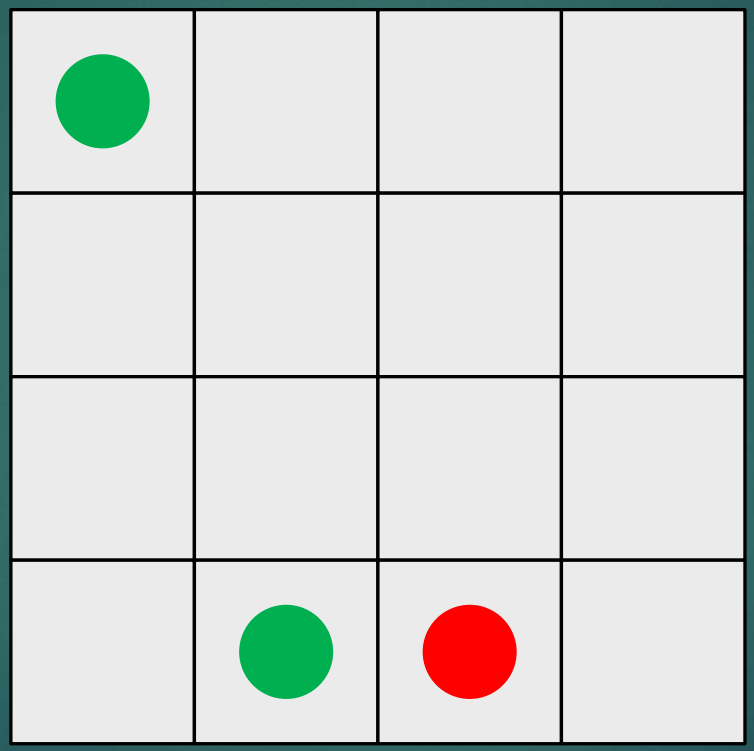


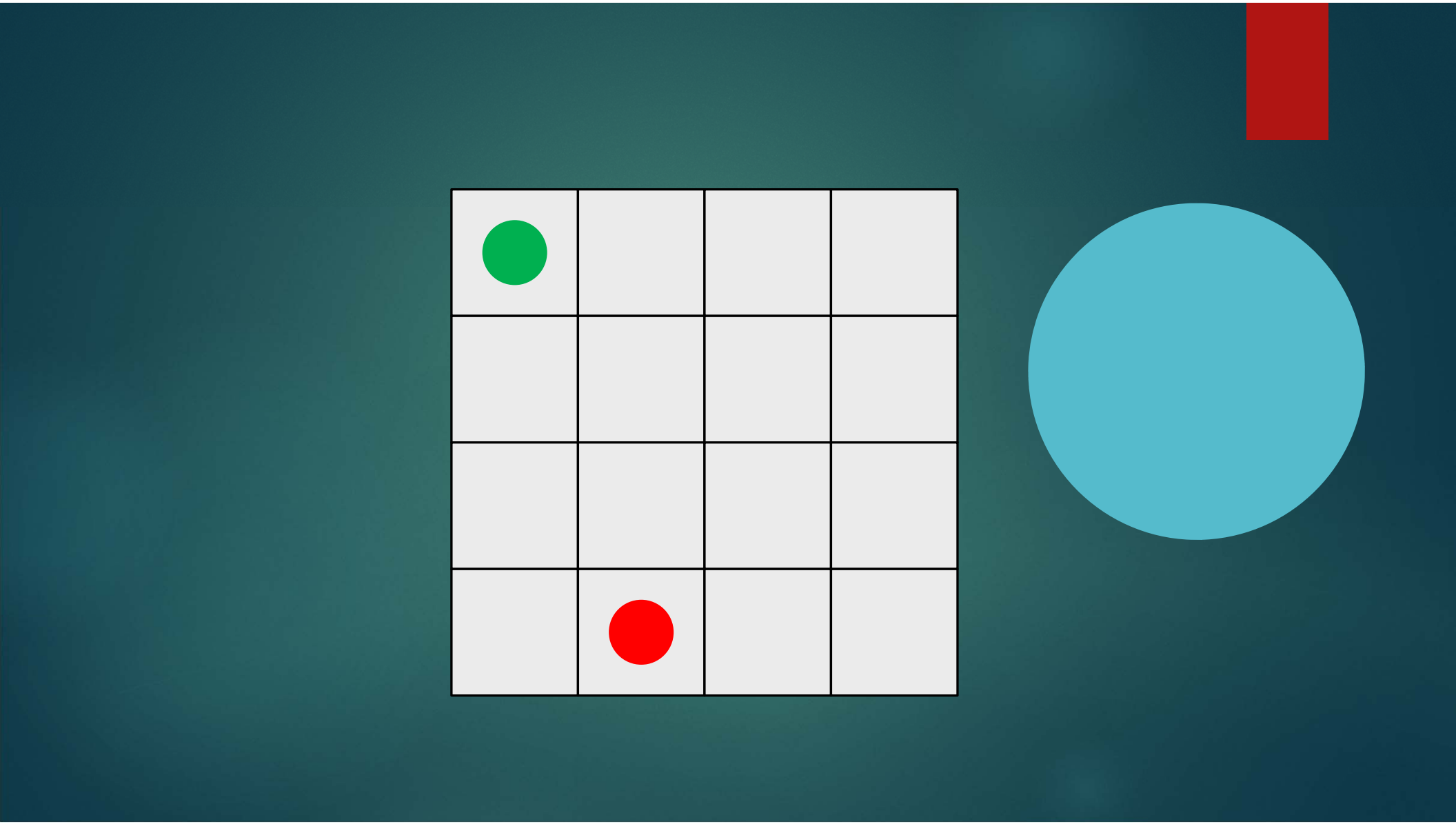
No feasible solution: have to step back ~ **BACKTRACK** !!!

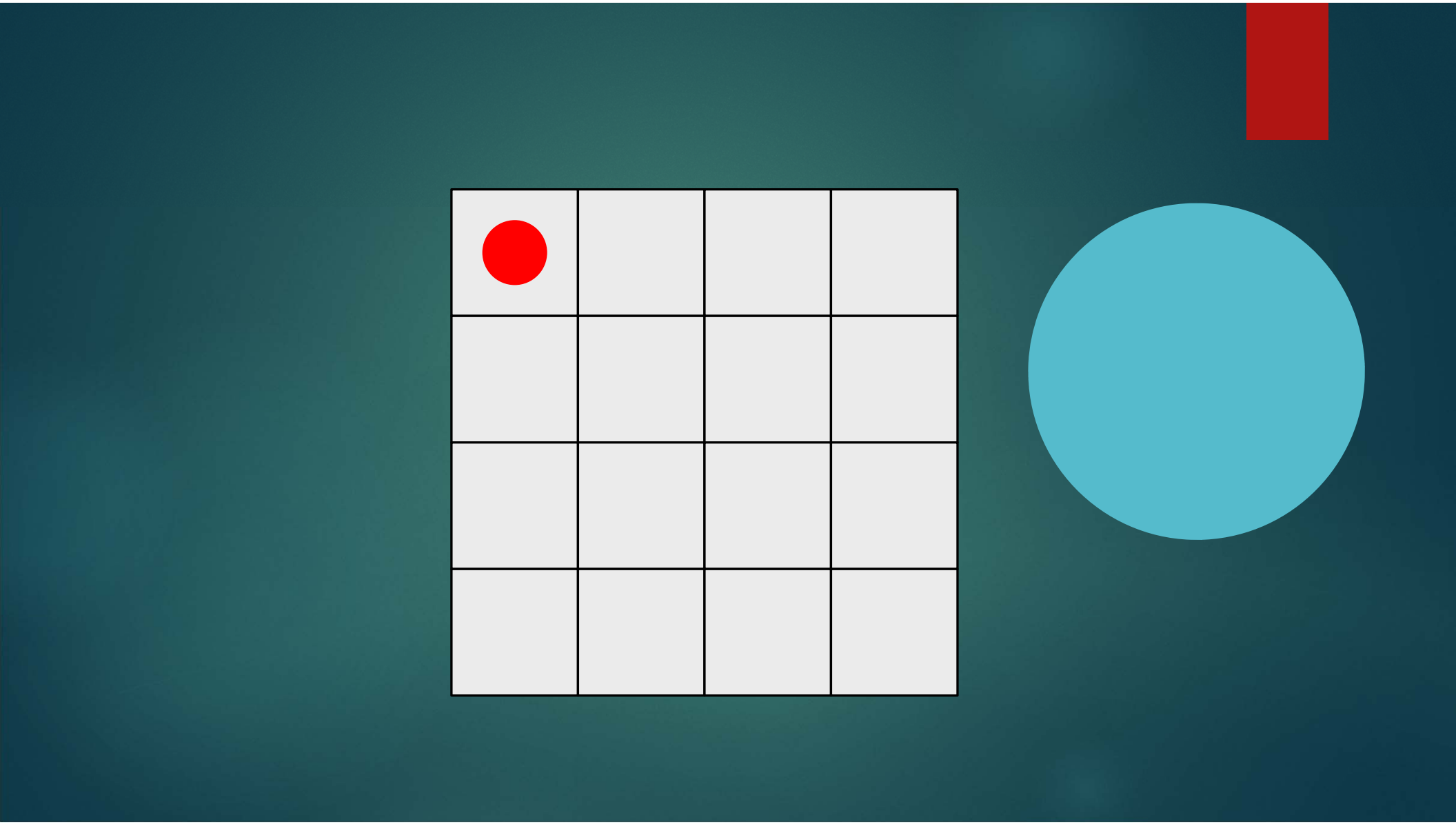


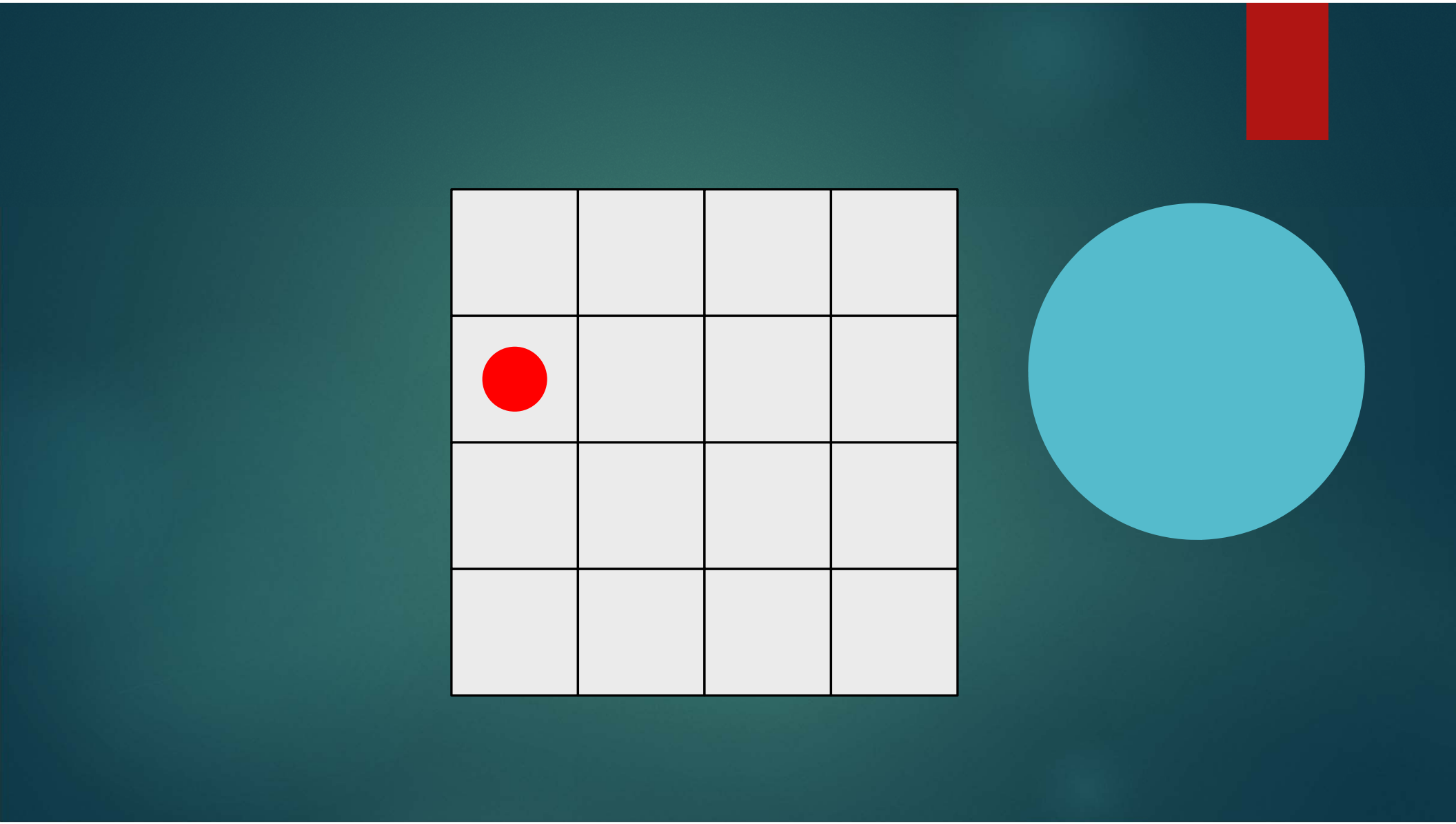


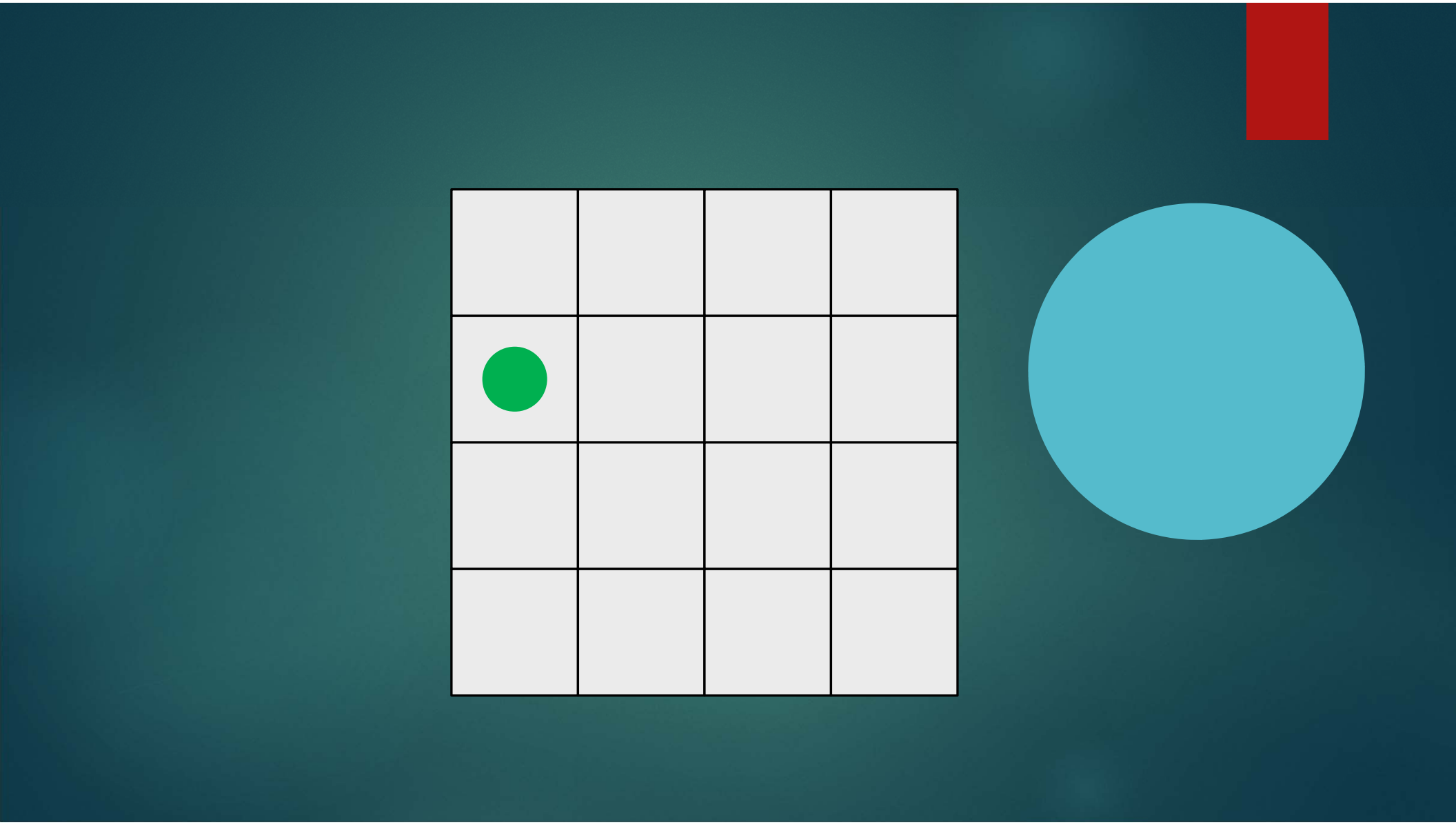


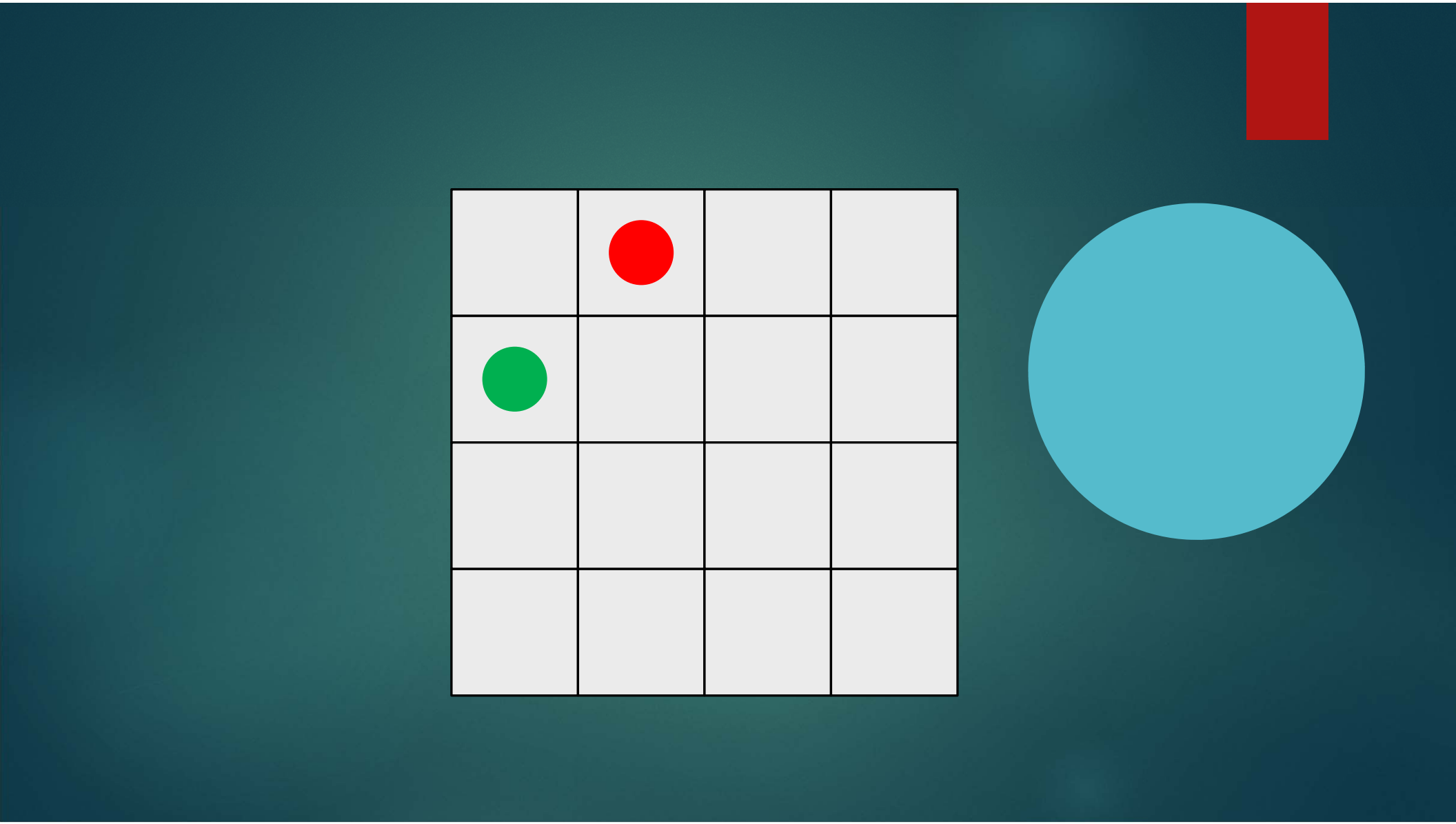


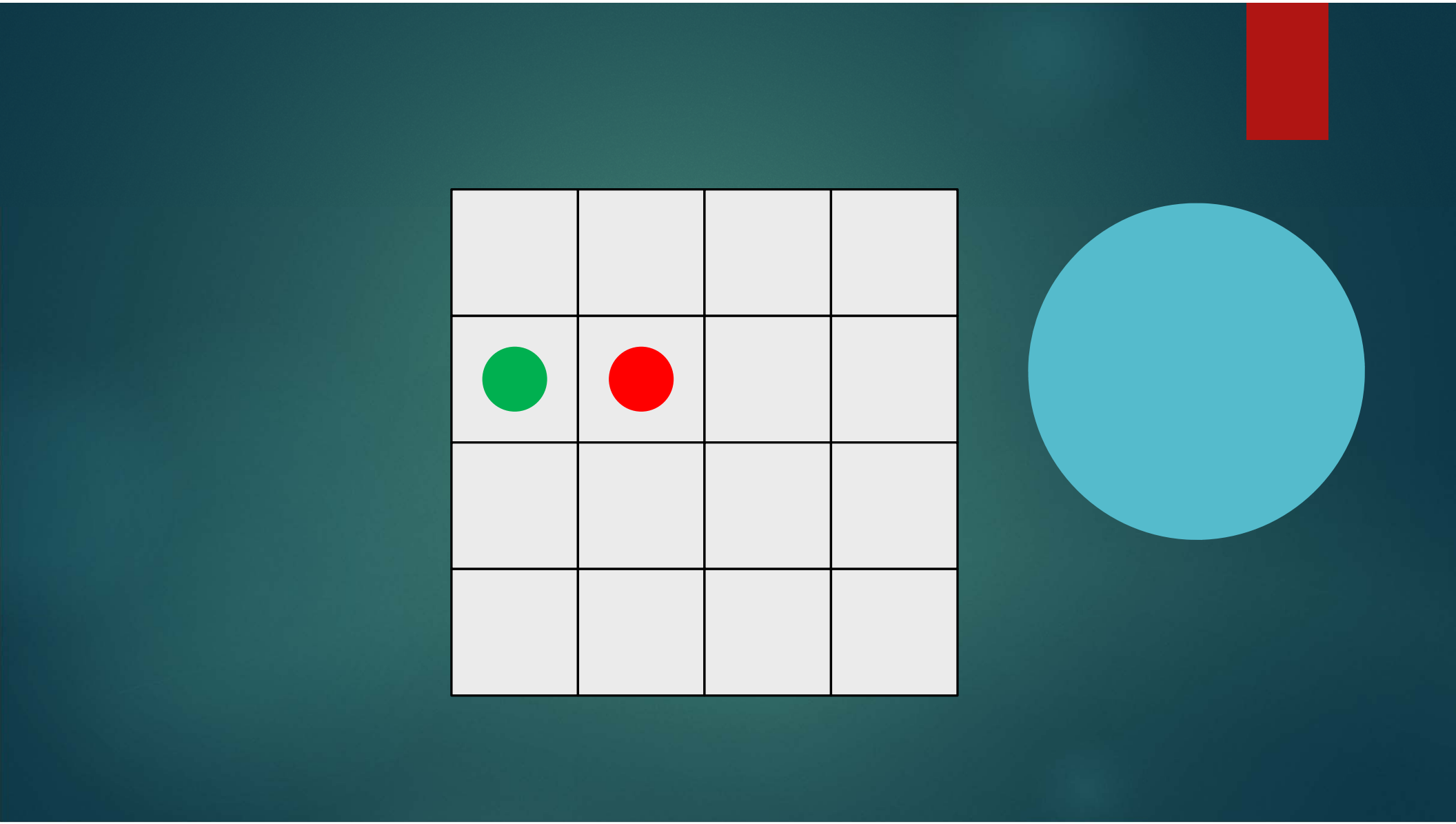


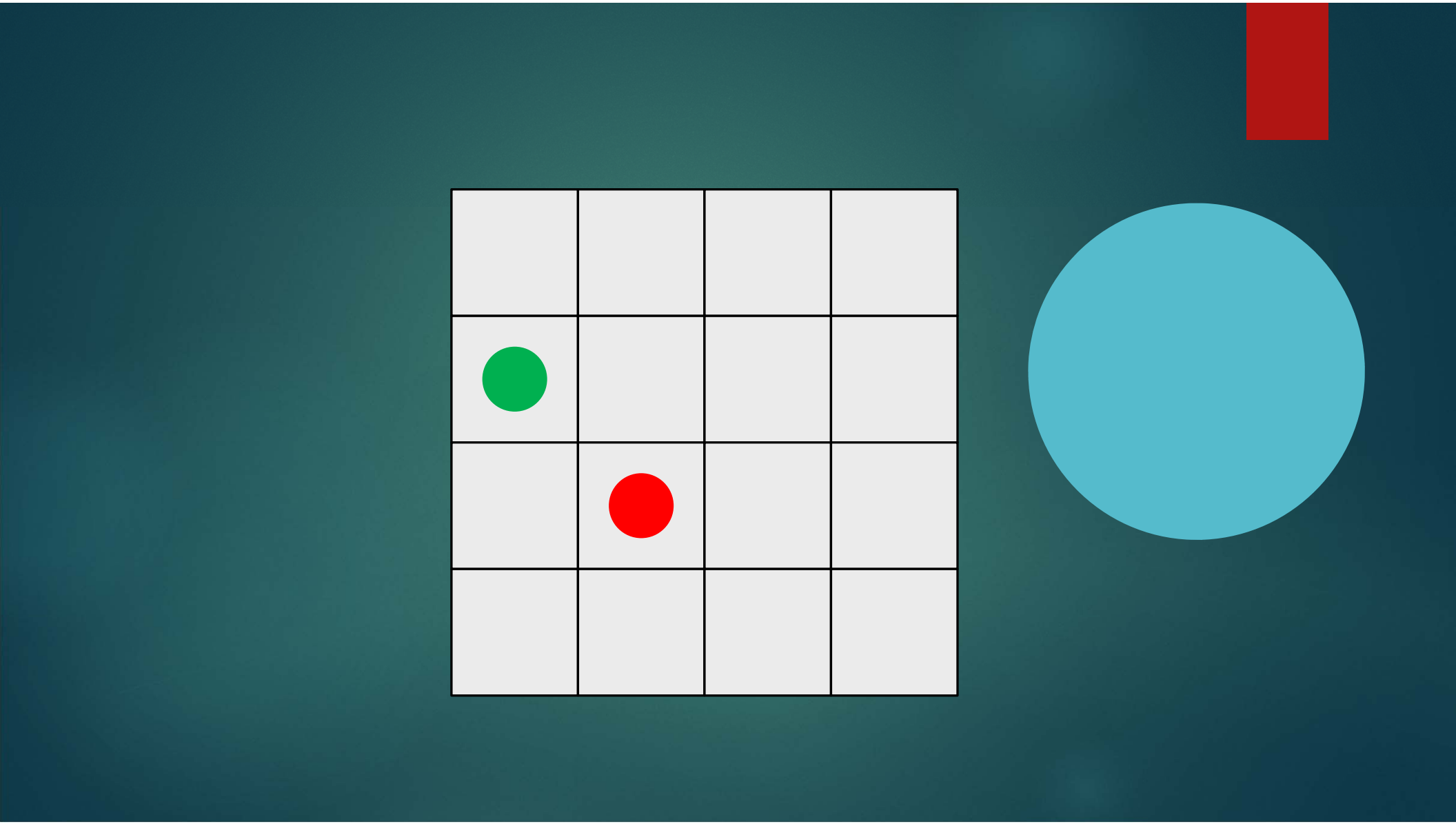


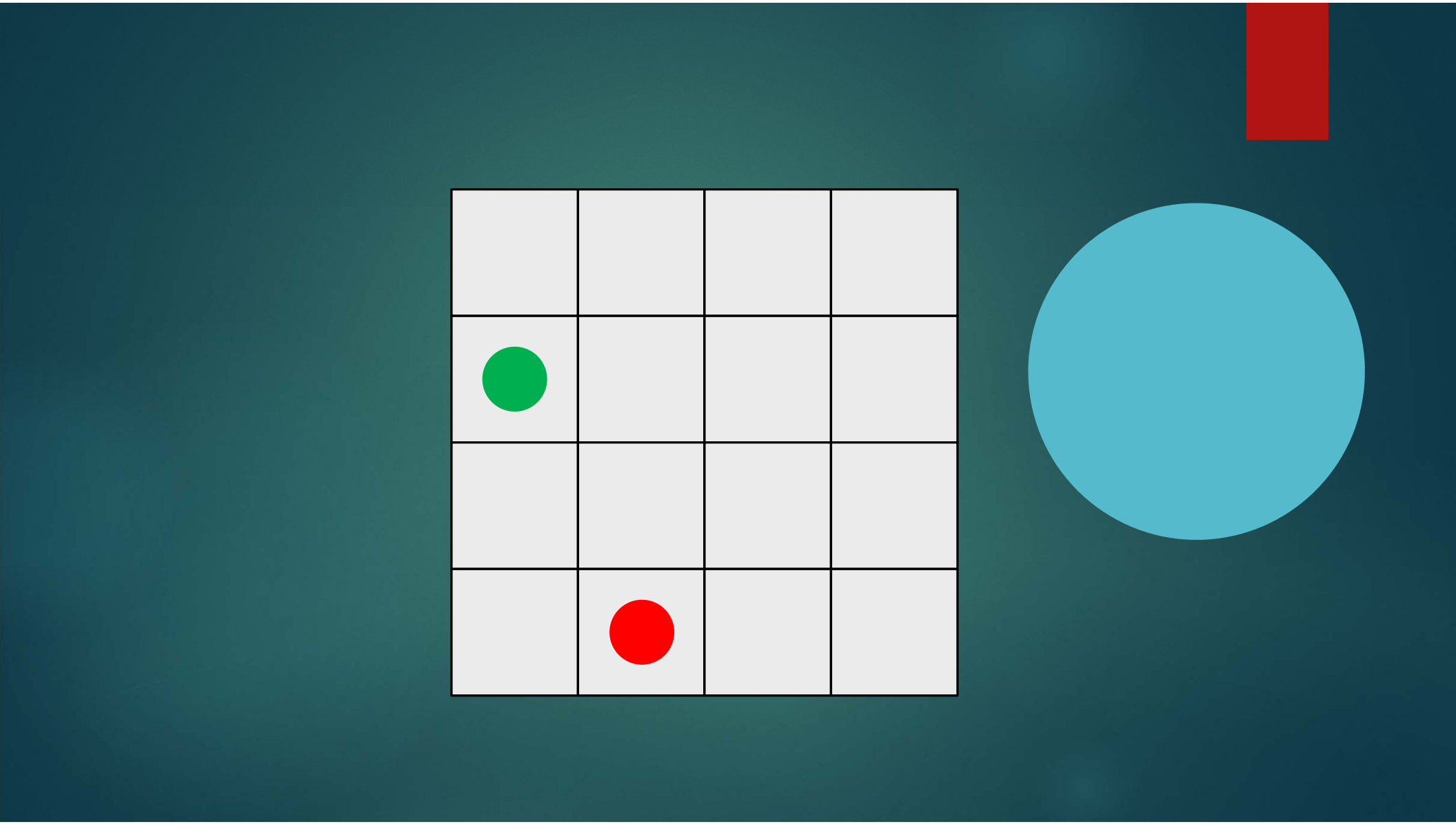


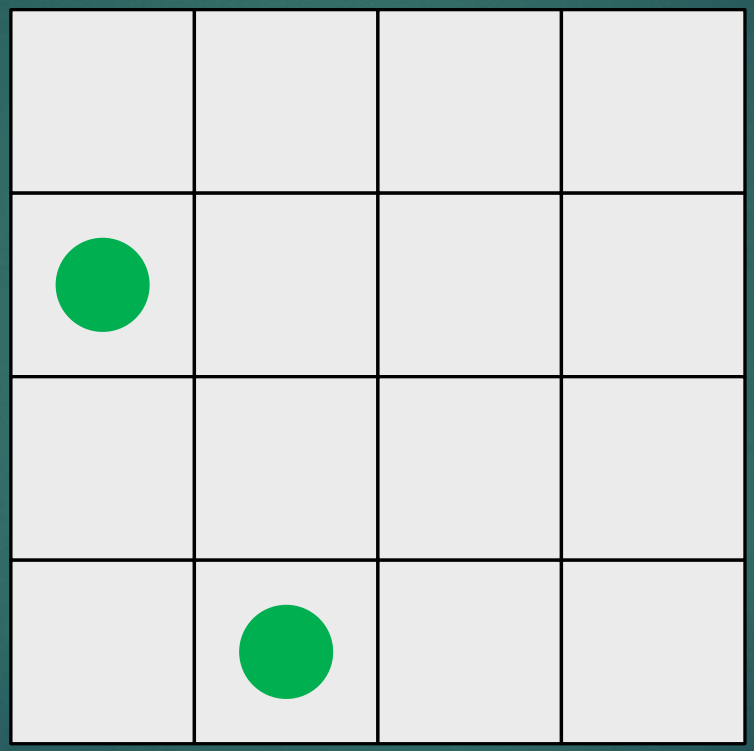


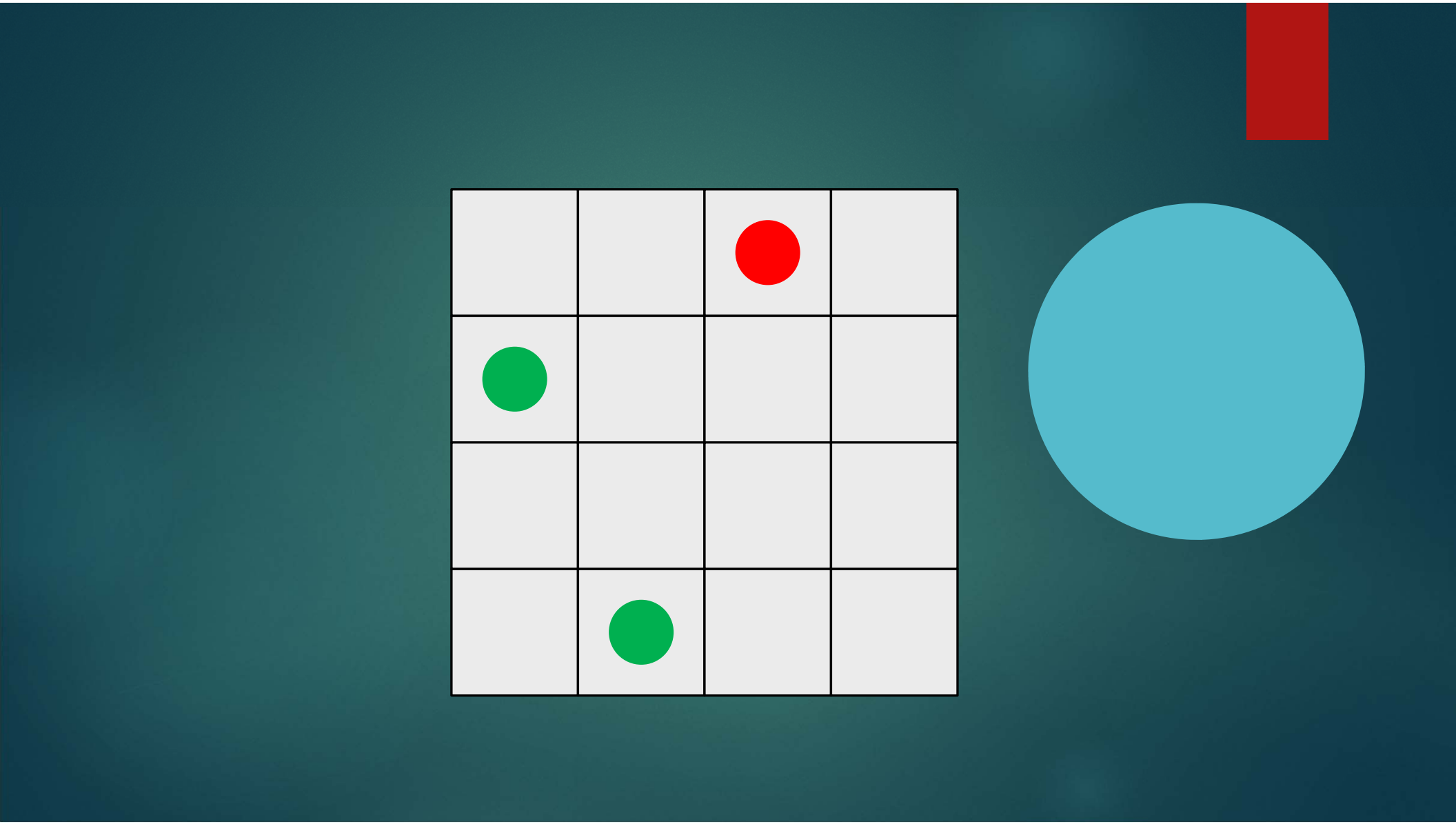


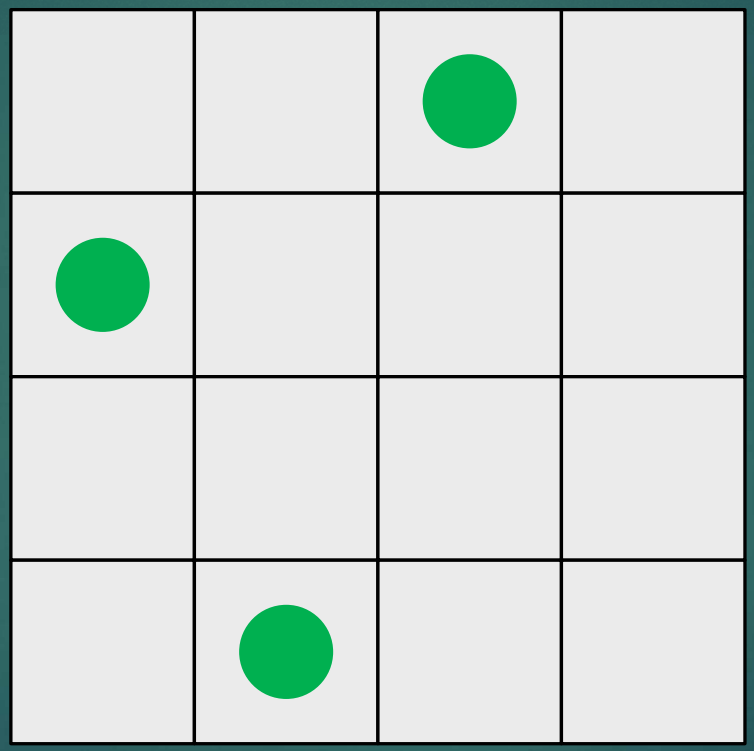


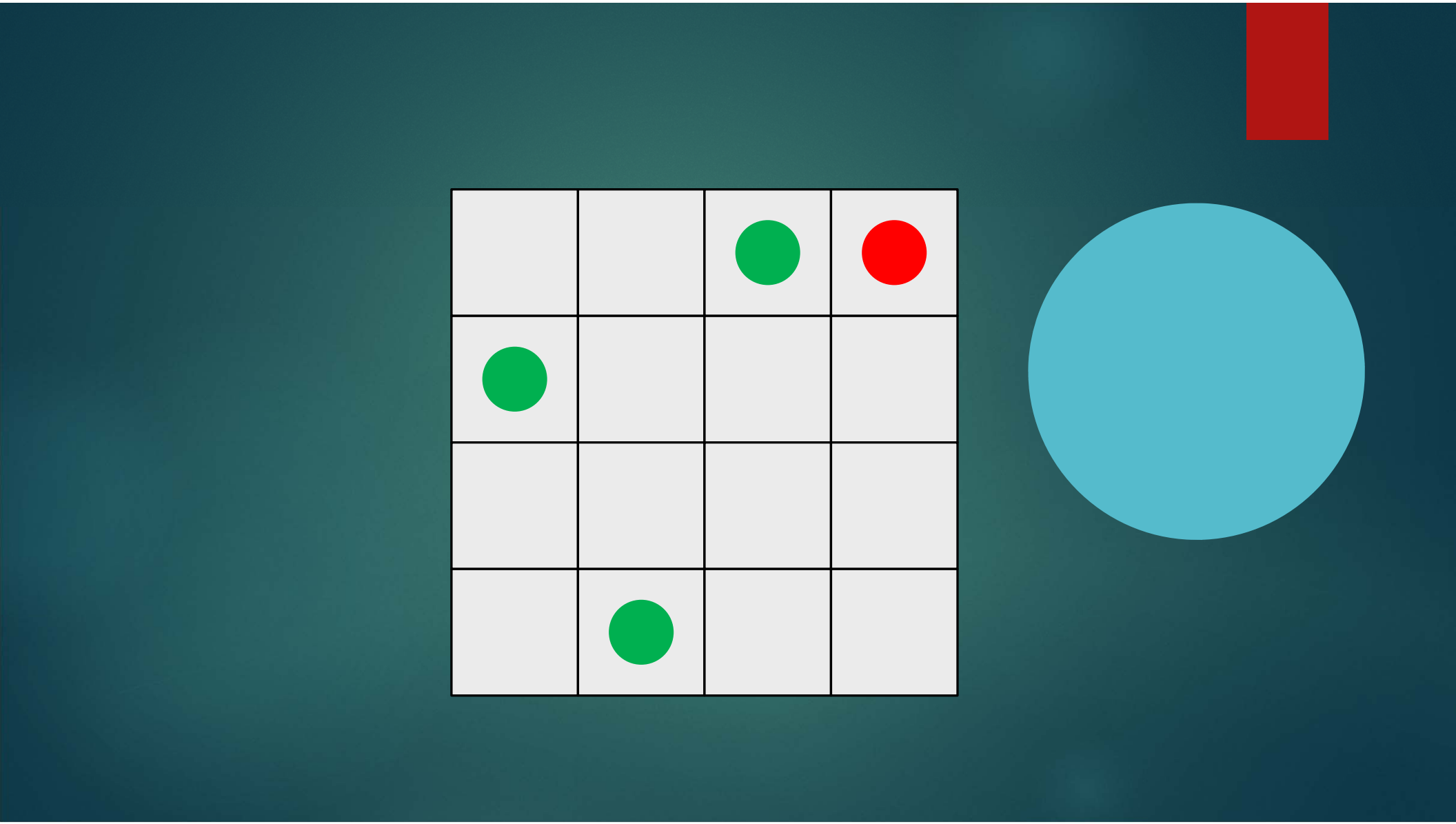


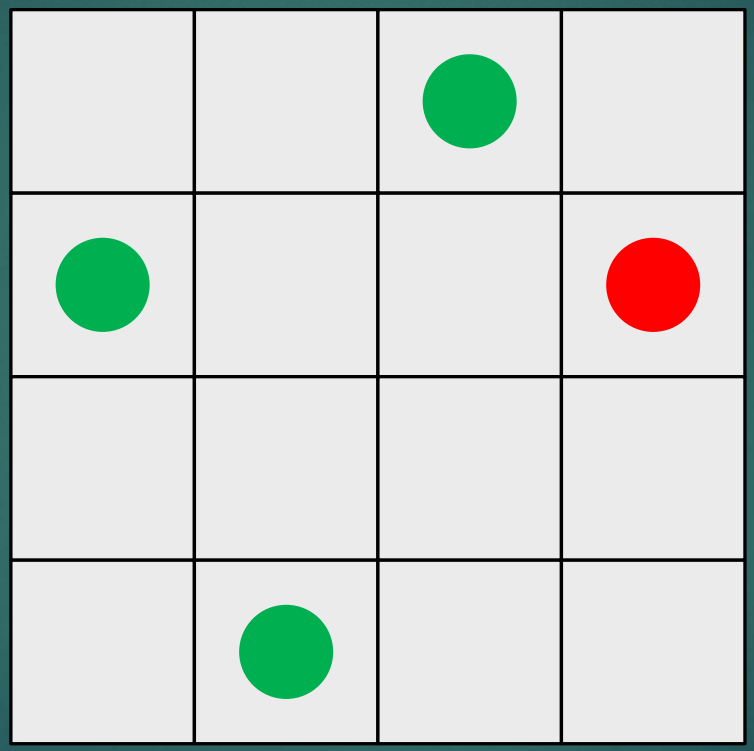


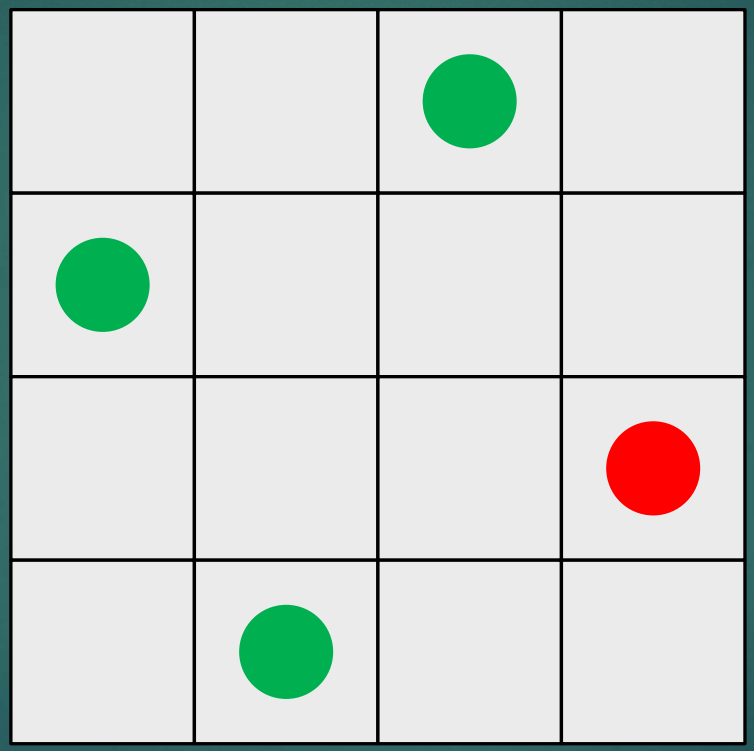


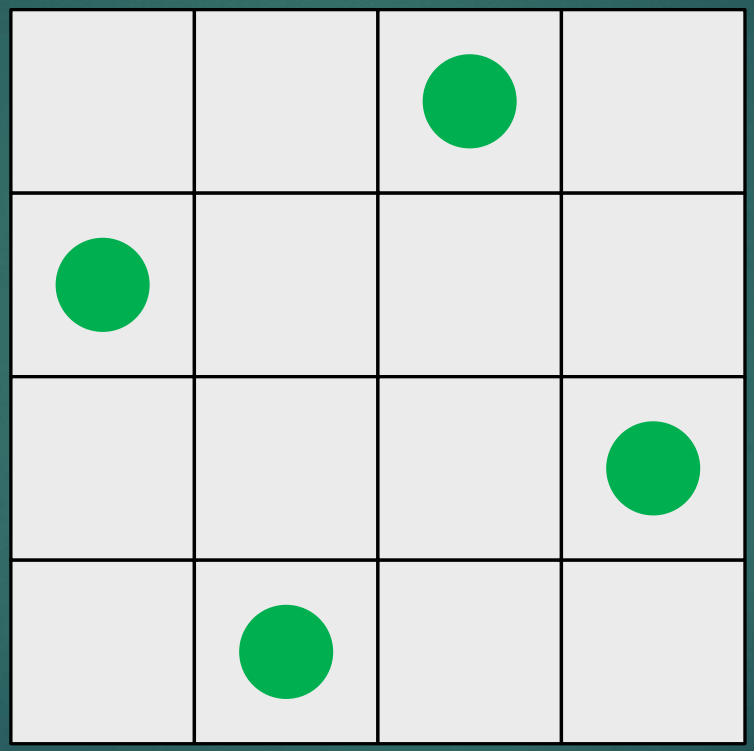




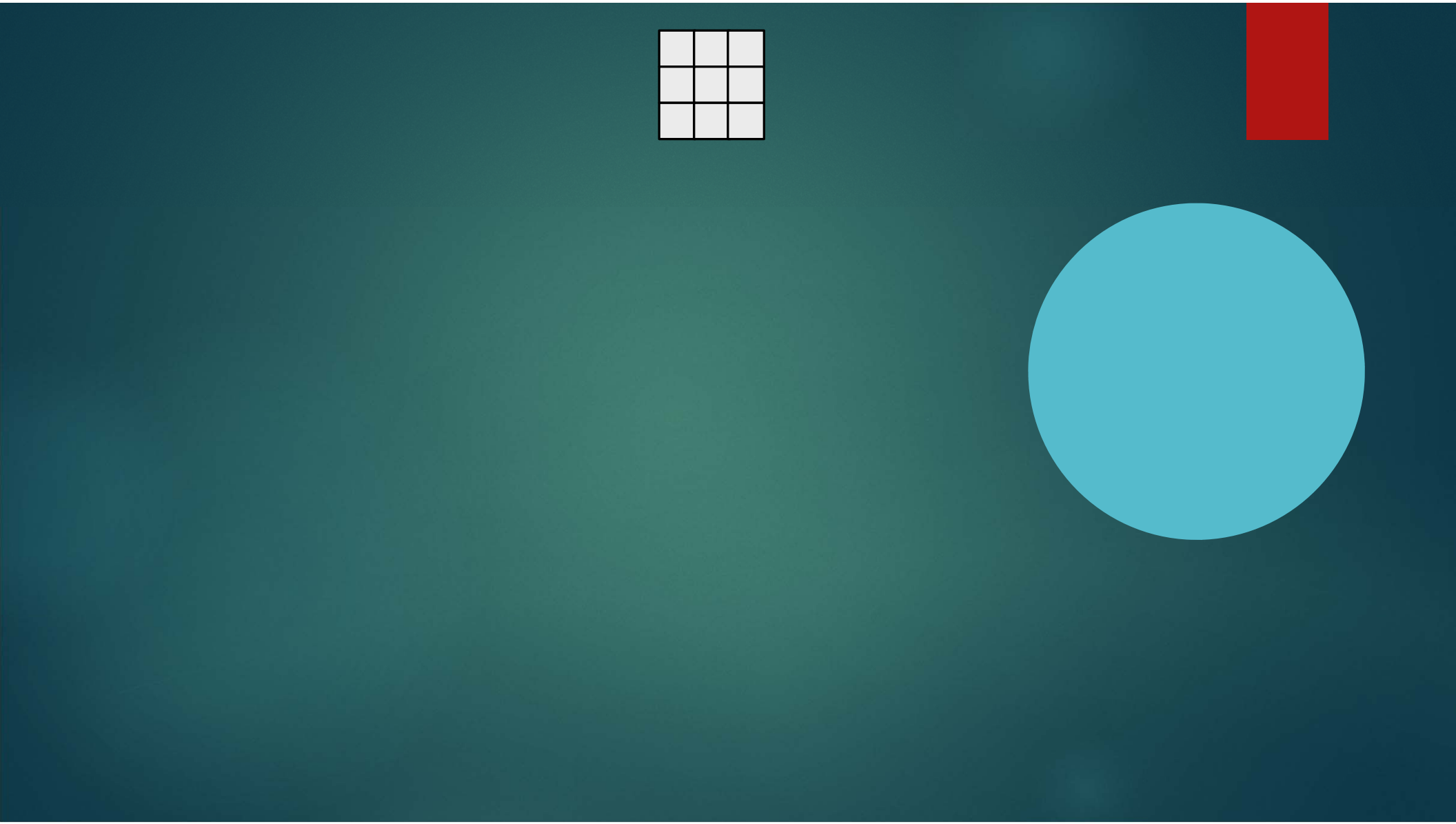


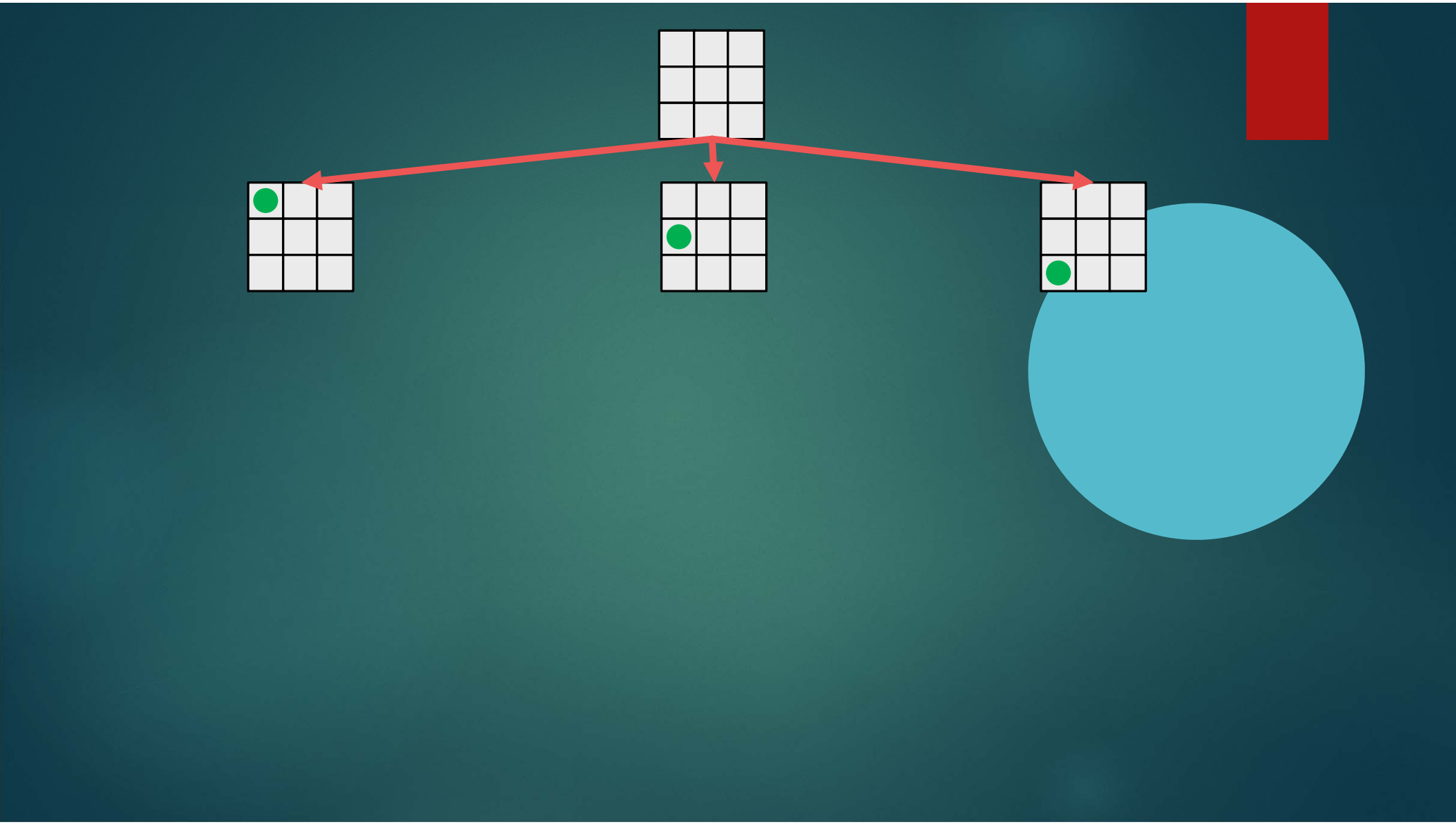


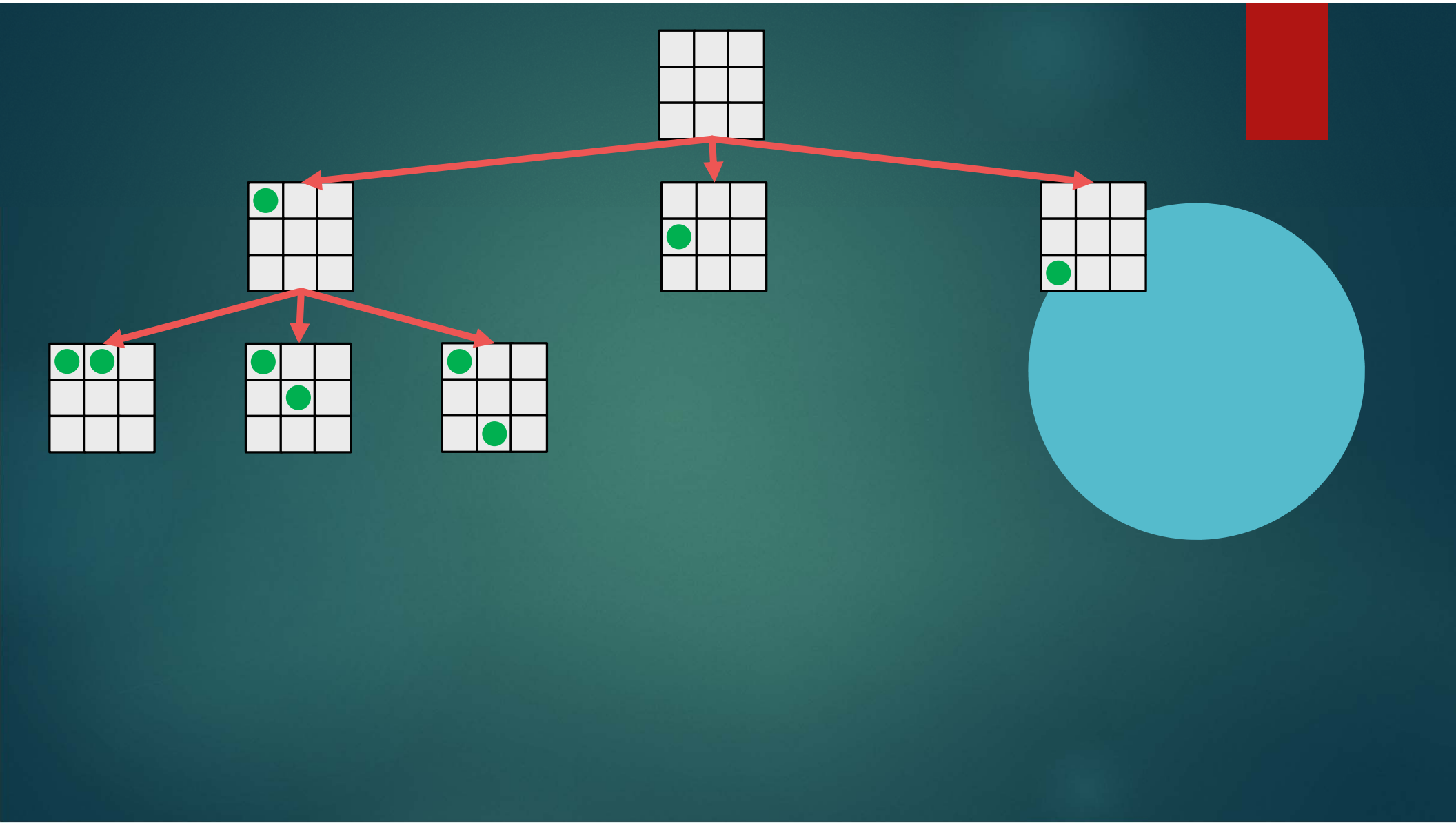


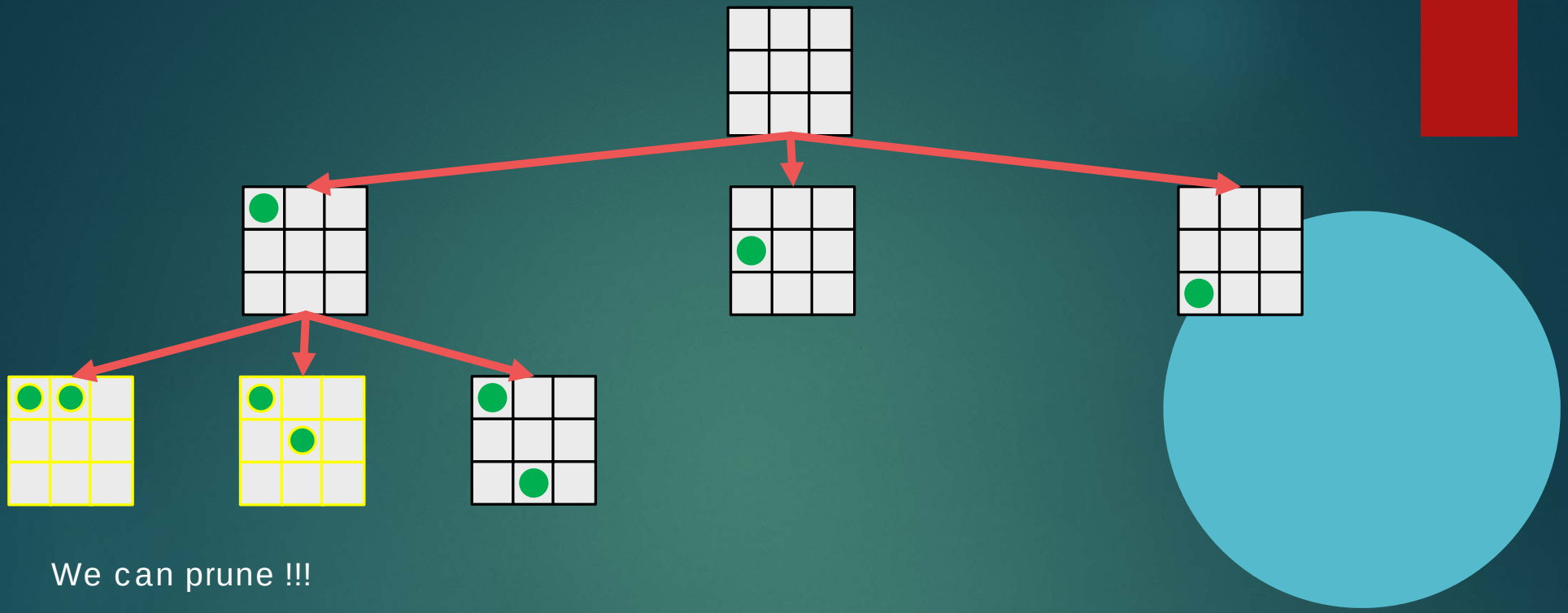


Search tree

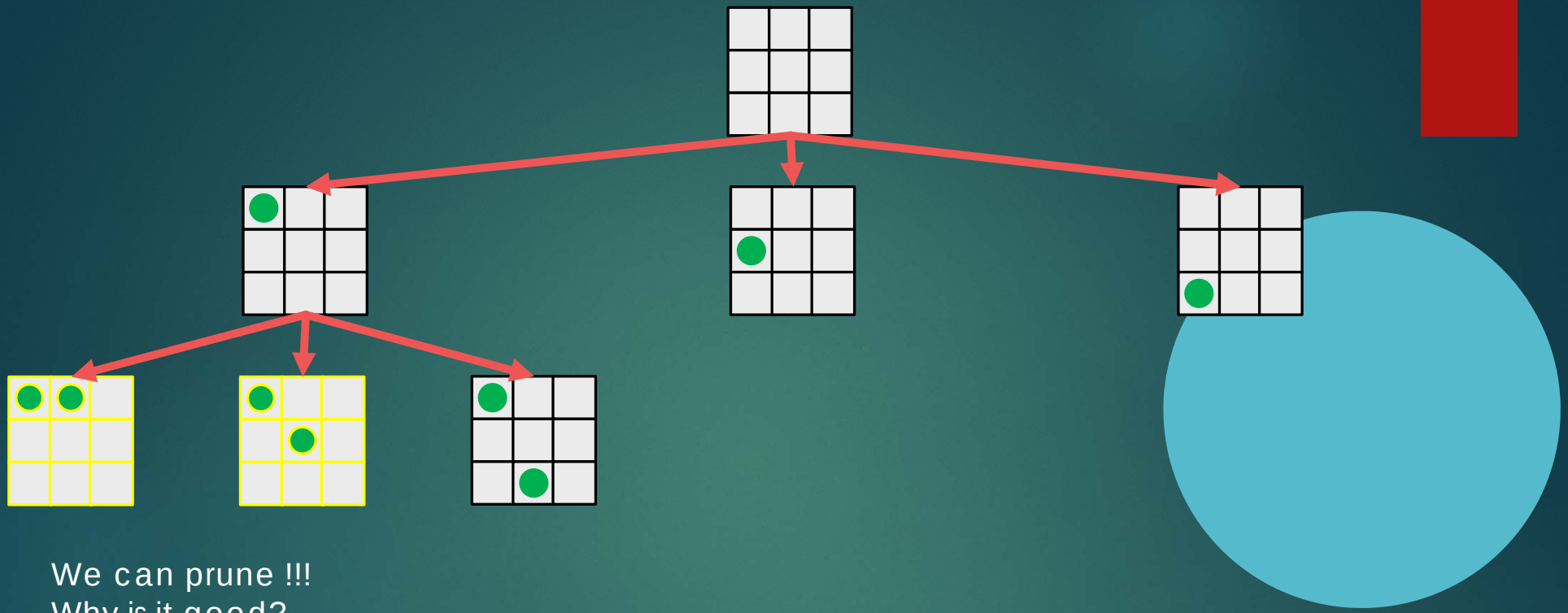




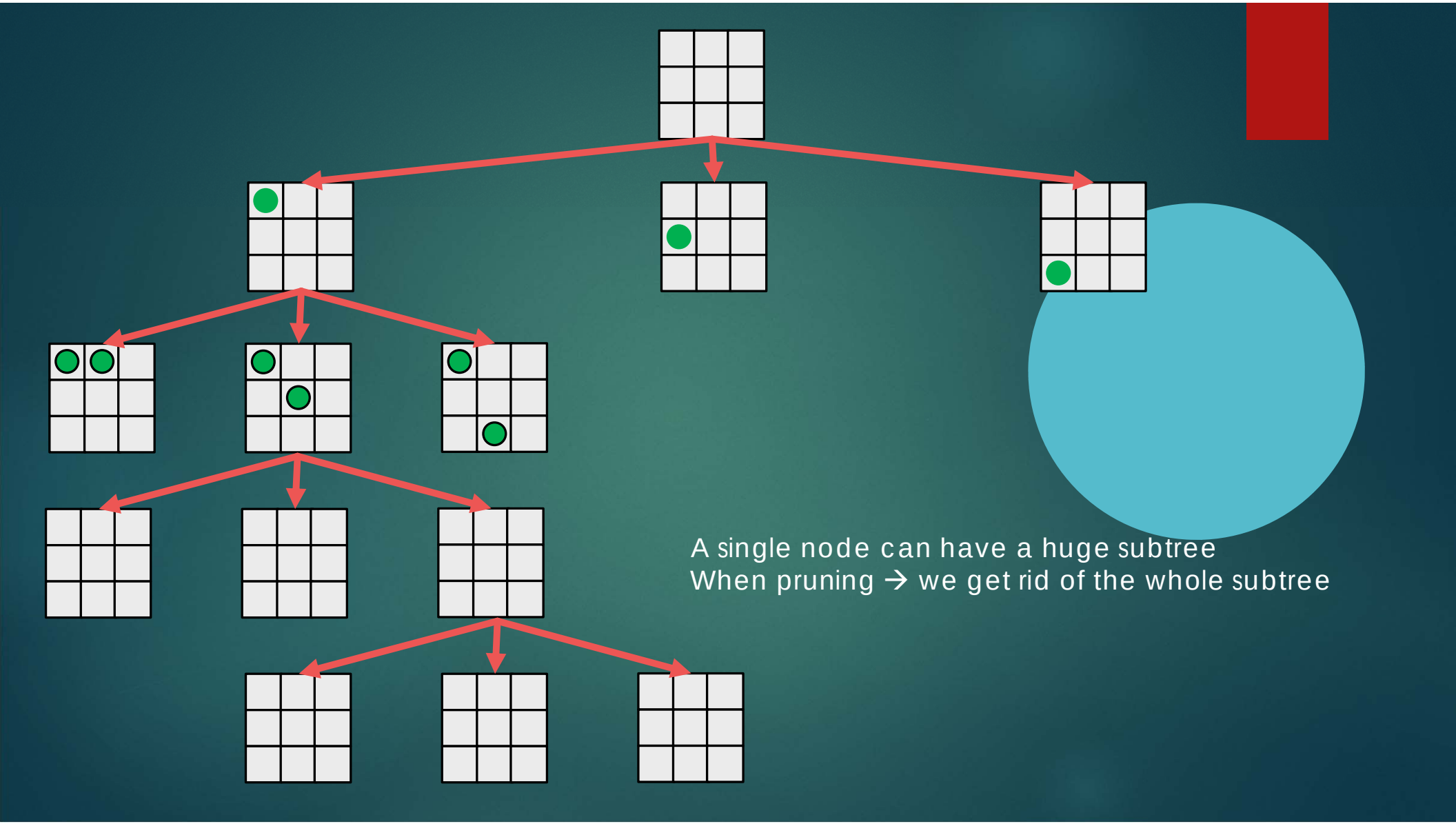


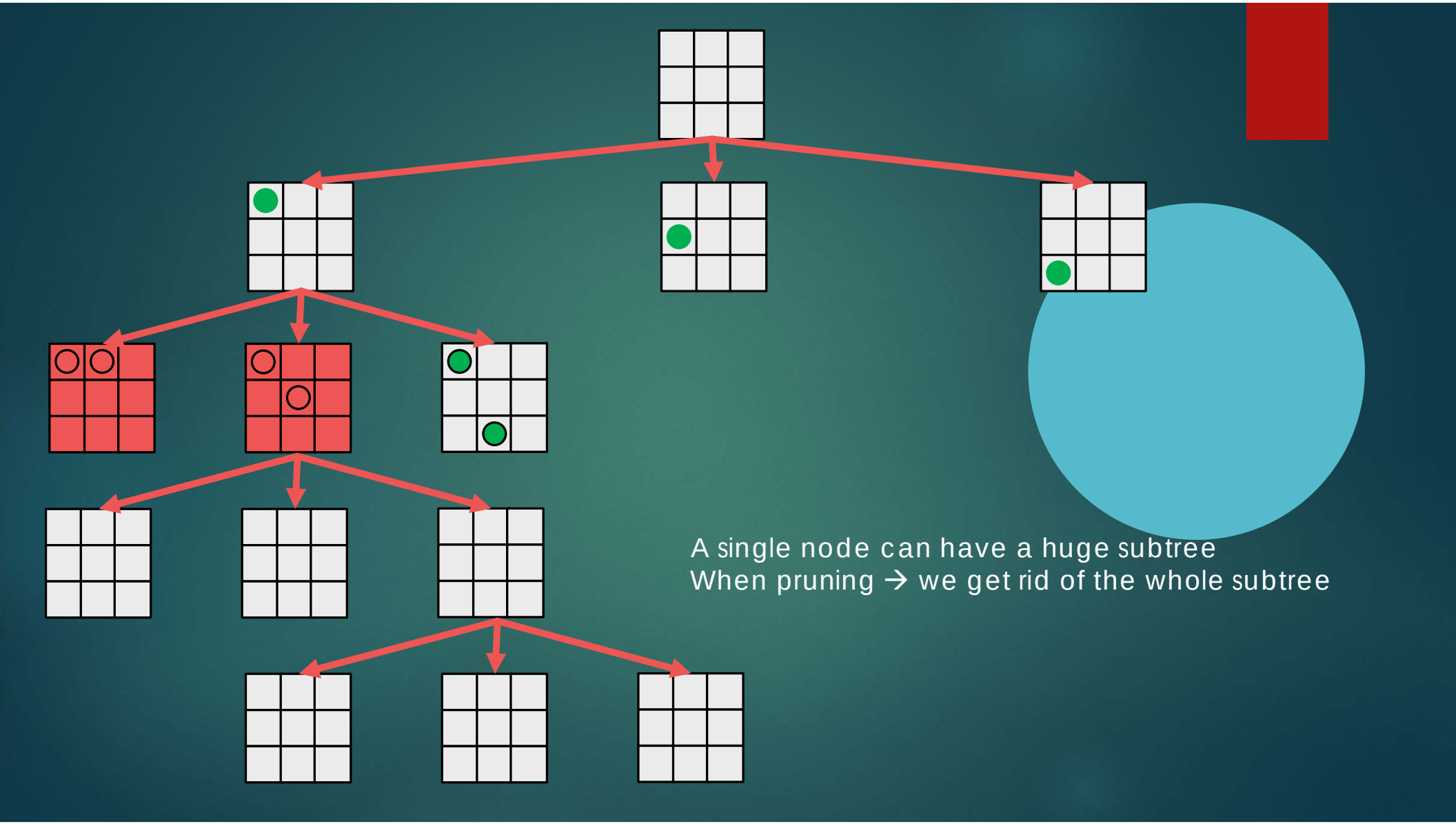


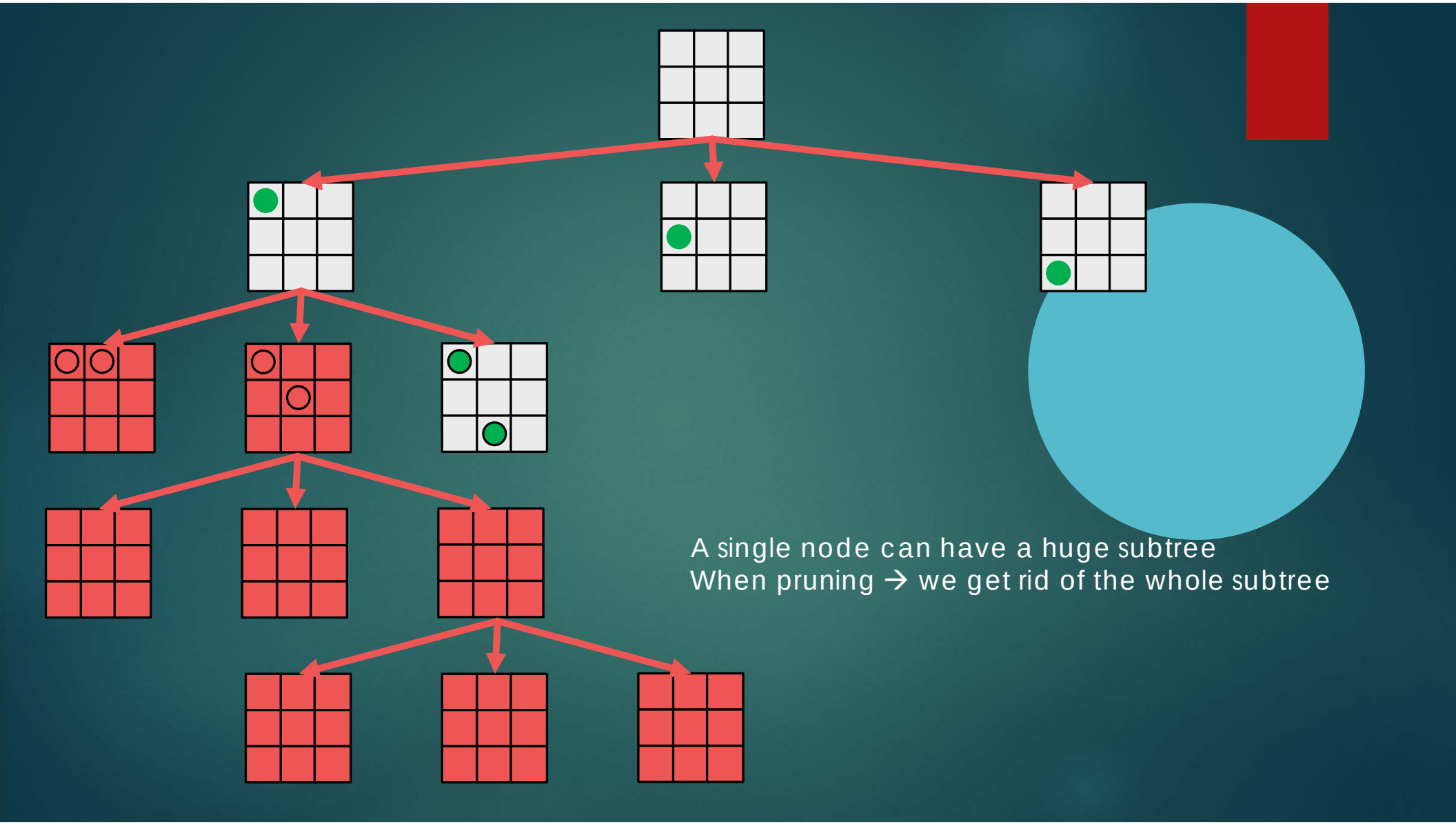
We can prune !!!



We can prune !!!
Why is it good?


A single node can have a huge subtree
When pruning → we get rid of the whole subtree


A single node can have a huge subtree
When pruning → we get rid of the whole subtree


A single node can have a huge subtree
When pruning → we get rid of the whole subtree