COMPACTION GAMES

Hugo Akitaya *University of Massachusetts*

Maarten Löffler *Utrecht University*

Anika Rounds *Amazon*

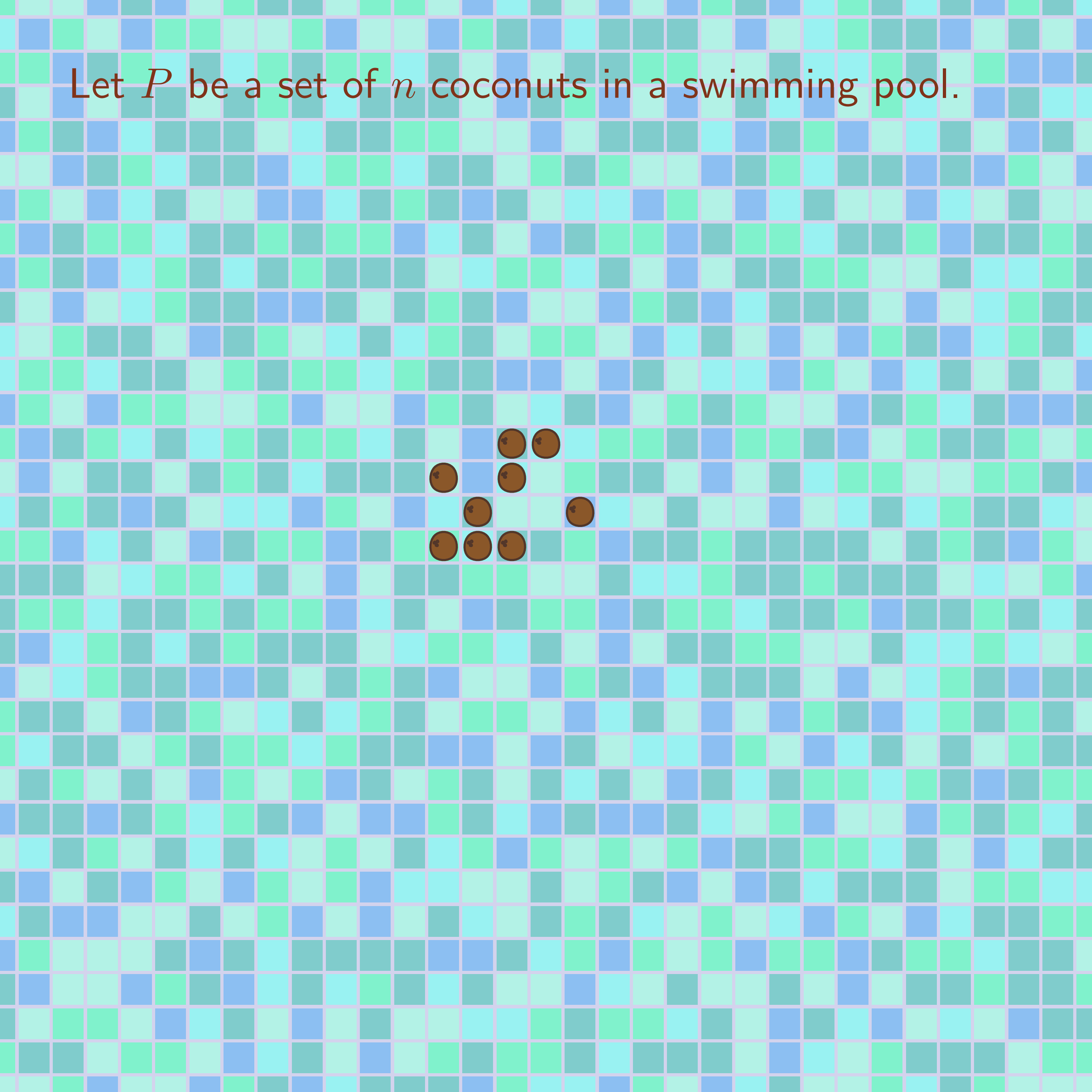
Giovanni Viglietta *JAIST*

Trash Cubes designed by Mary Bogdan, 2008. Brooklyn Art Project.

GRID COMPACTION

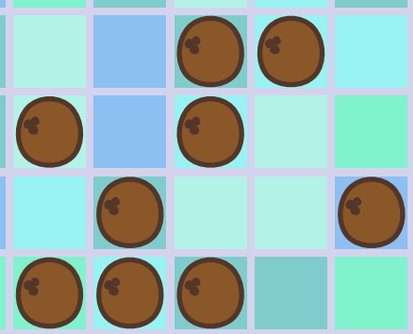
Let P be a set of n coconuts in a swimming pool.

Let P be a set of n coconuts in a swimming pool.



Let P be a set of n coconuts in a swimming pool.

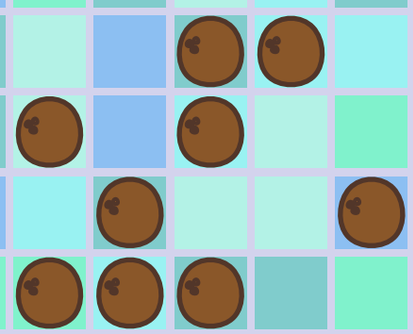
We assume the coconuts are aligned to a grid.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

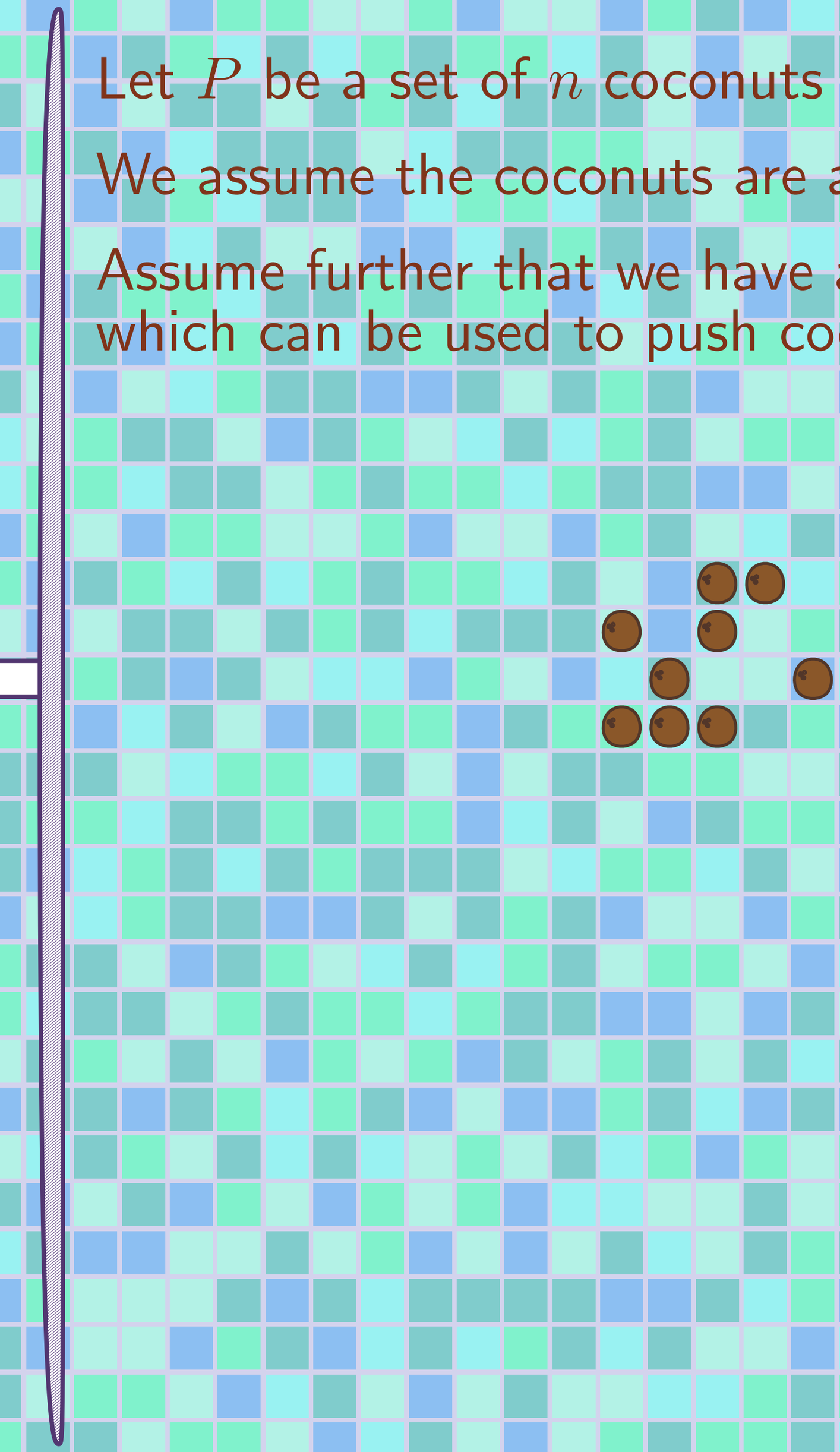
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

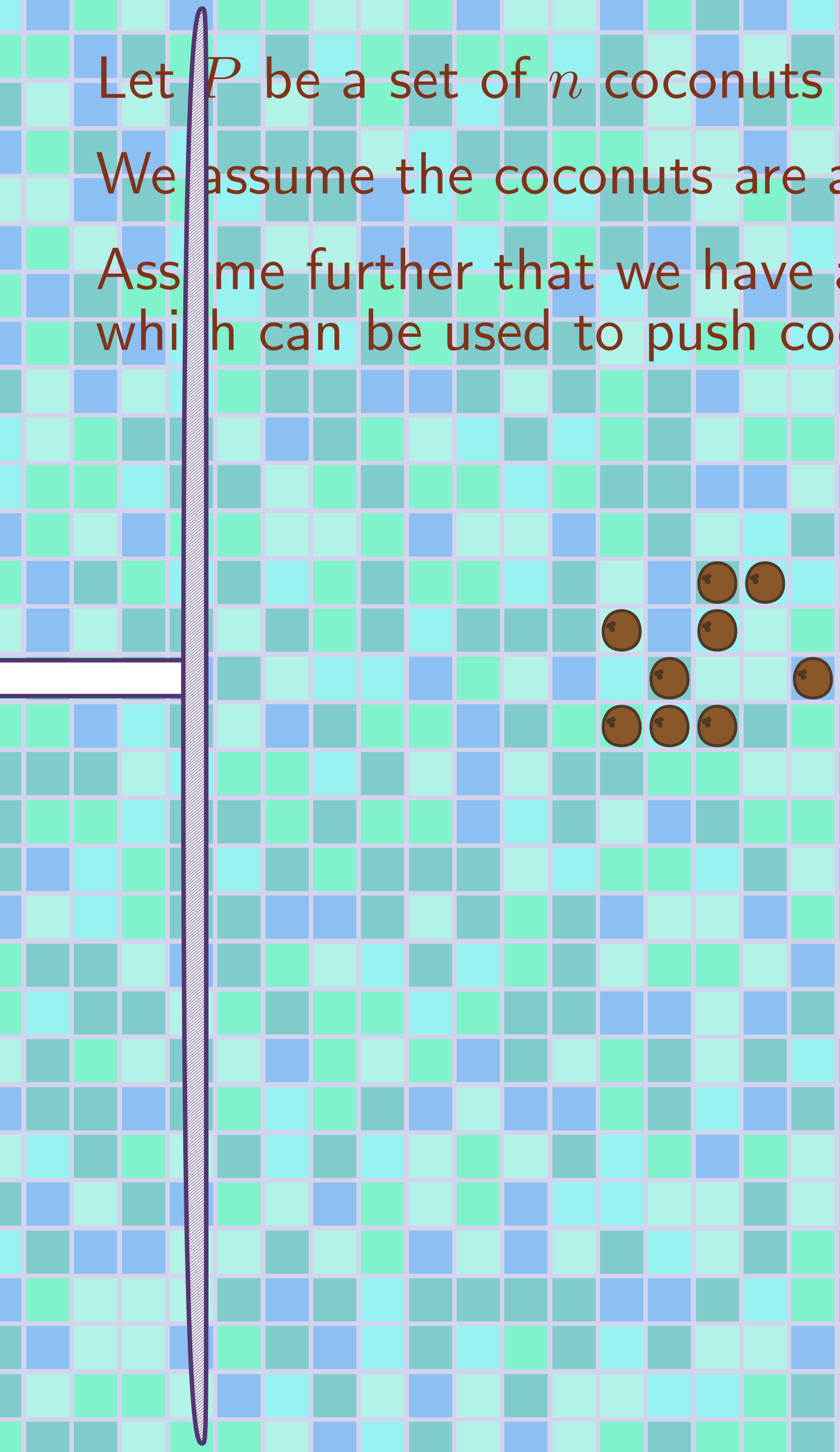
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

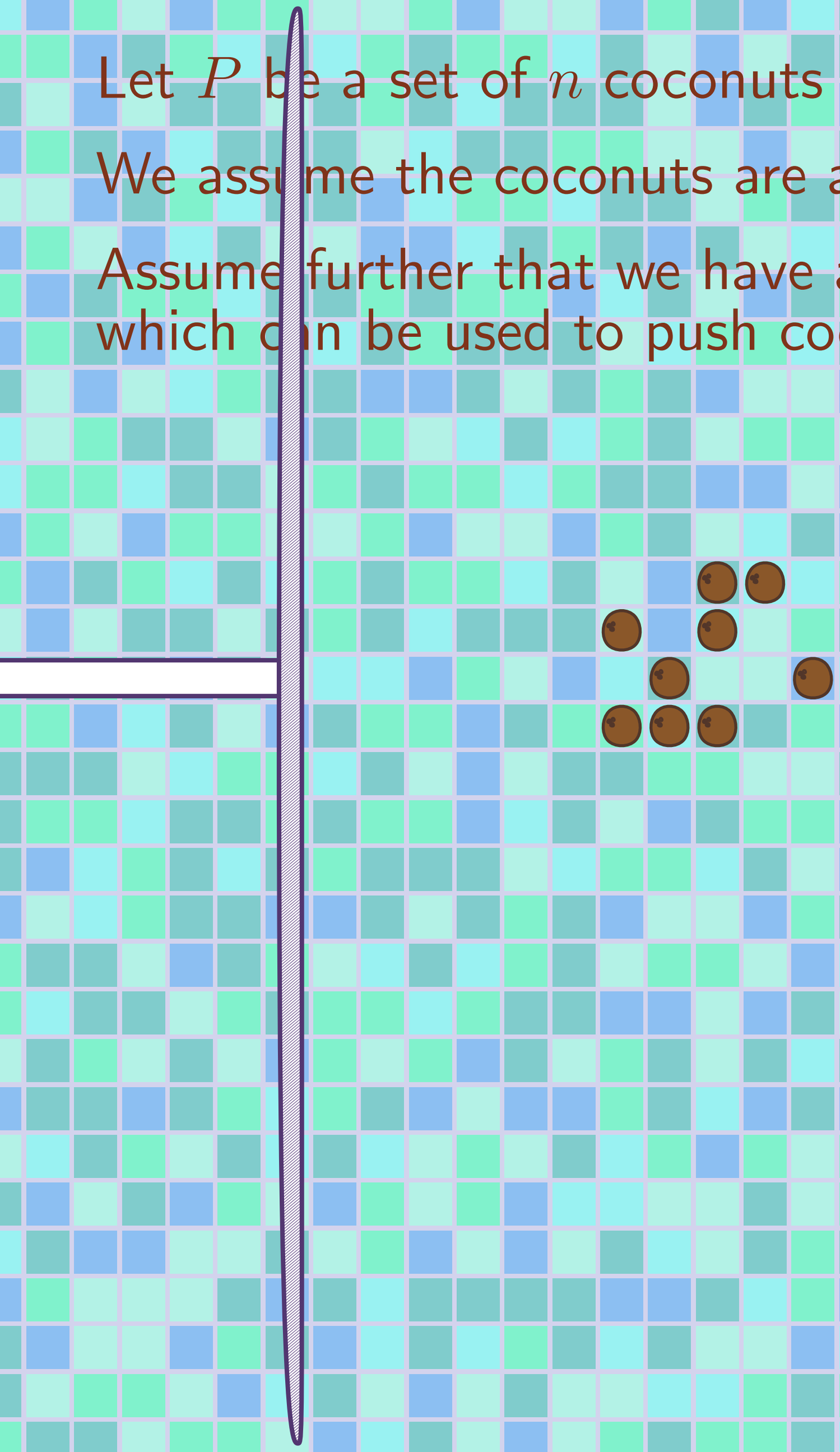
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

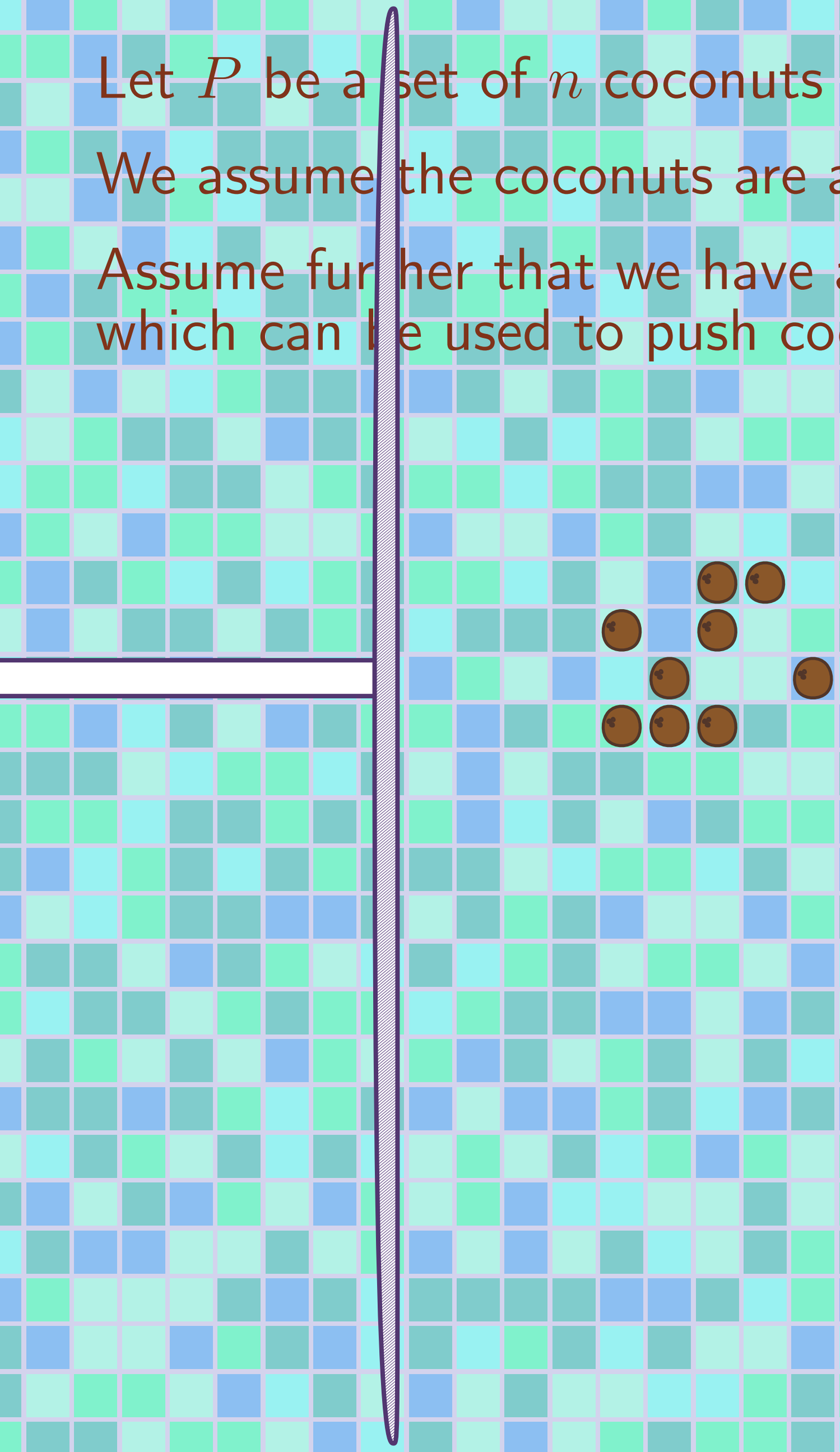
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

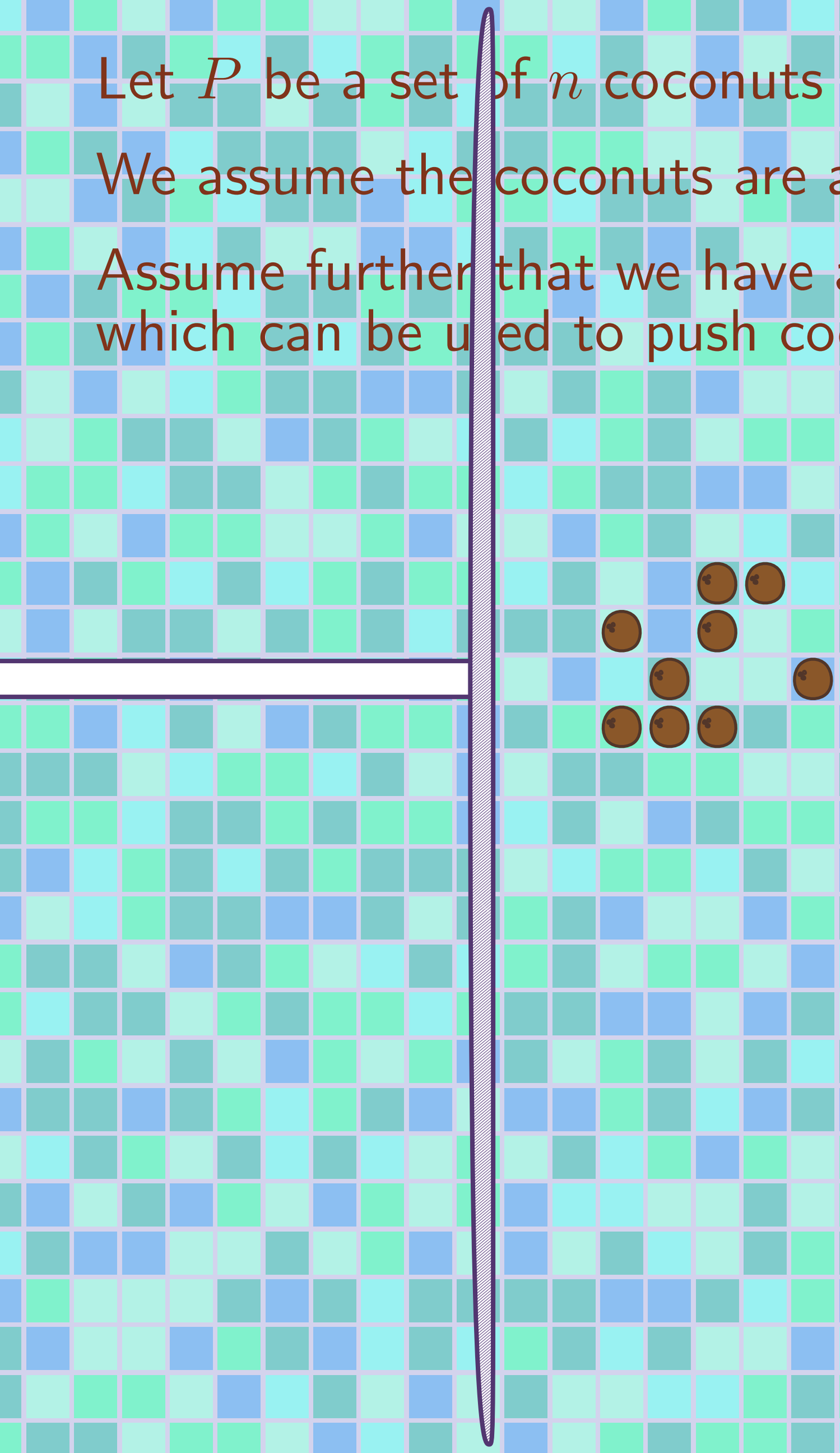
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

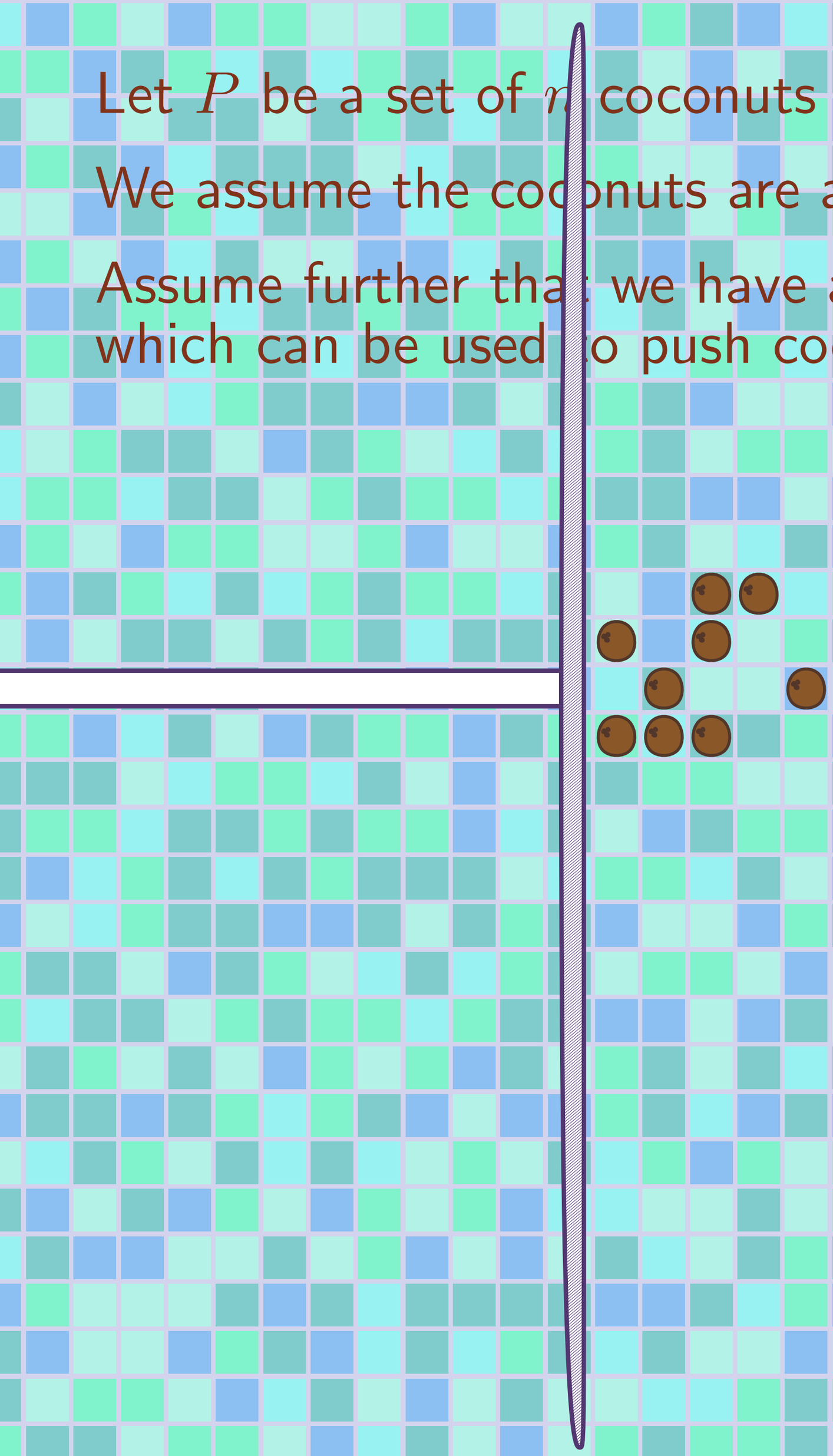
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

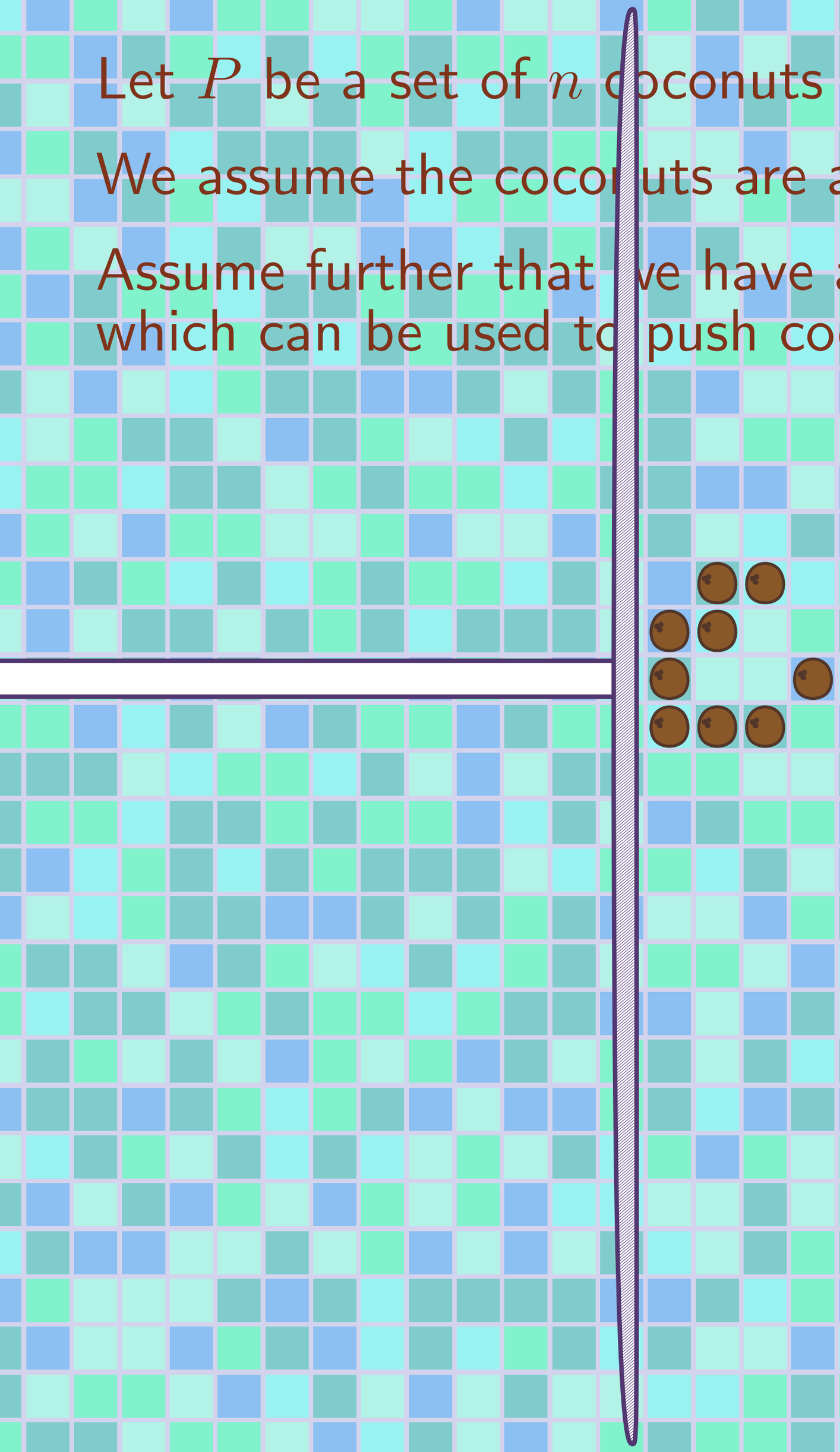
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

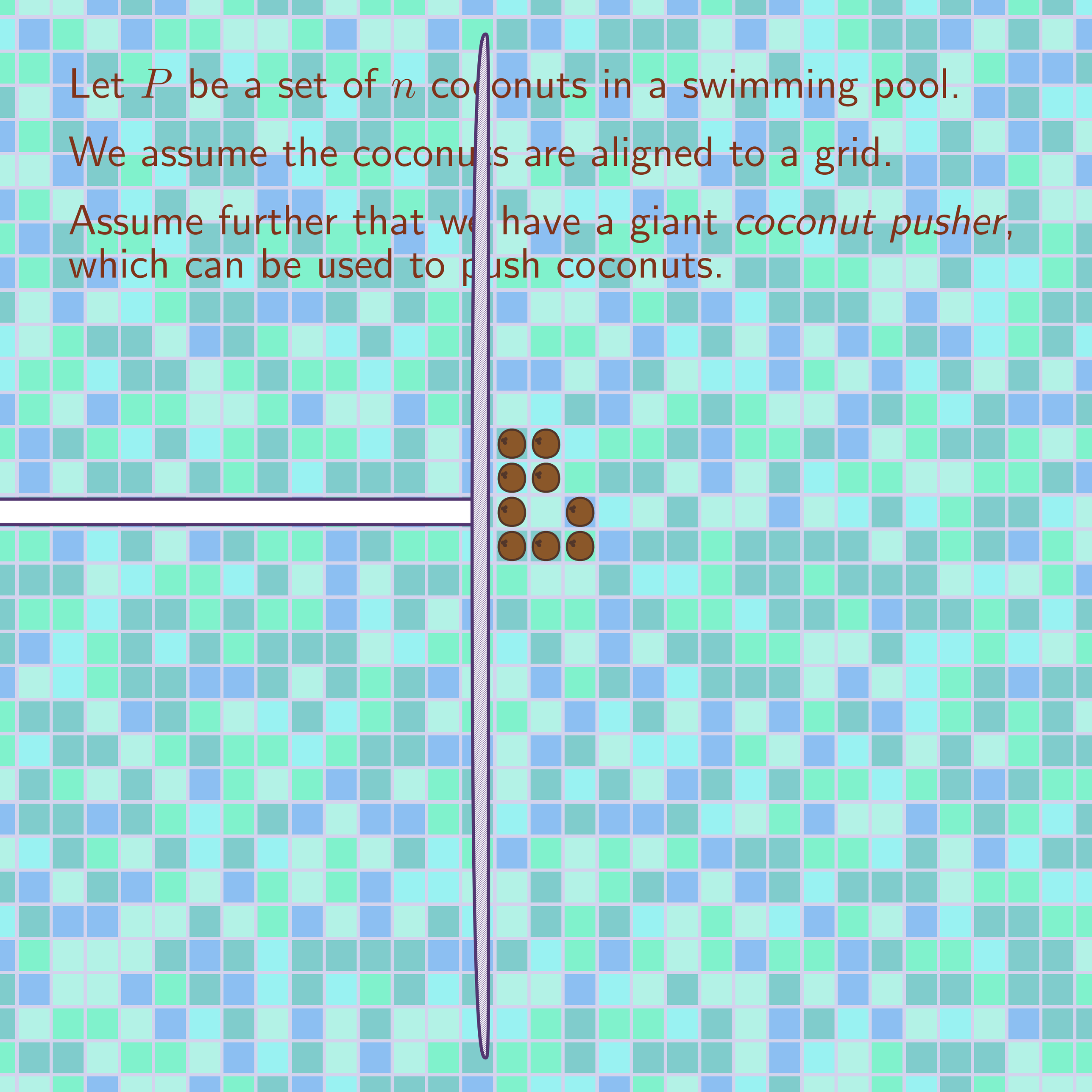
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

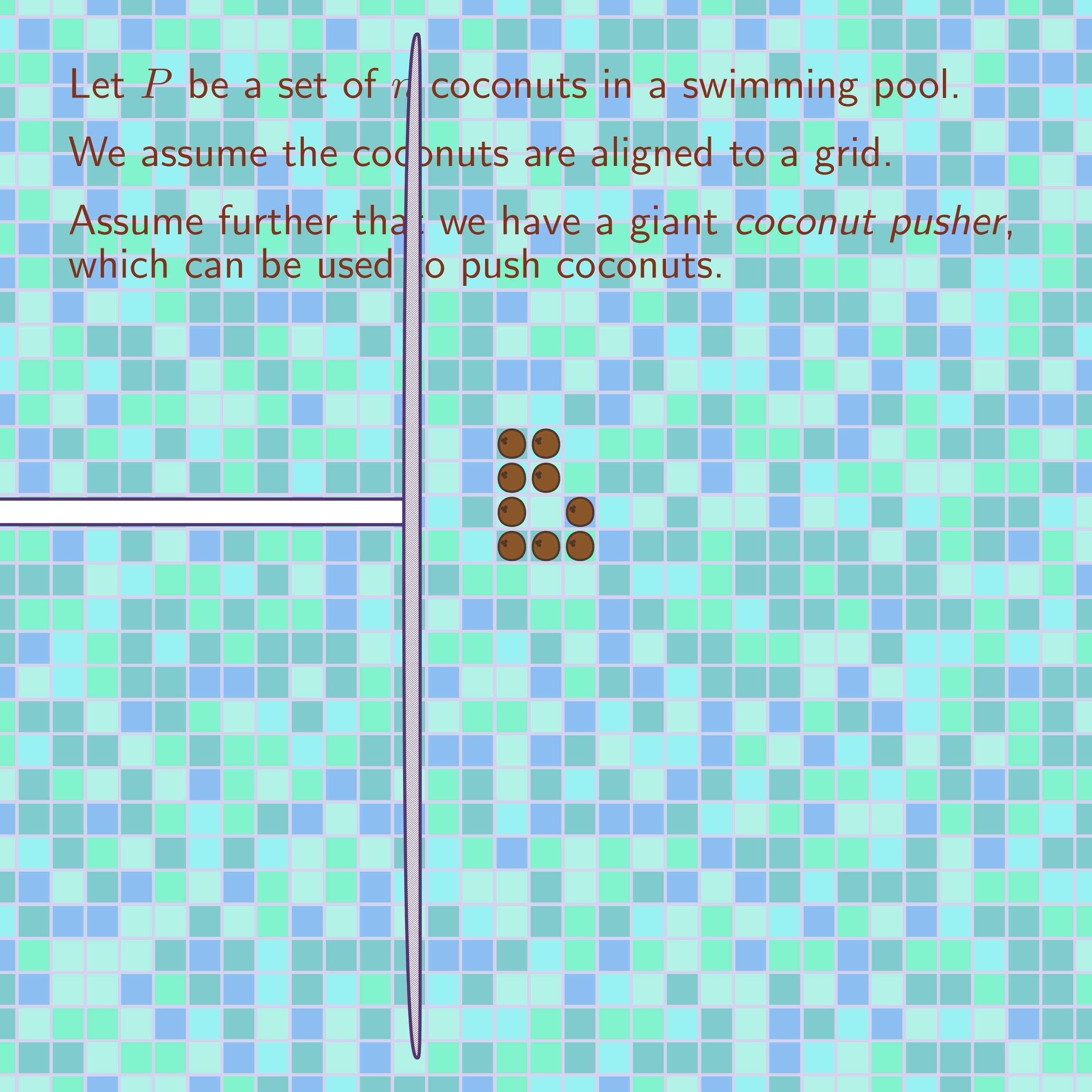
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

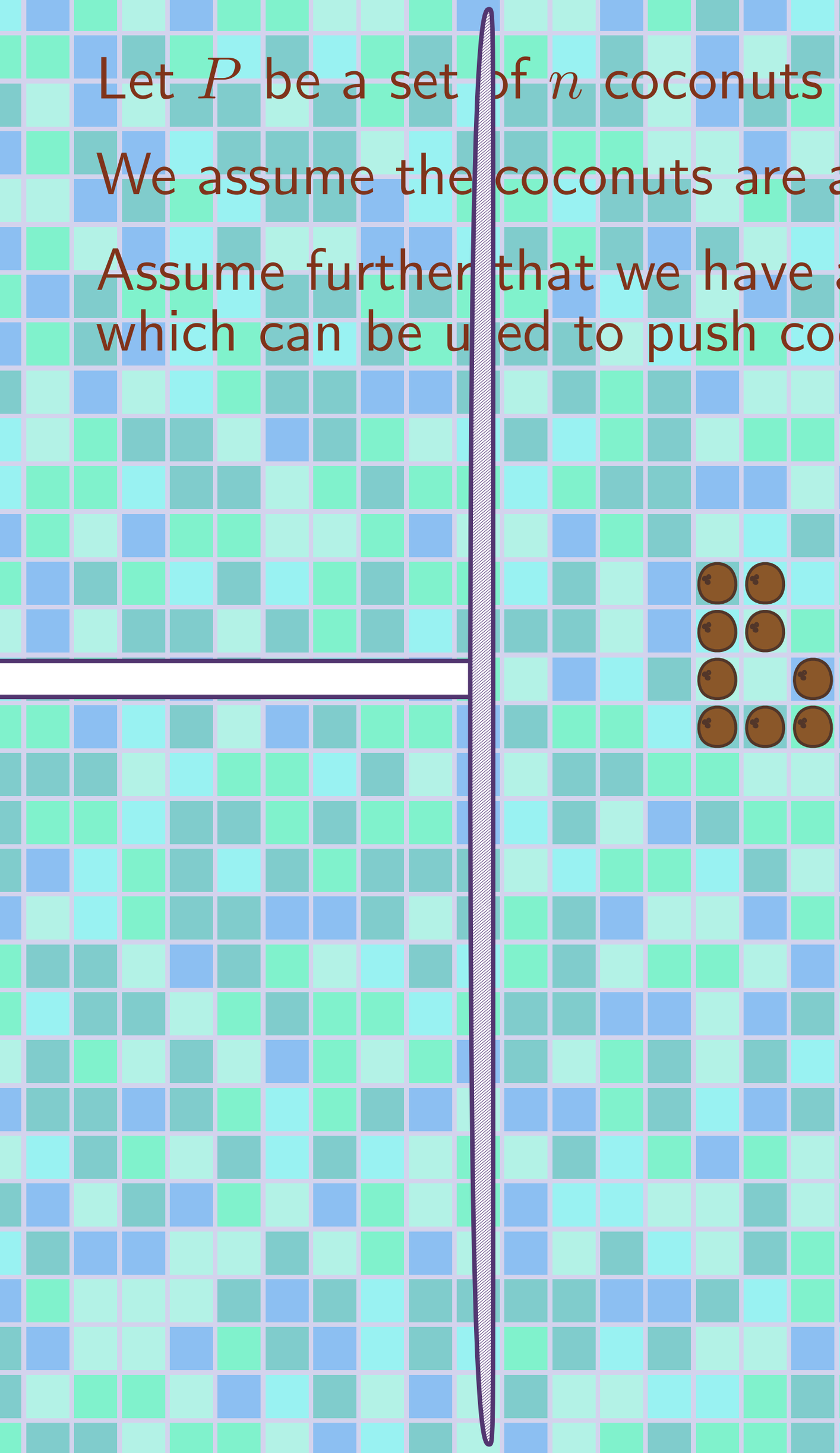
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

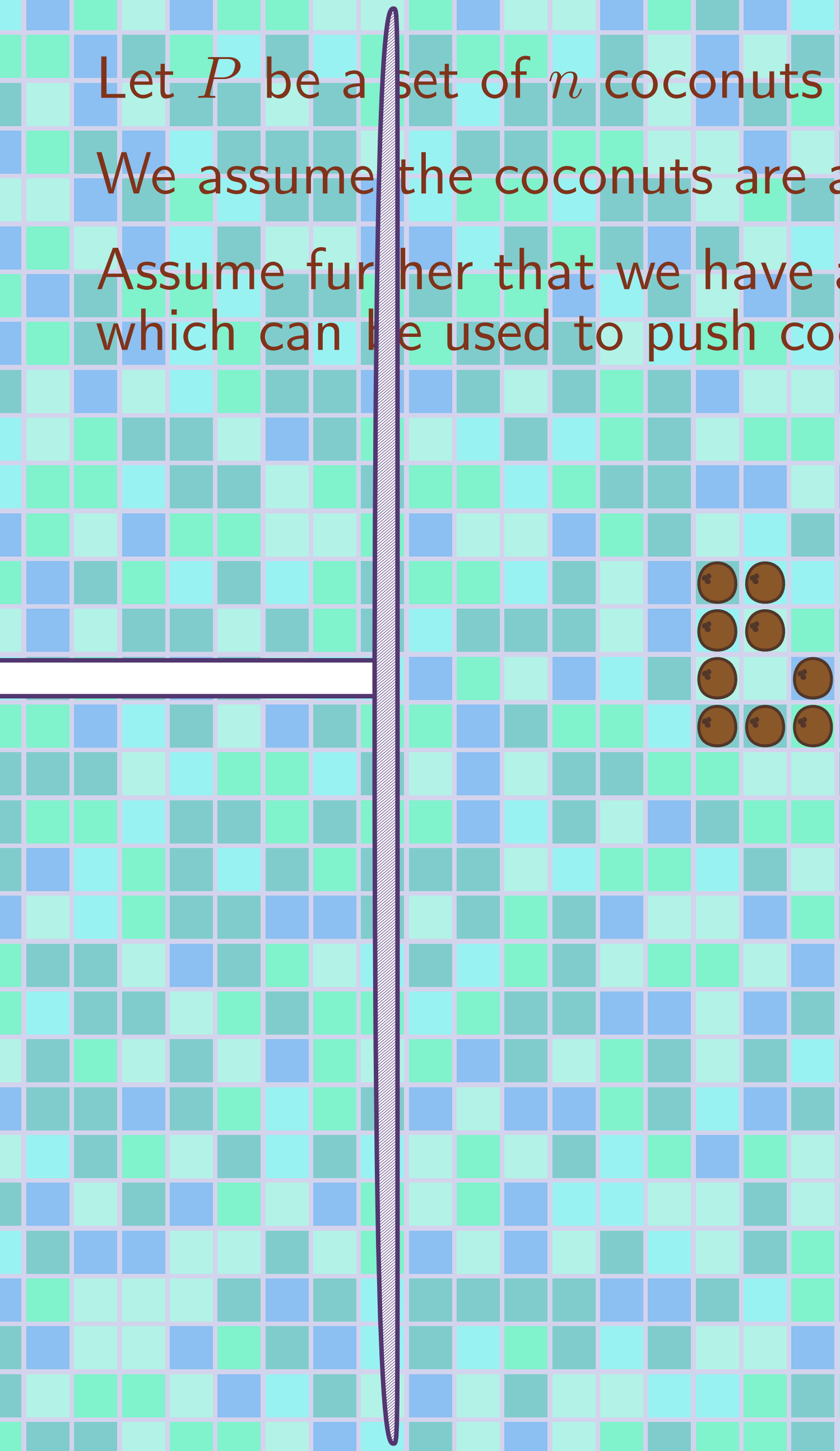
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

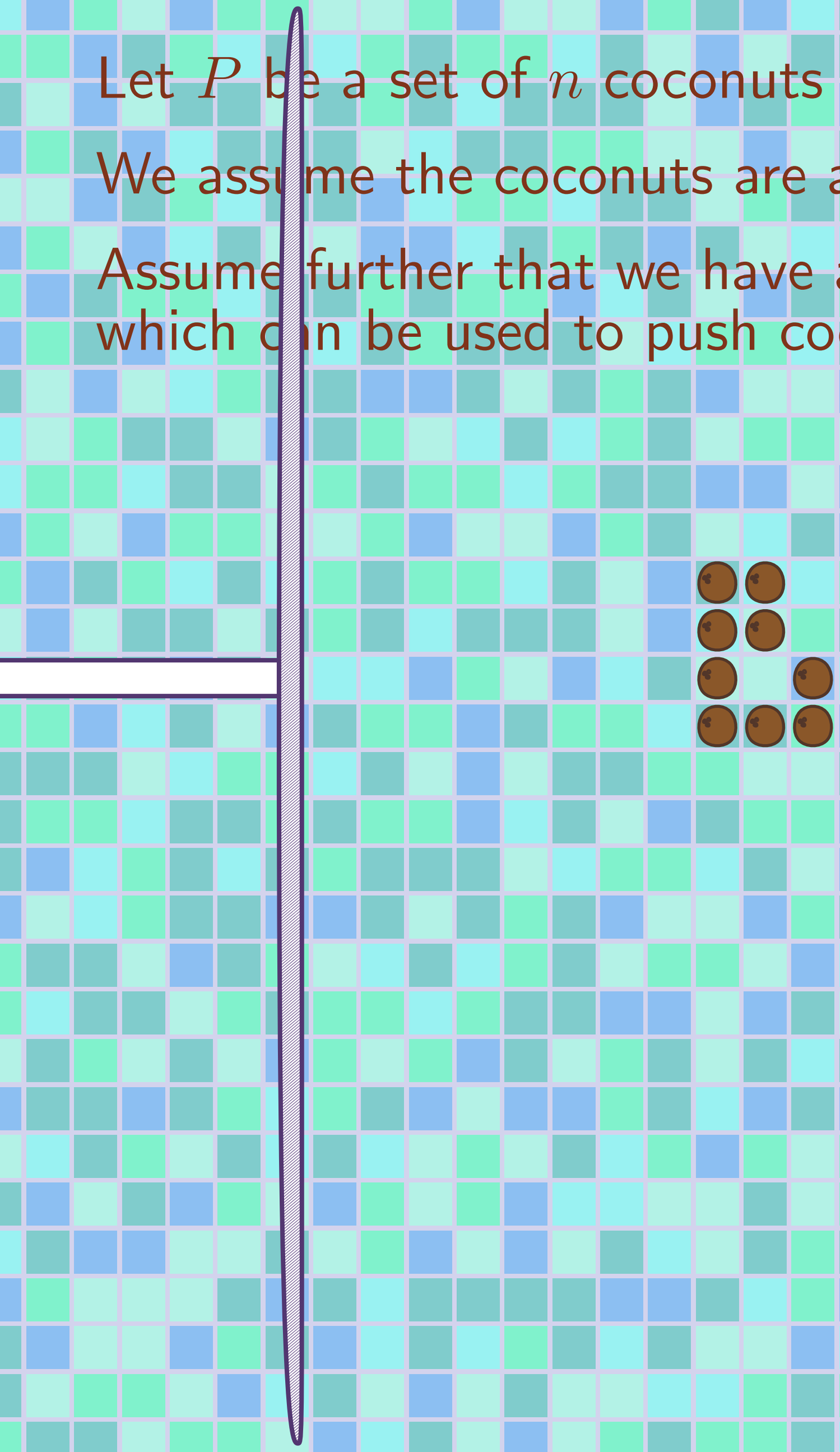
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

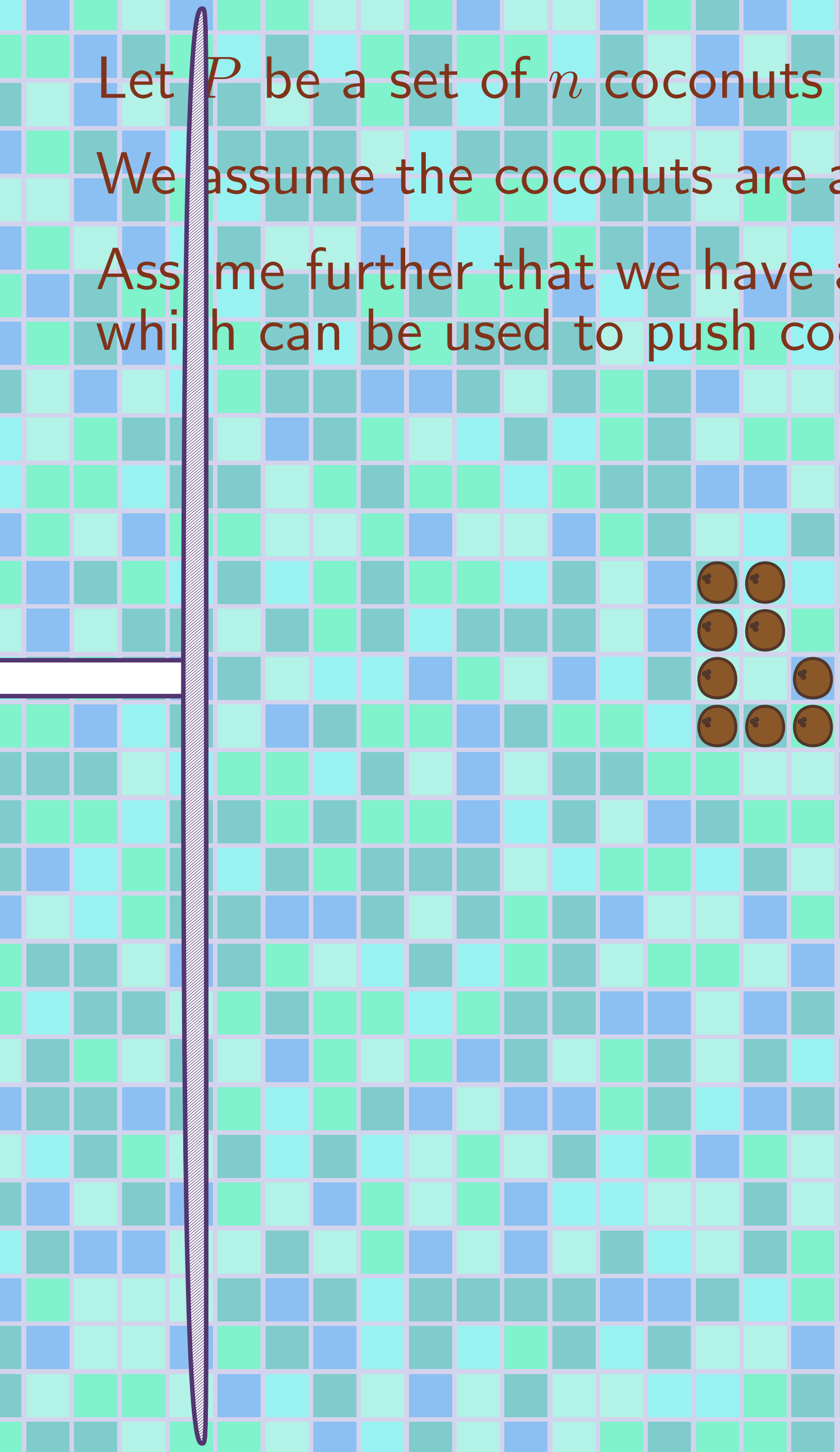
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

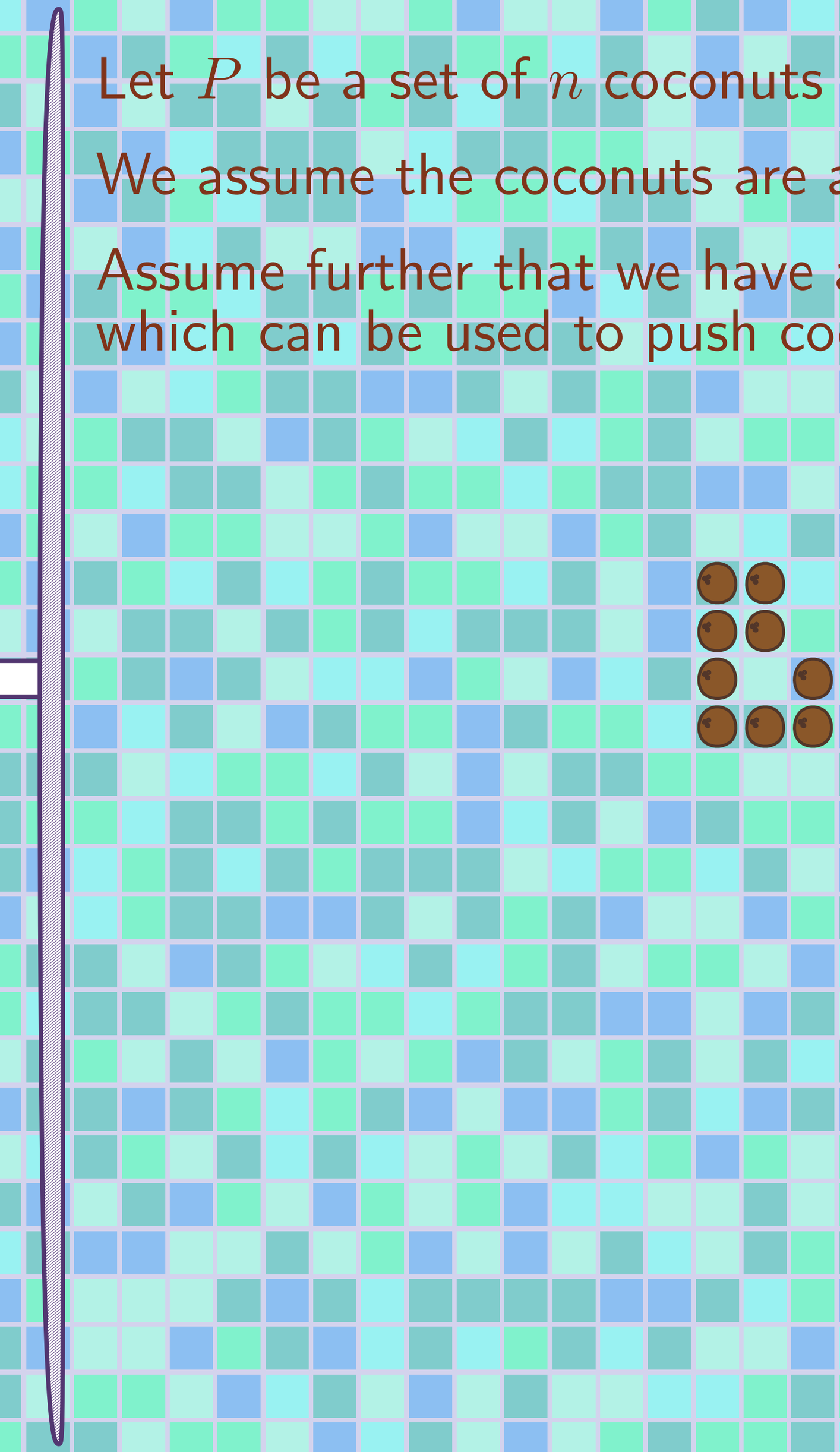
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts

Let P be a set of n coconuts in a swimming pool.

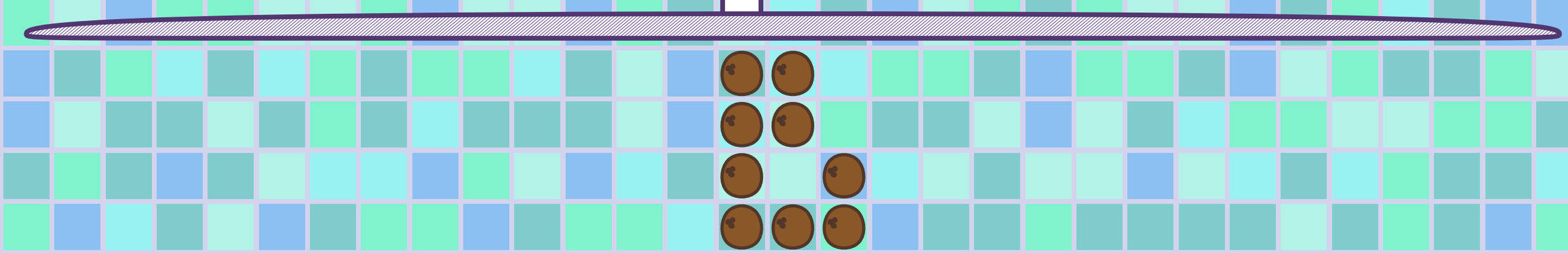
We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

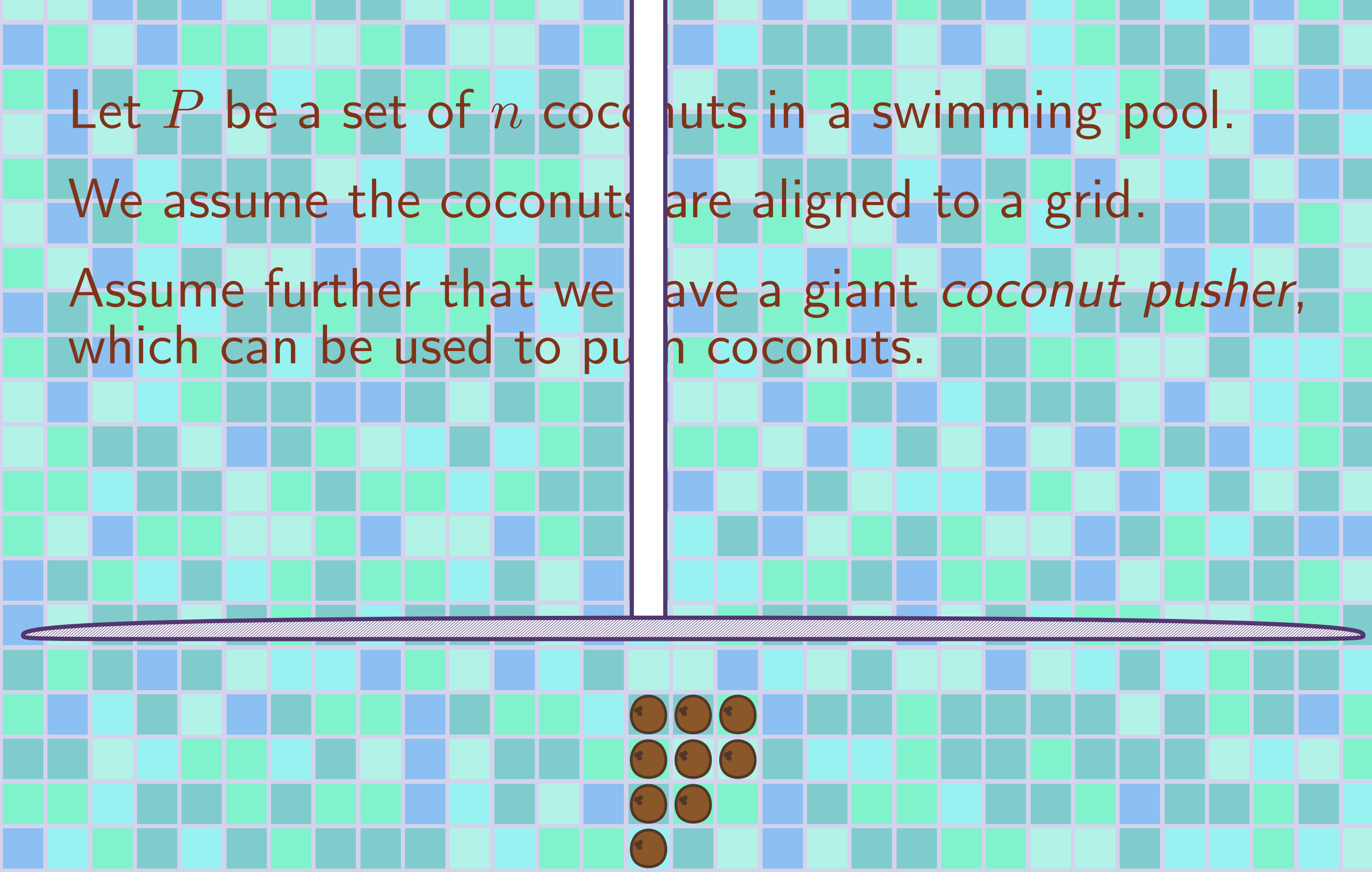
We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

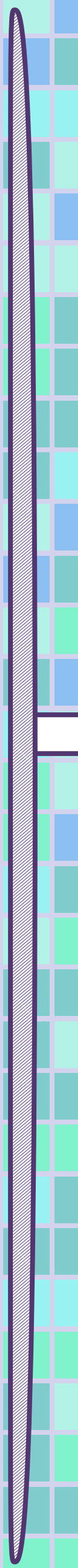
We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts

Let P be a set of n coconuts in a swimming pool.

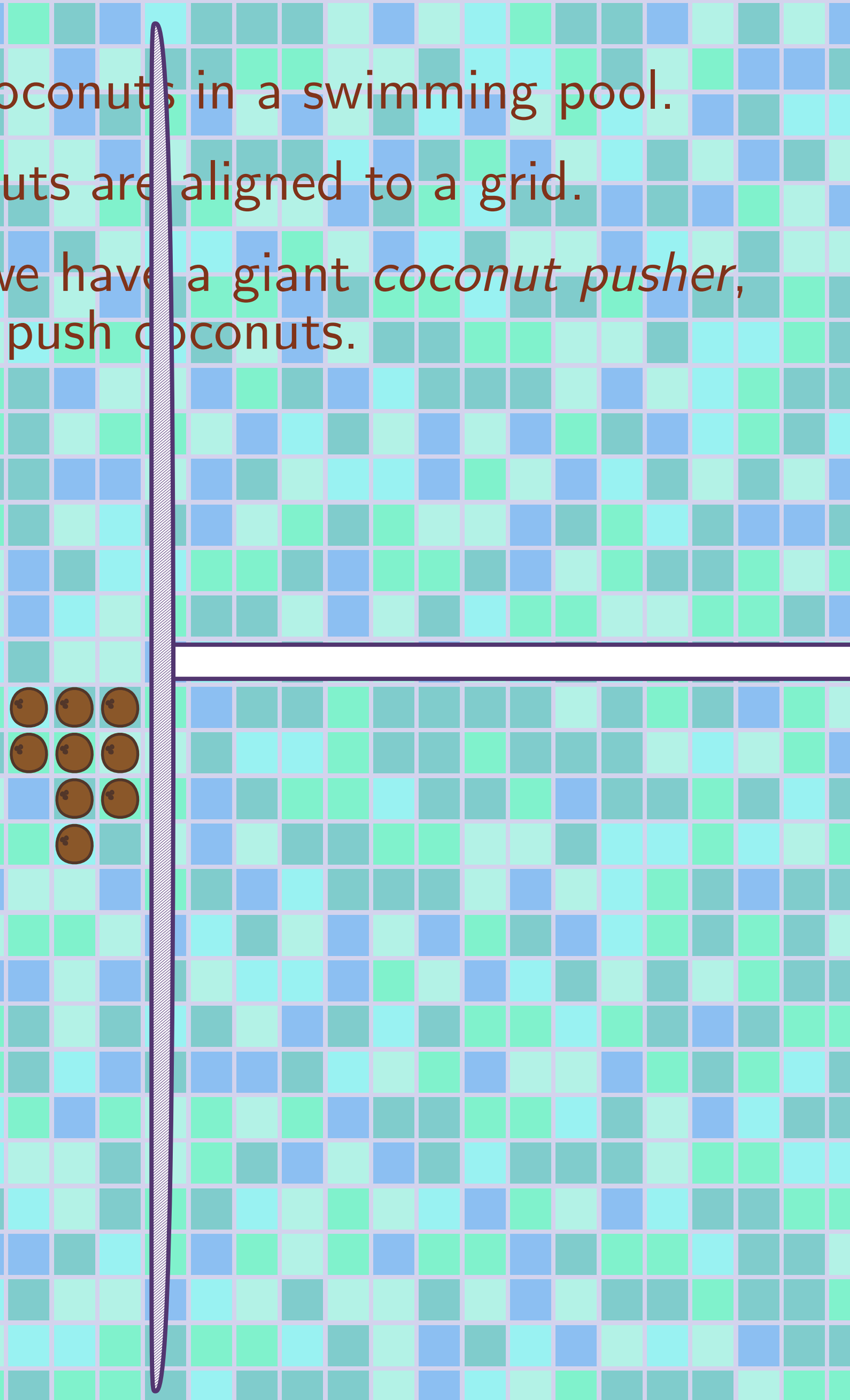
We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

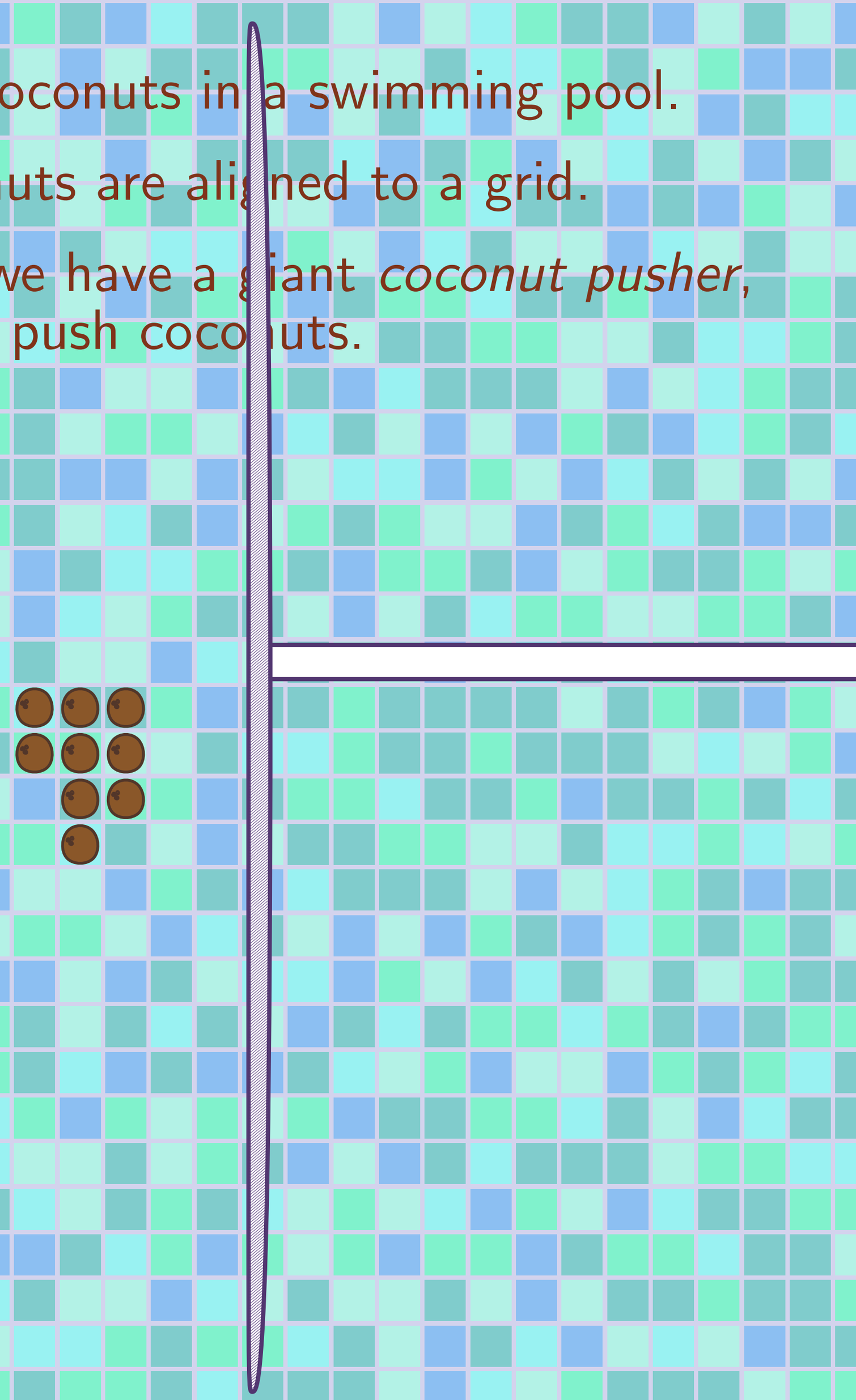
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

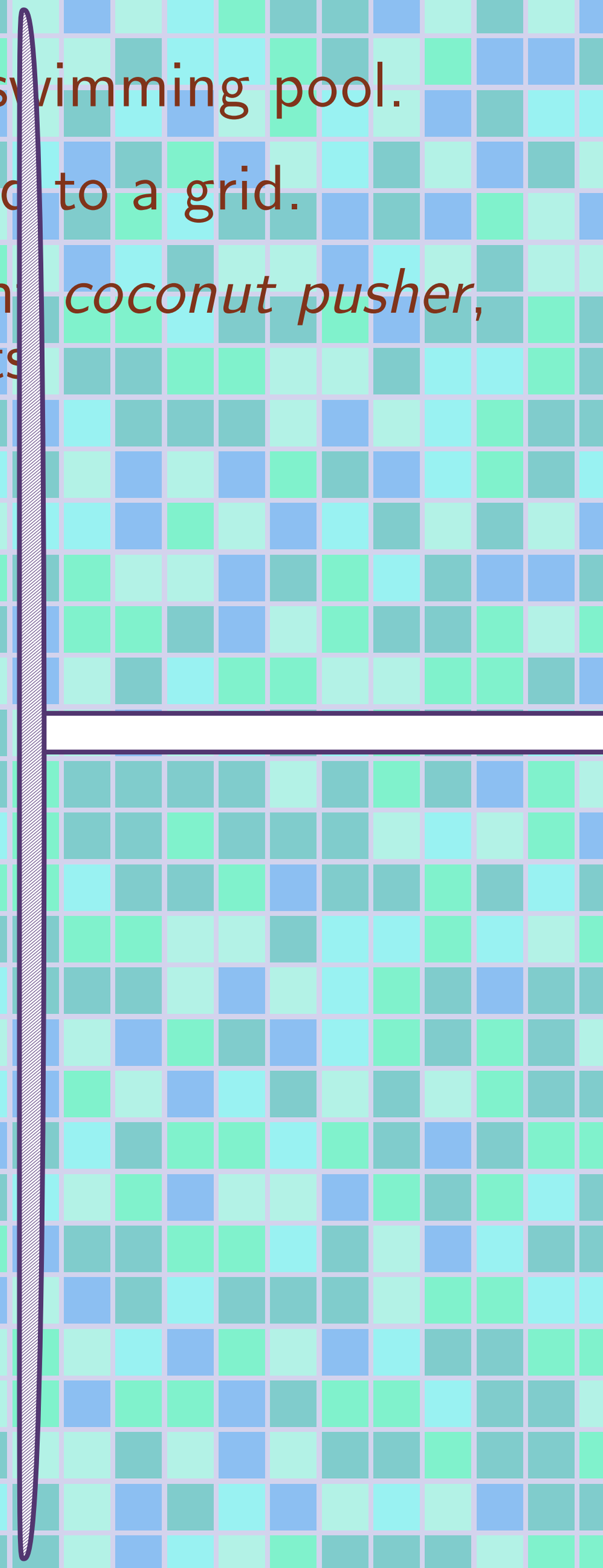
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

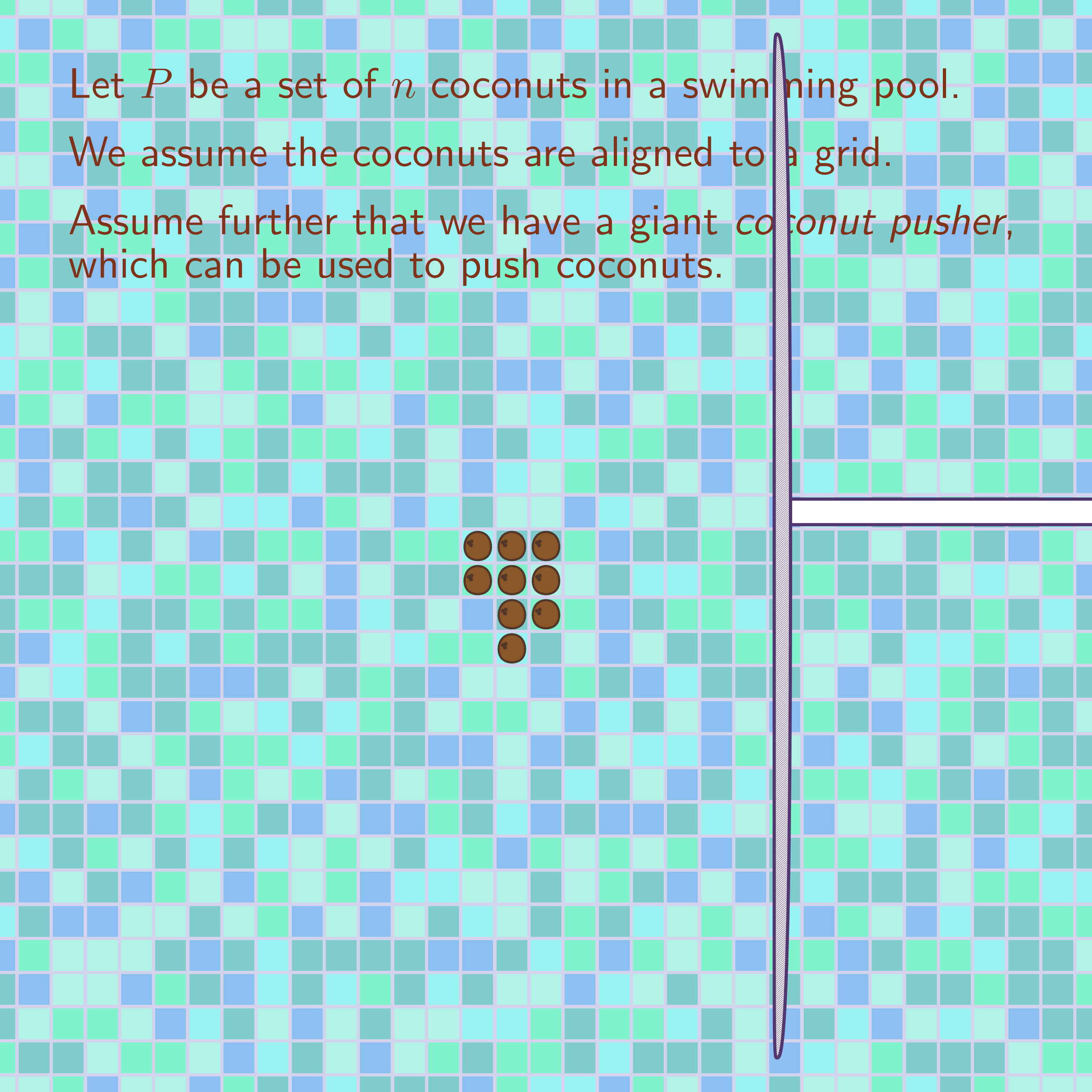
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

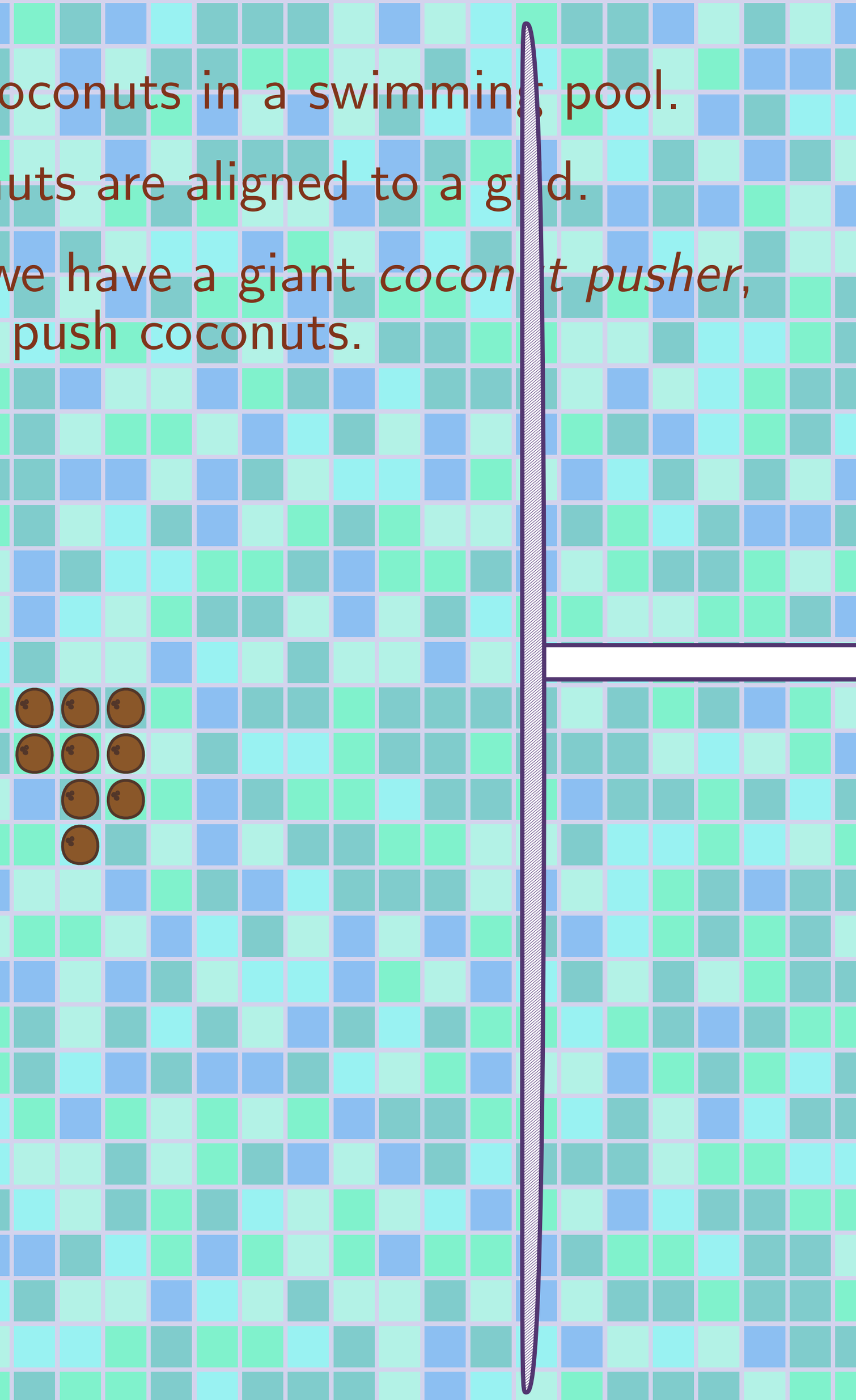
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

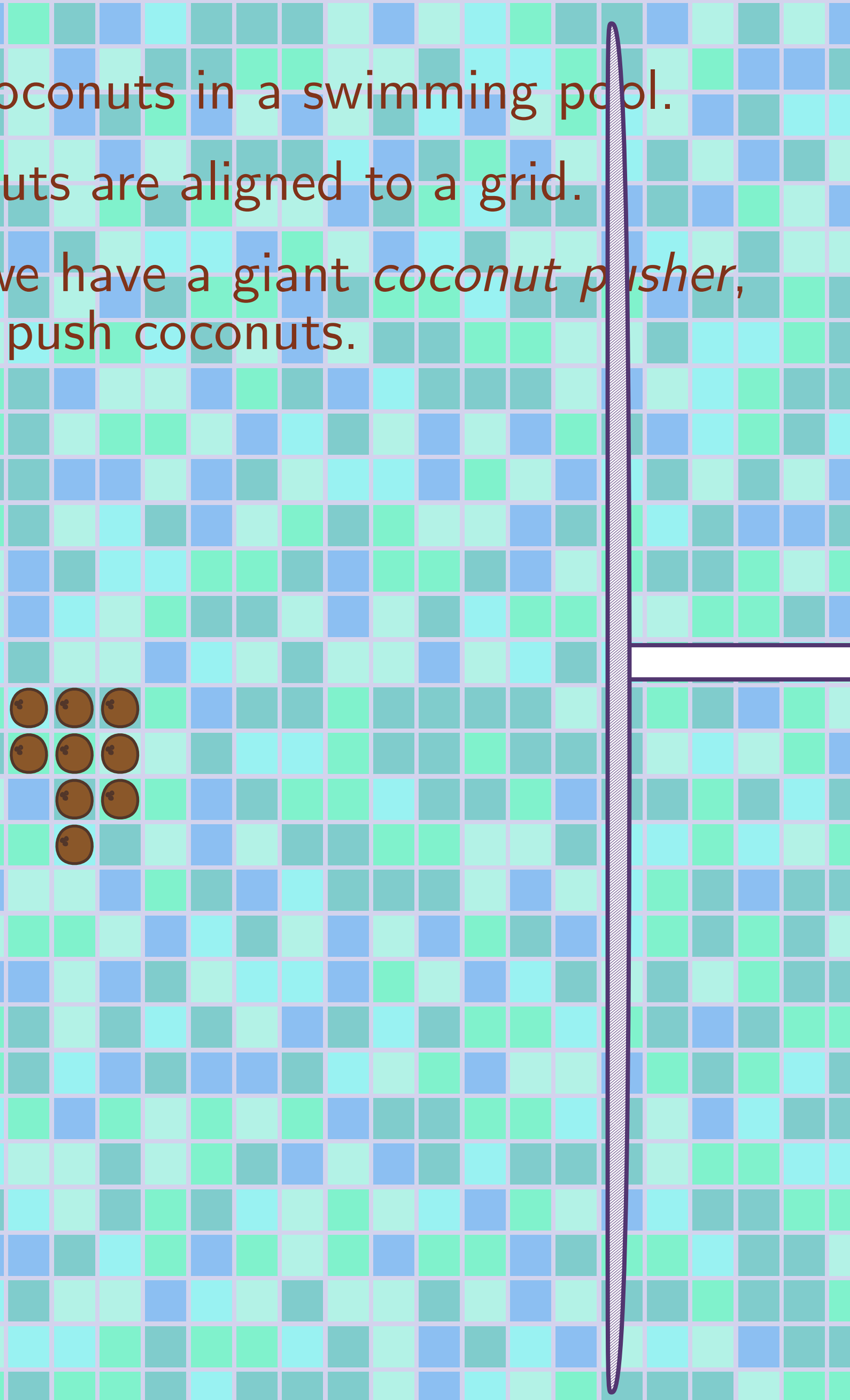
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

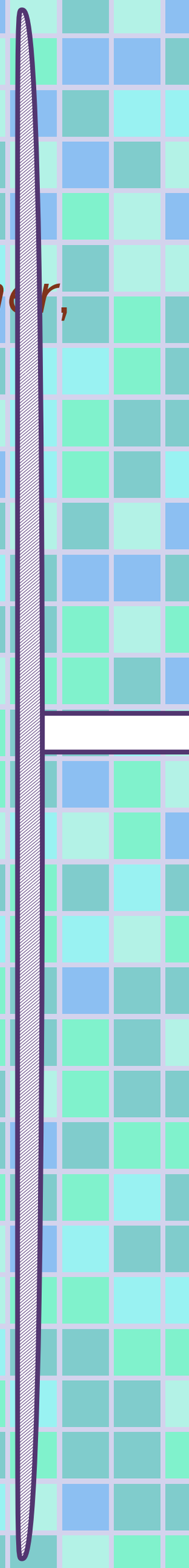
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

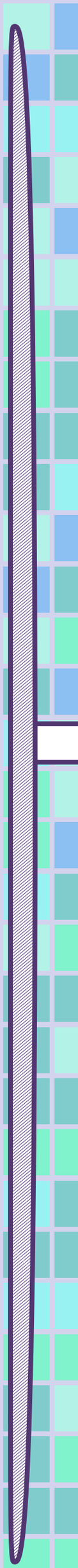
Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.



Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

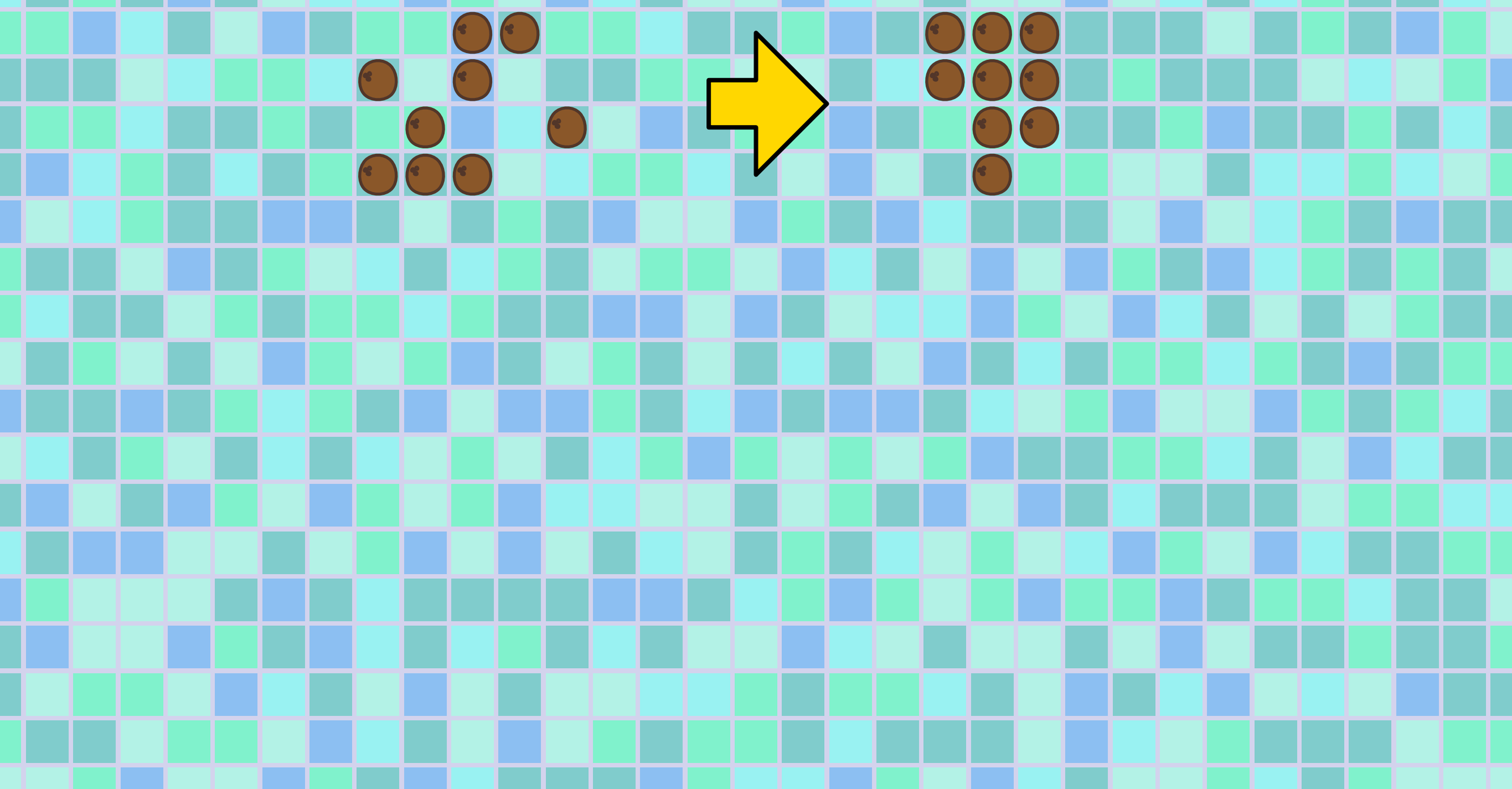
Any sequence of such coconut pushes leads to a *reconfiguration* of the coconuts.

Let P be a set of n coconuts in a swimming pool.

We assume the coconuts are aligned to a grid.

Assume further that we have a giant *coconut pusher*, which can be used to push coconuts.

Any sequence of such coconut pushes leads to a *reconfiguration* of the coconuts.



KNOWN RESULTS

Question

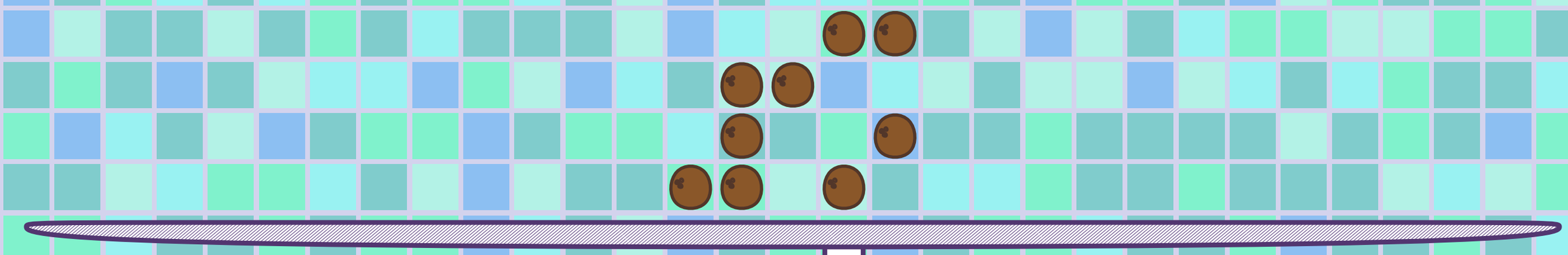
Can we always push our coconuts into tidy rectangles?

Question

Can we always push our coconuts into tidy rectangles?

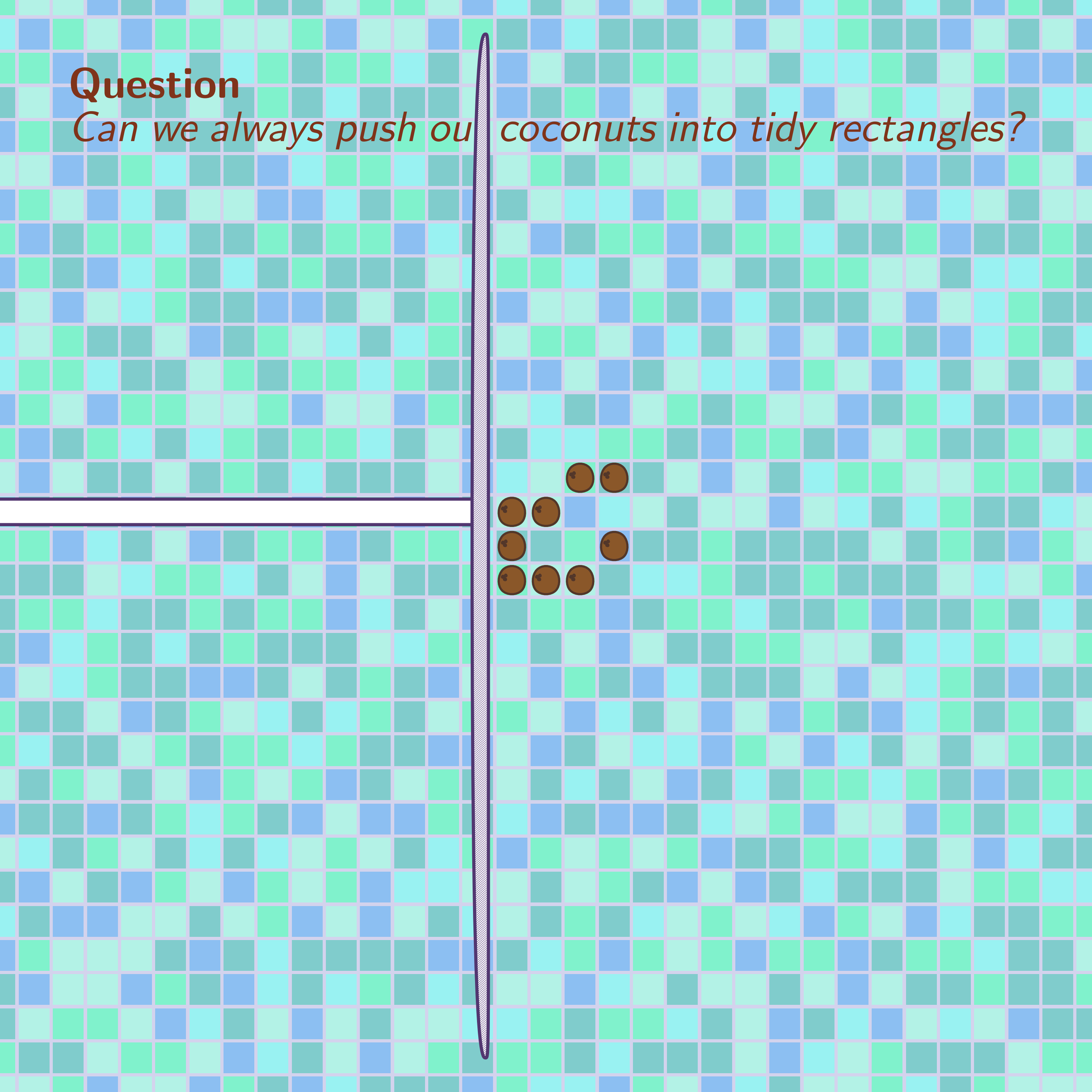
Question

Can we always push our coconuts into tidy rectangles?



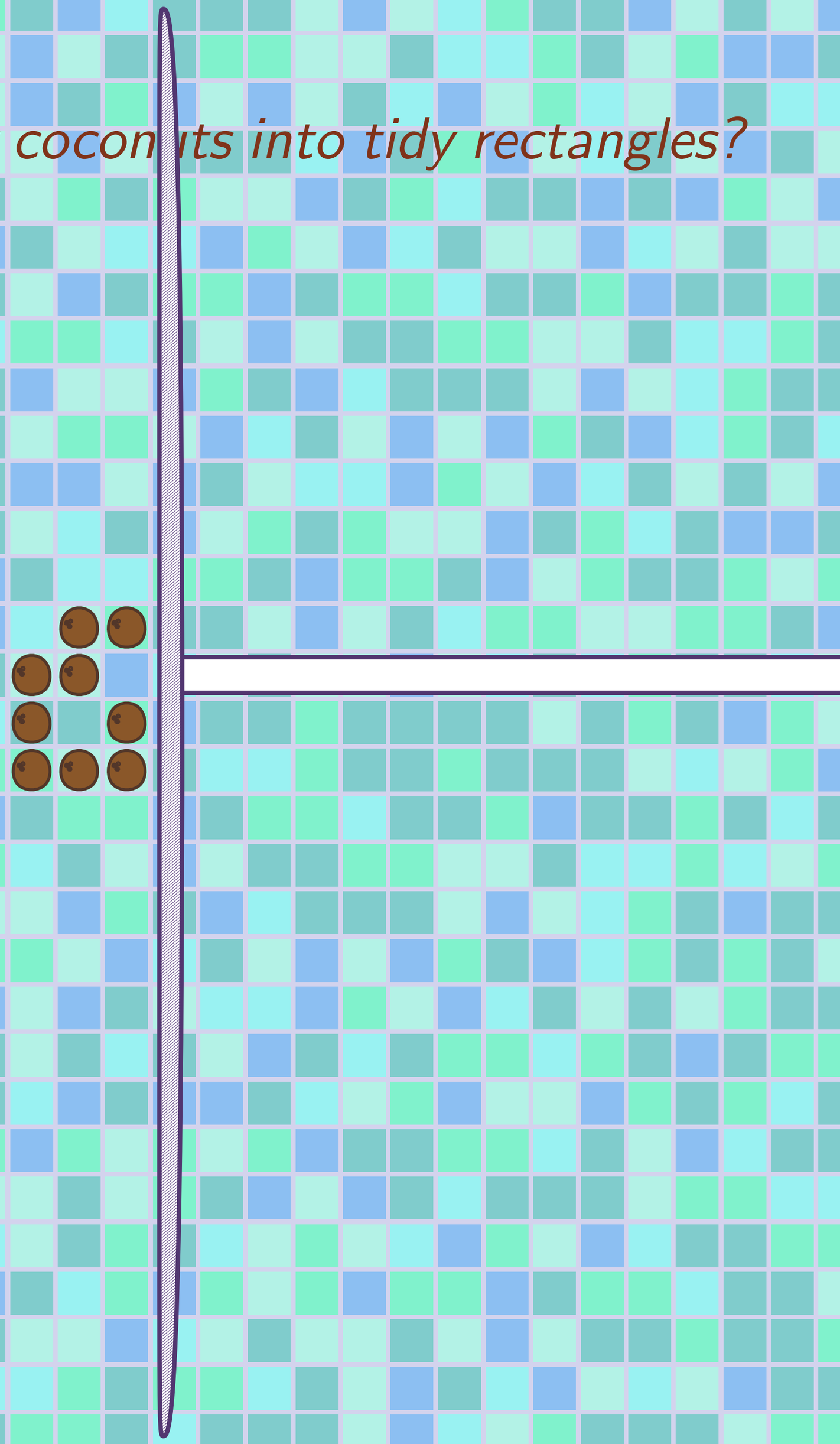
Question

Can we always push out coconuts into tidy rectangles?



Question

Can we always push our coconuts into tidy rectangles?

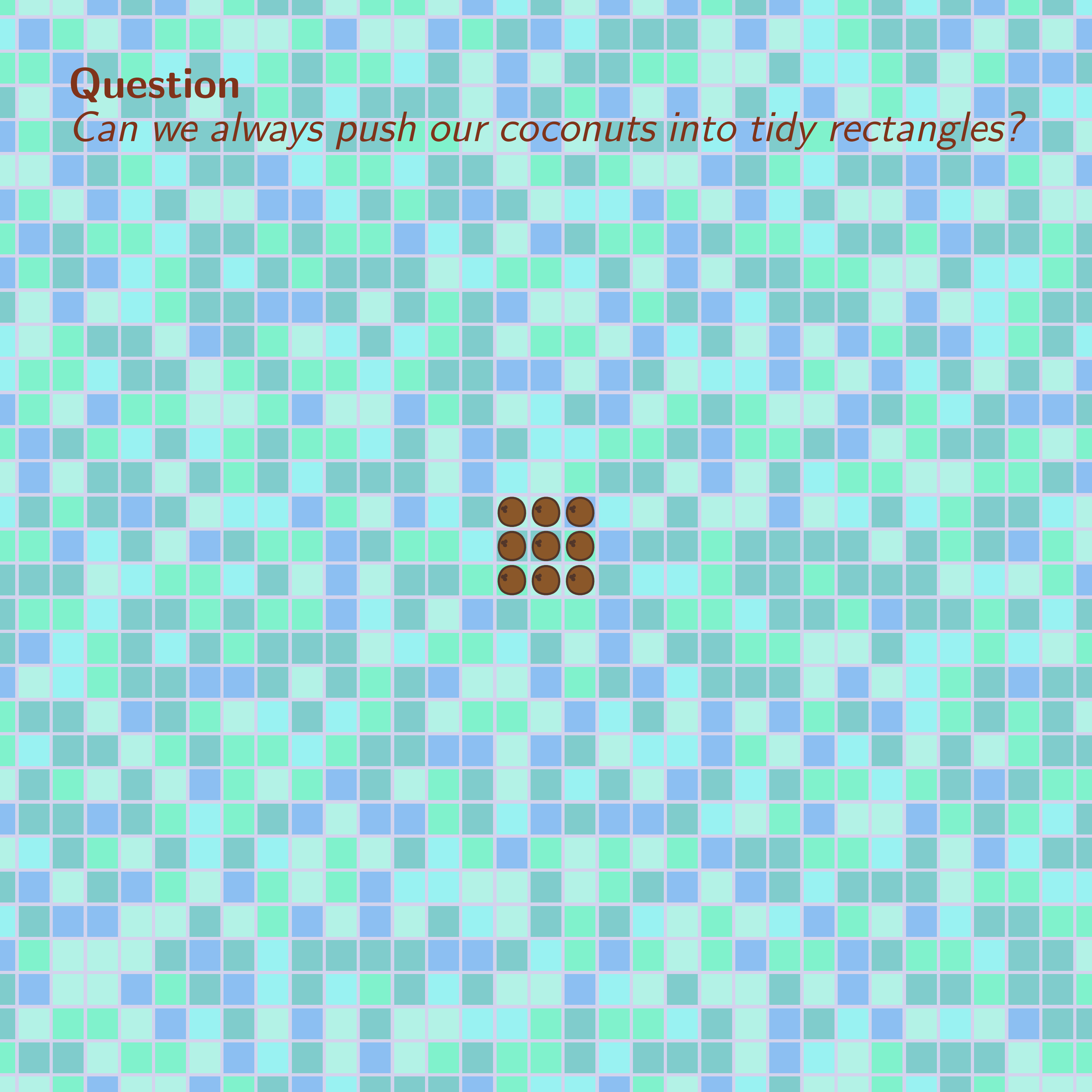


Question

Can we always push our coconuts into tidy rectangles?

Question

Can we always push our coconuts into tidy rectangles?



Question

Can we always push our coconuts into tidy rectangles?

Obviously, n coconuts can only be pushed into a $a \times b$ rectangle if $ab = n$!

Question

Can we always push our coconuts into tidy rectangles?

Obviously, n coconuts can only be pushed into a $a \times b$ rectangle if $ab = n$!

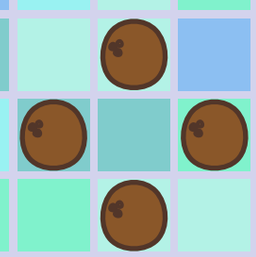
But is it always possible?

Question

Can we always push our coconuts into tidy rectangles?

Obviously, n coconuts can only be pushed into a $a \times b$ rectangle if $ab = n$!

But is it always possible?



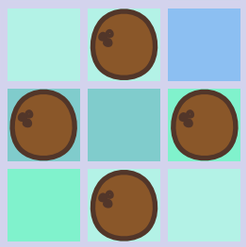
Question

Can we always push our coconuts into tidy rectangles?

Obviously, n coconuts can only be pushed into a $a \times b$ rectangle if $ab = n$!

But is it always possible?

No!



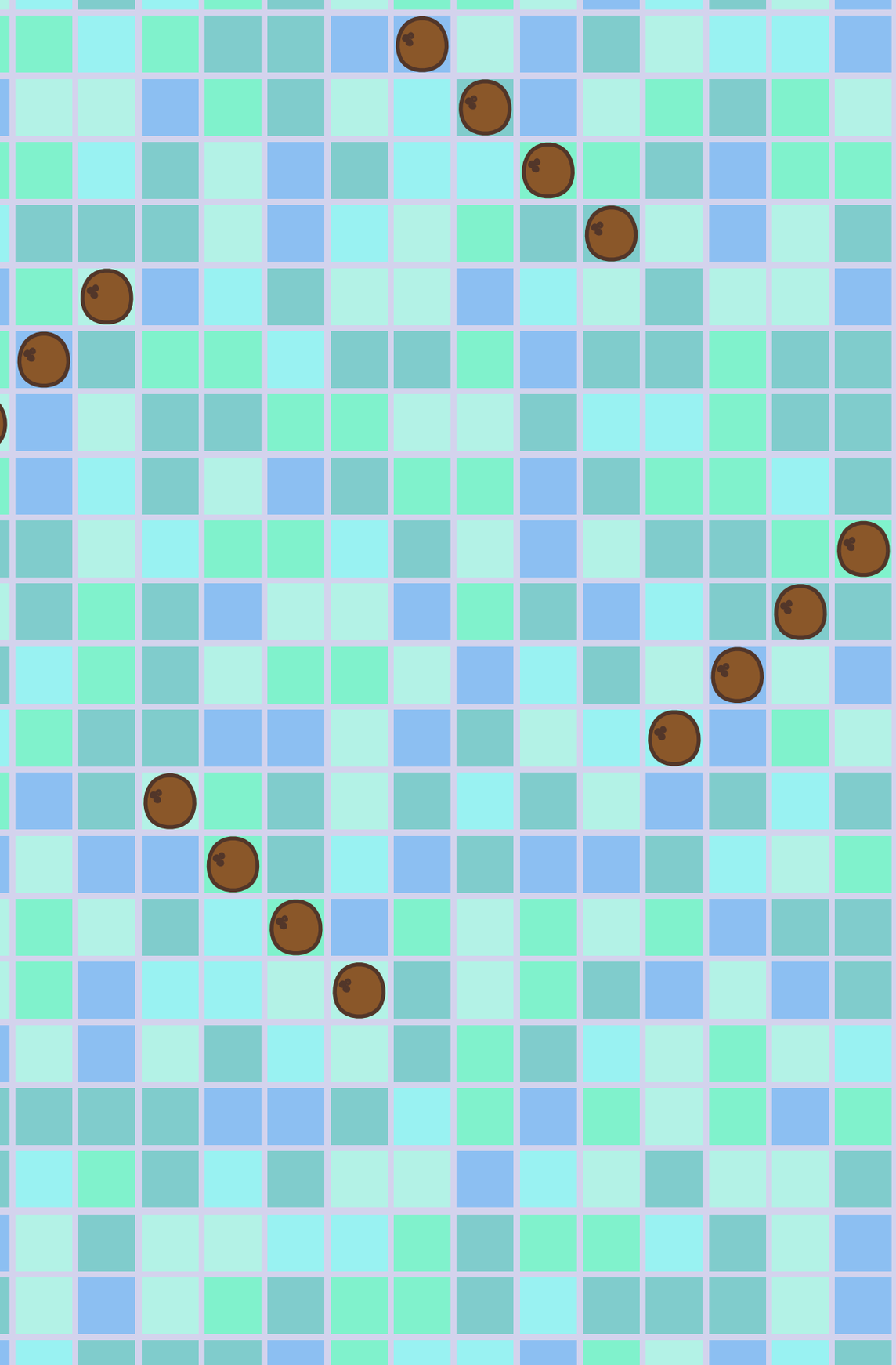
Question

Can we always push our coconuts into tidy rectangles?

Obviously, n coconuts can only be pushed into a $a \times b$ rectangle if $ab = n$!

But is it always possible?

No!



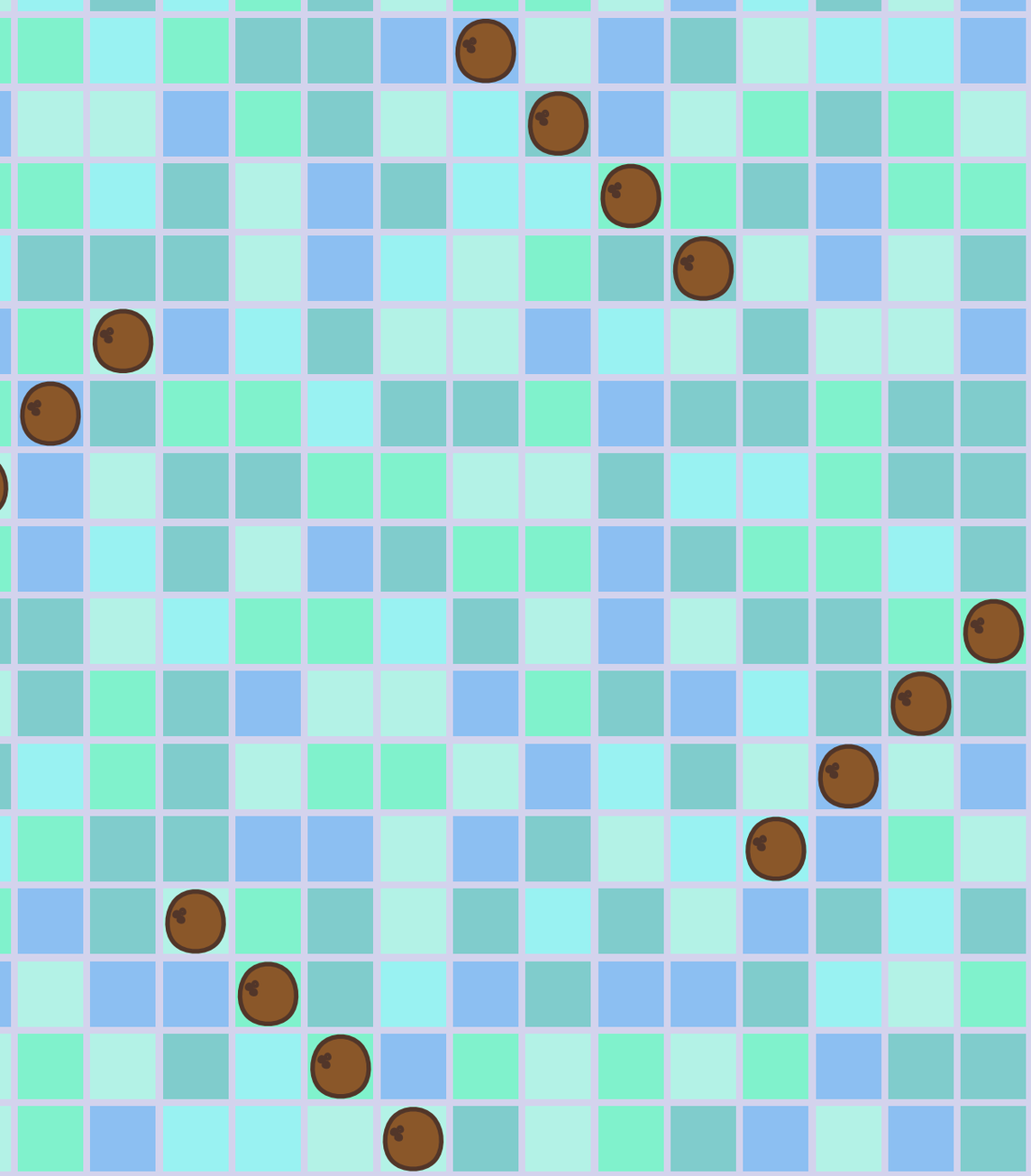
Question

Can we always push our coconuts into tidy rectangles?

Obviously, n coconuts can only be pushed into a $a \times b$ rectangle if $ab = n$!

But is it always possible?

No!



...and *some* aspect ratios are not even possible if we start with at most one coconut per row and column.

Results

Results

Deciding whether n coconuts can be pushed into an $a \times b$ rectangle is NP-complete.

Results

Deciding whether n coconuts can be pushed into an $a \times b$ rectangle is NP-complete.

Deciding whether n coconuts which occupy at most k rows can be pushed into a $2 \times n/2$ rectangle is polynomial.

Results

Deciding whether n coconuts can be pushed into an $a \times b$ rectangle is NP-complete.

Deciding whether n coconuts which occupy at most k rows can be pushed into a $2 \times n/2$ rectangle is polynomial.

Everything inbetween is still open!

Results

Deciding whether n coconuts can be pushed into an $a \times b$ rectangle is NP-complete.

Deciding whether n coconuts which occupy at most k rows can be pushed into a $2 \times n/2$ rectangle is polynomial.

Everything inbetween is still open!

[Akitaya, Aloupis, Löffler, Rounds, 2016]

GAMES

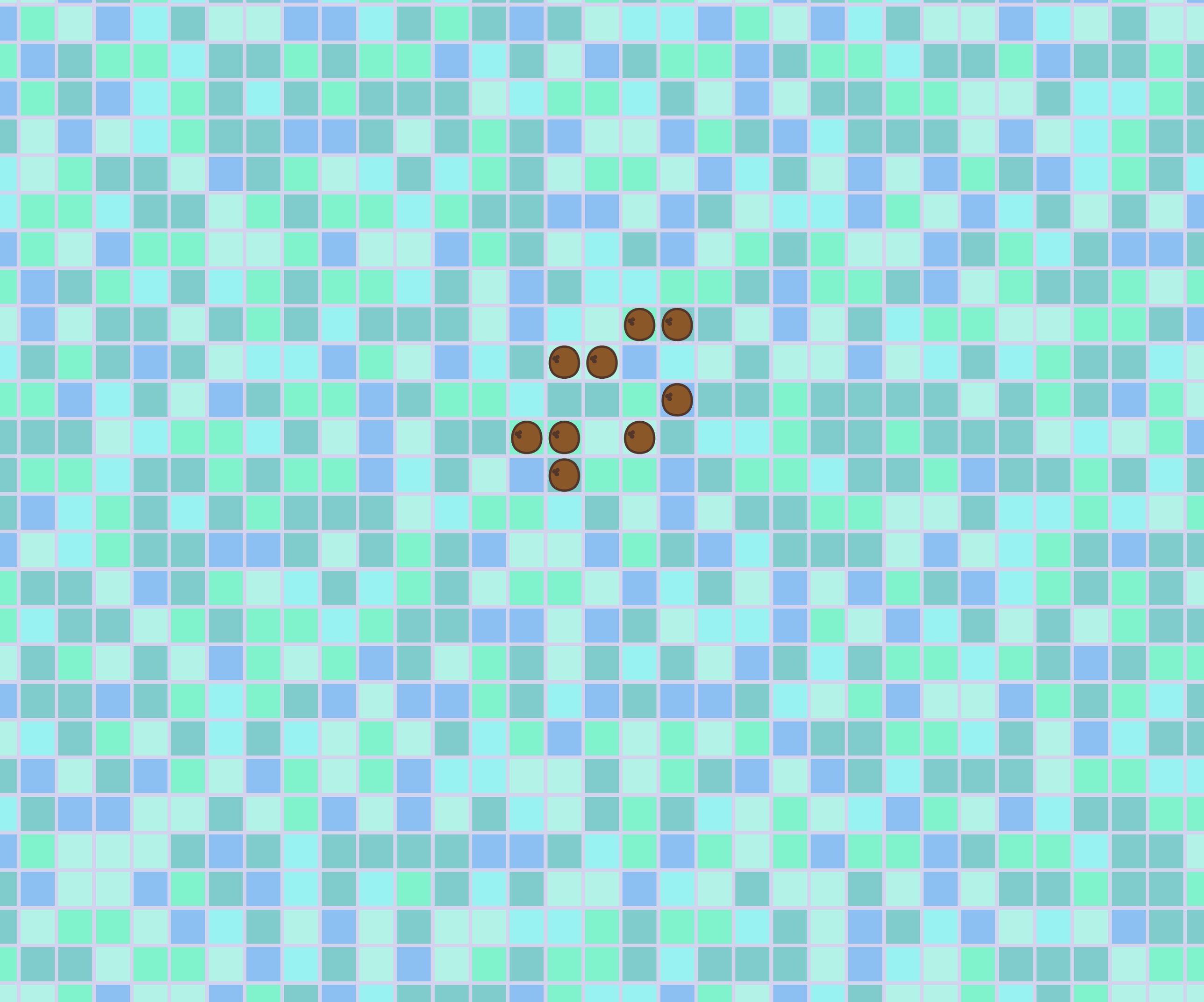
In compaction *games*, players take turns pushing.

In compaction *games*, players take turns pushing.

We will consider 2-player games, *red* against *blue*.

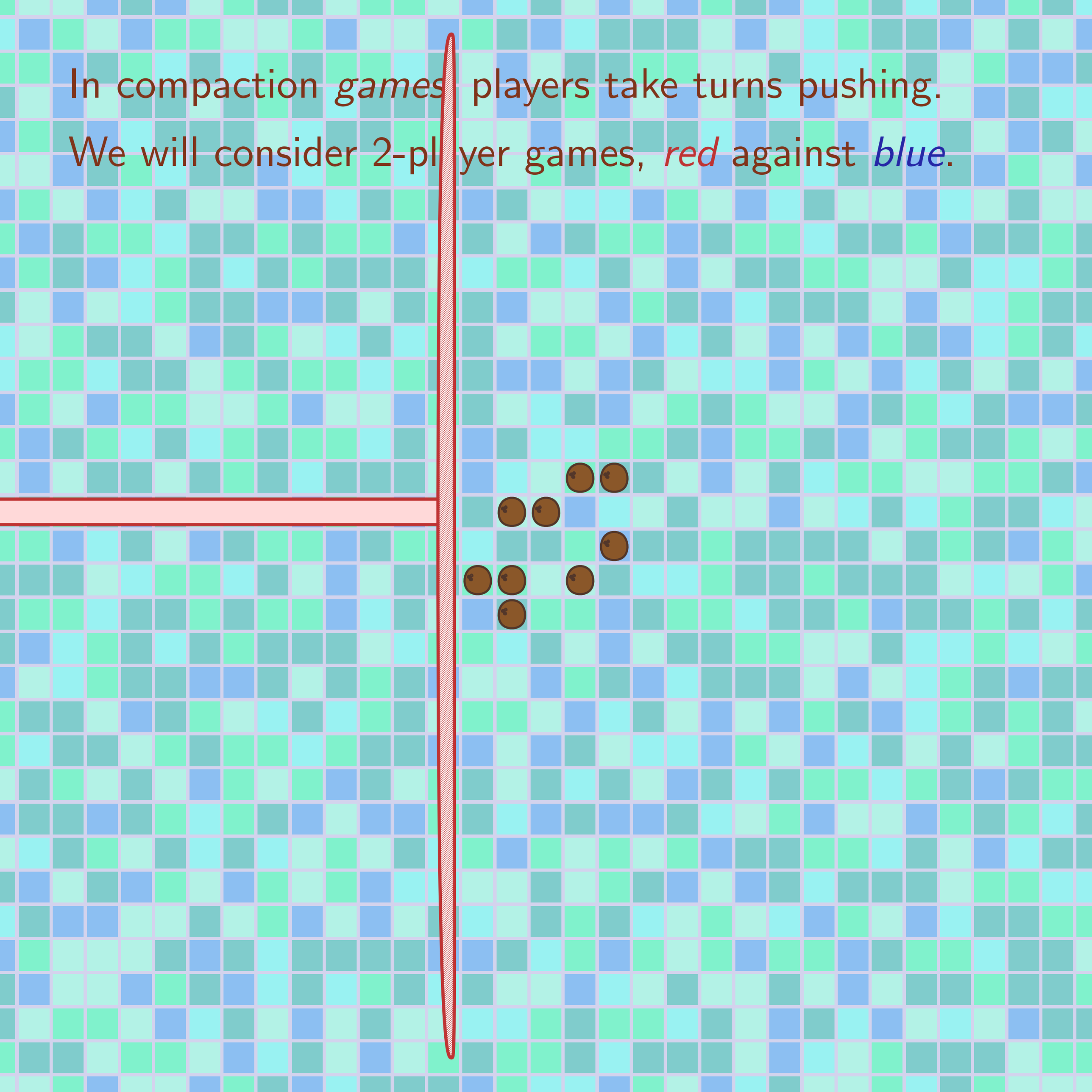
In compaction *games*, players take turns pushing.

We will consider 2-player games, *red* against *blue*.



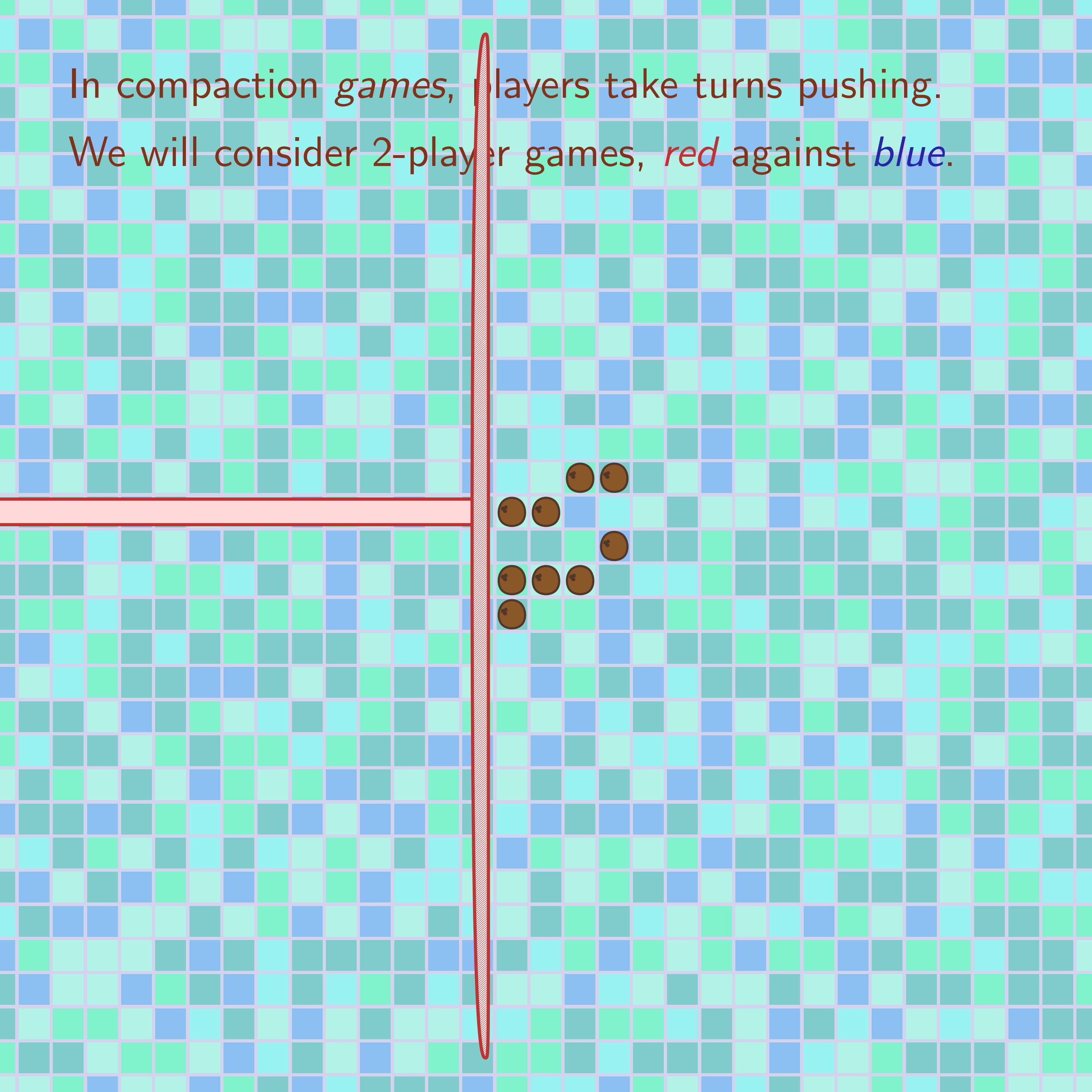
In compaction *games* players take turns pushing.

We will consider 2-player games, *red* against *blue*.



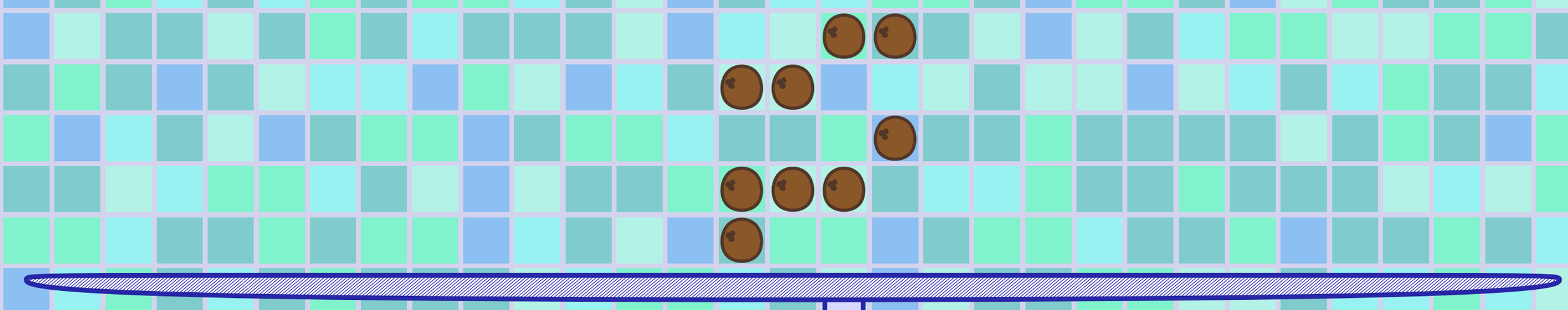
In compaction *games*, players take turns pushing.

We will consider 2-player games, *red* against *blue*.



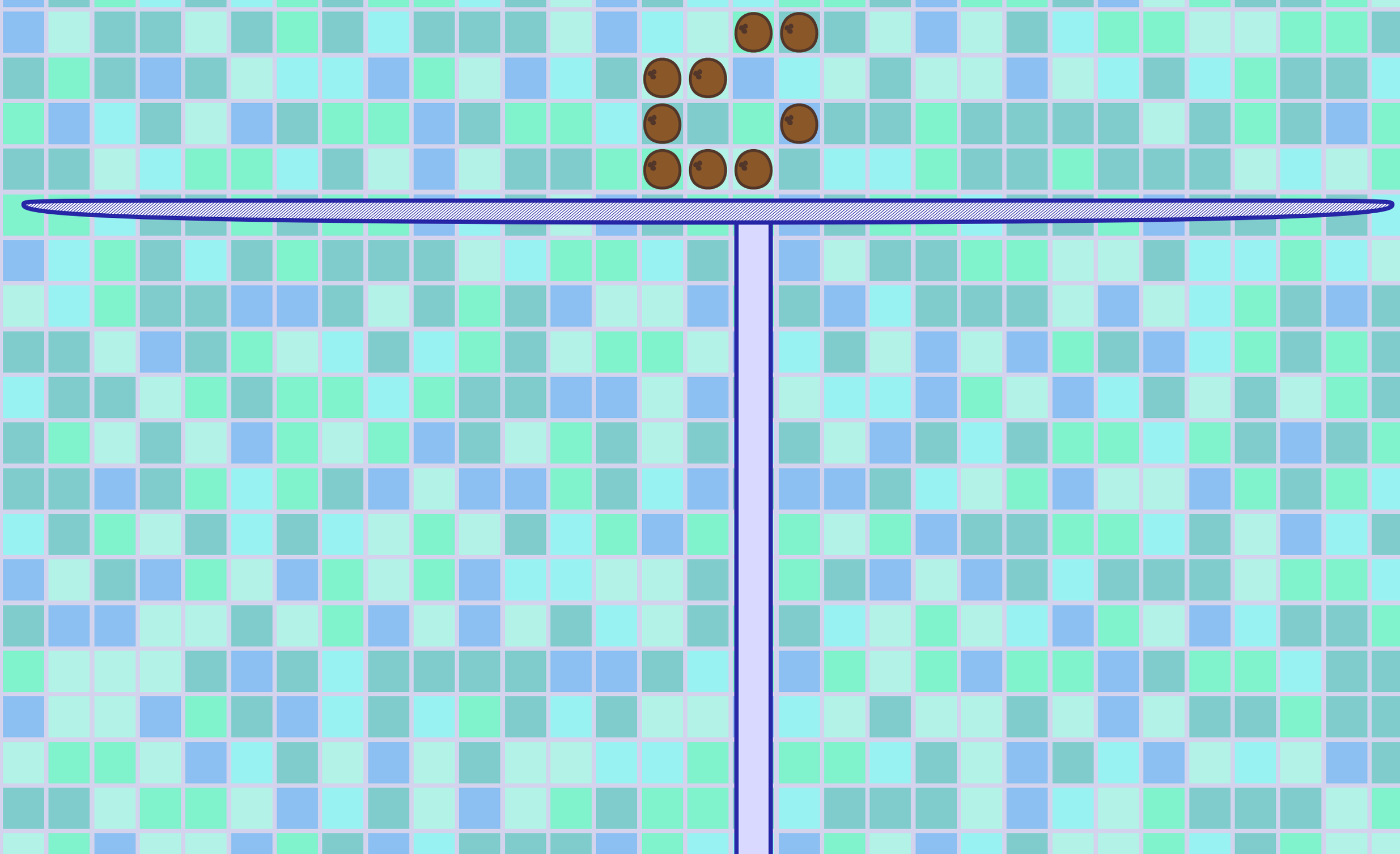
In compaction *games*, players take turns pushing.

We will consider 2-player games, *red* against *blue*.



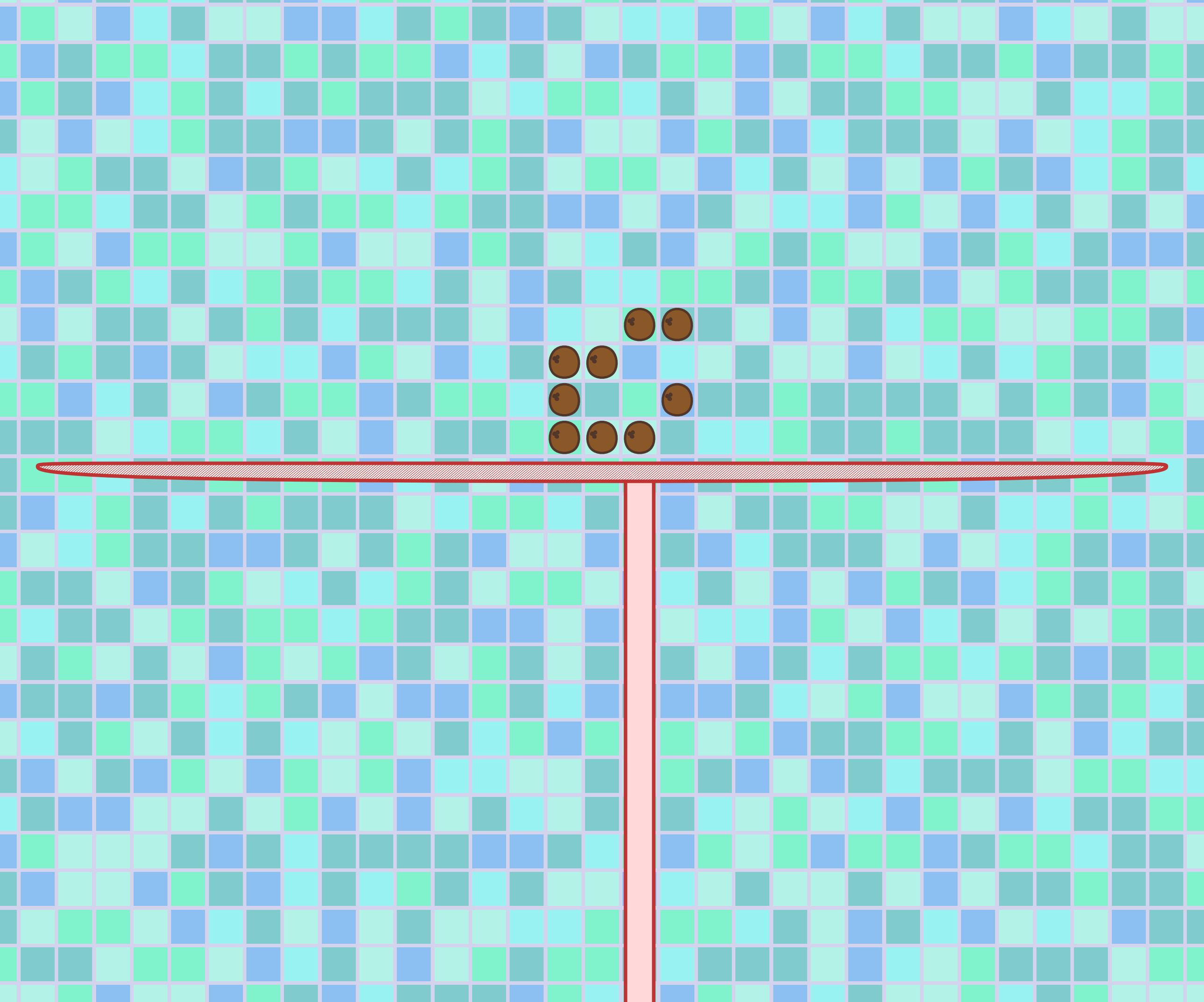
In compaction *games*, players take turns pushing.

We will consider 2-player games, *red* against *blue*.



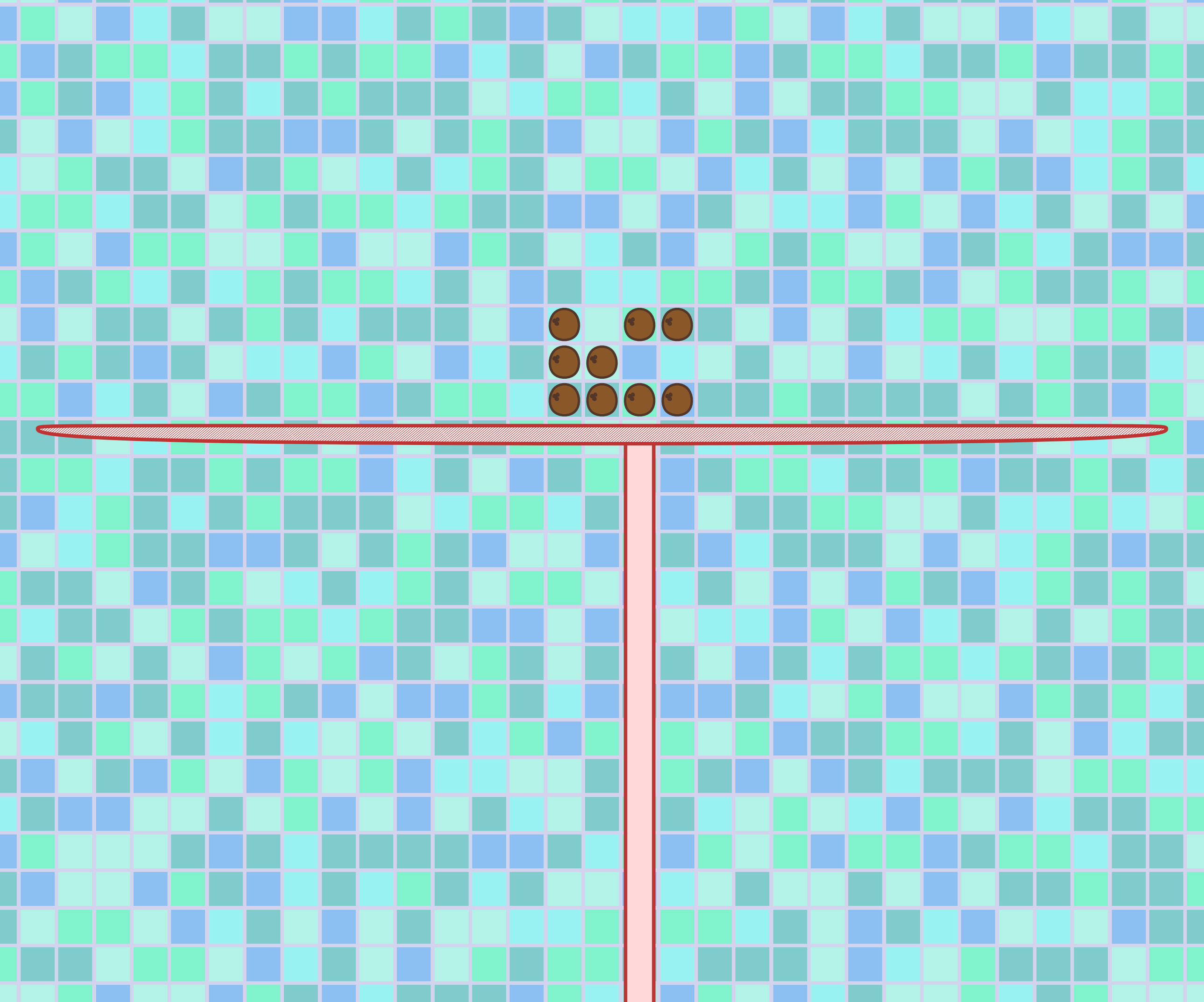
In compaction *games*, players take turns pushing.

We will consider 2-player games, *red* against *blue*.



In compaction *games*, players take turns pushing.

We will consider 2-player games, *red* against *blue*.



GAME 1

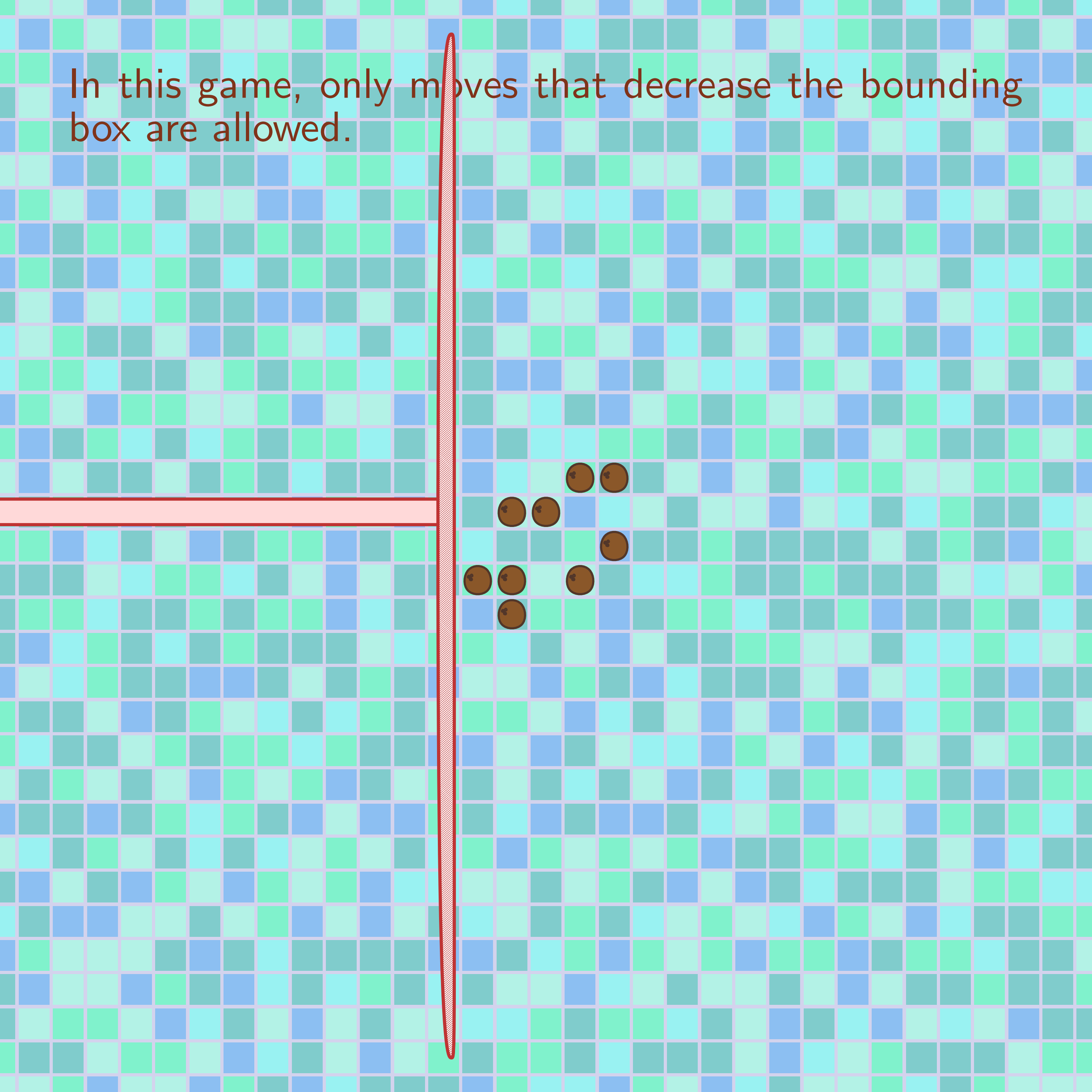
LAST MOVE WINS

In this game, only moves that decrease the bounding box are allowed.

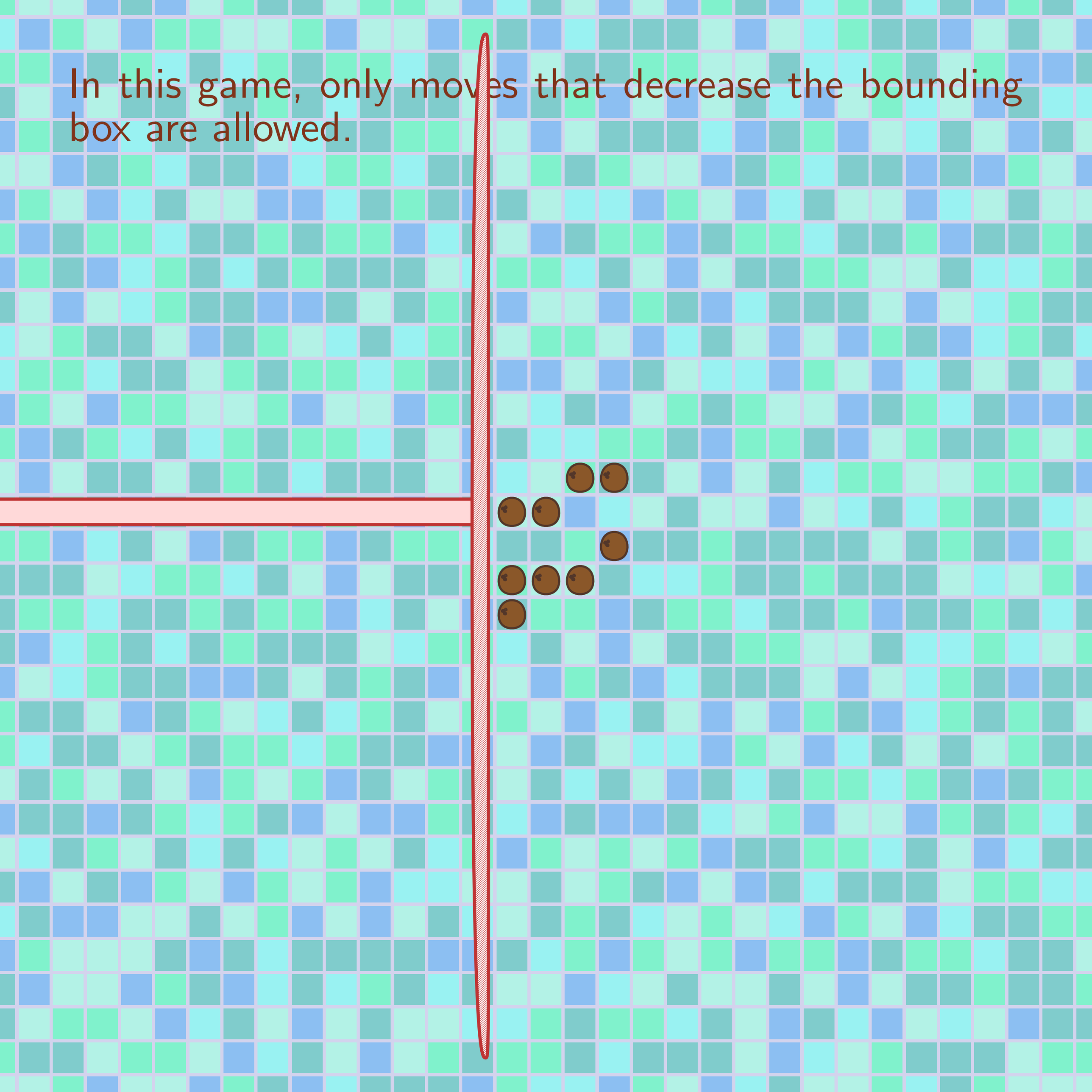
In this game, only moves that decrease the bounding box are allowed.

In this game, only moves that decrease the bounding box are allowed.

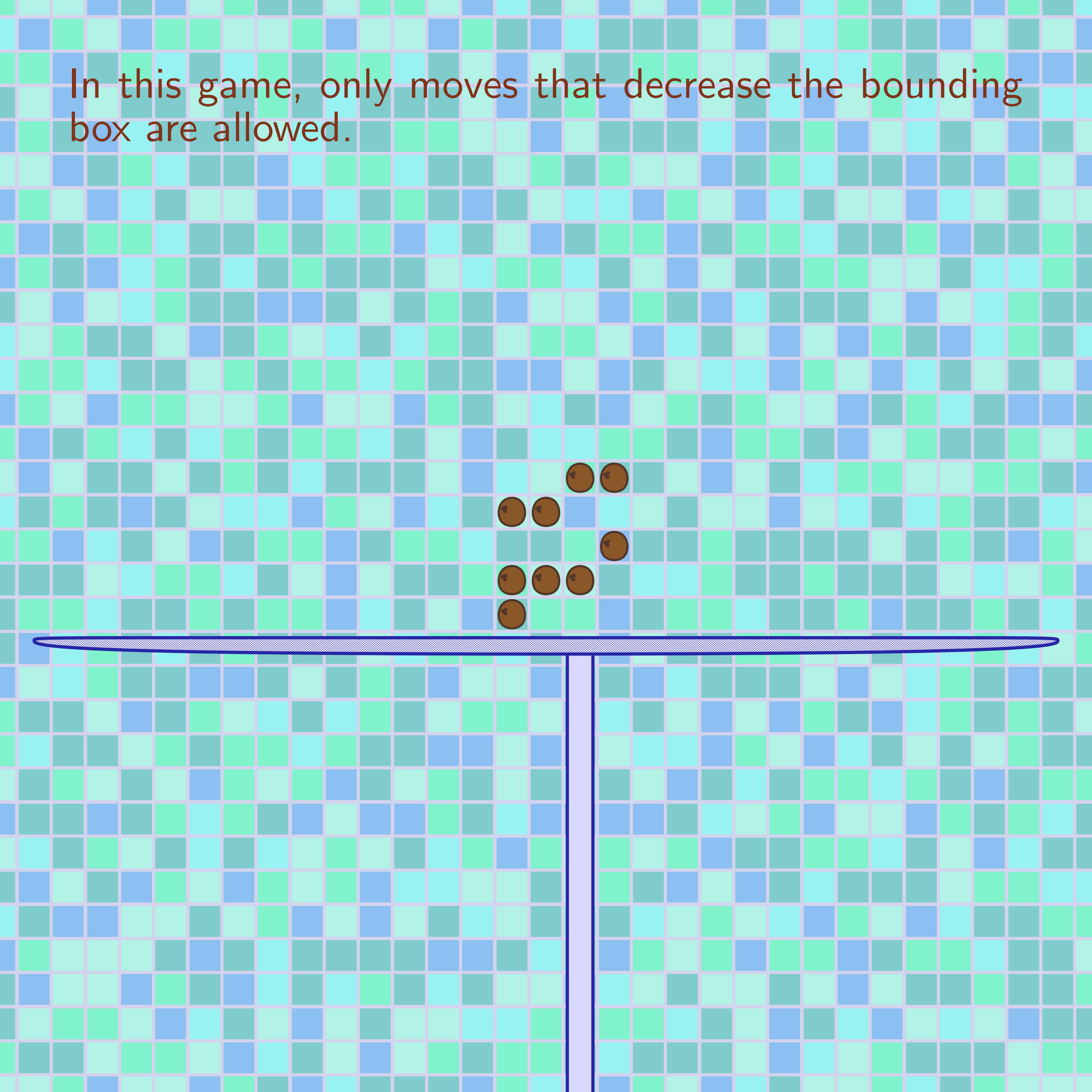
In this game, only moves that decrease the bounding box are allowed.



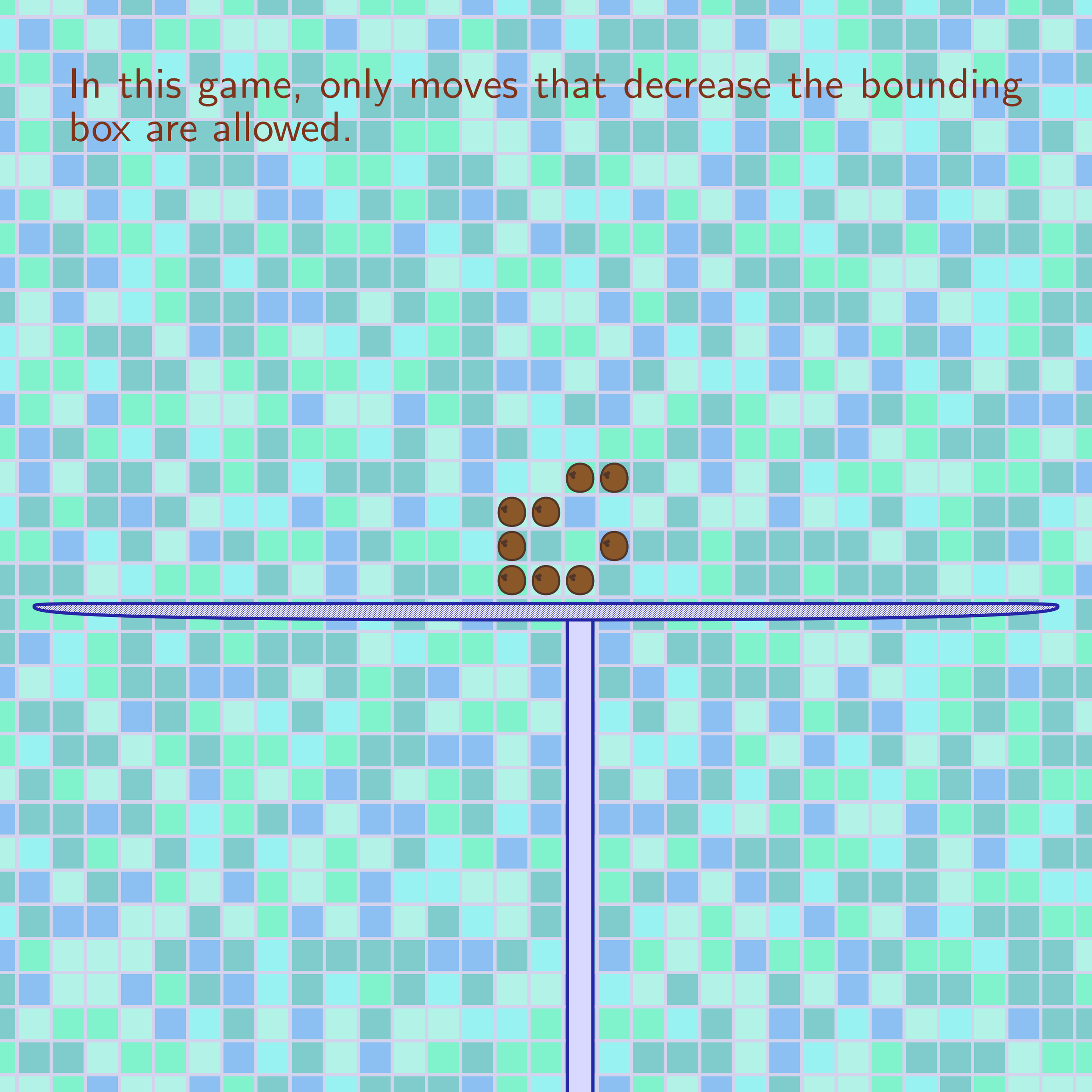
In this game, only moves that decrease the bounding box are allowed.



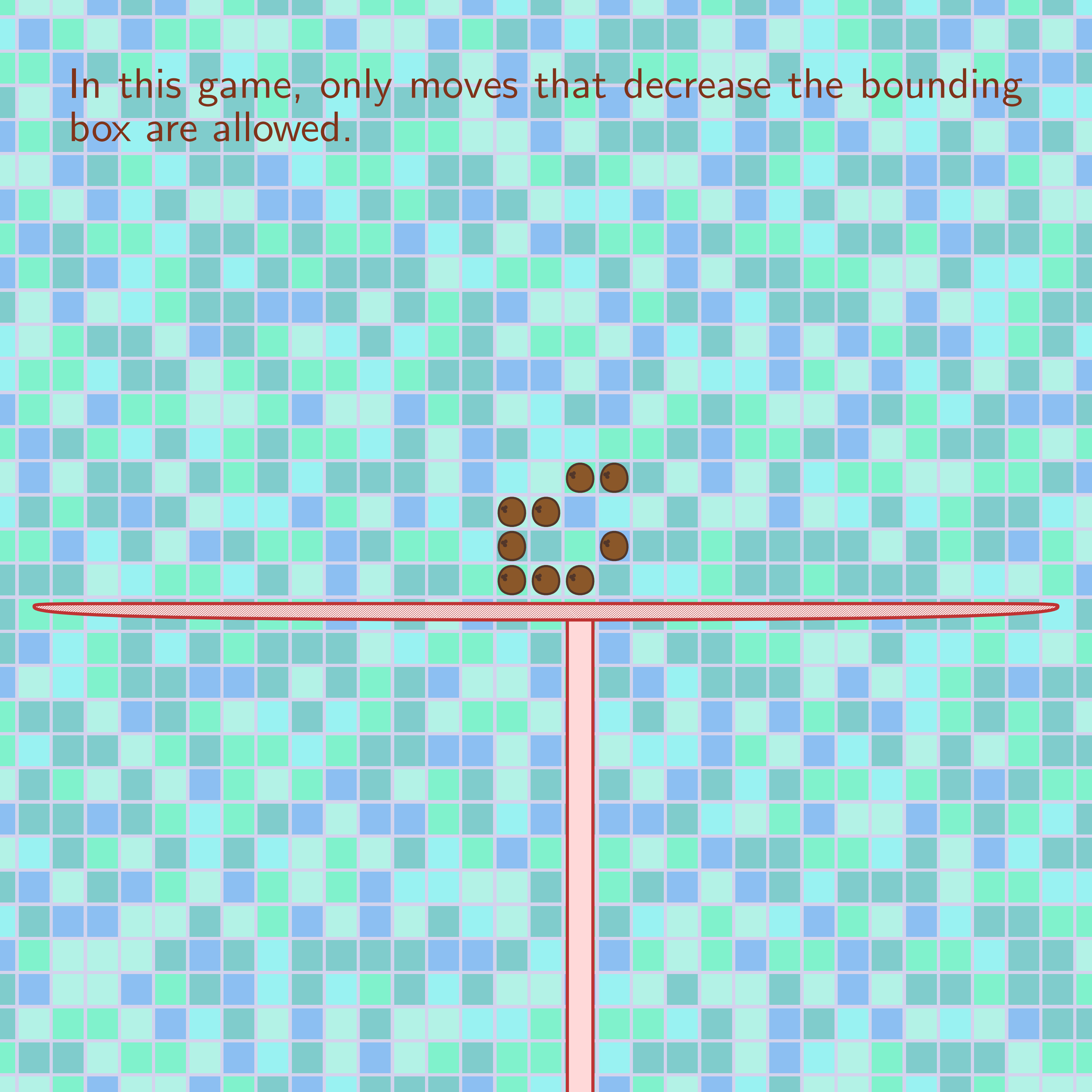
In this game, only moves that decrease the bounding box are allowed.



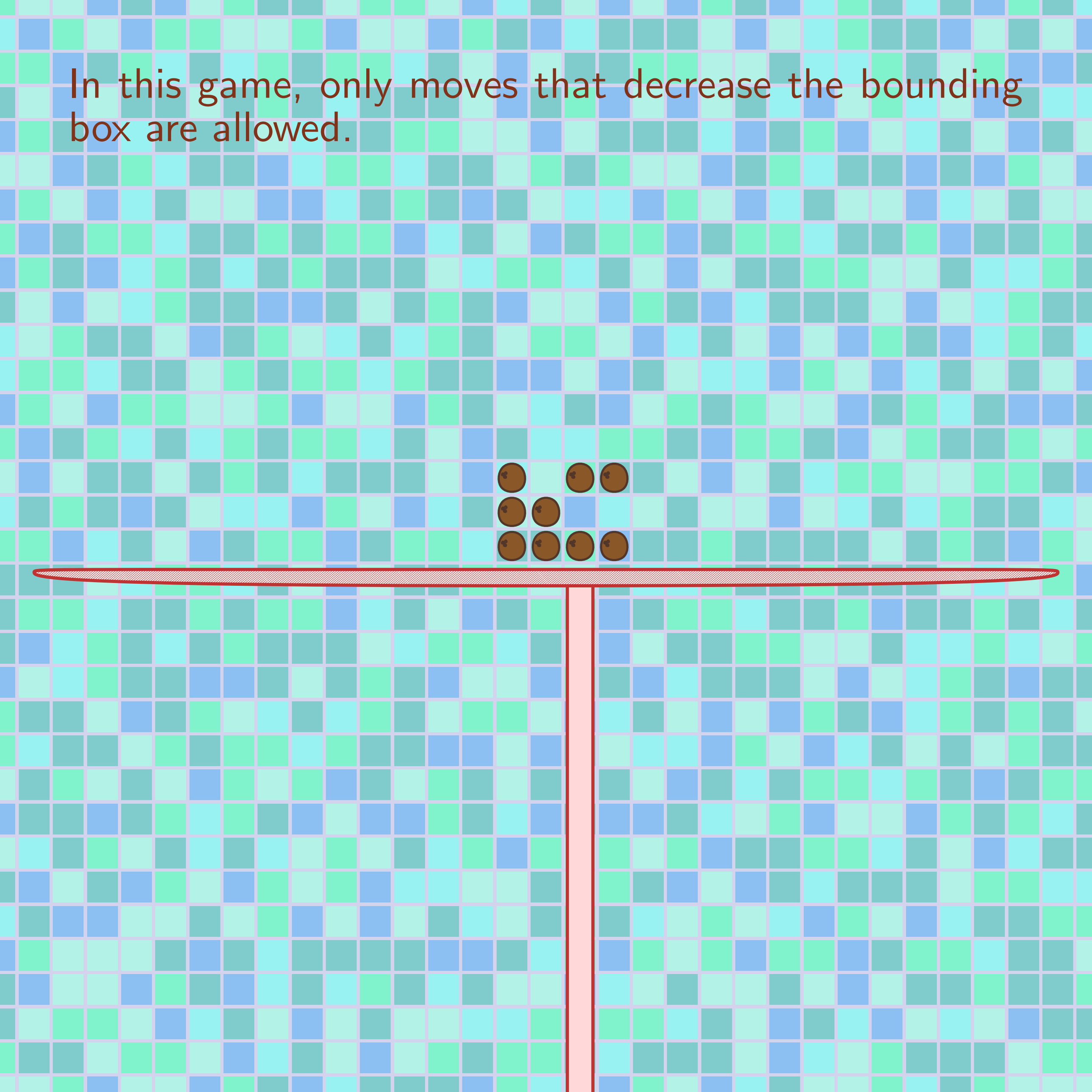
In this game, only moves that decrease the bounding box are allowed.



In this game, only moves that decrease the bounding box are allowed.



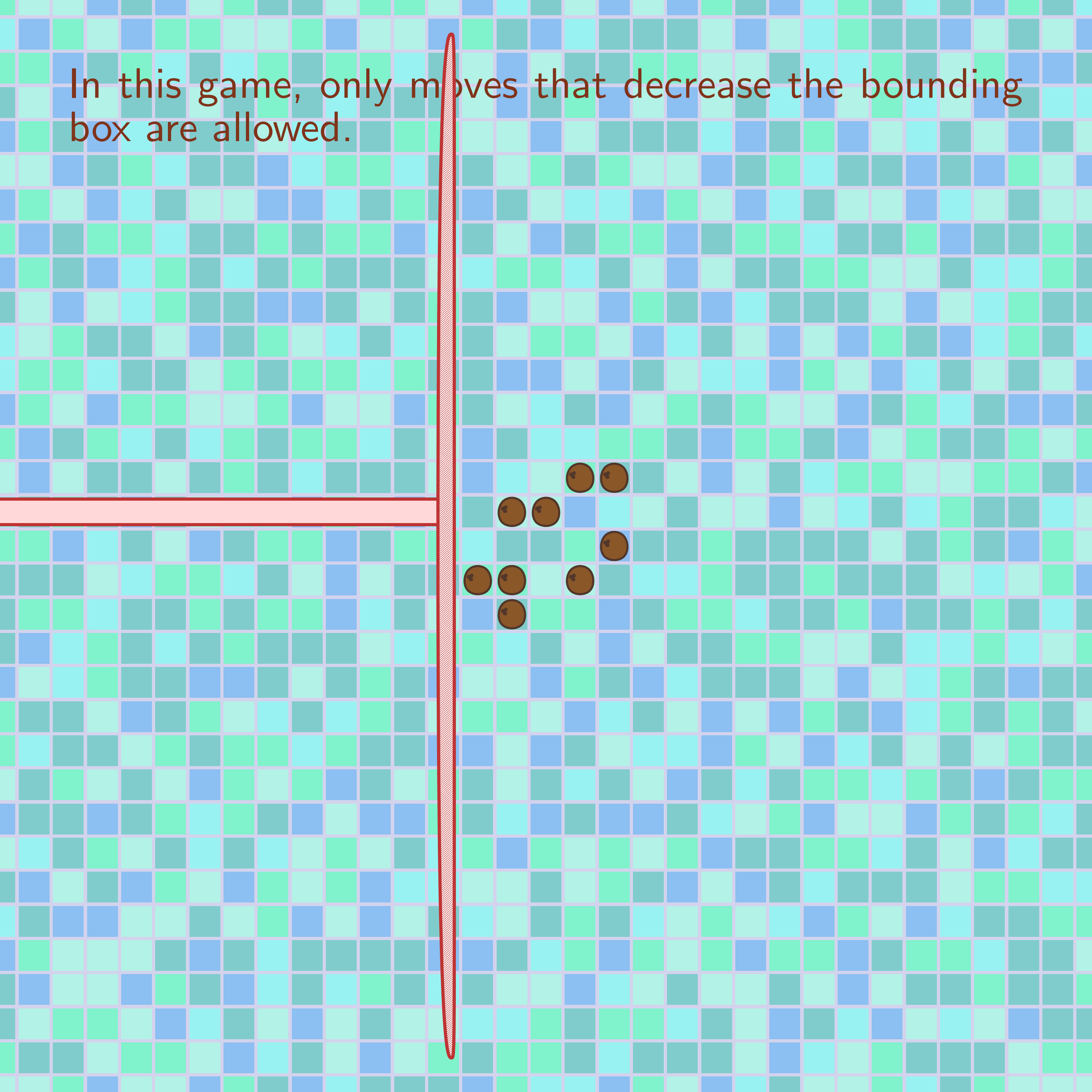
In this game, only moves that decrease the bounding box are allowed.



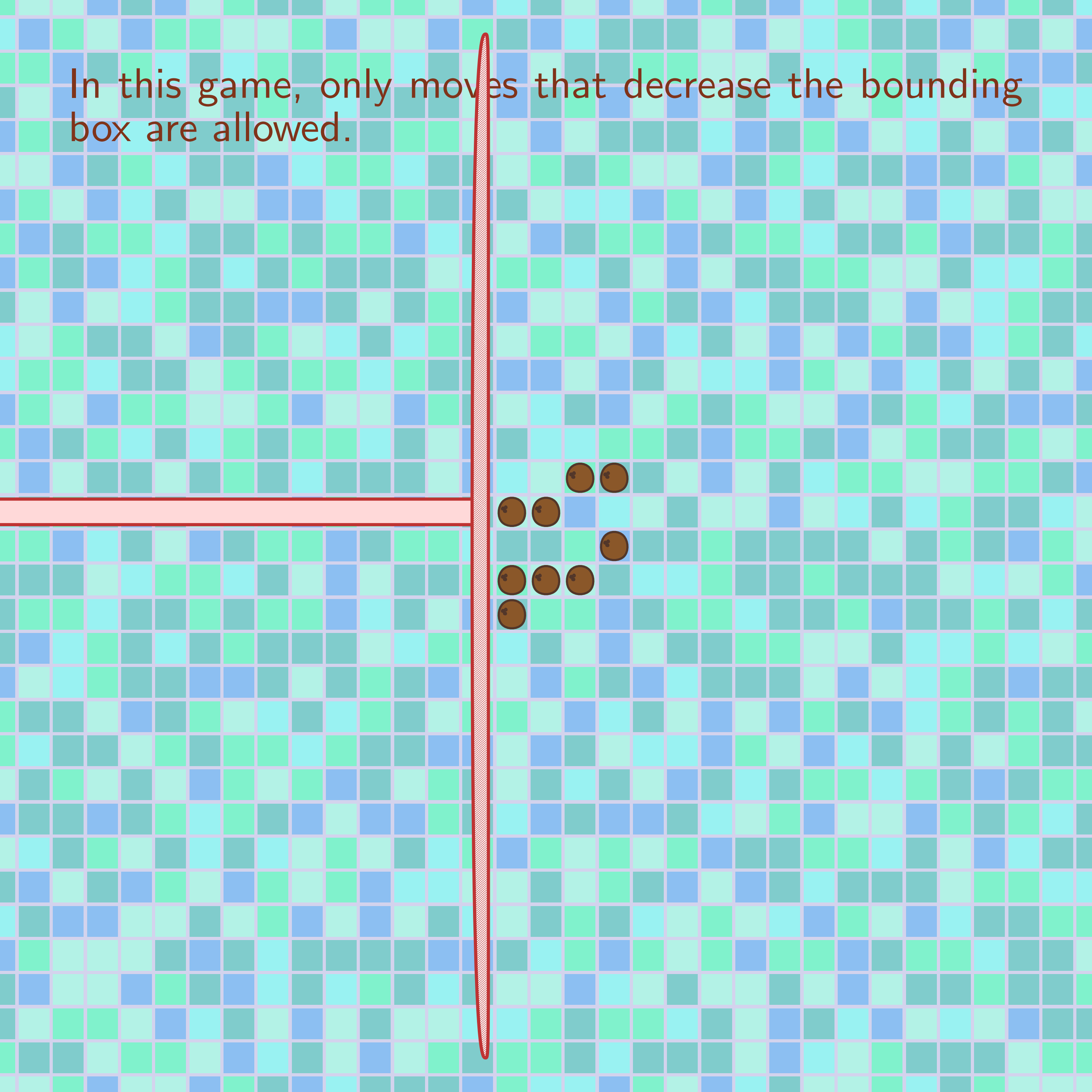
In this game, only moves that decrease the bounding box are allowed.

RED WINS!

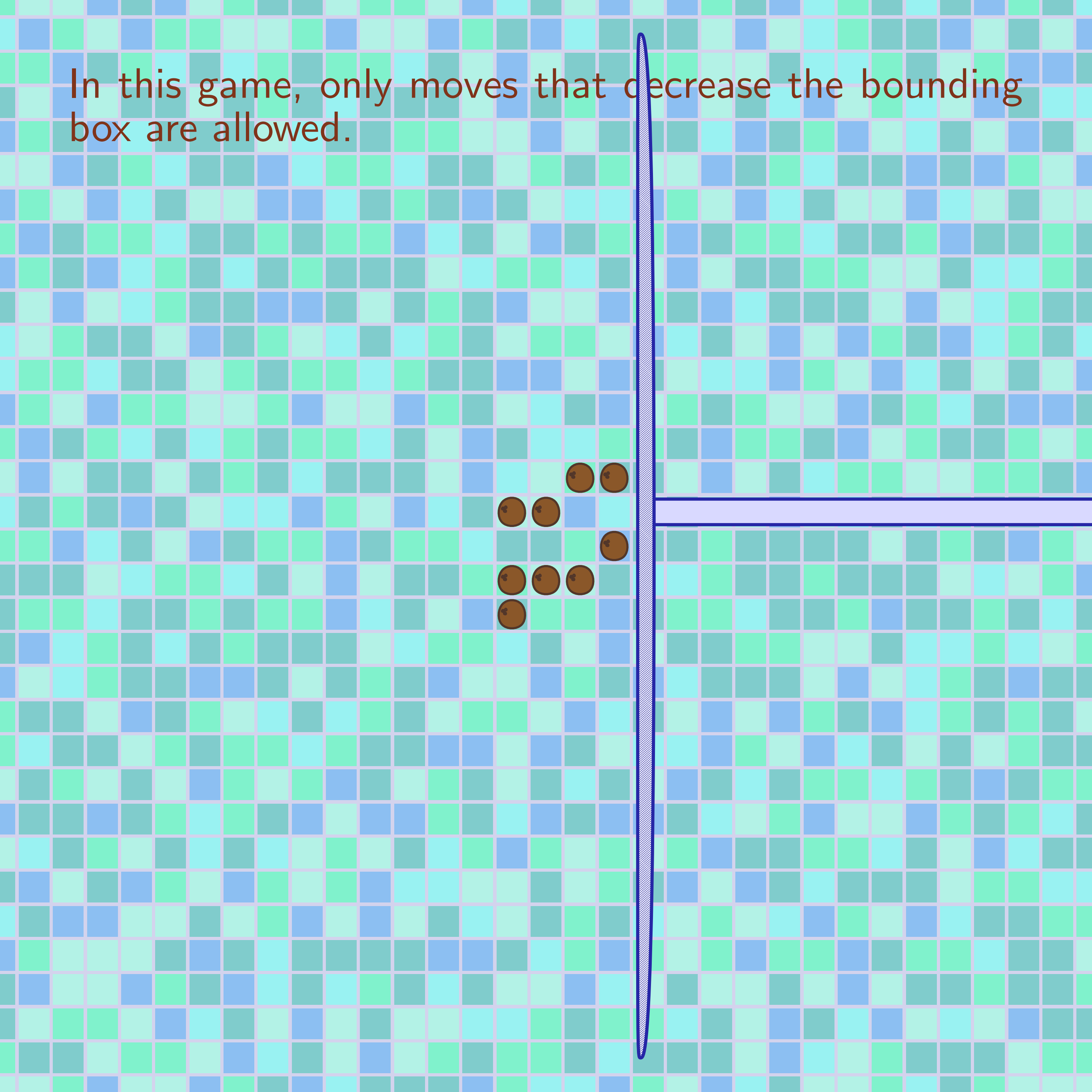
In this game, only moves that decrease the bounding box are allowed.



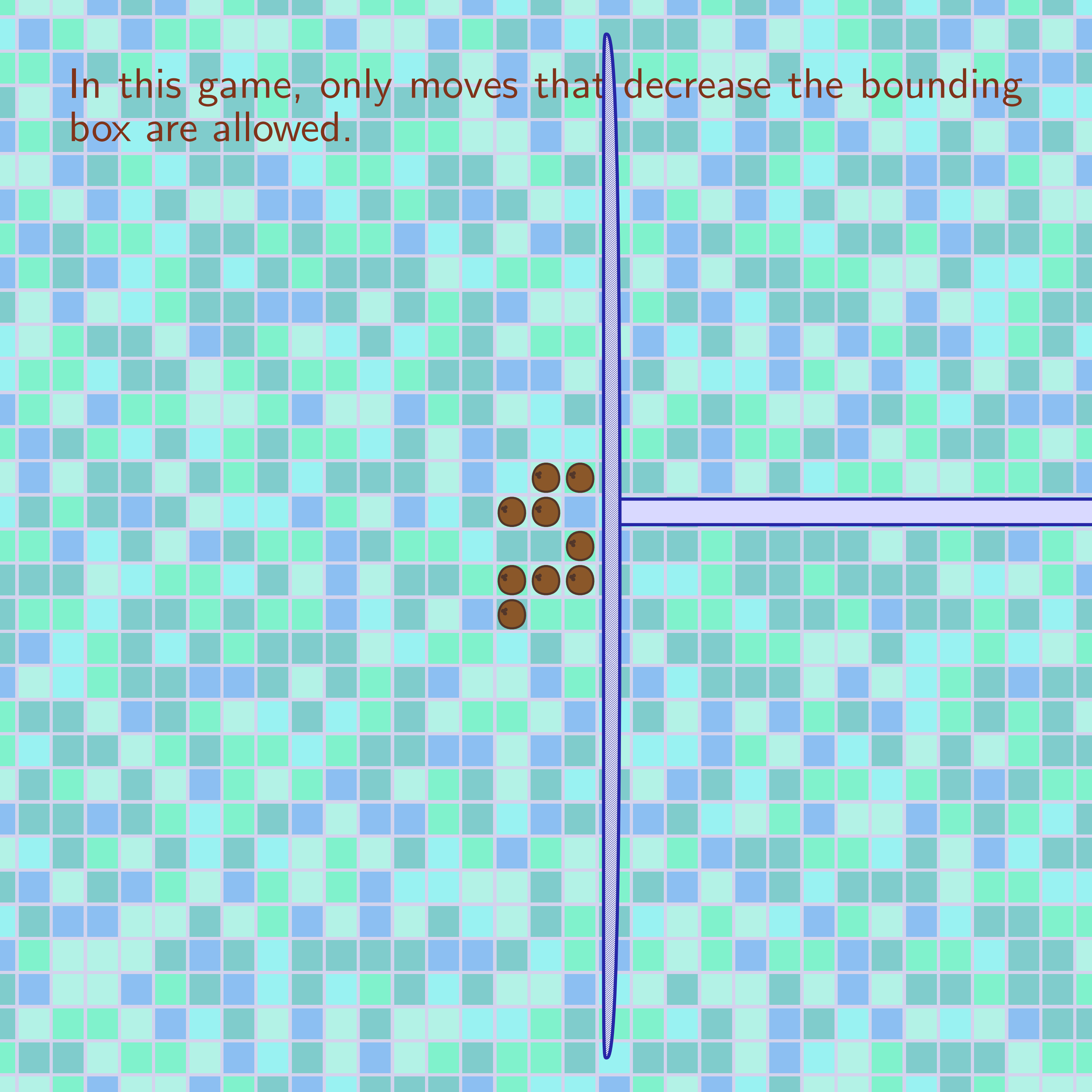
In this game, only moves that decrease the bounding box are allowed.



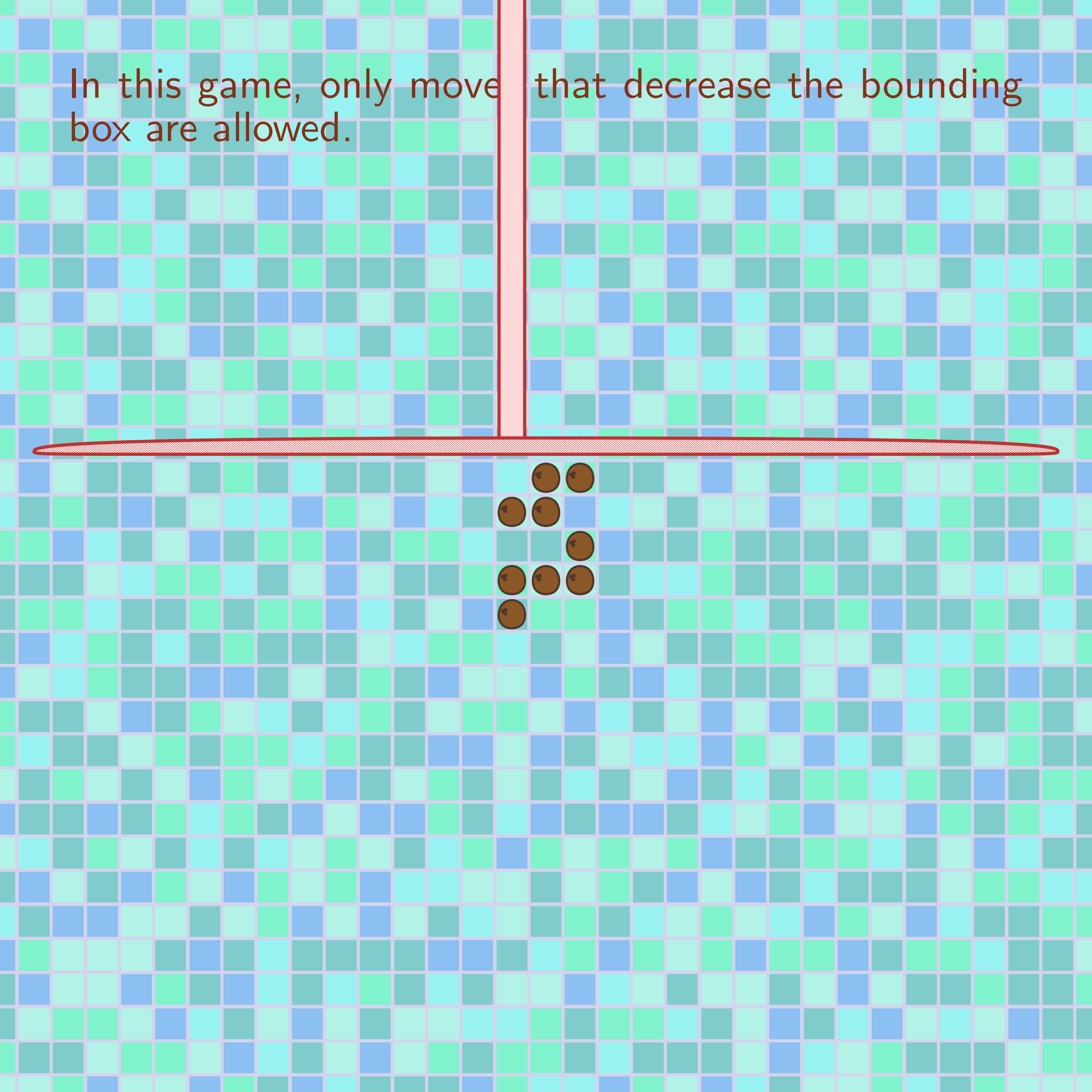
In this game, only moves that decrease the bounding box are allowed.



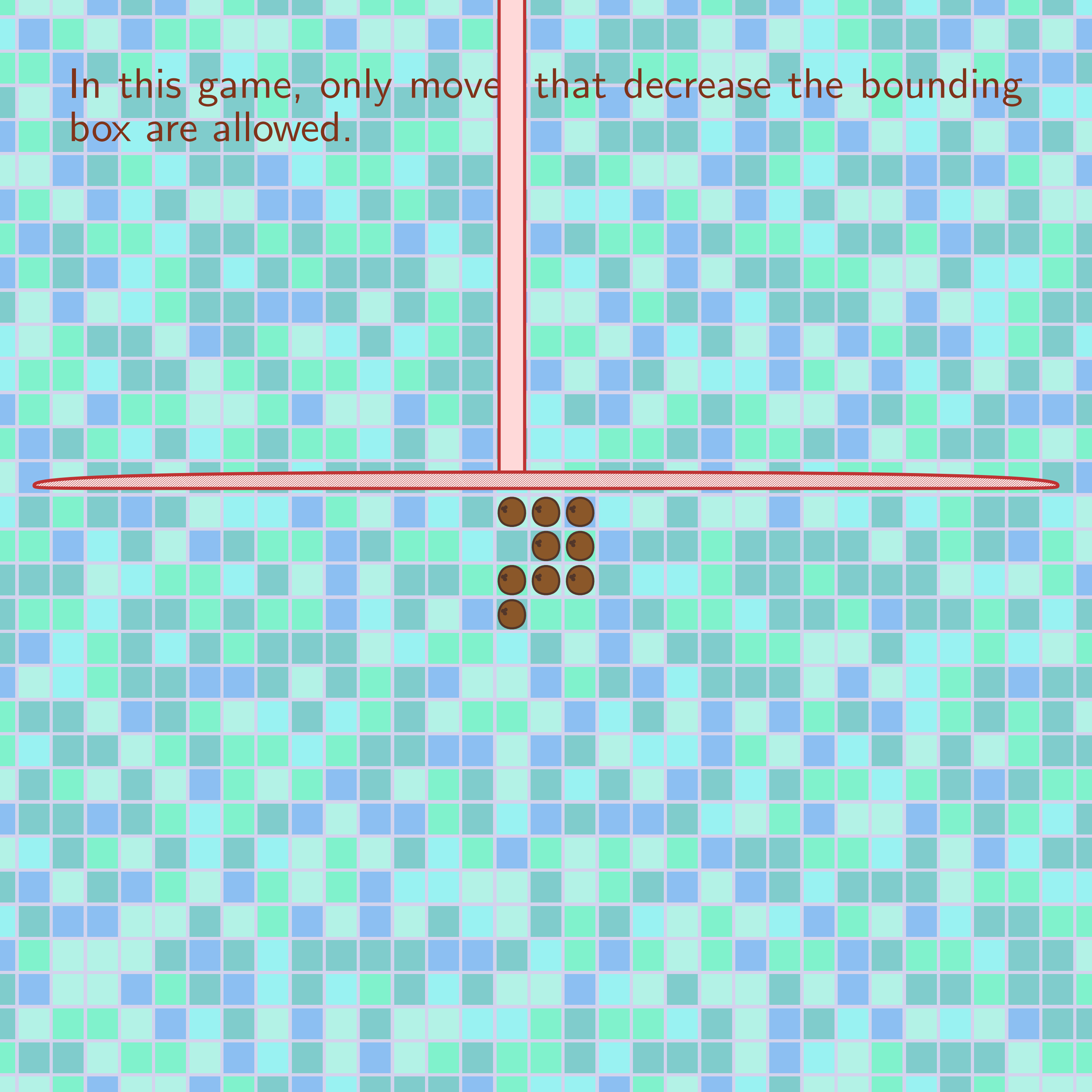
In this game, only moves that decrease the bounding box are allowed.



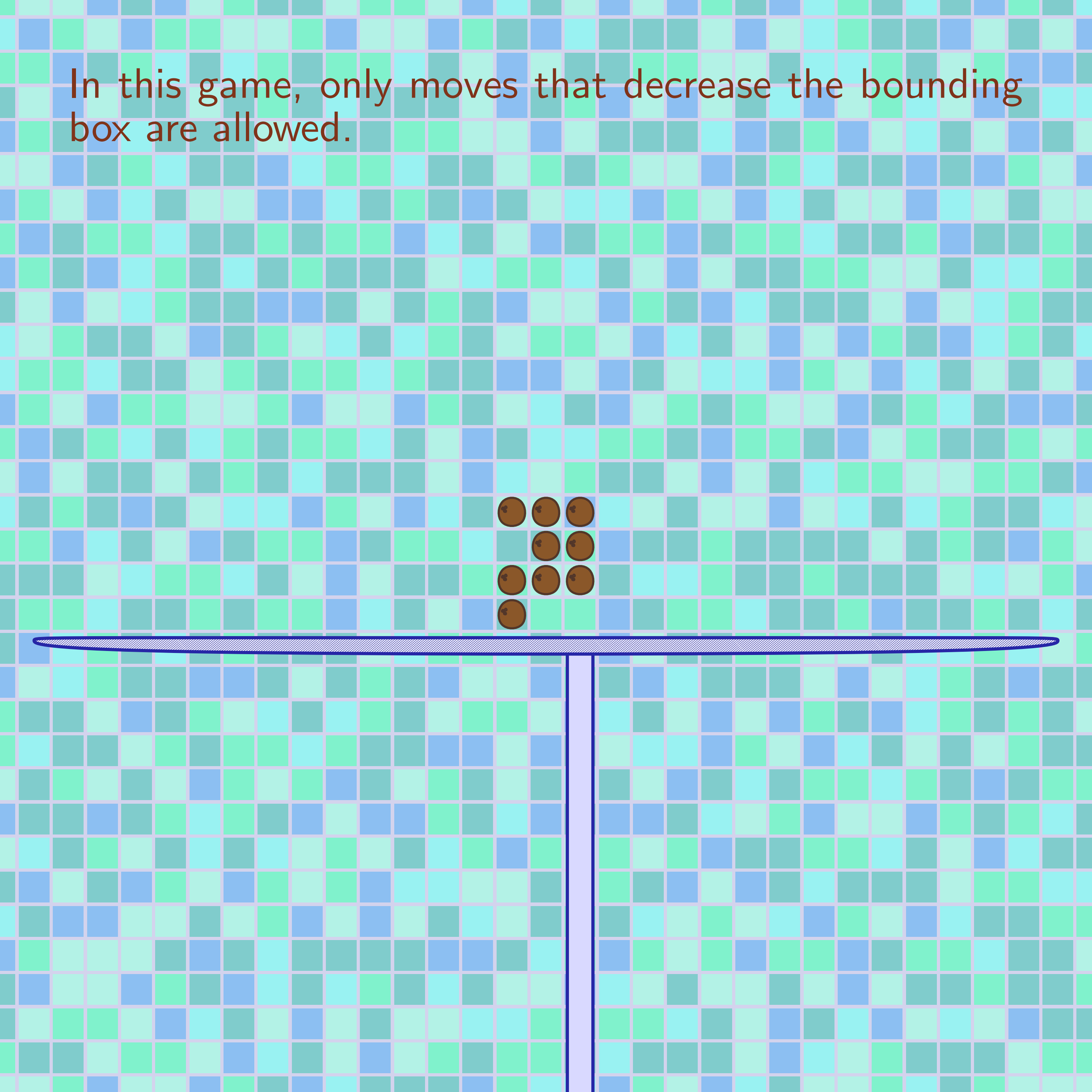
In this game, only moves that decrease the bounding box are allowed.



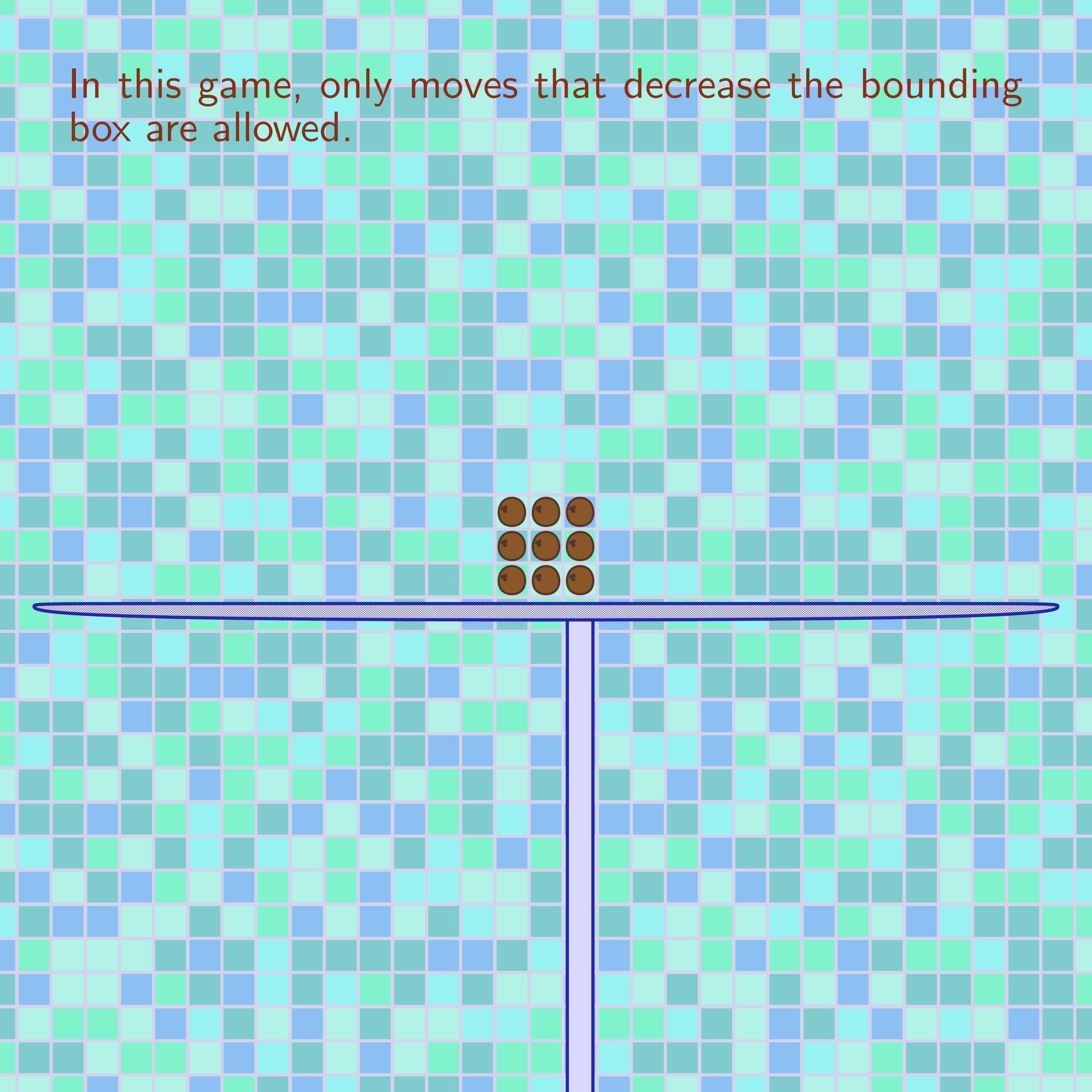
In this game, only moves that decrease the bounding box are allowed.



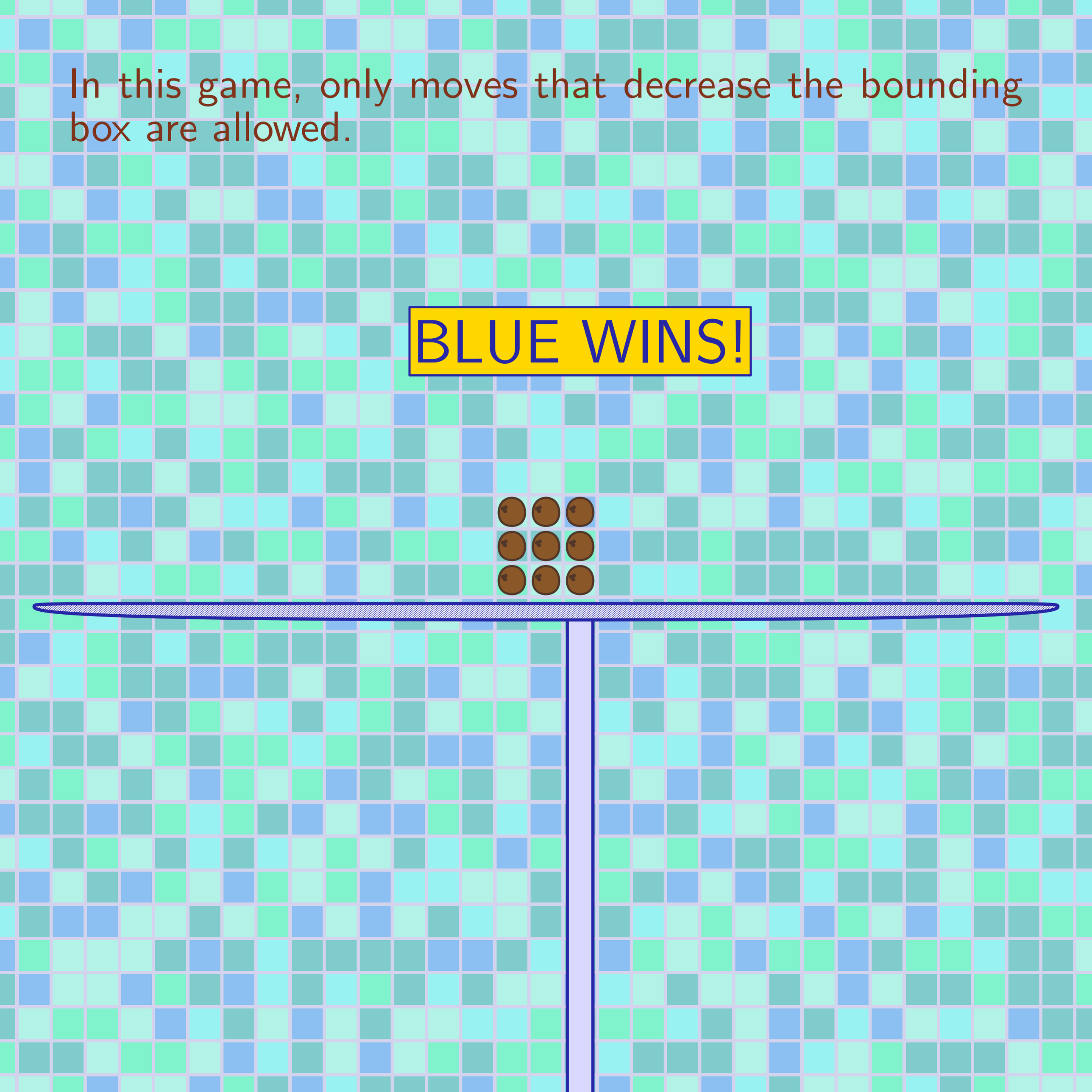
In this game, only moves that decrease the bounding box are allowed.



In this game, only moves that decrease the bounding box are allowed.



In this game, only moves that decrease the bounding box are allowed.



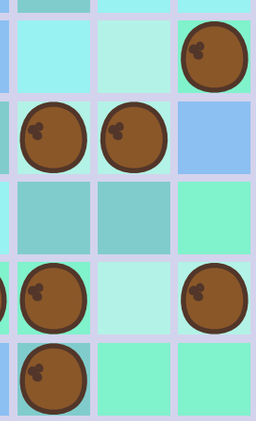
This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.

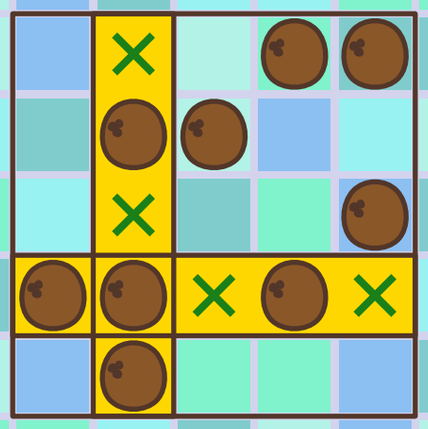
This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.



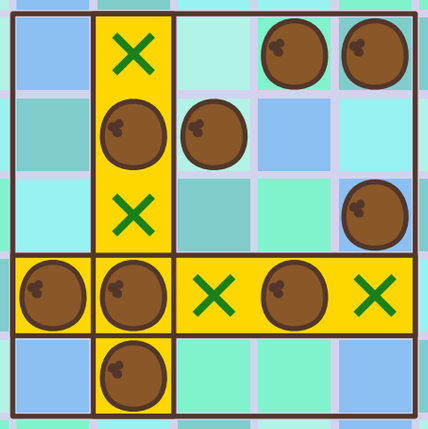
This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.



This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

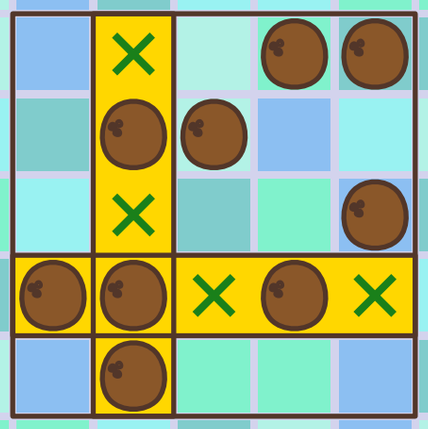
We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.



Observation
Every move decreases the potential by at least 1.

This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.

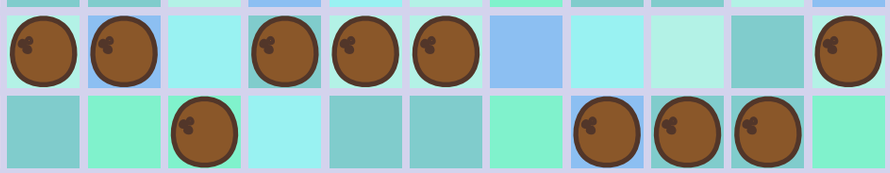


Observation
Every move decreases the potential by at least 1.

But it can decrease arbitrarily!

This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.

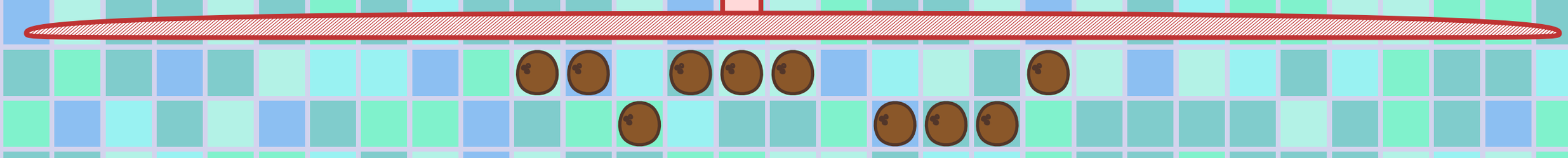


Observation
Every move decreases the potential by at least 1.

But it can decrease arbitrarily!

This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.

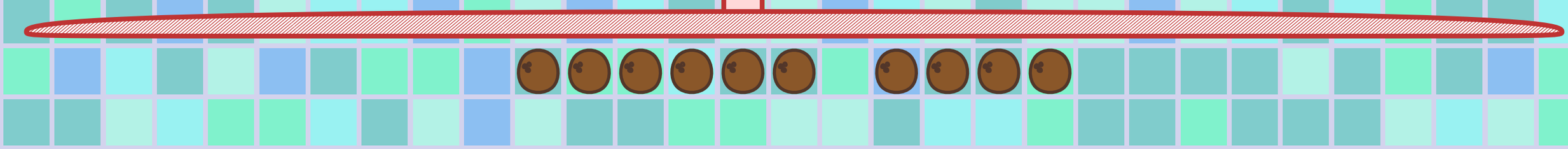


Observation
Every move decreases the potential by at least 1.

But it can decrease arbitrarily!

This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.

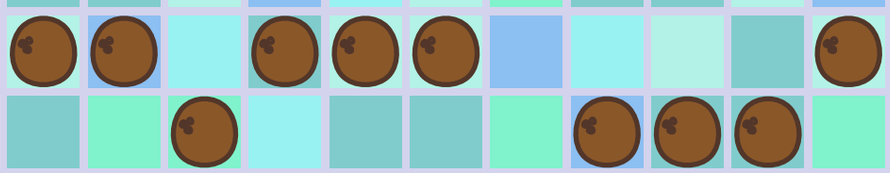


Observation
Every move decreases the potential by at least 1.

But it can decrease arbitrarily!

This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.

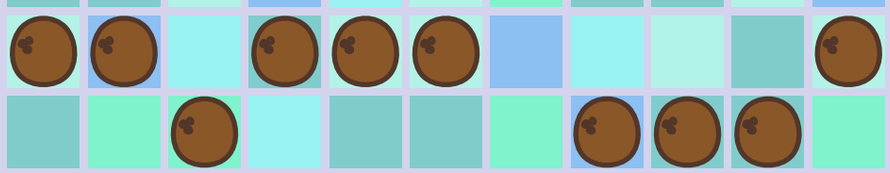


Observation
Every move decreases the potential by at least 1.

But it can decrease arbitrarily!

This compaction game is *impartial*: in any configuration, the set of legal moves does not depend on which player's turn it is.

We define the *potential* of a configuration as the minimum number of empty cells in any row plus the minimum number of empty cells in any column.



Observation
Every move decreases the potential by at least 1.

But it can decrease arbitrarily!

Controlling the potential is non-trivial.

Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

We did some experiments on random inputs.

Idea

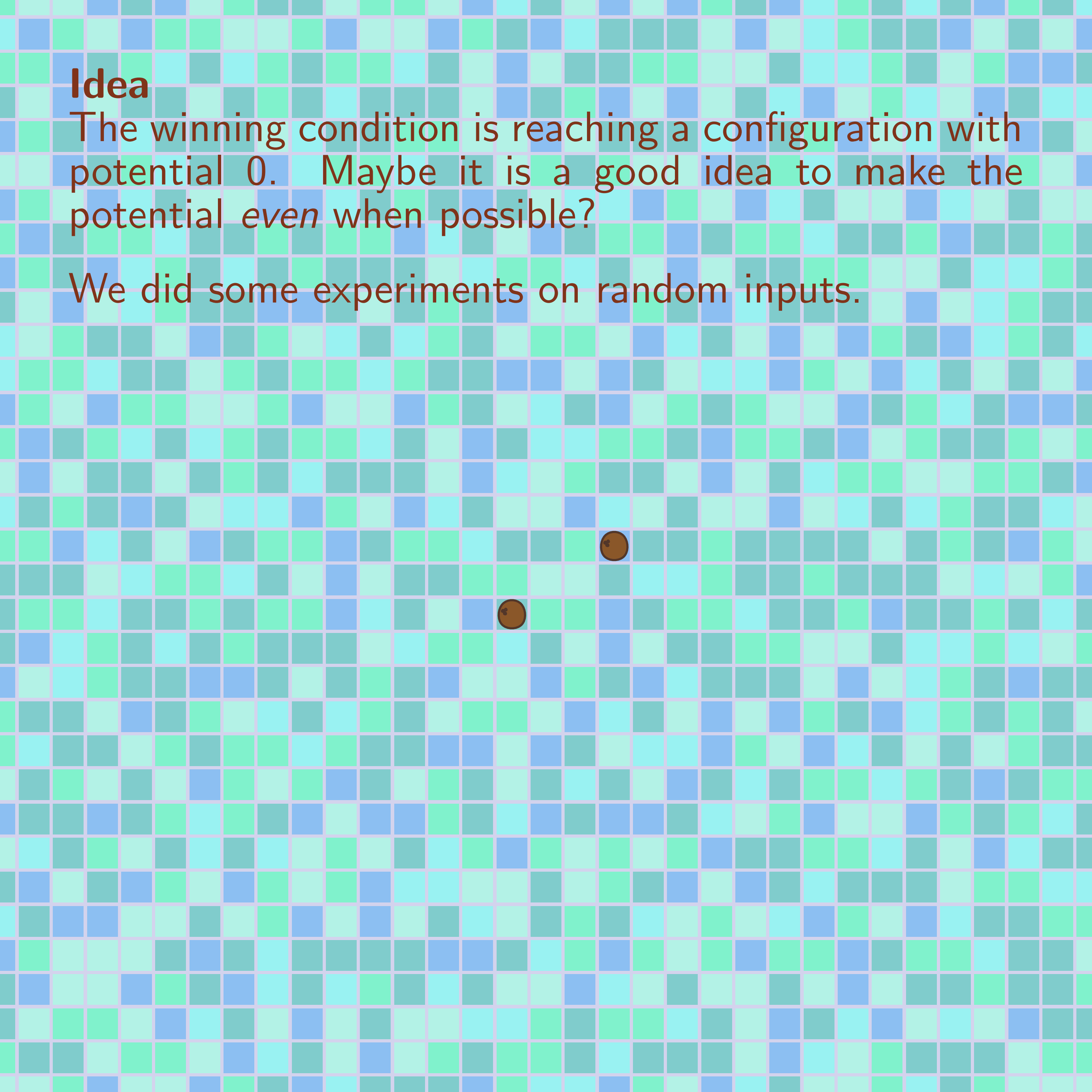
The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

We did some experiments on random inputs.

Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

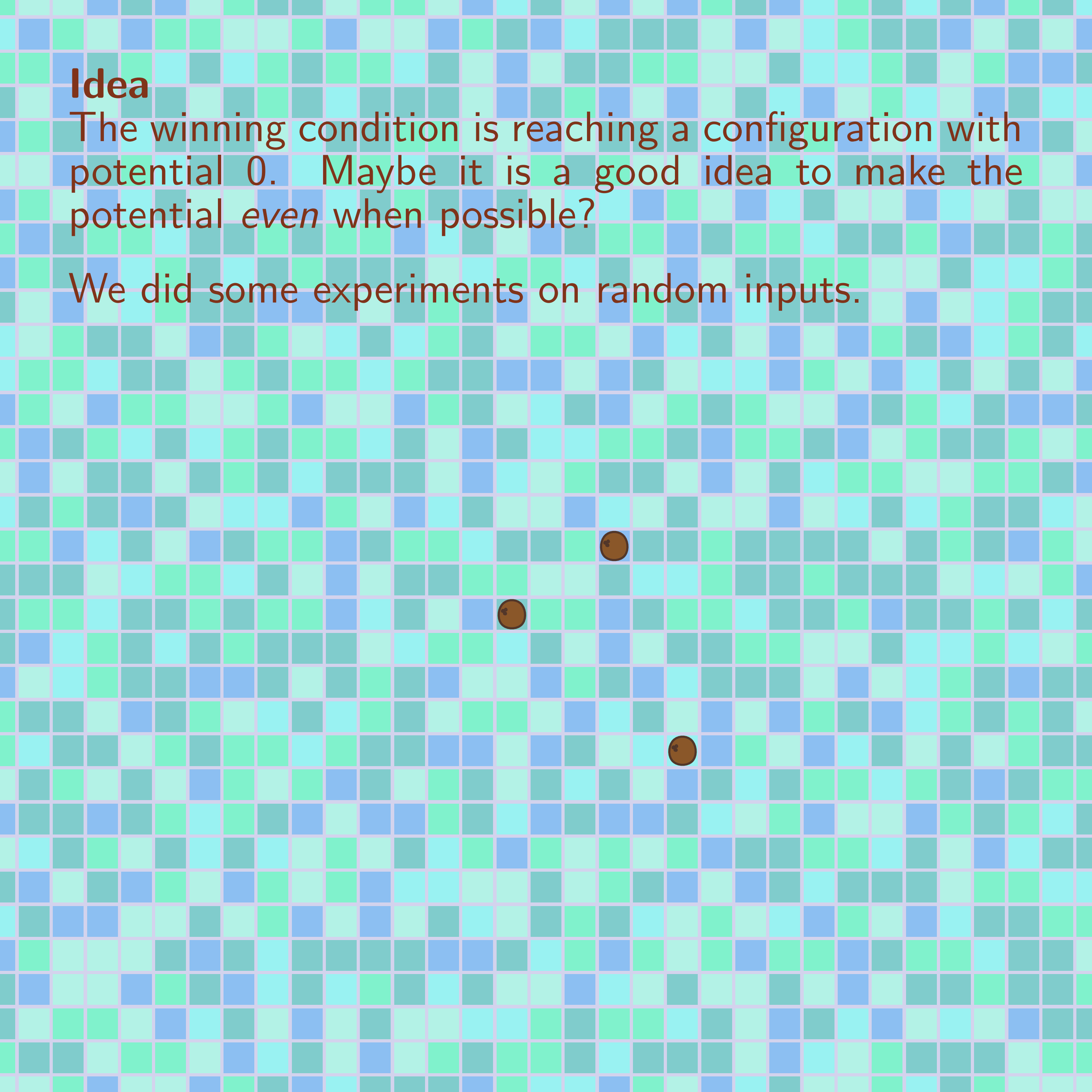
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

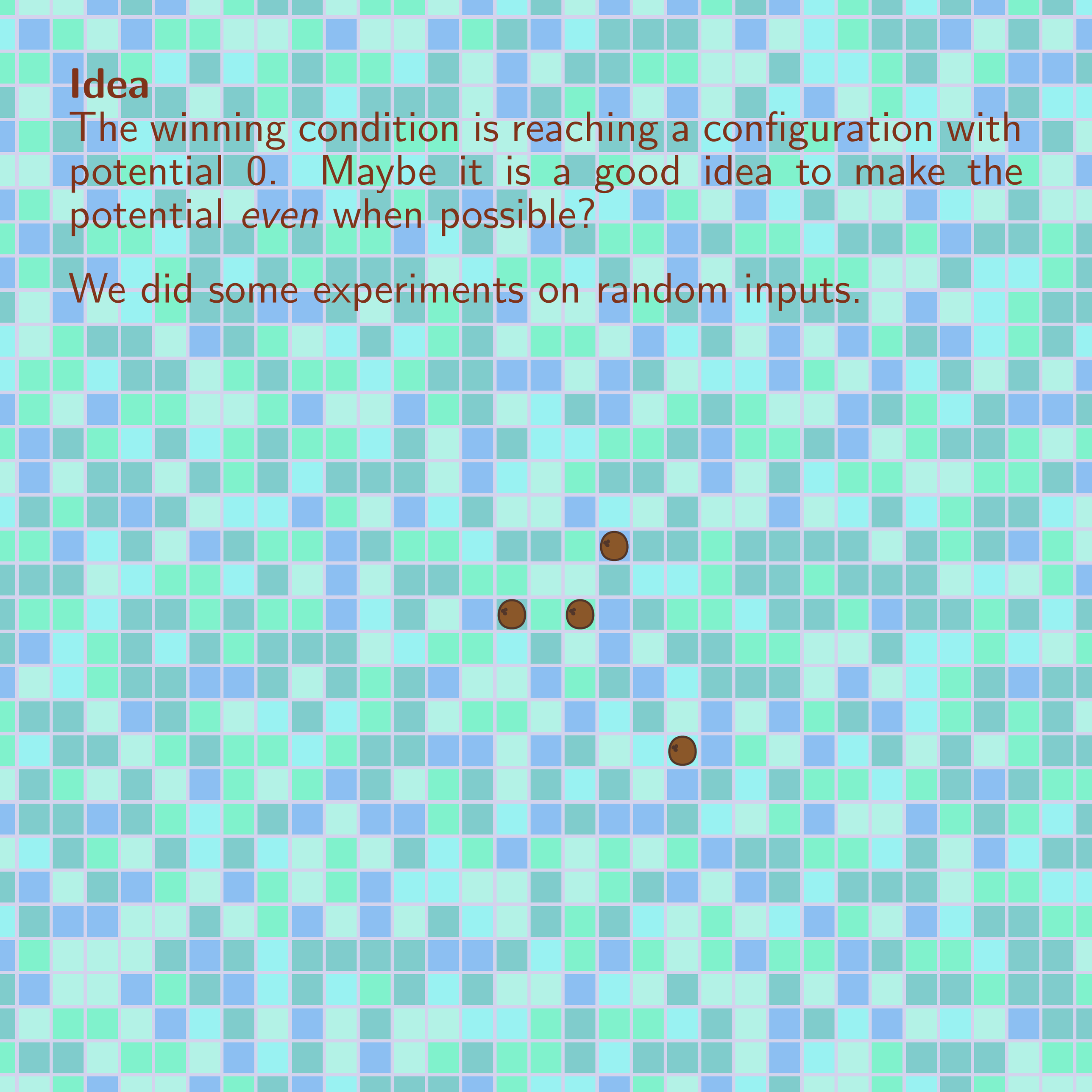
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

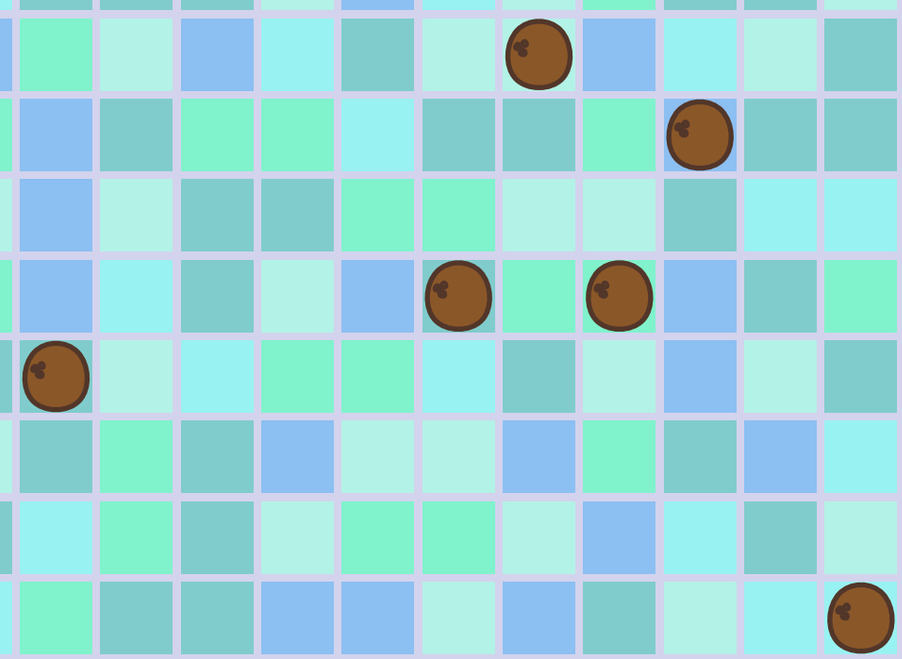
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

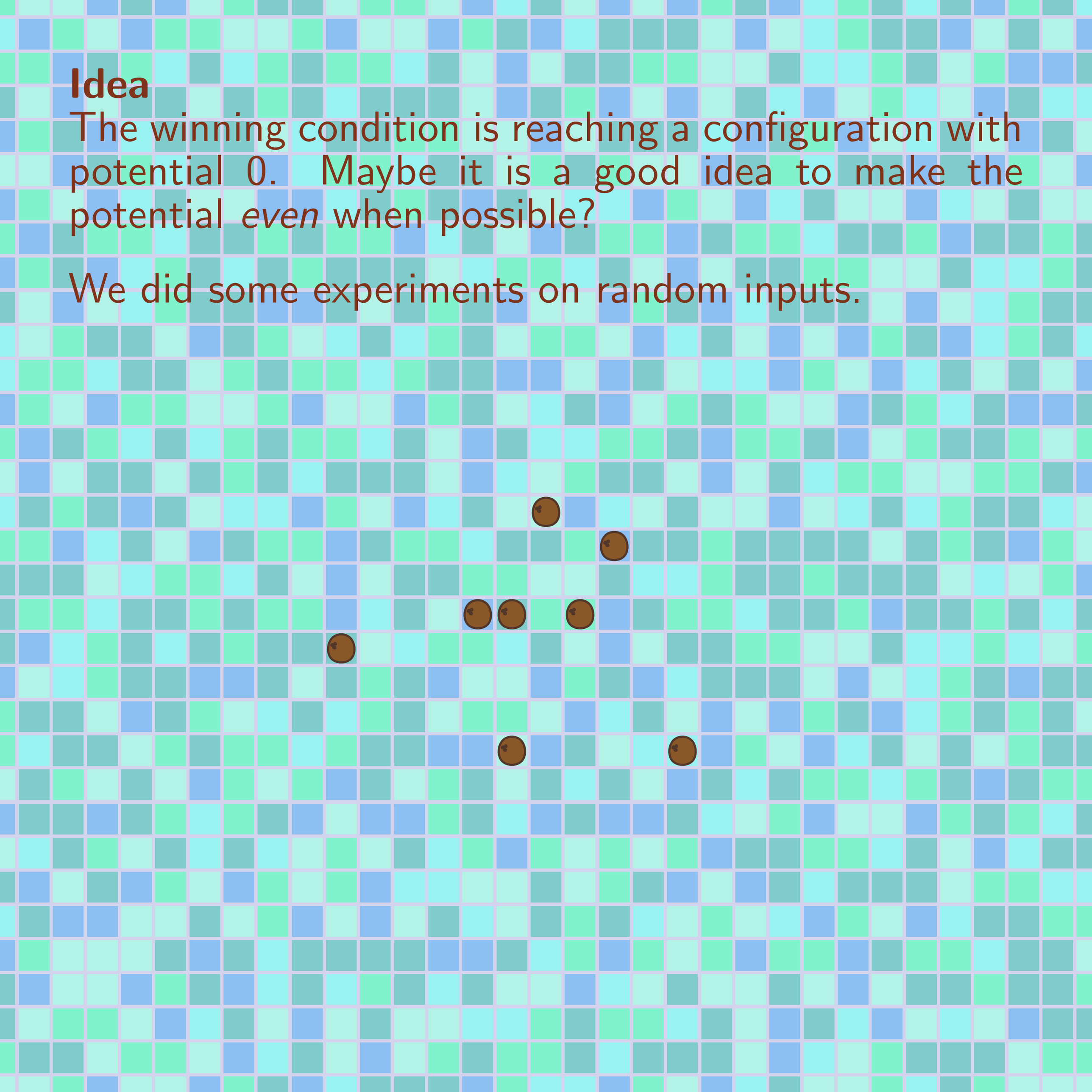
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

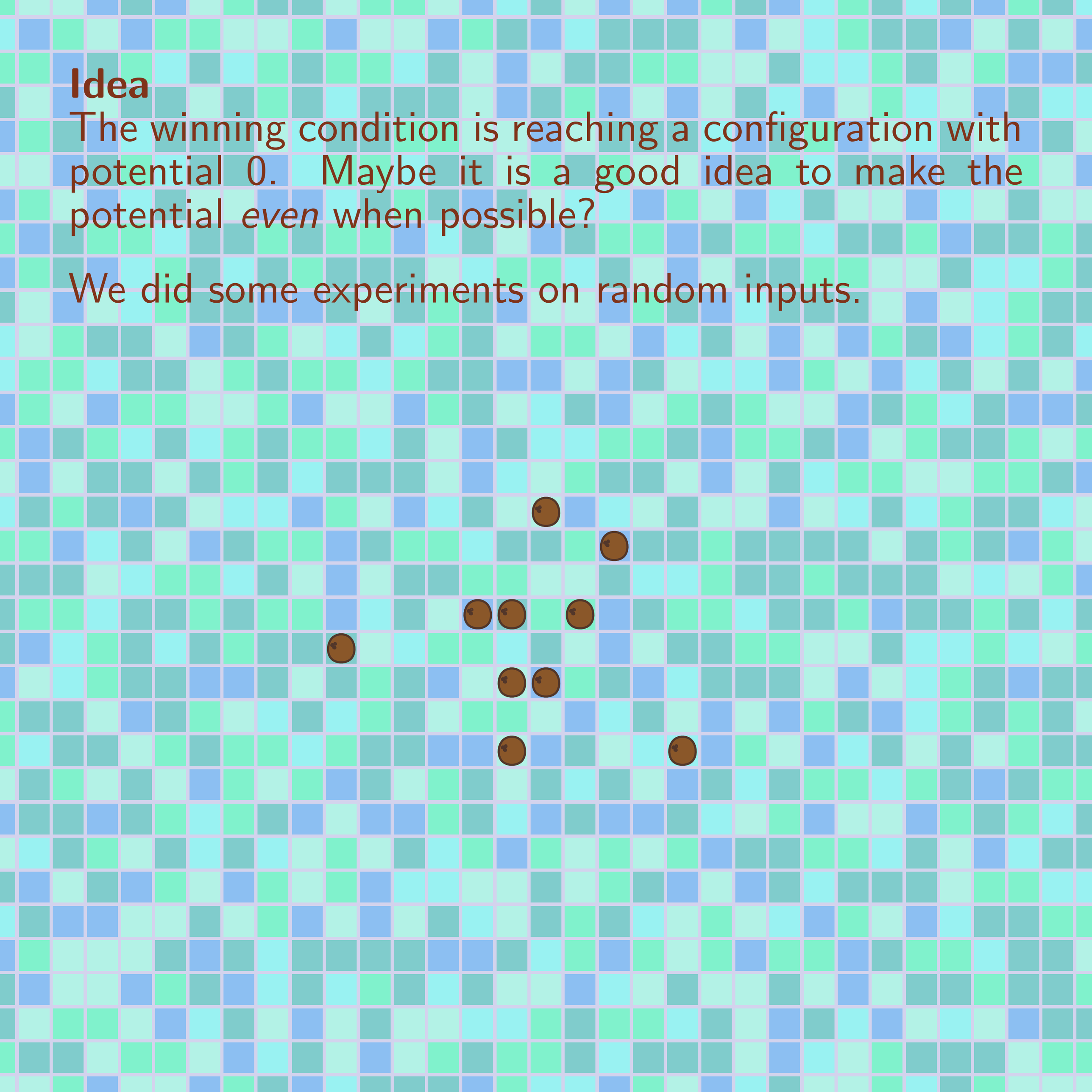
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

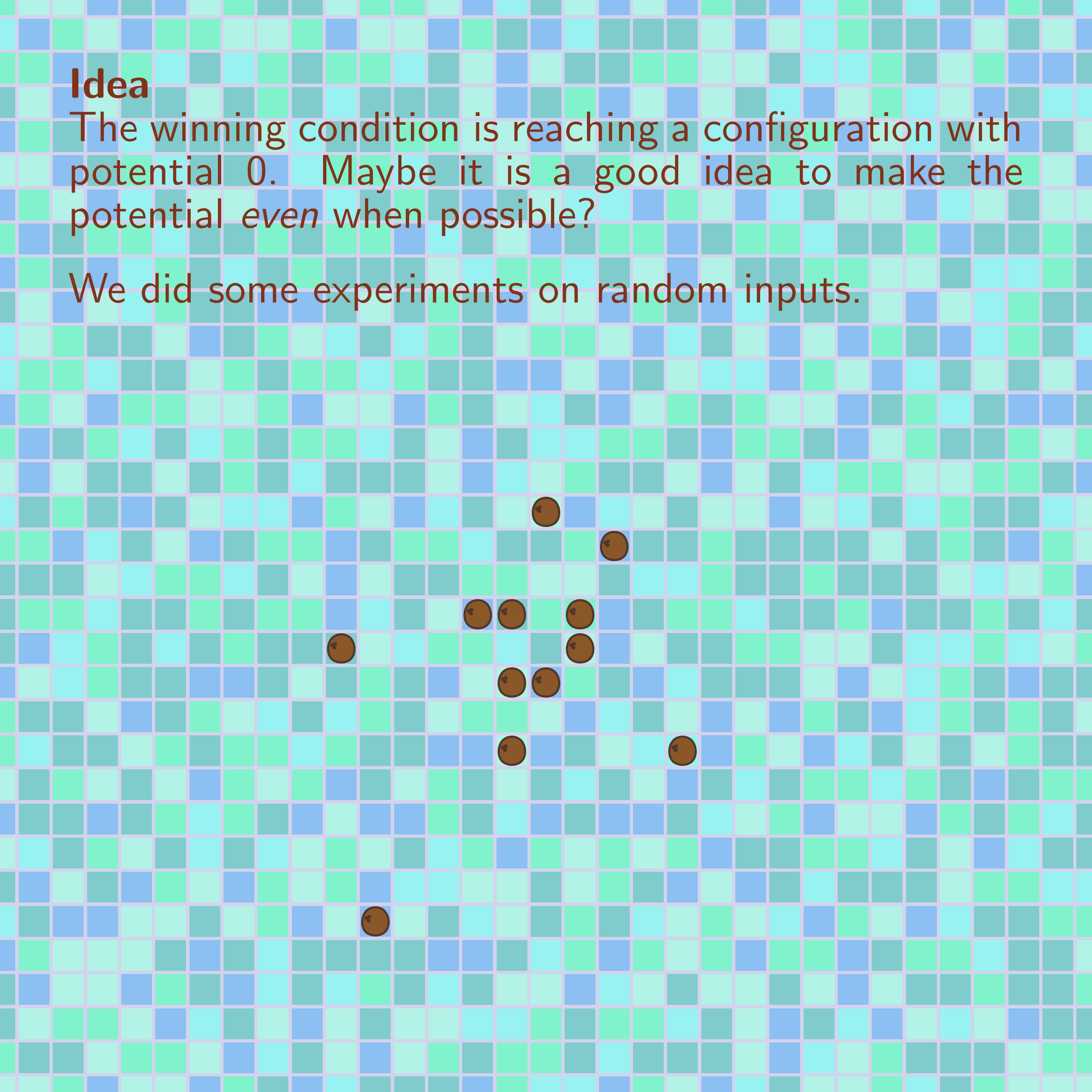
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

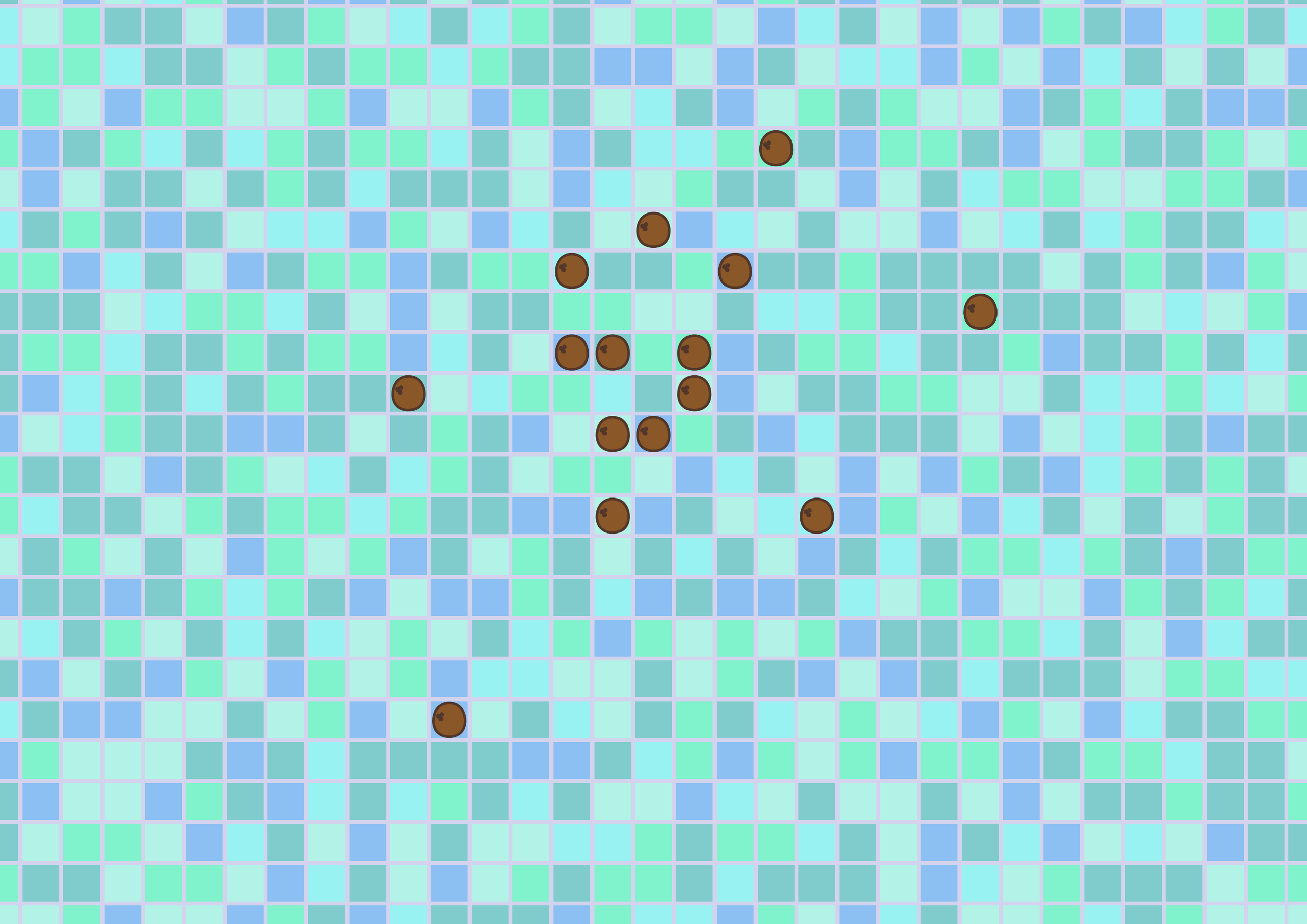
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

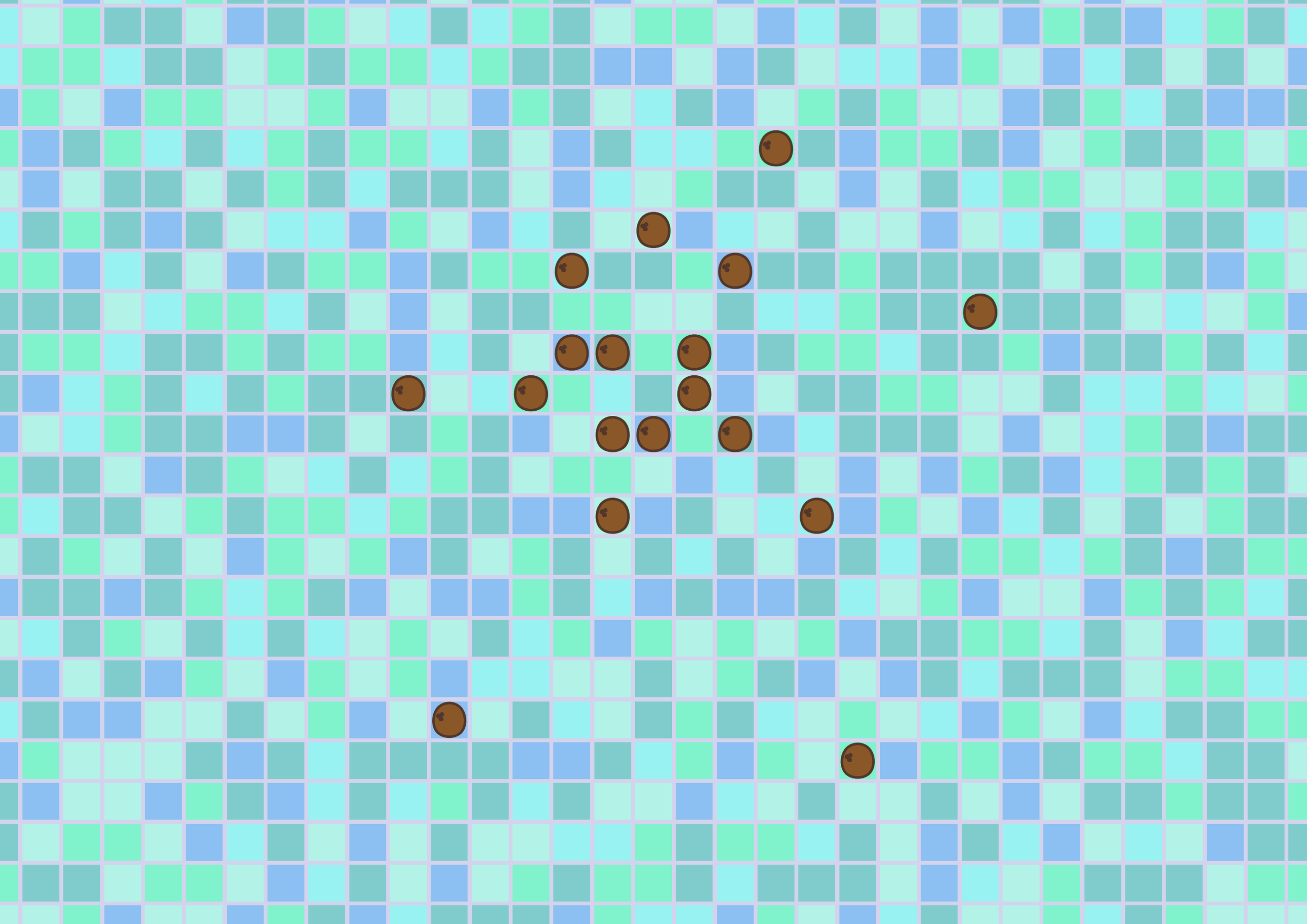
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

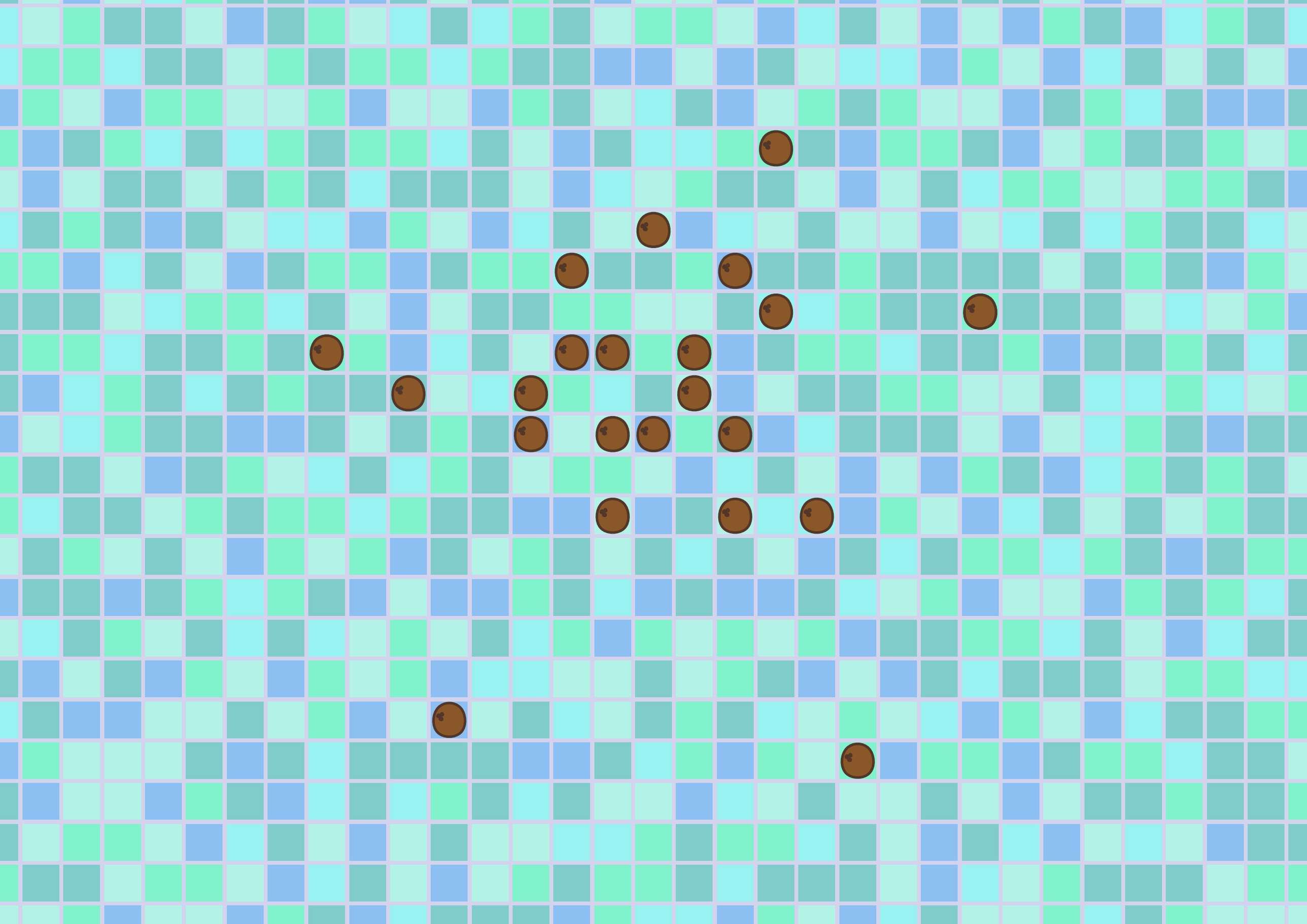
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

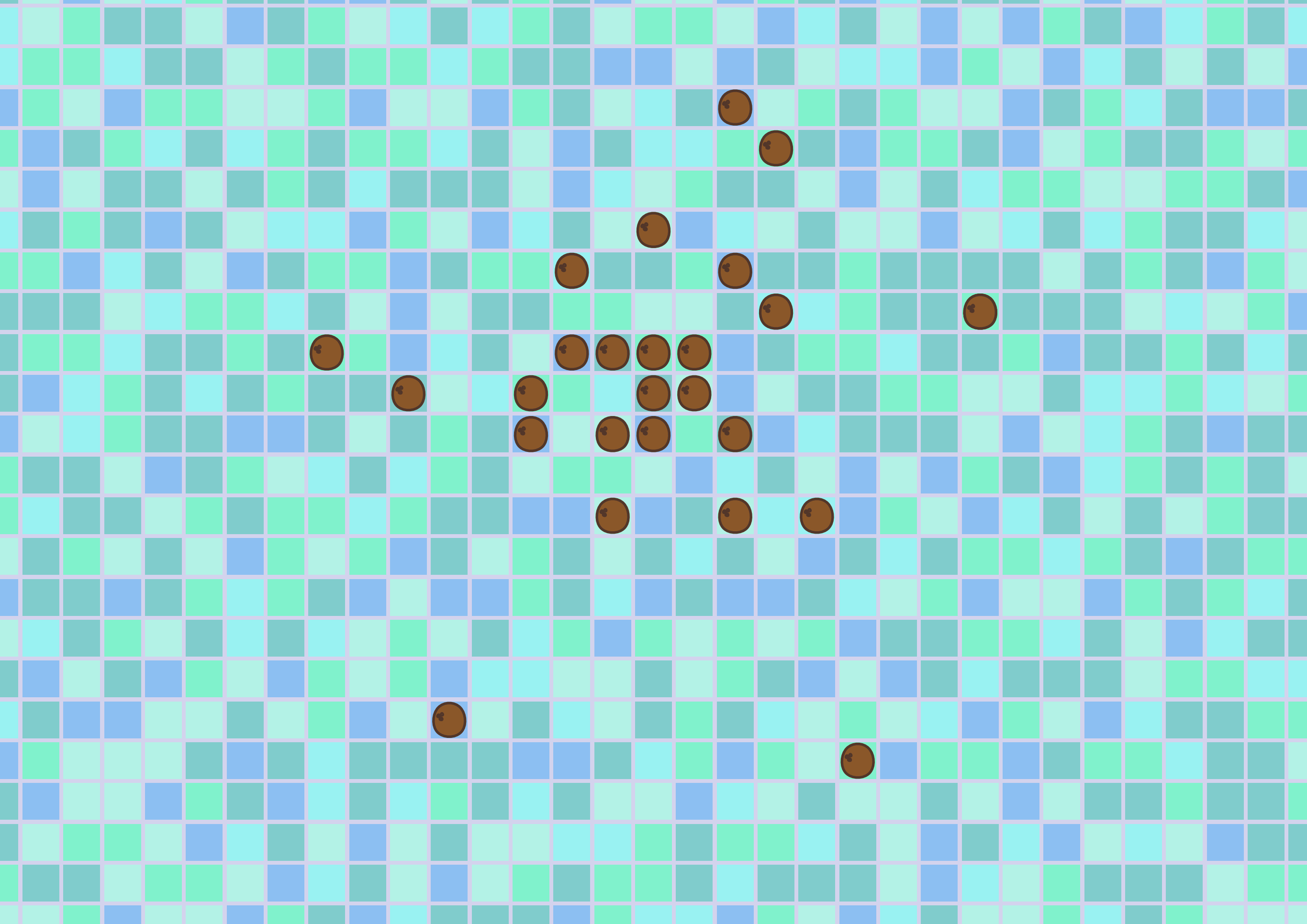
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

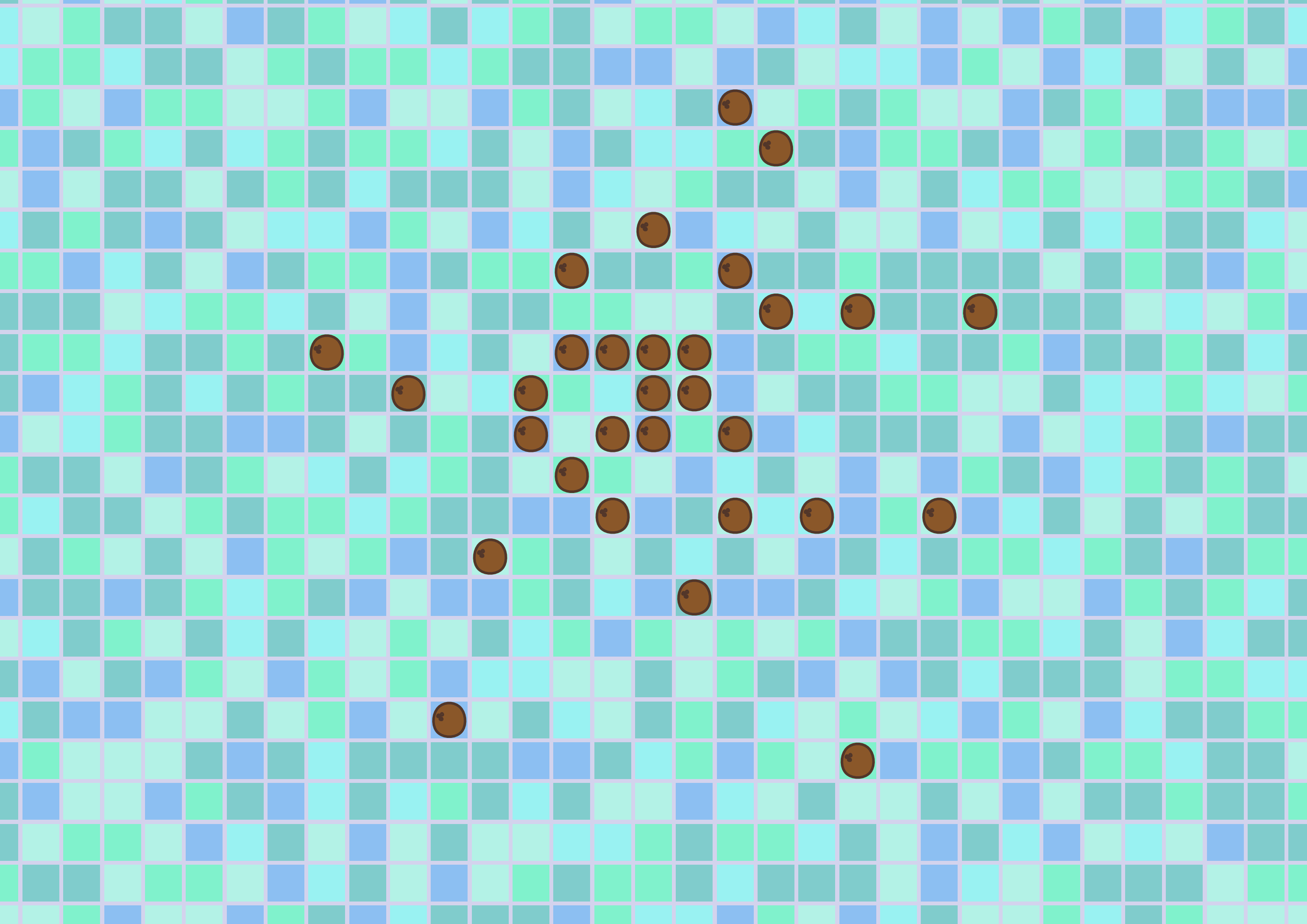
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

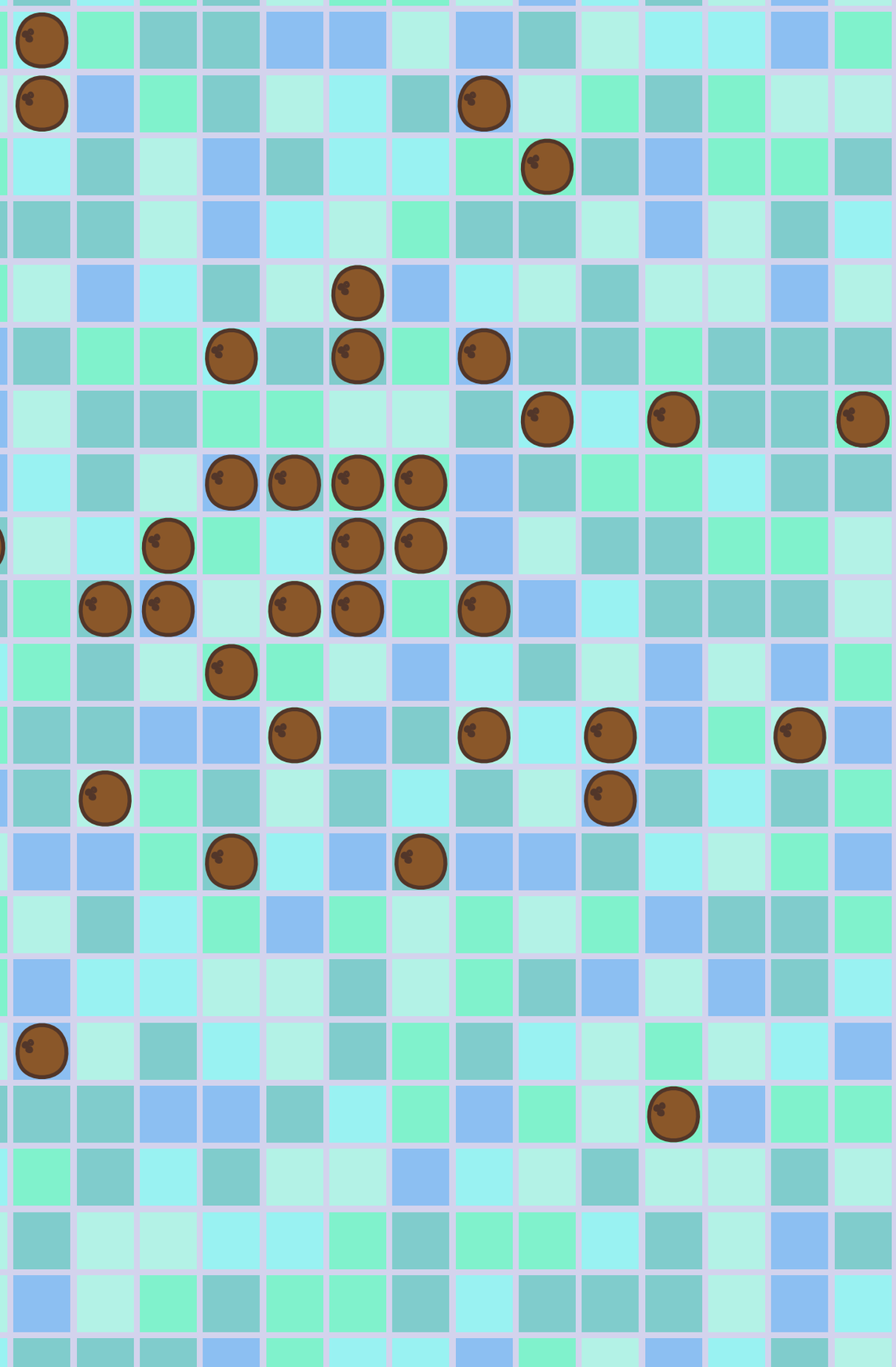
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

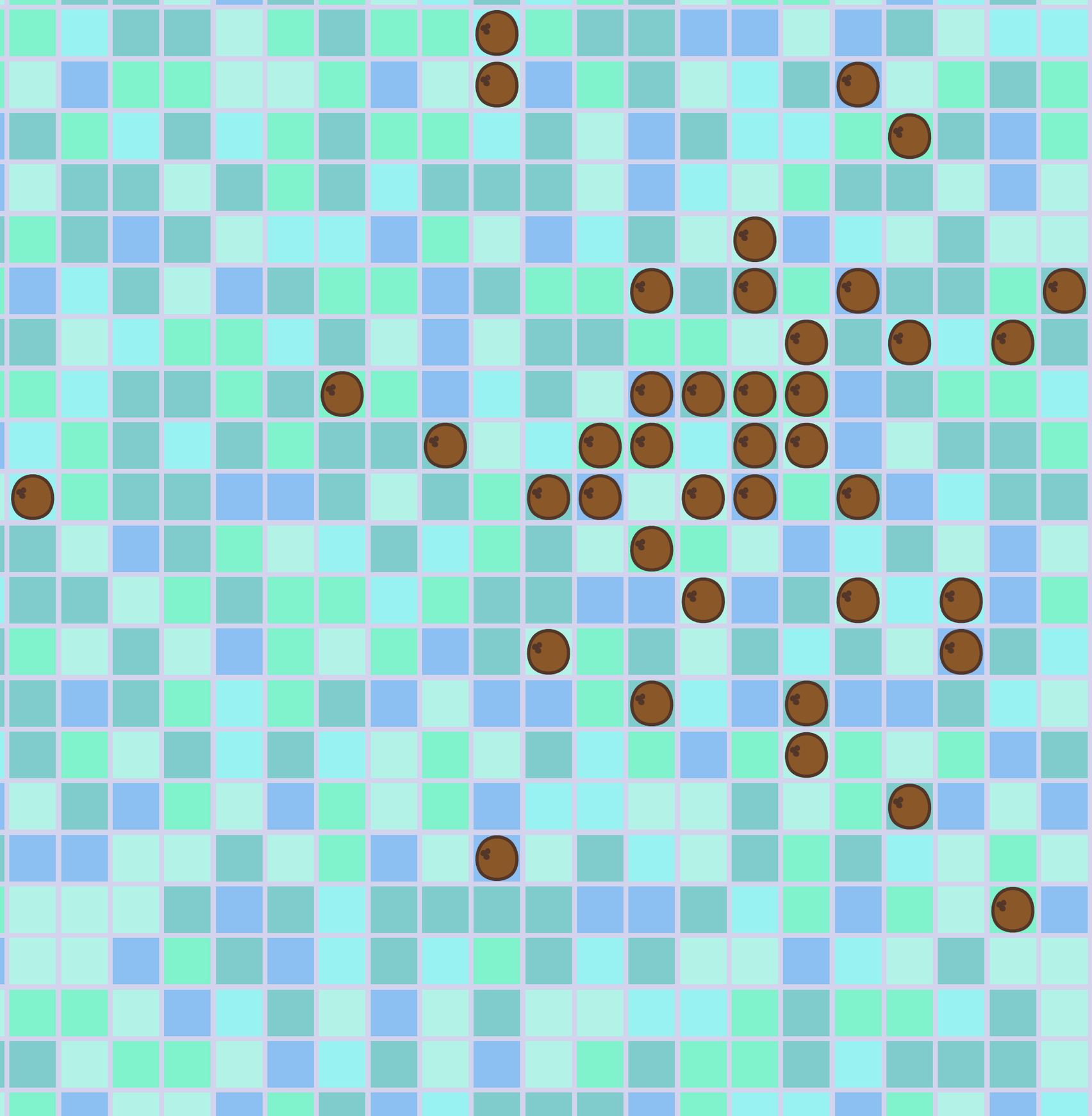
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

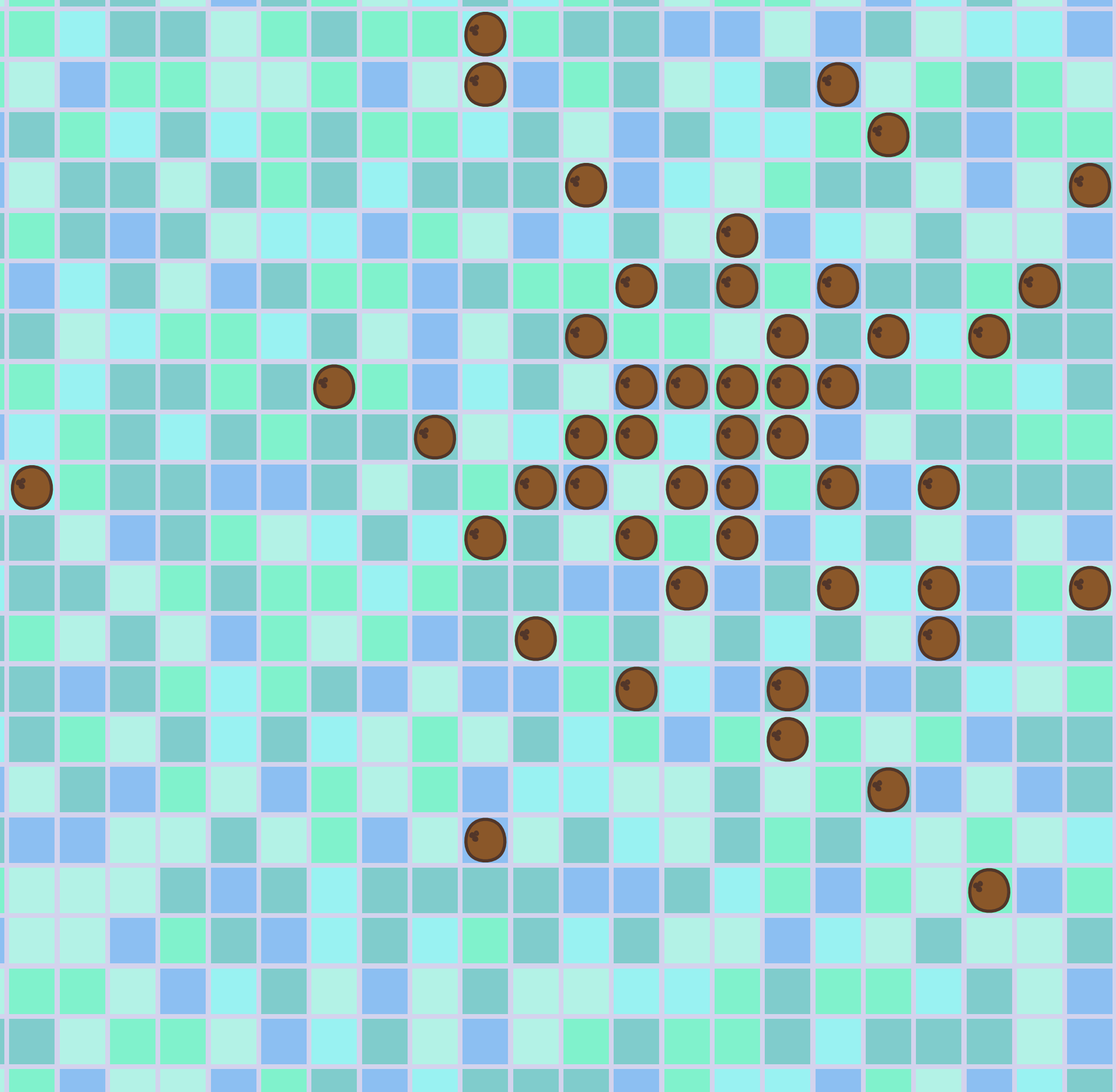
We did some experiments on random inputs.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

We did some experiments on random inputs.

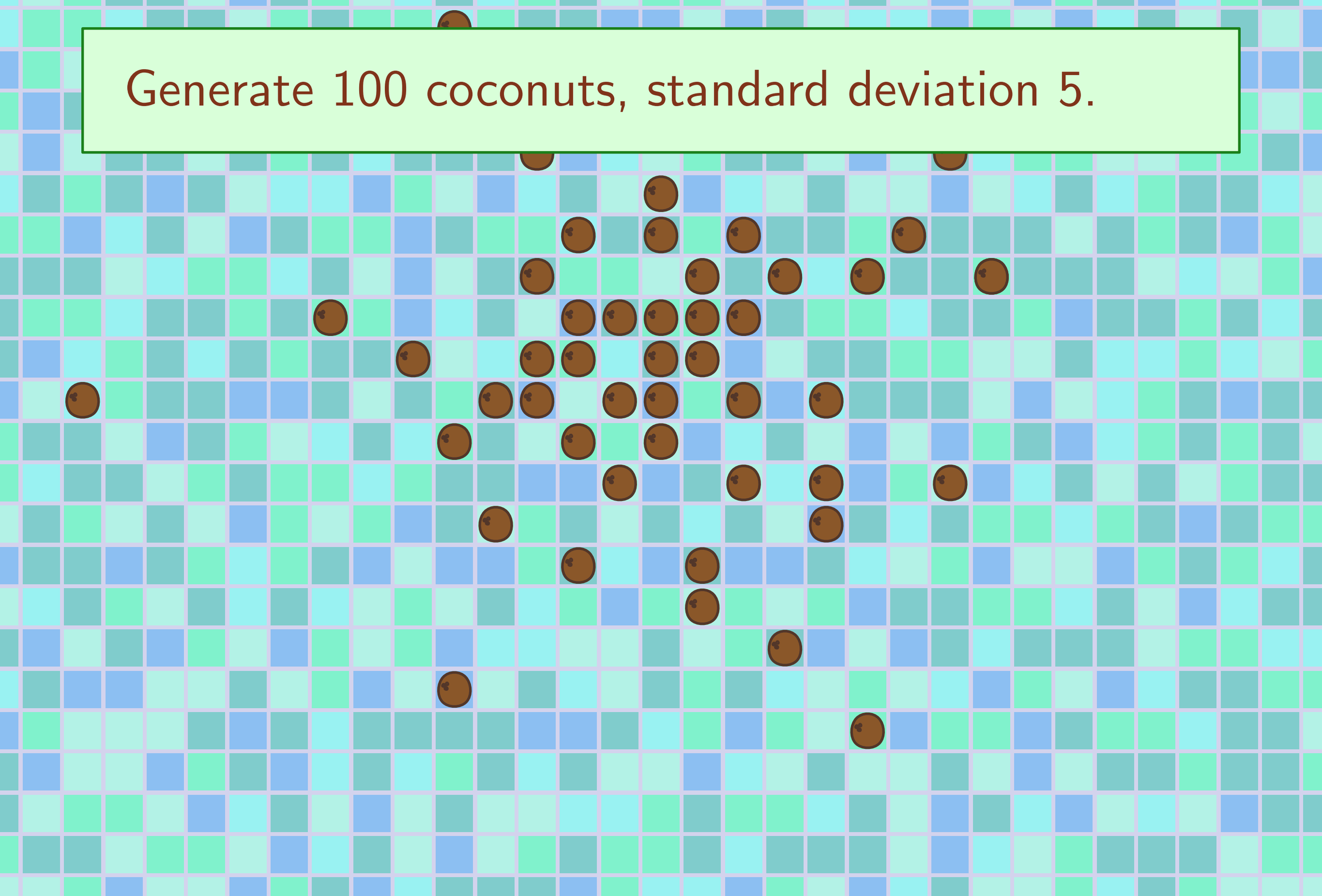


Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

We did some experiments on random inputs.

Generate 100 coconuts, standard deviation 5.



Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

We did some experiments on random inputs.

Generate 100 coconuts, standard deviation 5.

Player 1

Make potential even when possible. Otherwise random.

Player 2

Just play randomly.

Idea

The winning condition is reaching a configuration with potential 0. Maybe it is a good idea to make the potential *even* when possible?

We did some experiments on random inputs.

Generate 100 coconuts, standard deviation 5.

Player 1

Make potential even when possible. Otherwise random.

Player 2

Just play randomly.

Player 1 wins $\approx 70\%$ of the games!

Open Question

Is there a better measure of the state of the game than the potential?

Open Question
Is there a better measure of the state of the game than the potential?

Open Question
Is LAST MOVE WINS *hard?*

GAME 2

K IN A ROW

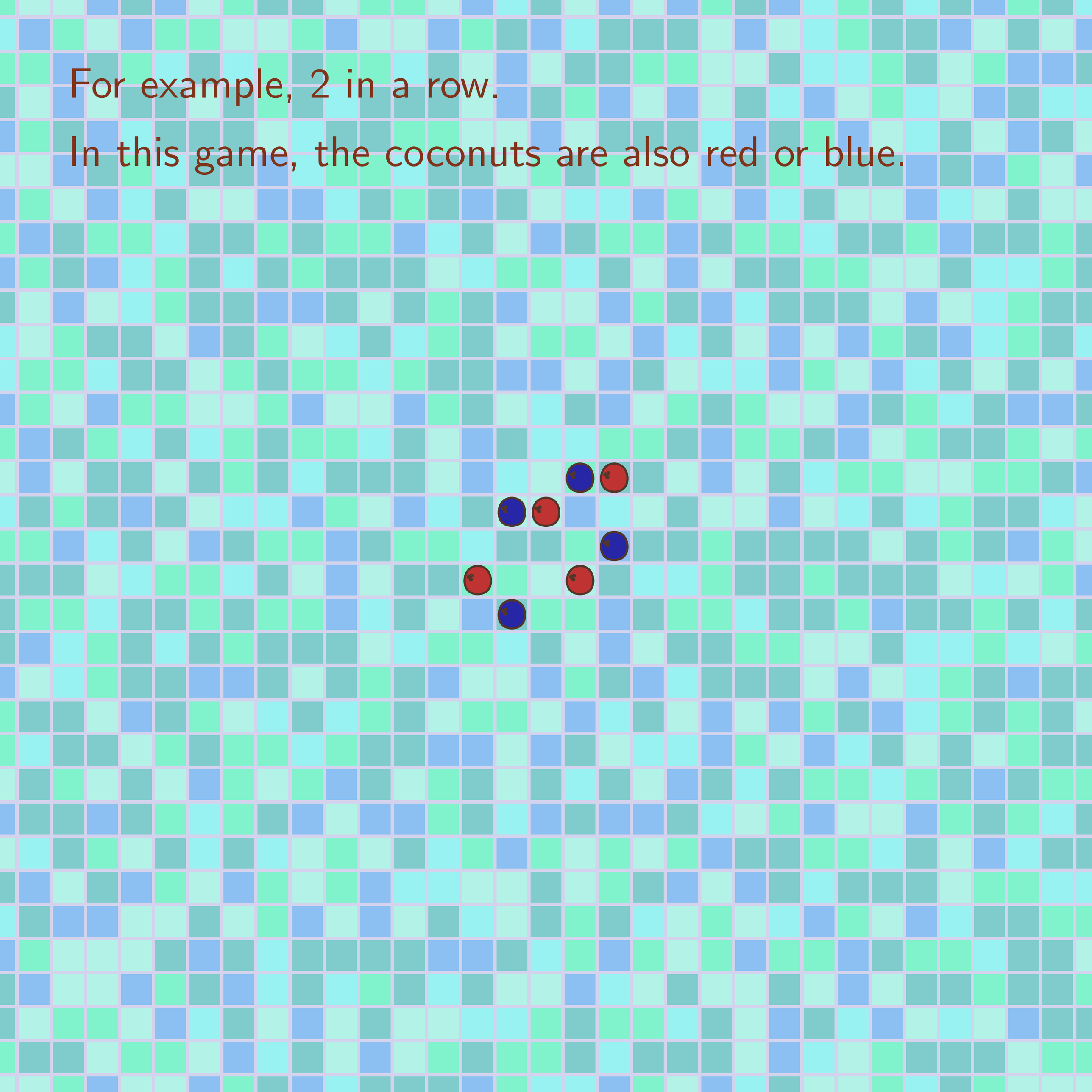
For example, 2 in a row.

For example, 2 in a row.

In this game, the coconuts are also red or blue.

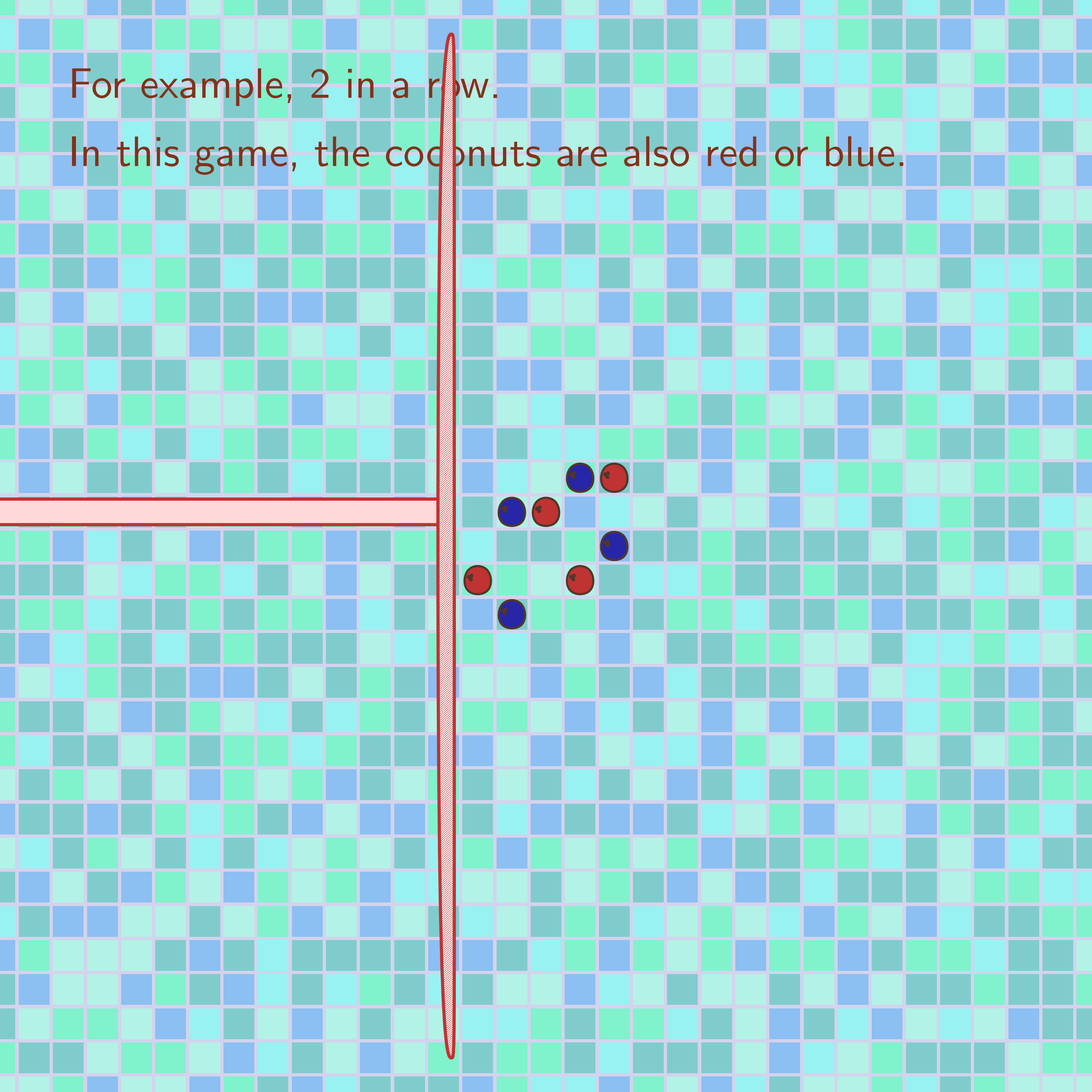
For example, 2 in a row.

In this game, the coconuts are also red or blue.



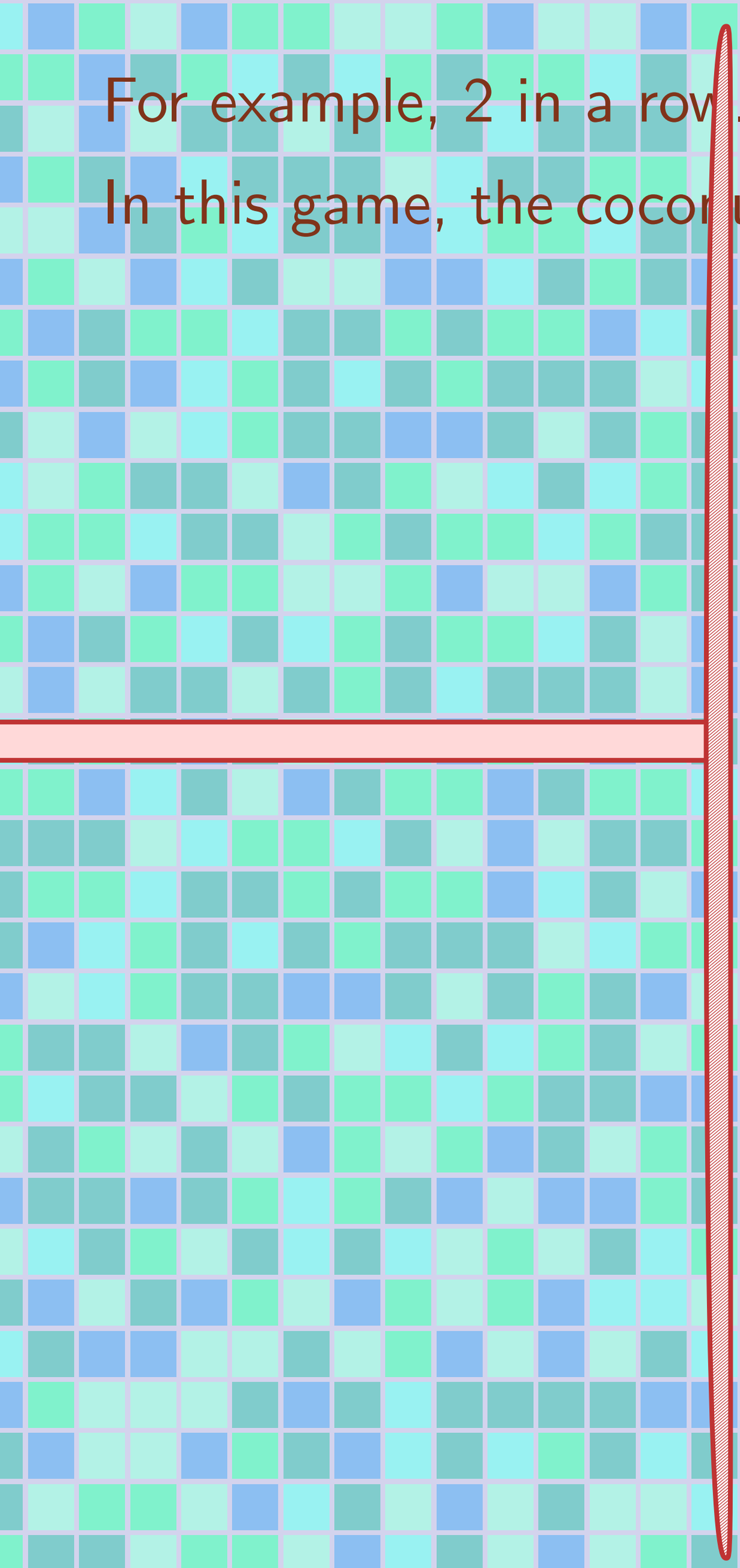
For example, 2 in a row.

In this game, the coconuts are also red or blue.



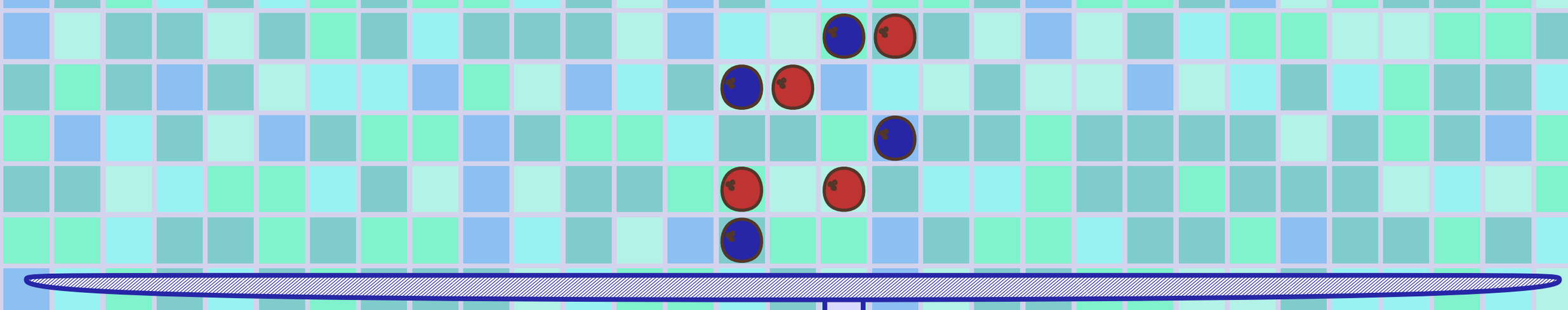
For example, 2 in a row.

In this game, the coconuts are also red or blue.



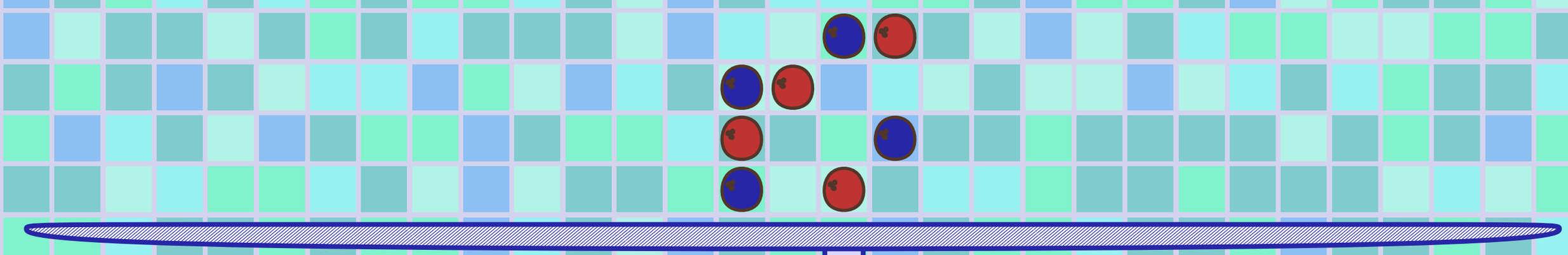
For example, 2 in a row.

In this game, the coconuts are also red or blue.



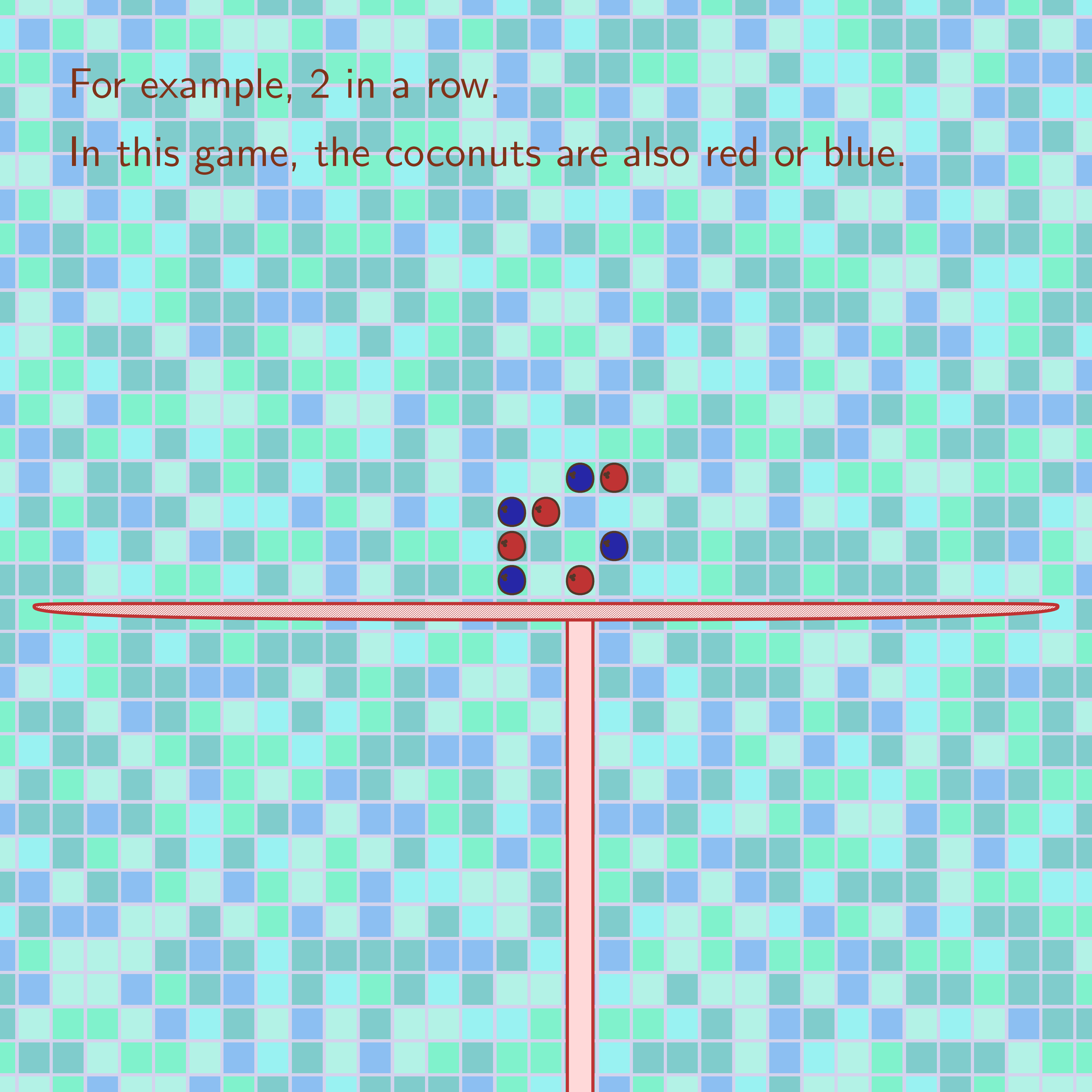
For example, 2 in a row.

In this game, the coconuts are also red or blue.



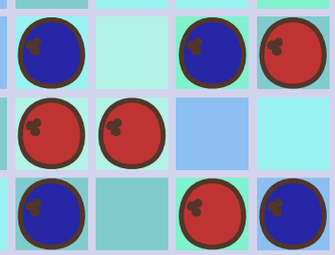
For example, 2 in a row.

In this game, the coconuts are also red or blue.



For example, 2 in a row.

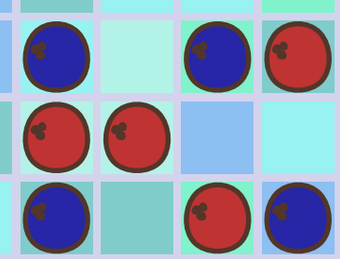
In this game, the coconuts are also red or blue.



For example, 2 in a row.

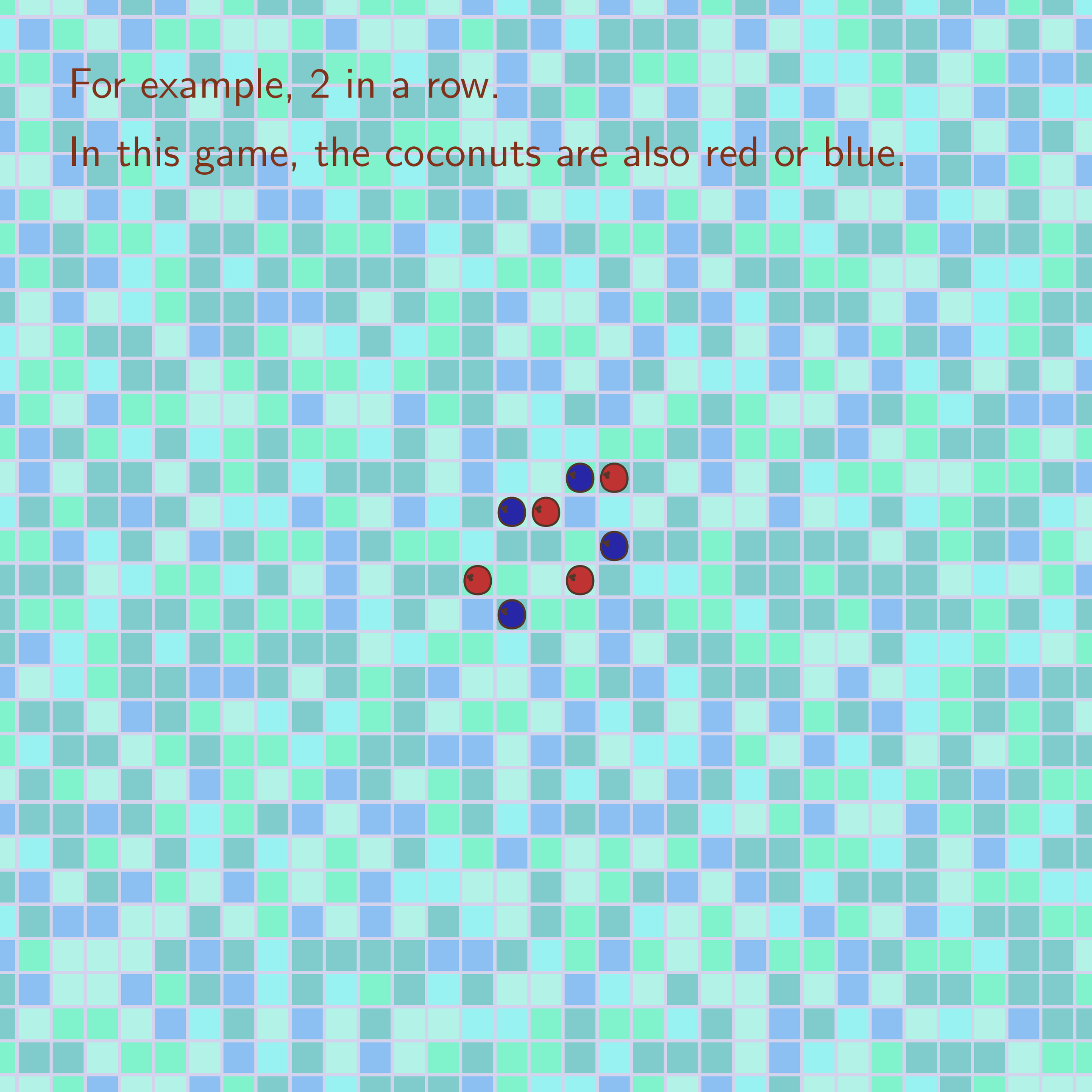
In this game, the coconuts are also red or blue.

RED WINS!



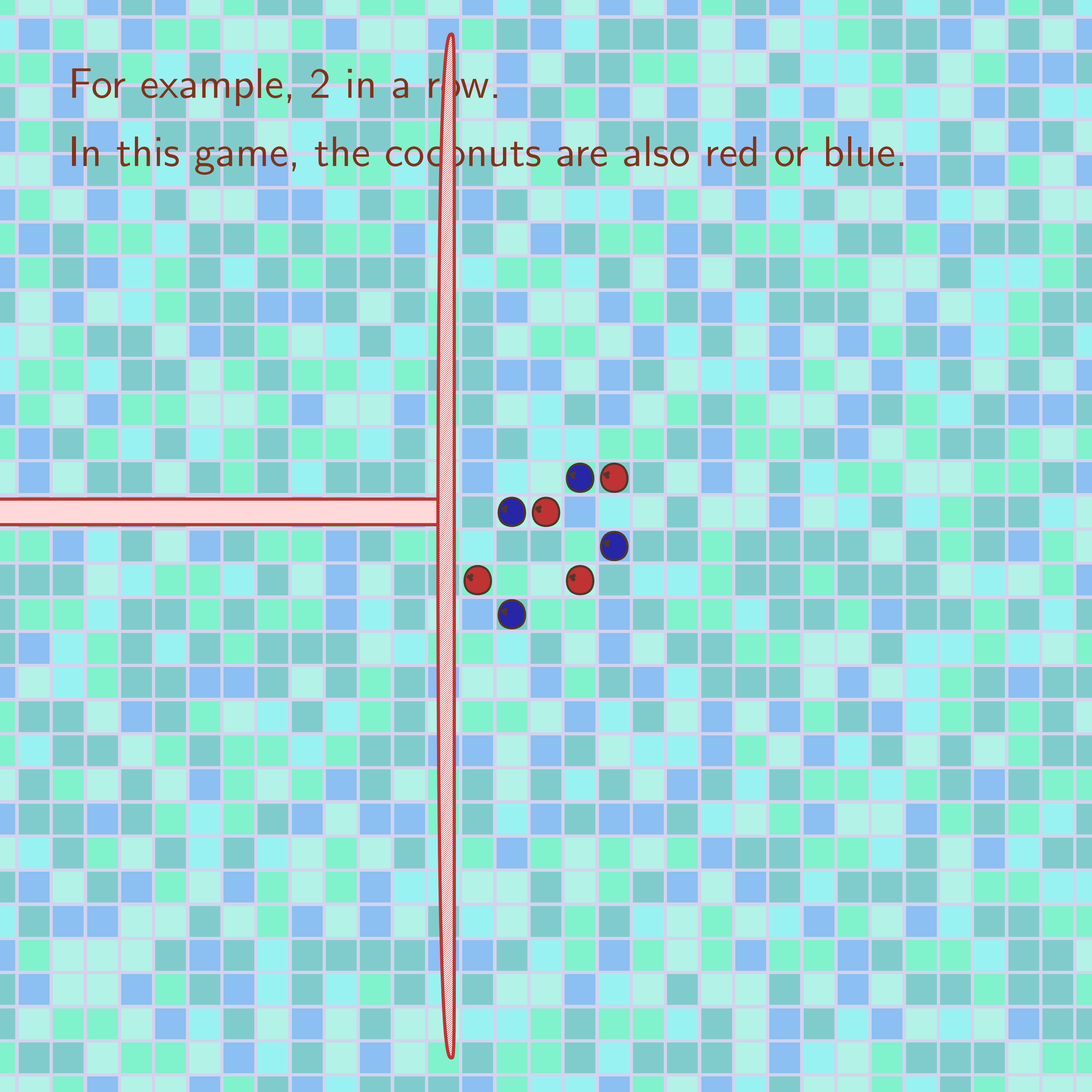
For example, 2 in a row.

In this game, the coconuts are also red or blue.



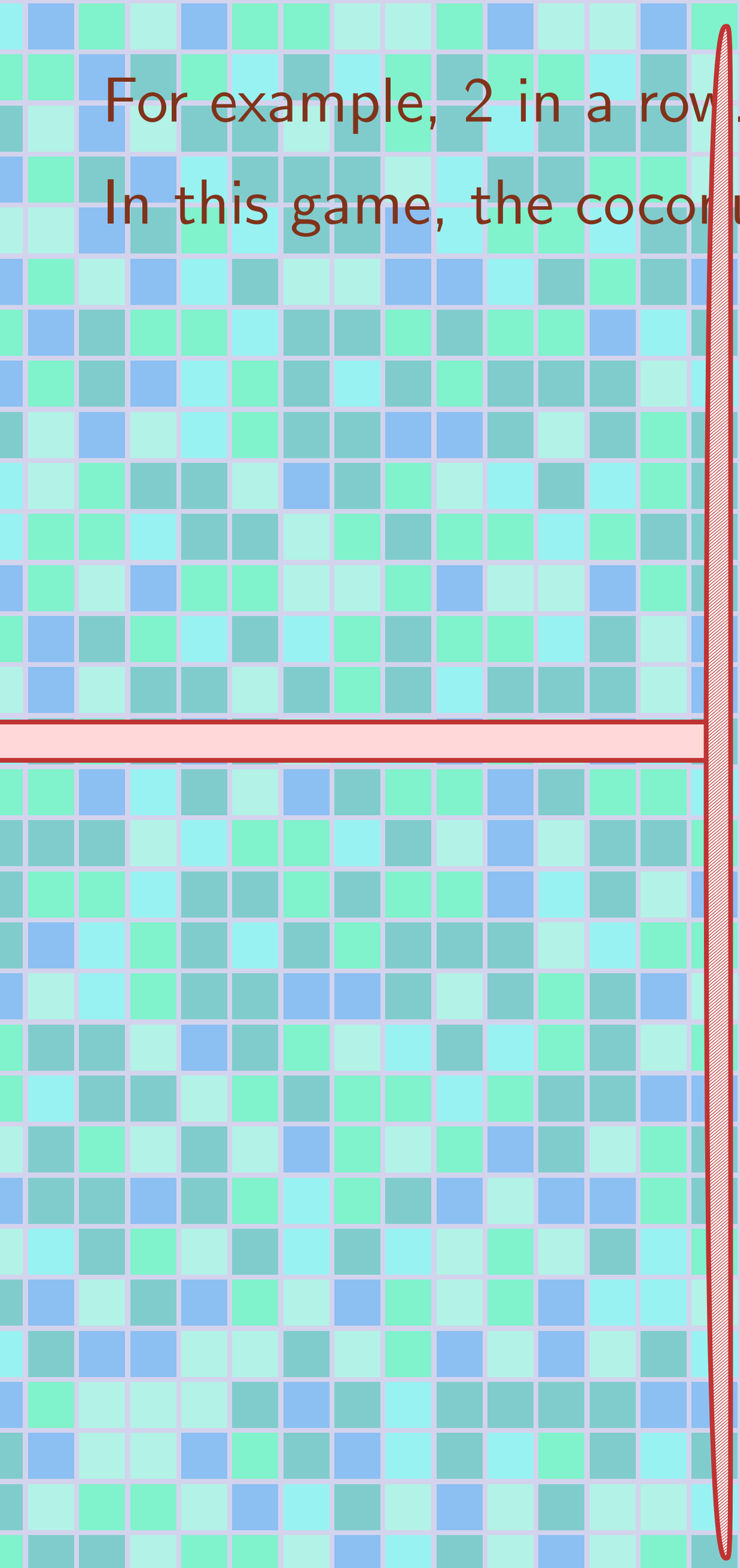
For example, 2 in a row.

In this game, the coconuts are also red or blue.



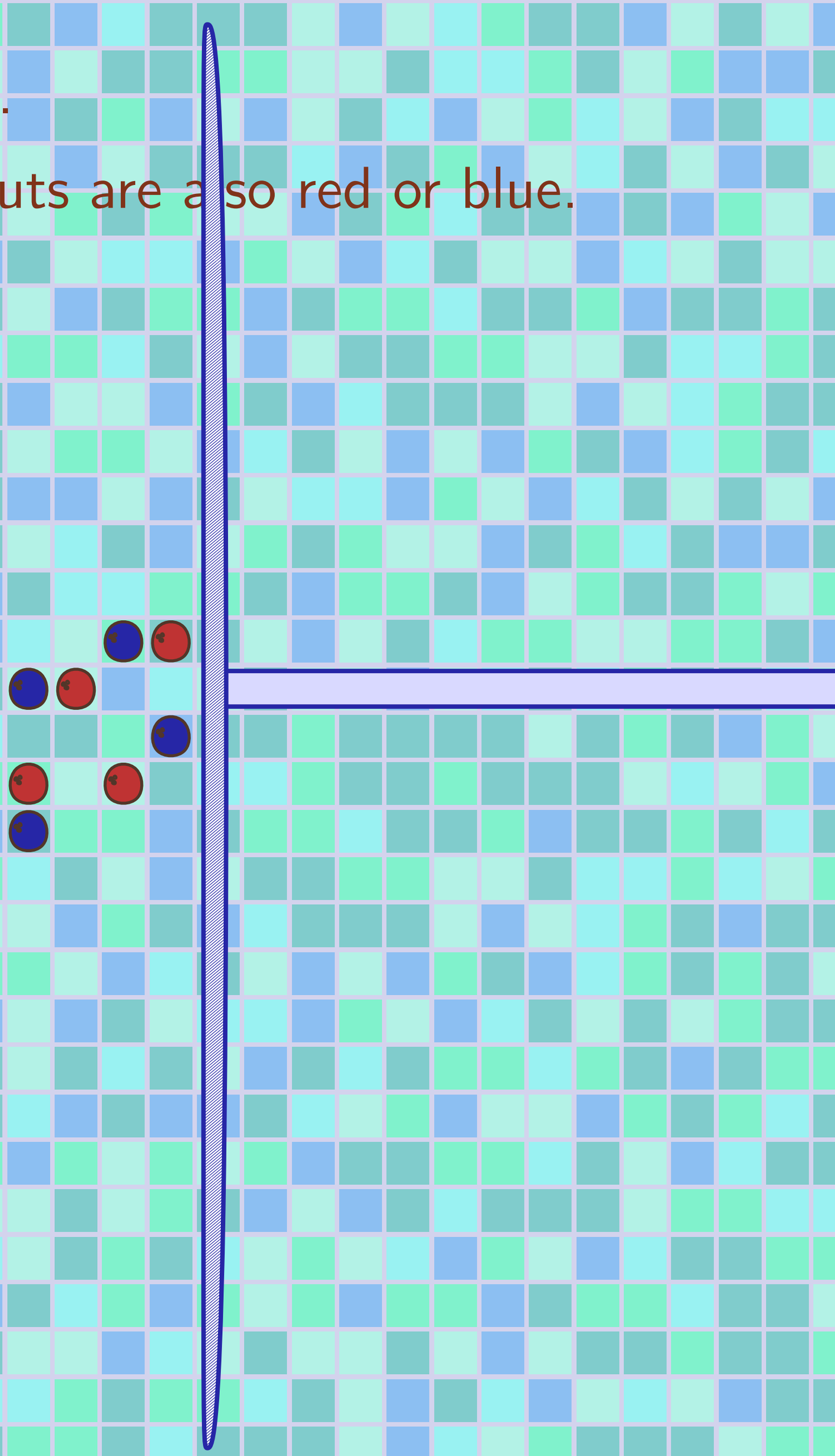
For example, 2 in a row.

In this game, the coconuts are also red or blue.



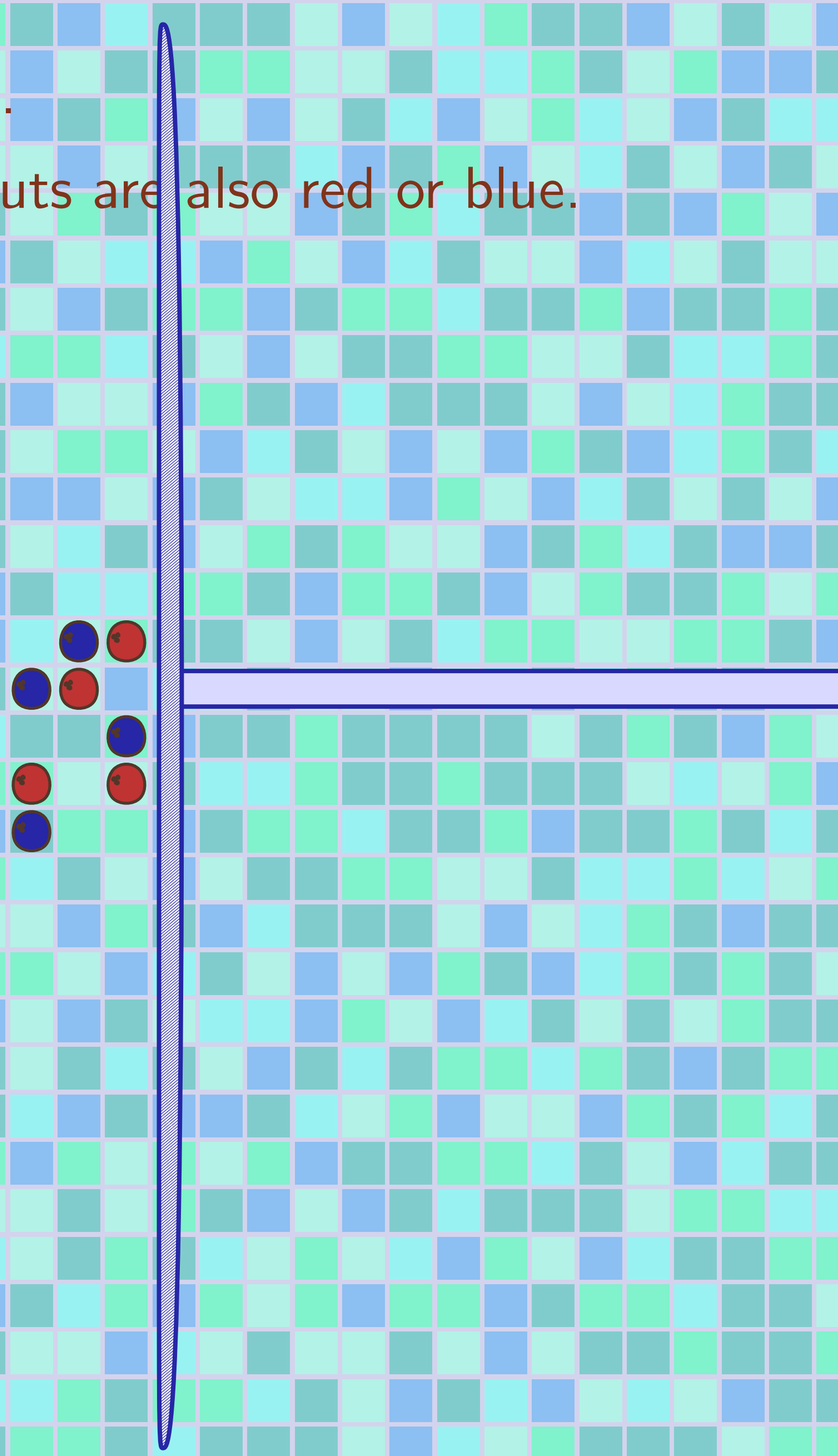
For example, 2 in a row.

In this game, the coconuts are also red or blue.



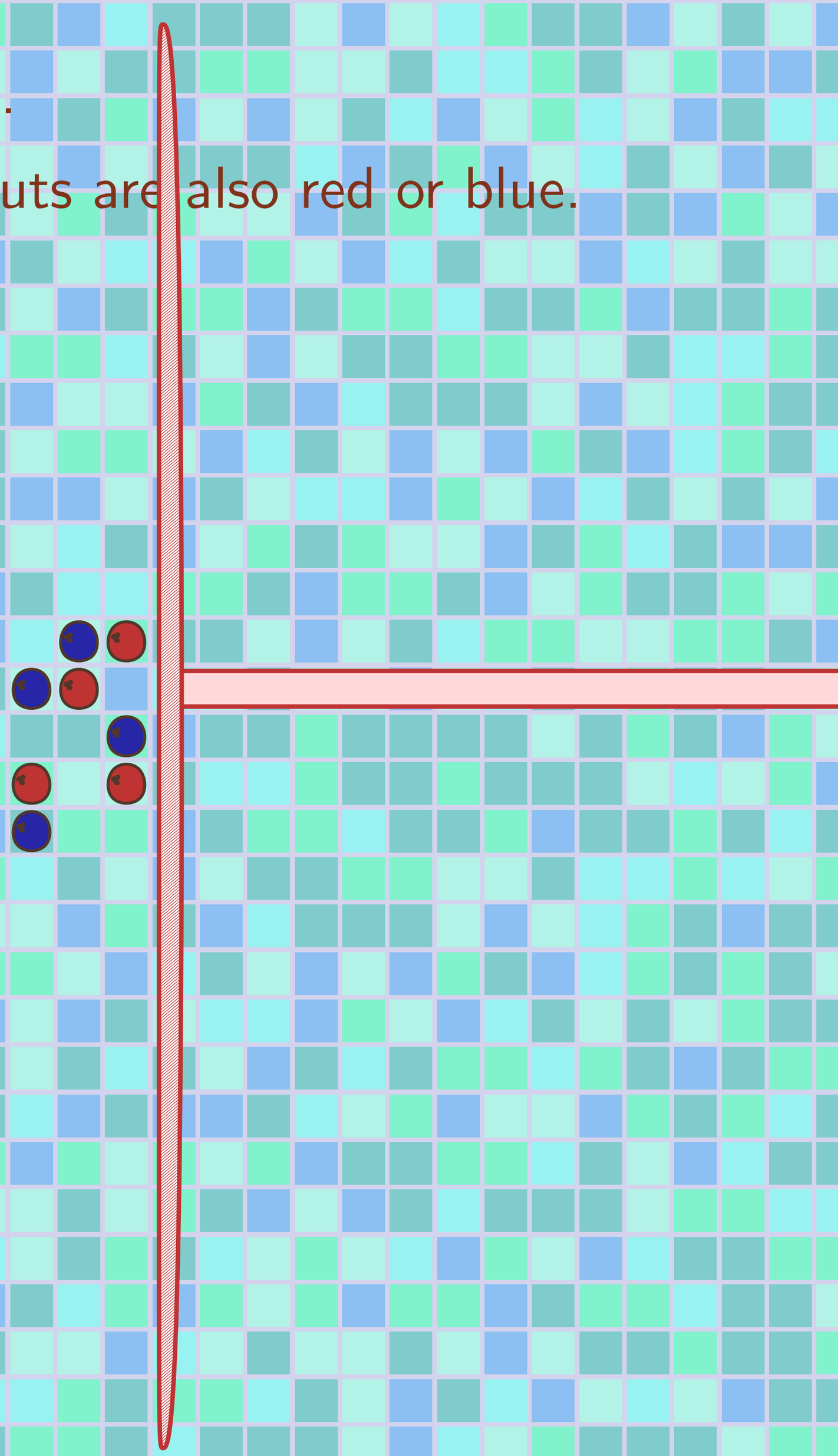
For example, 2 in a row.

In this game, the coconuts are also red or blue.



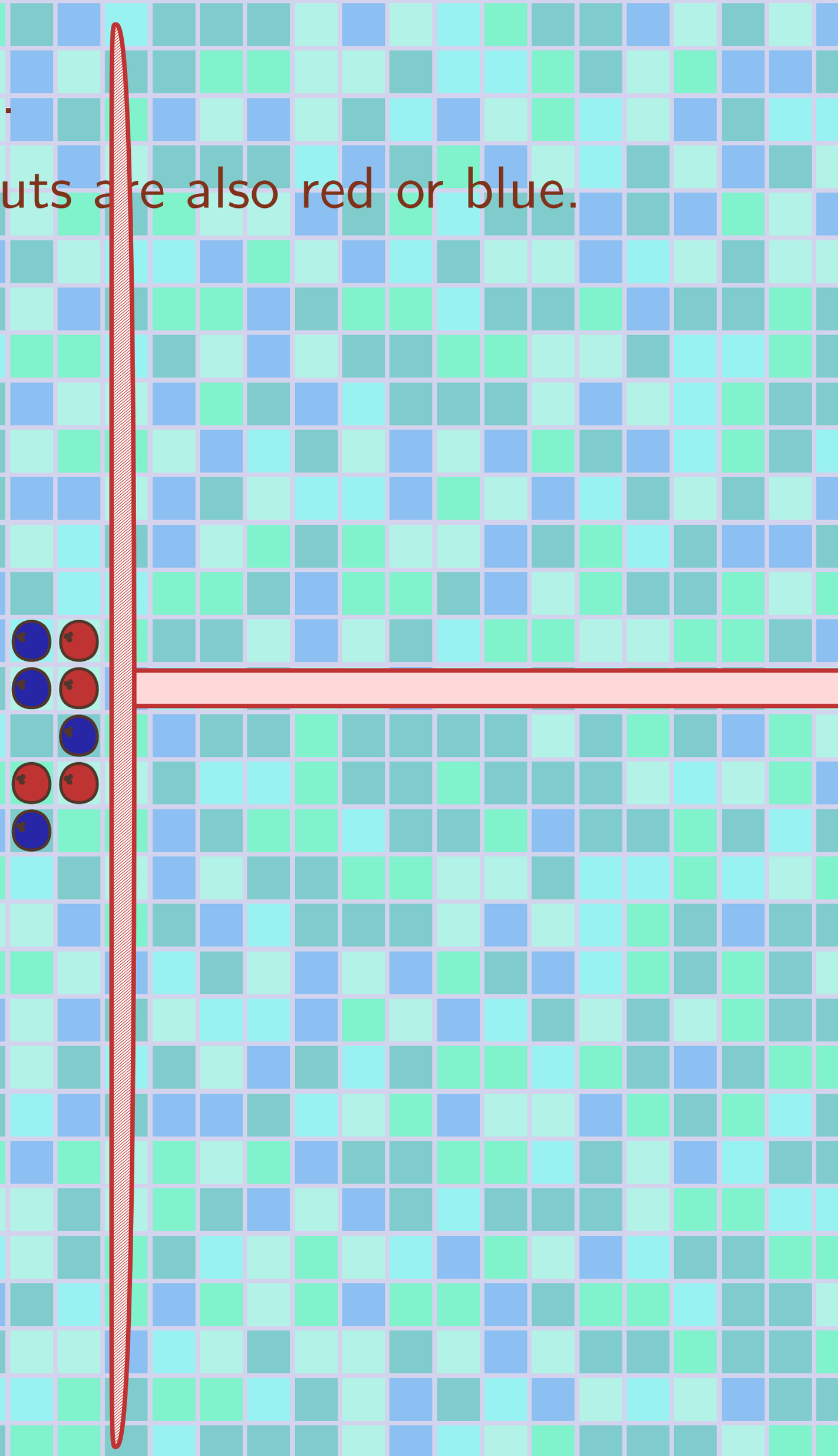
For example, 2 in a row.

In this game, the coconuts are also red or blue.



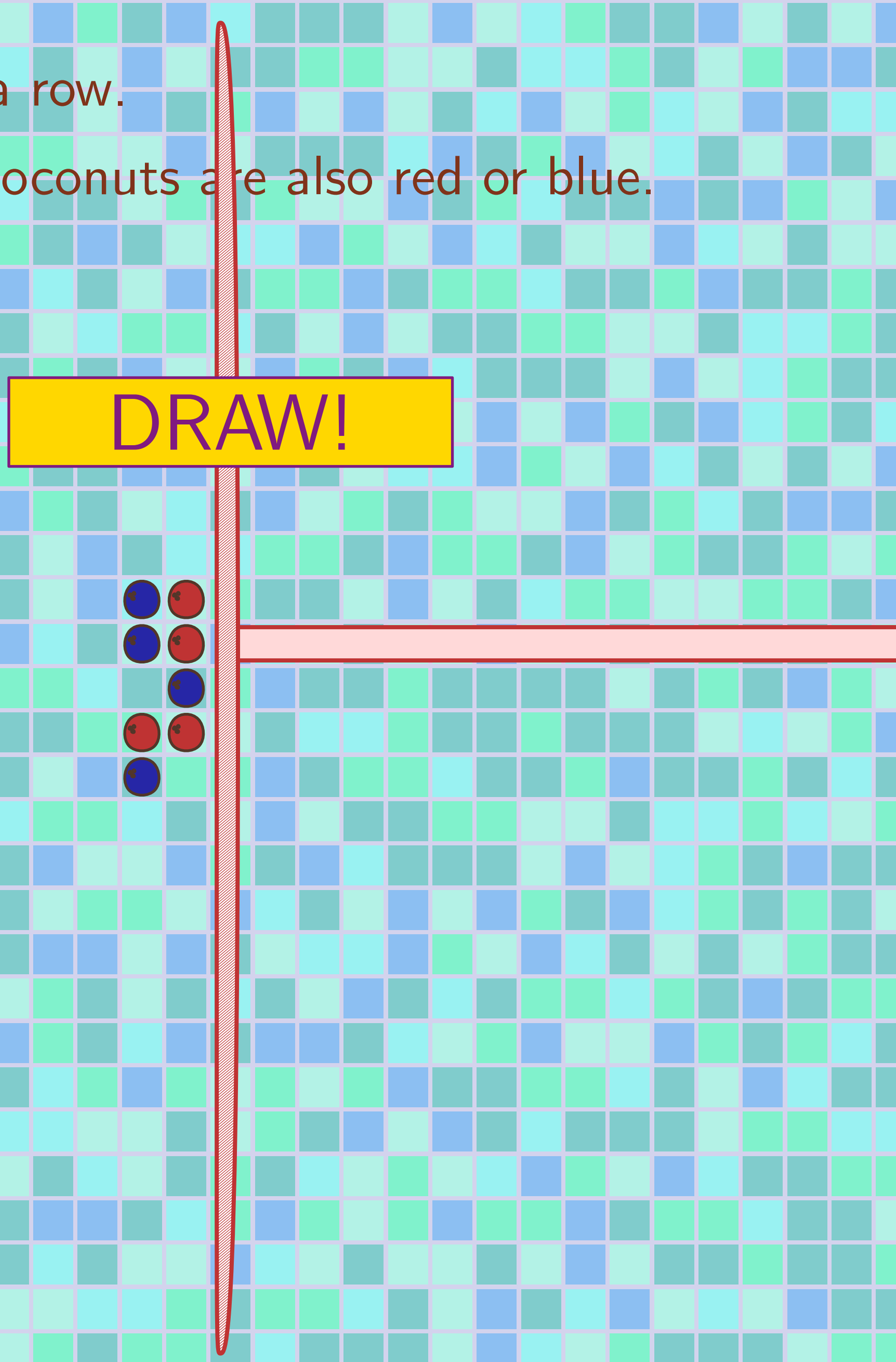
For example, 2 in a row.

In this game, the coconuts are also red or blue.



For example, 2 in a row.

In this game, the coconuts are also red or blue.



Question

Does this game always terminate?

Question

Does this game always terminate?

No!

Question

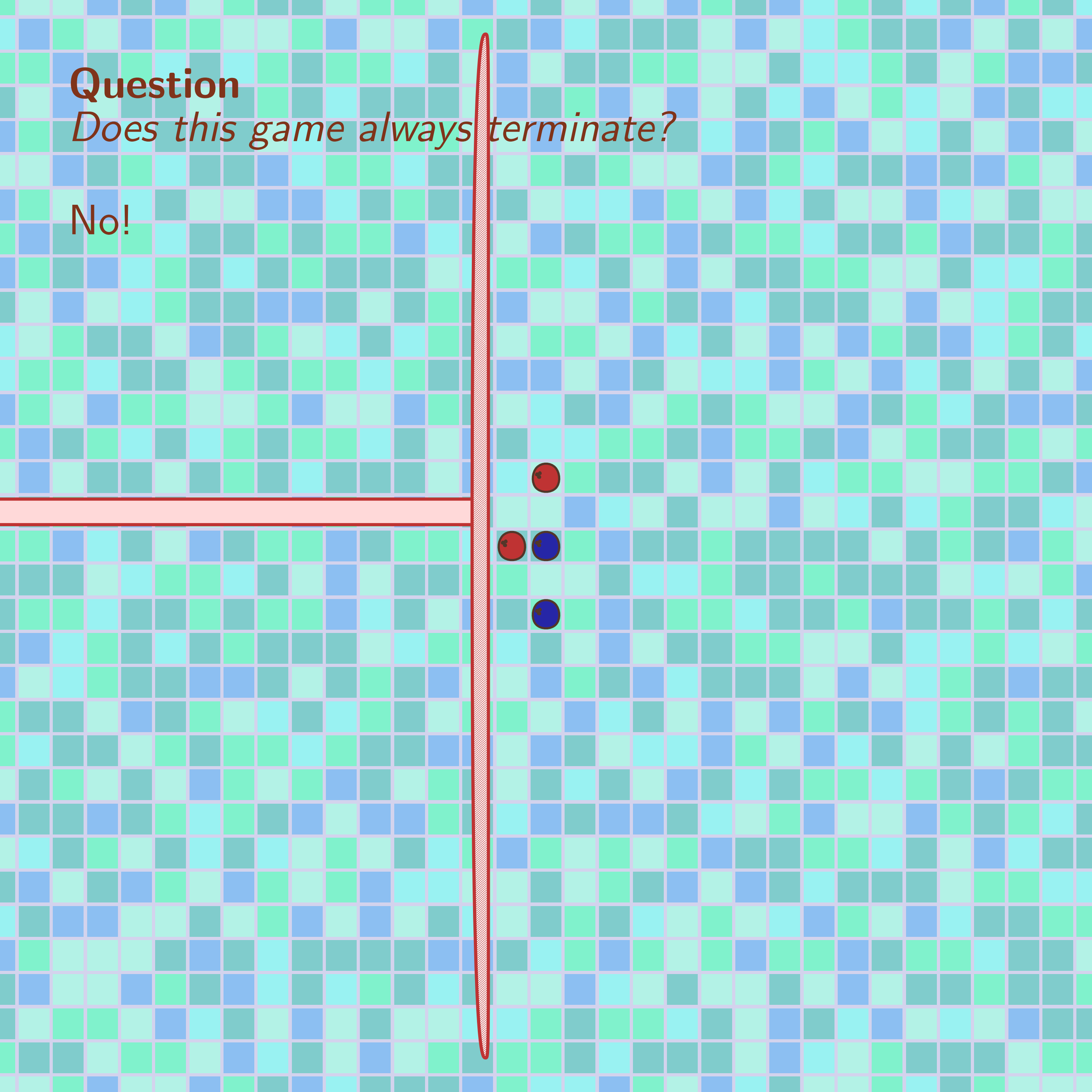
Does this game always terminate?

No!

Question

Does this game always terminate?

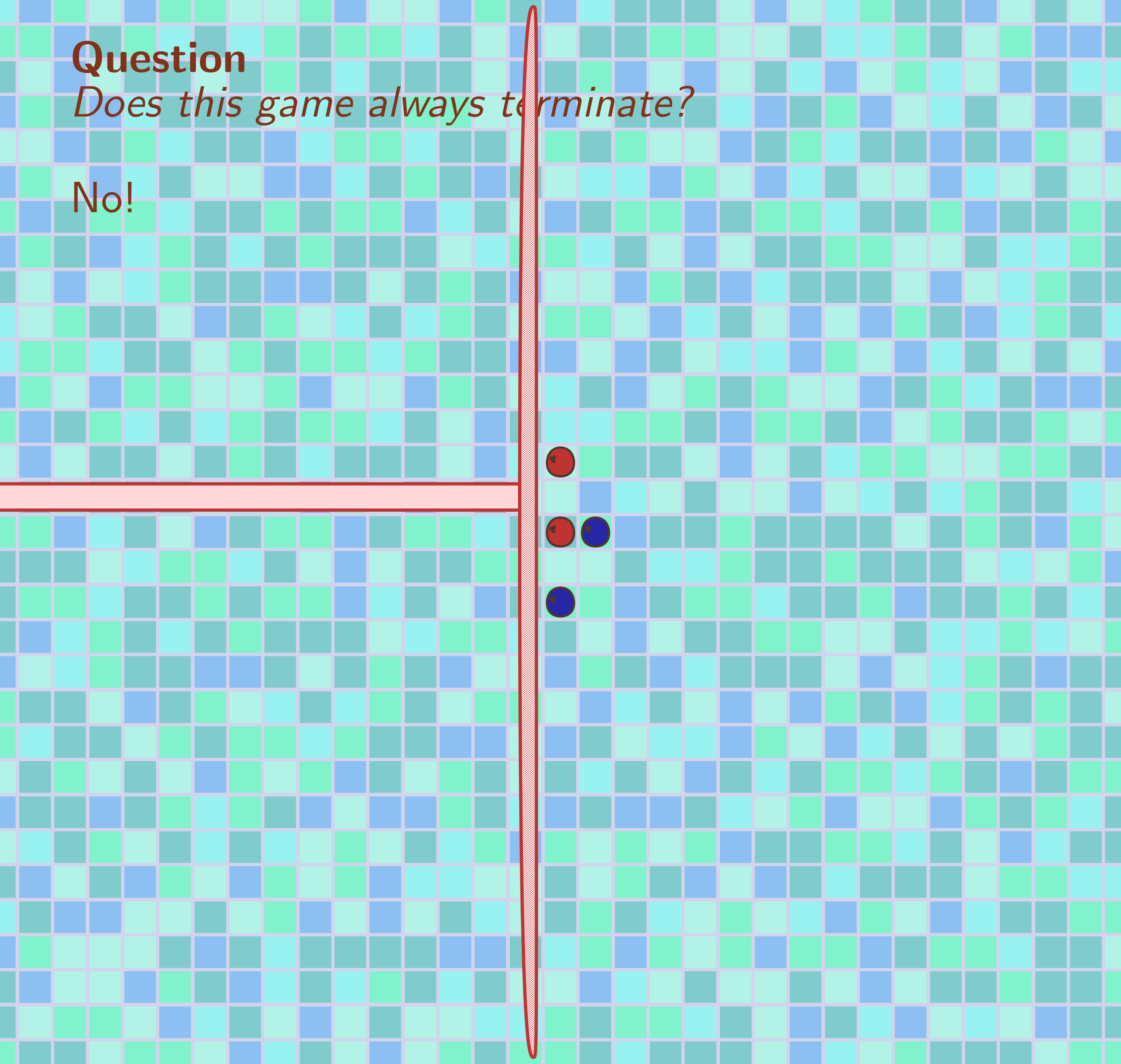
No!



Question

Does this game always terminate?

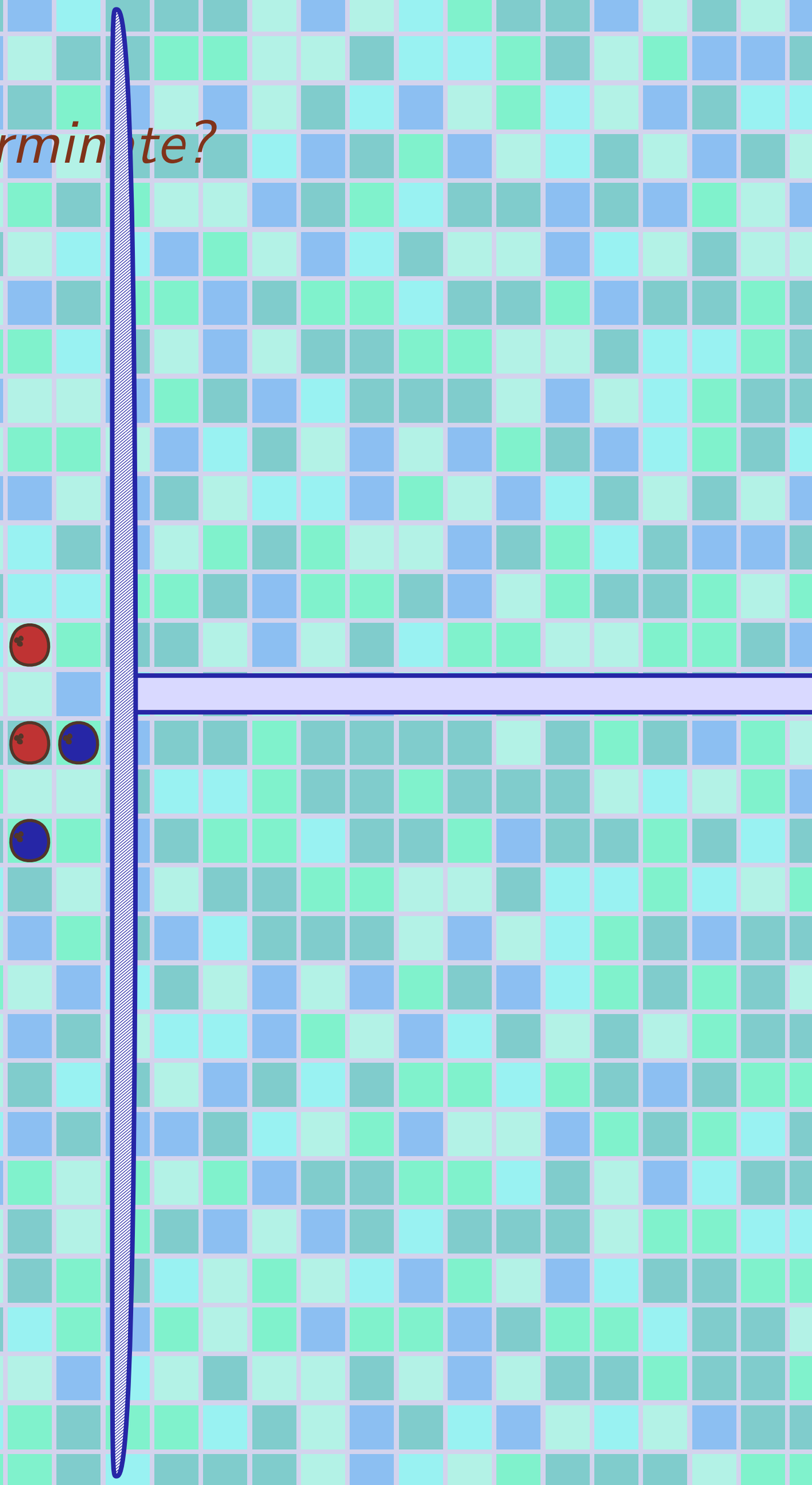
No!



Question

Does this game always terminate?

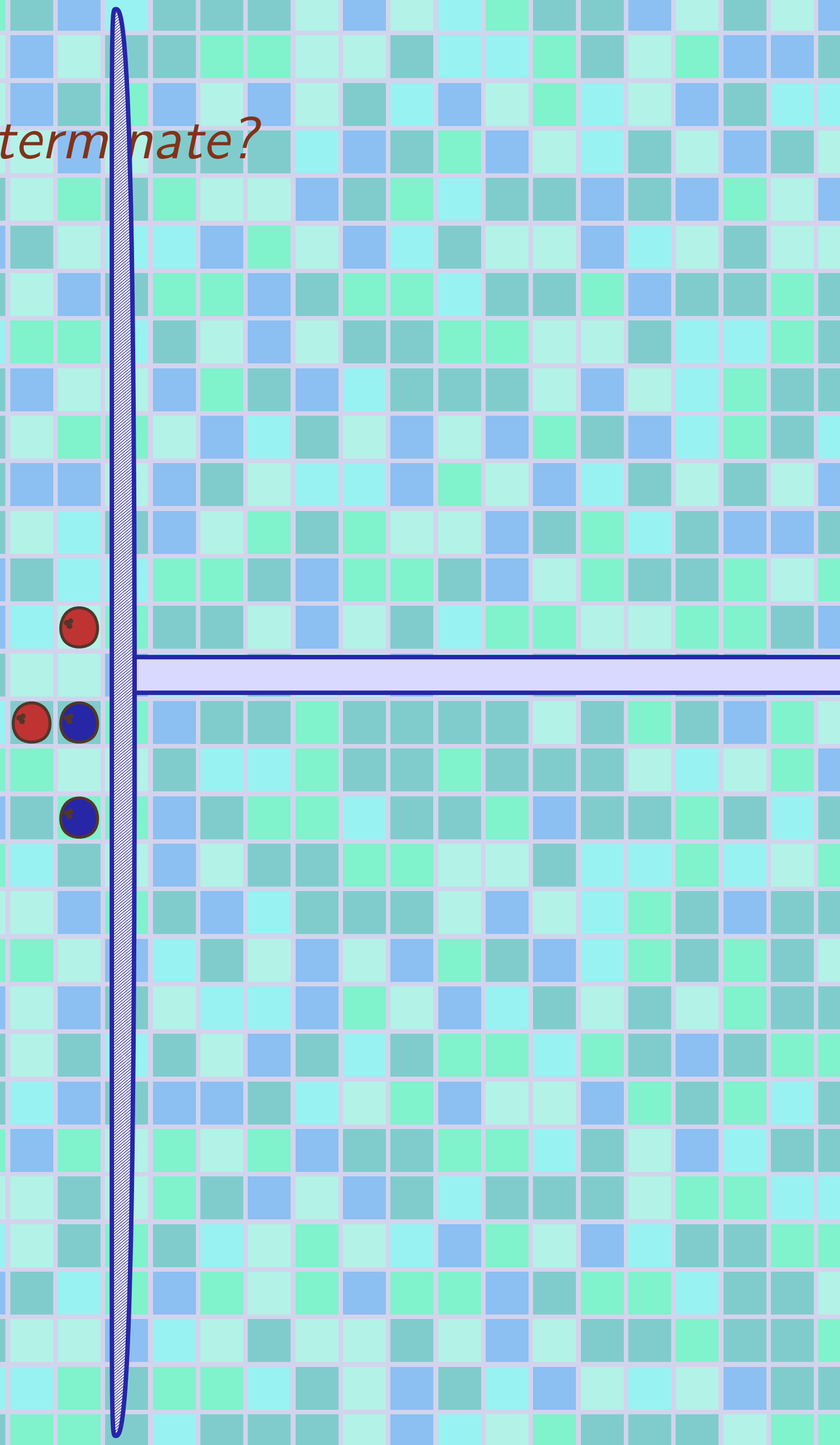
No!



Question

Does this game always terminate?

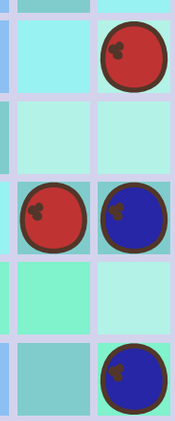
No!



Question
Does this game always terminate?

No!

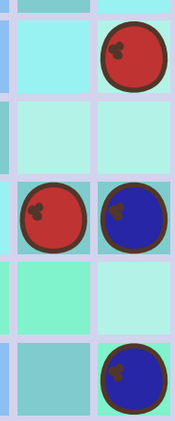
Open Question
Is it possible to get into a cycle of more than 2 states?



Question
Does this game always terminate?

No!

Open Question
Is it possible to get into a cycle of more than 2 states?



Open Question
Is 2-IN-A-ROW *hard?*

QUESTIONS?