

Luckasaurus Rex

Presentation by
Orson Teodoro



Ranked #2 out of 46

Overview

Placed **second** out of 46 teams

Java had better support with BoardModel

Used BoardModel.clone() and exchanged tiles for testing

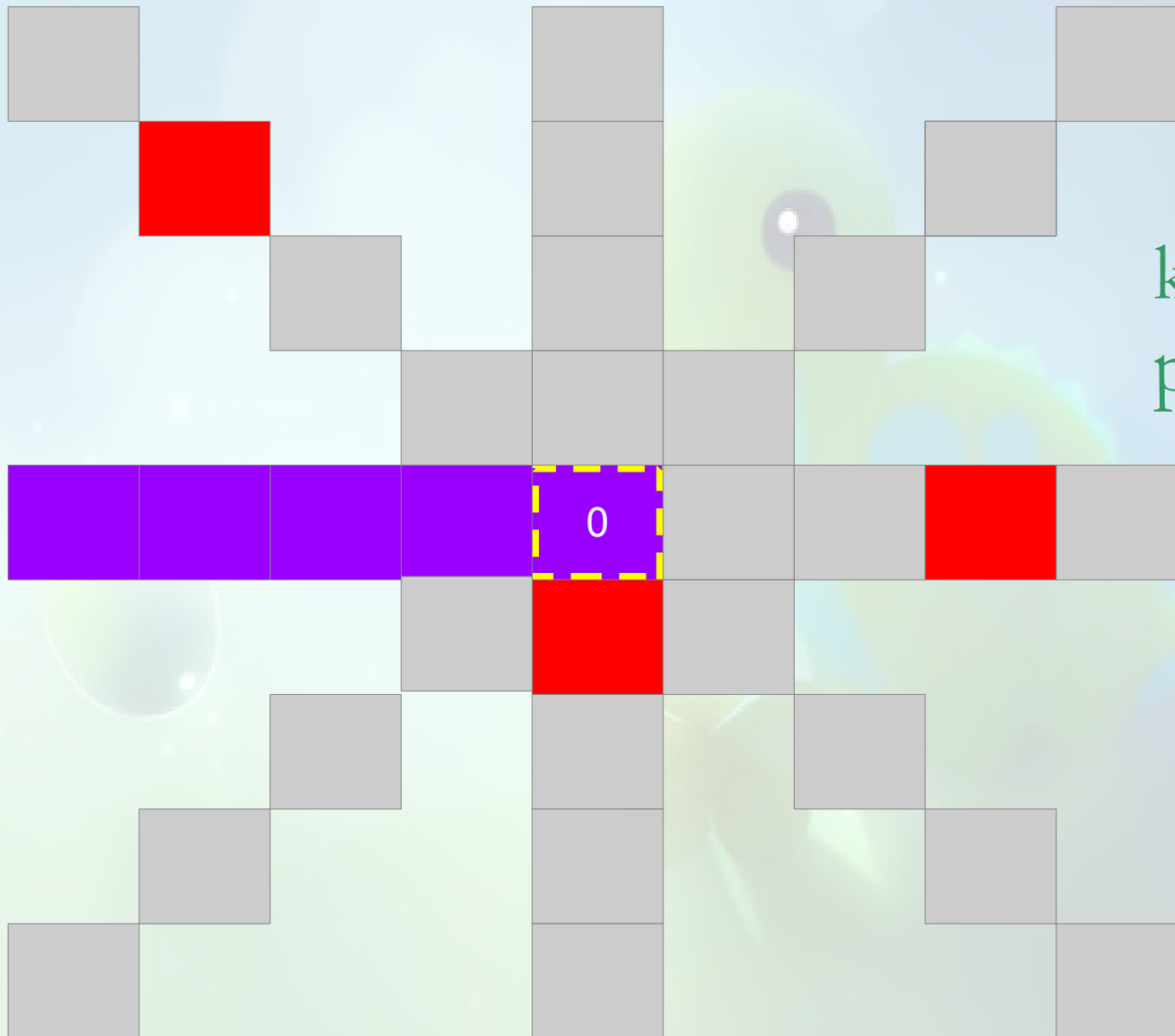
Focused on defaults and non-gravity

Testing frequency: PoorAI > AverageAI > GoodAI

Testing order: PoorAI → AverageAI → GoodAI

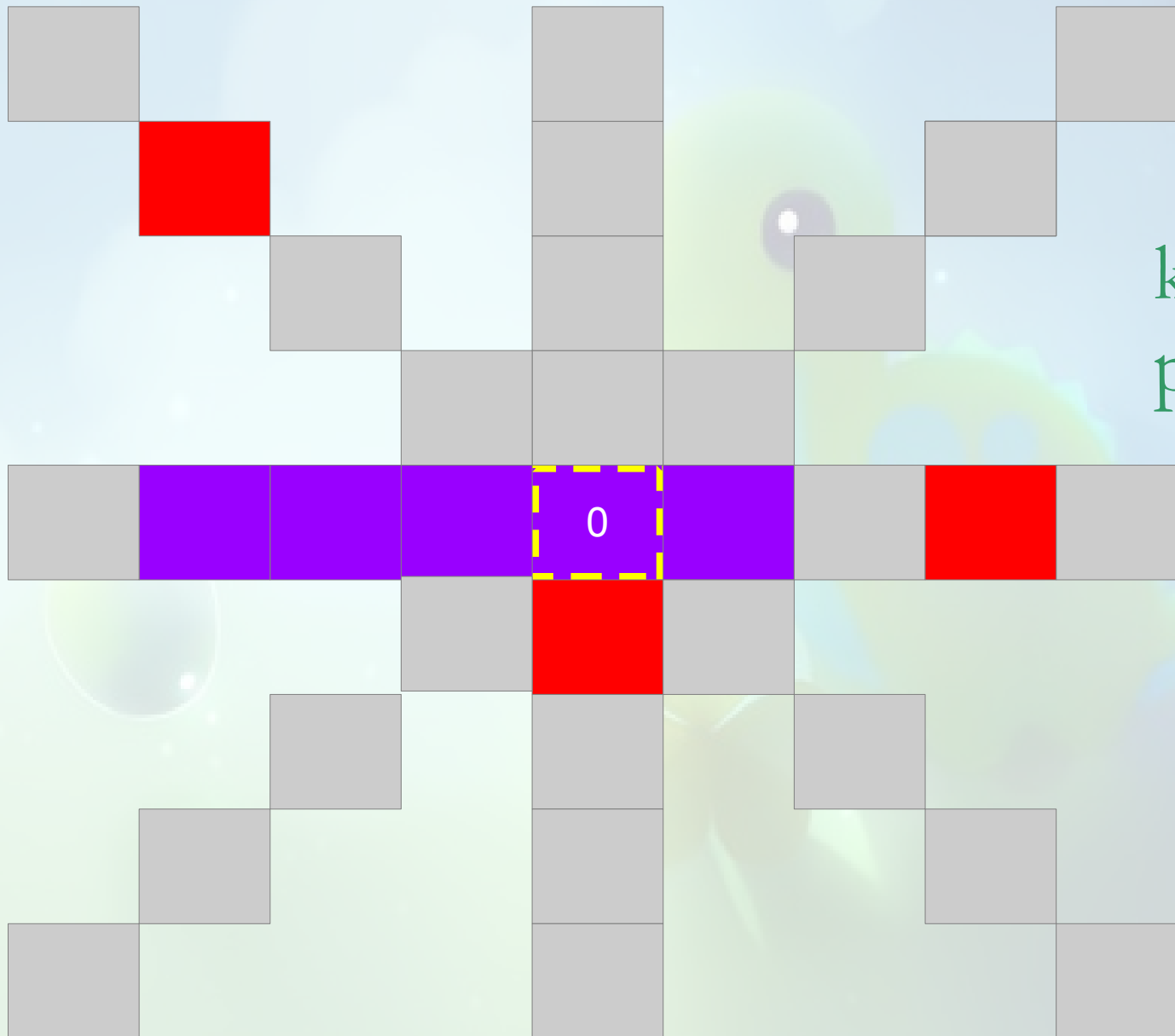
Weaknesses: (1) 2 steps ahead attacks and
(2) forced moves

Counting Saturated Viable K-Connects



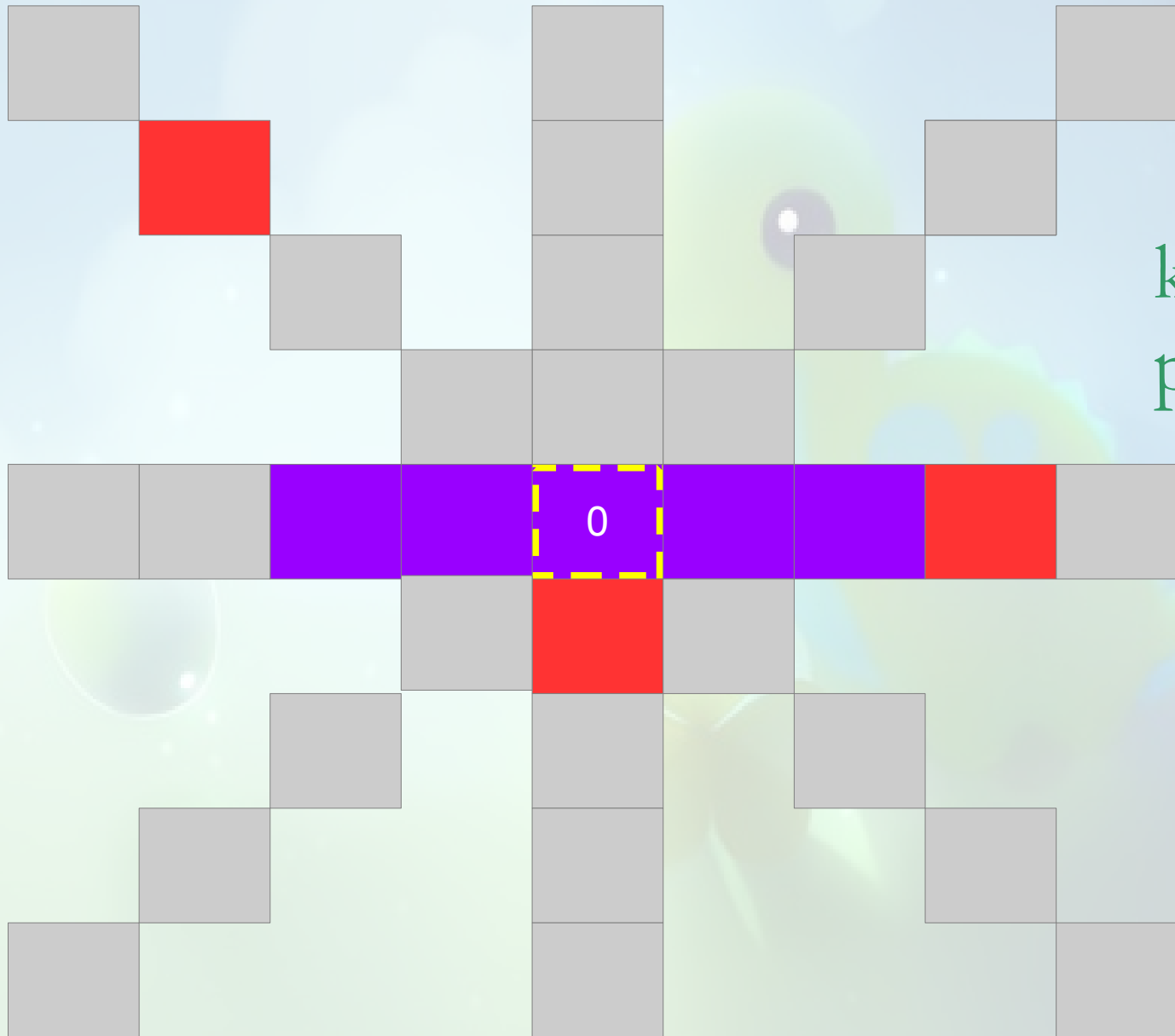
k-connects
possible: 0

Counting Saturated Viable K-Connects



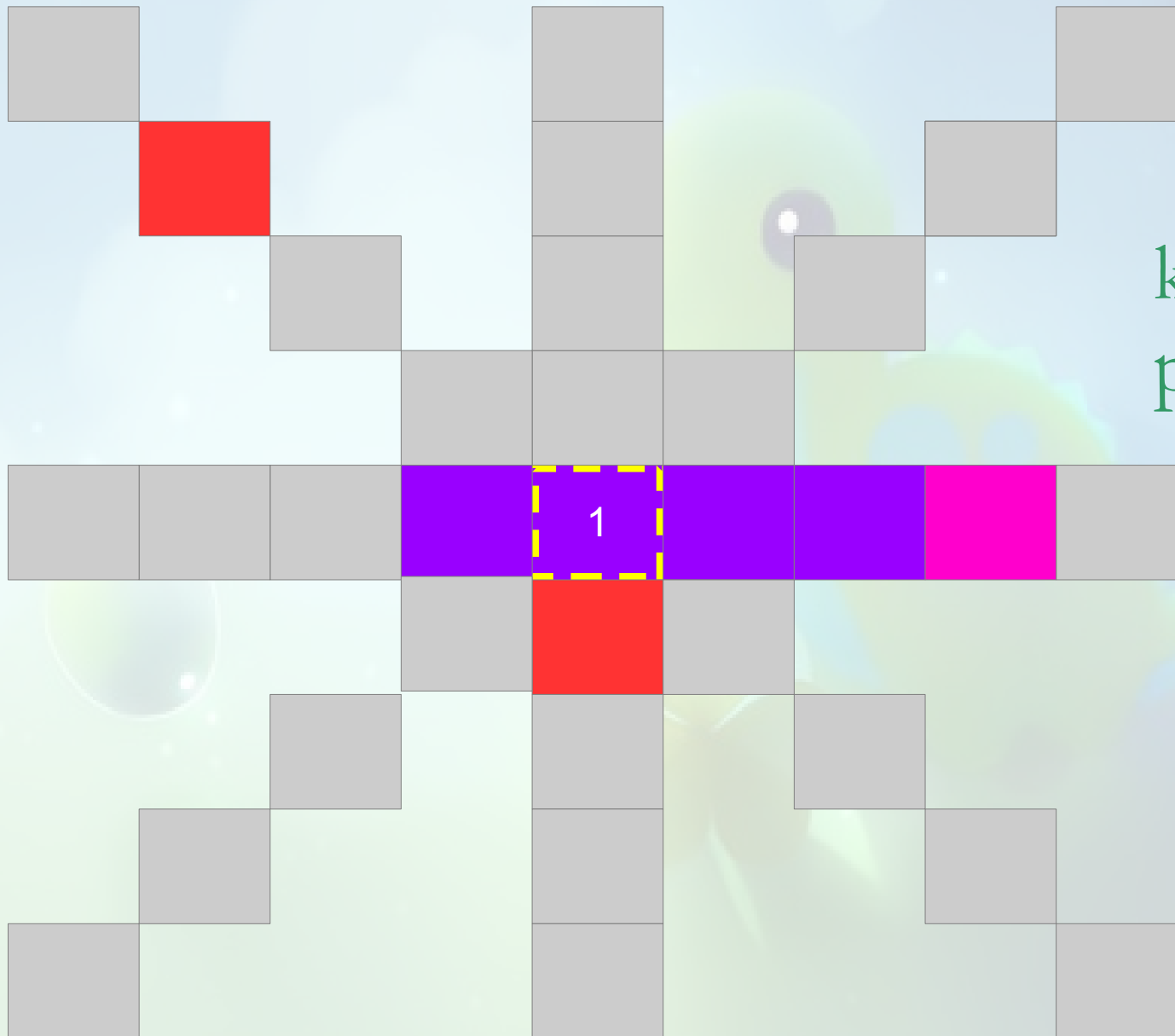
k-connects
possible: 0

Counting Saturated Viable K-Connects



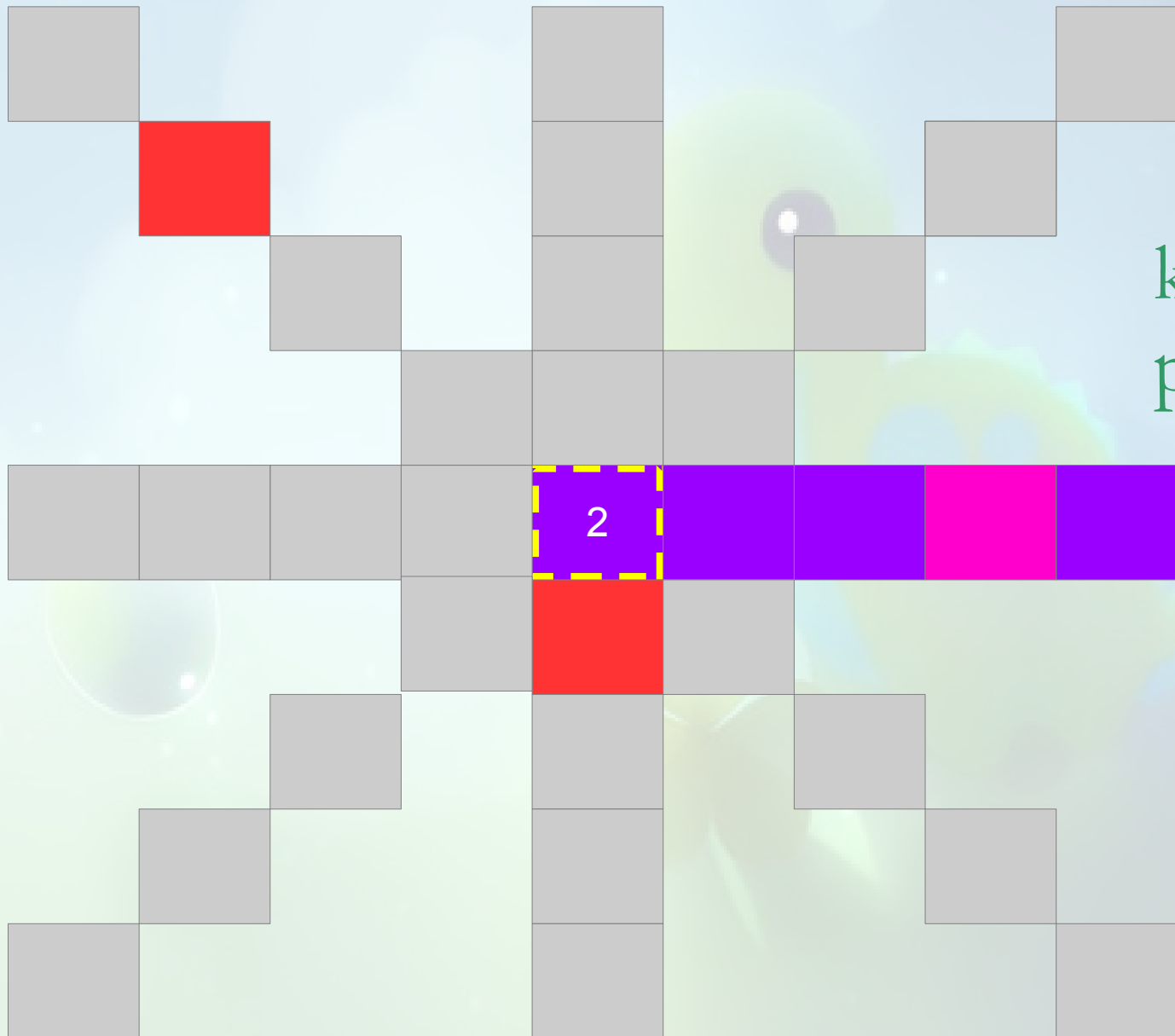
k-connects
possible: 0

Counting Saturated Viable K-Connects



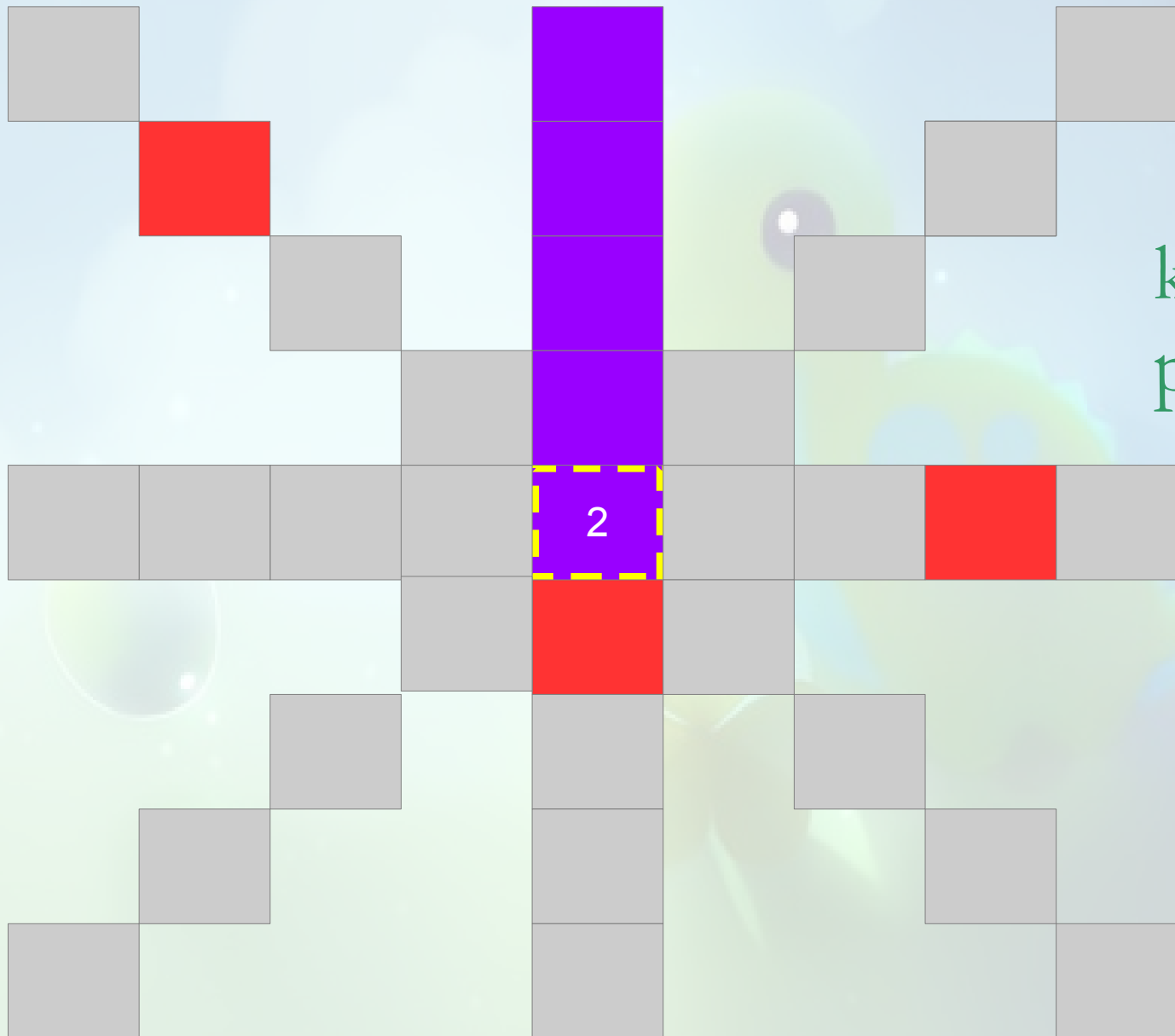
k-connects
possible: 1

Counting Saturated Viable K-Connects



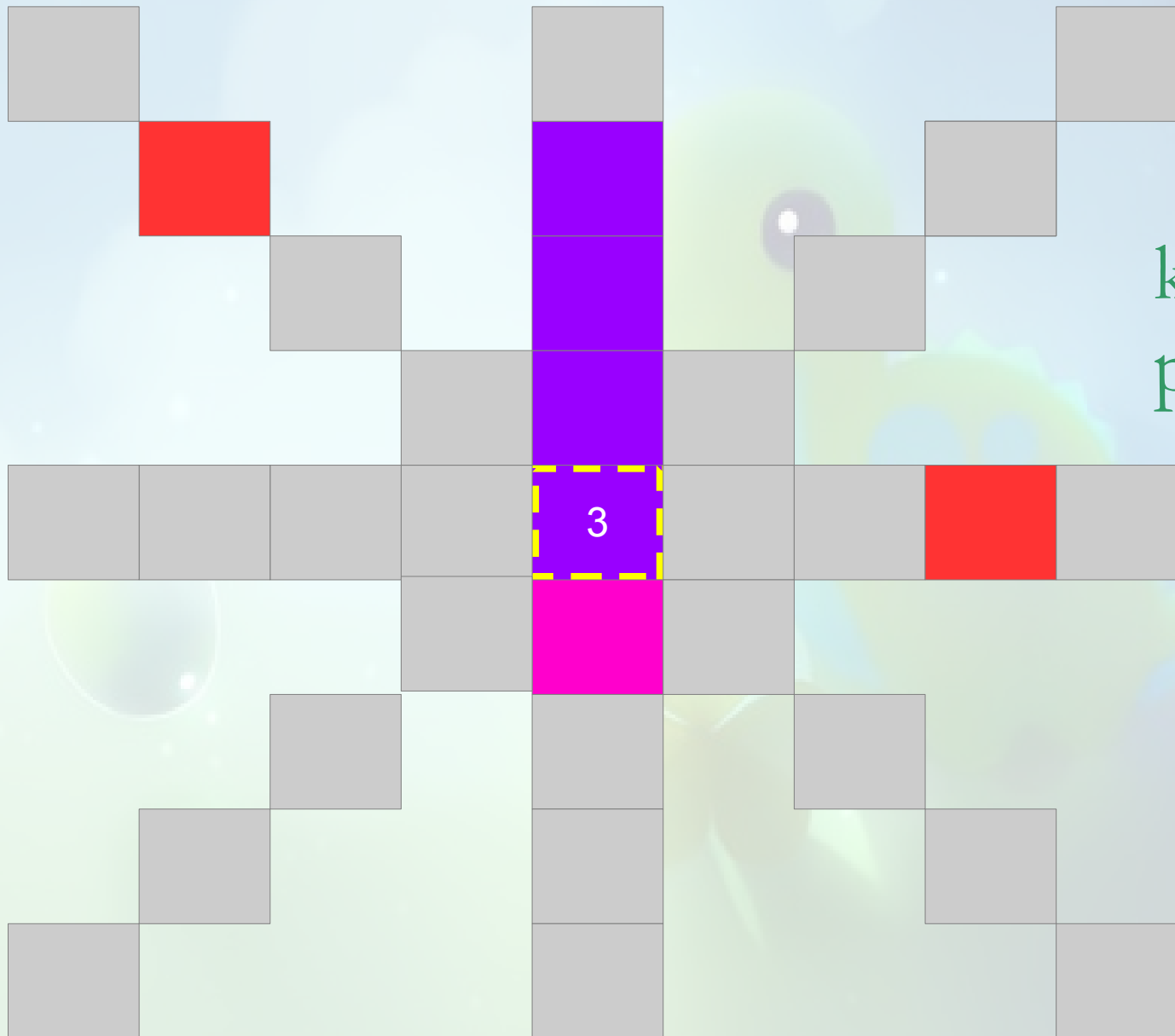
k-connects
possible: 2

Counting Saturated Viable K-Connects



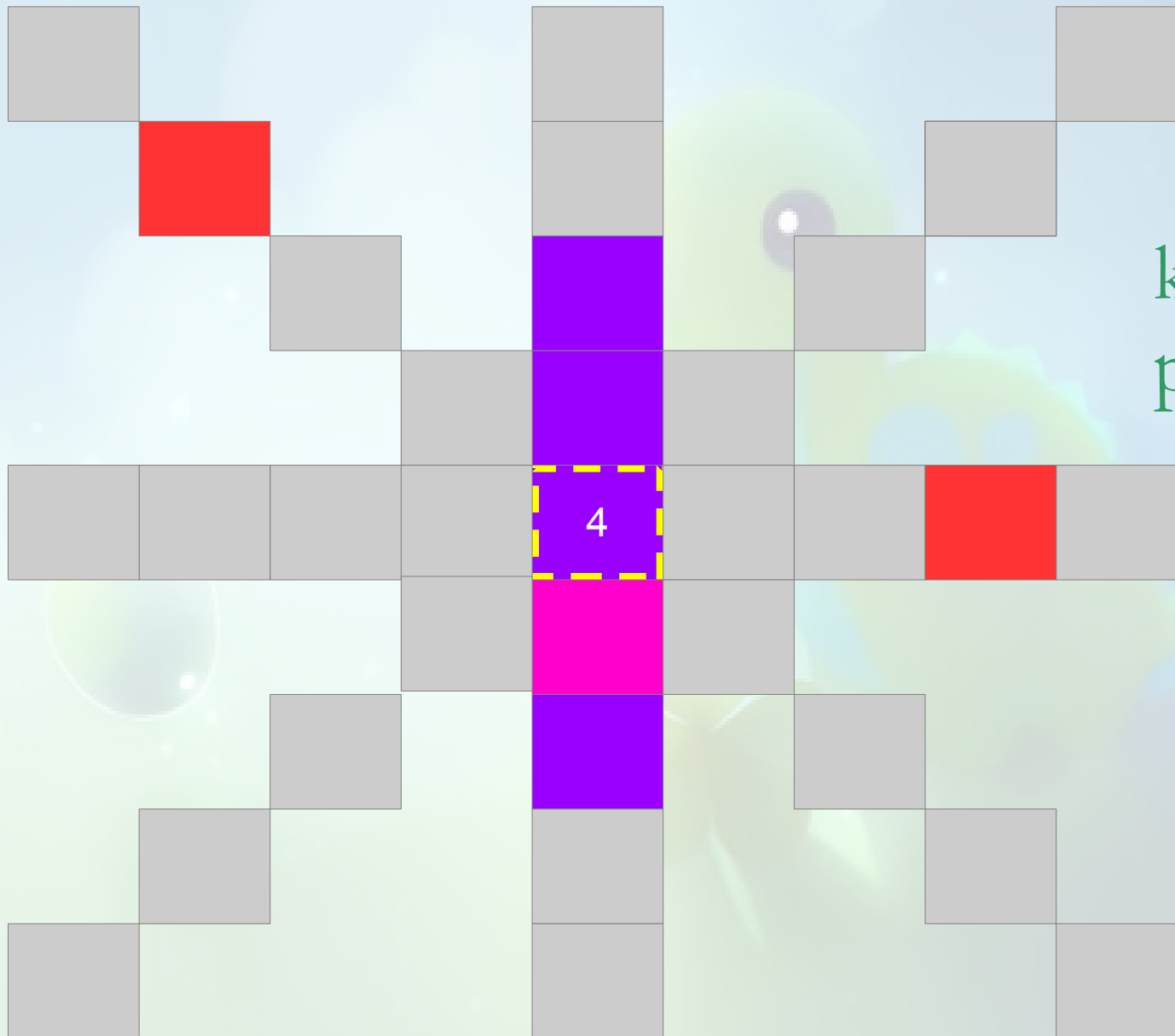
k-connects
possible: 2

Counting Saturated Viable K-Connects



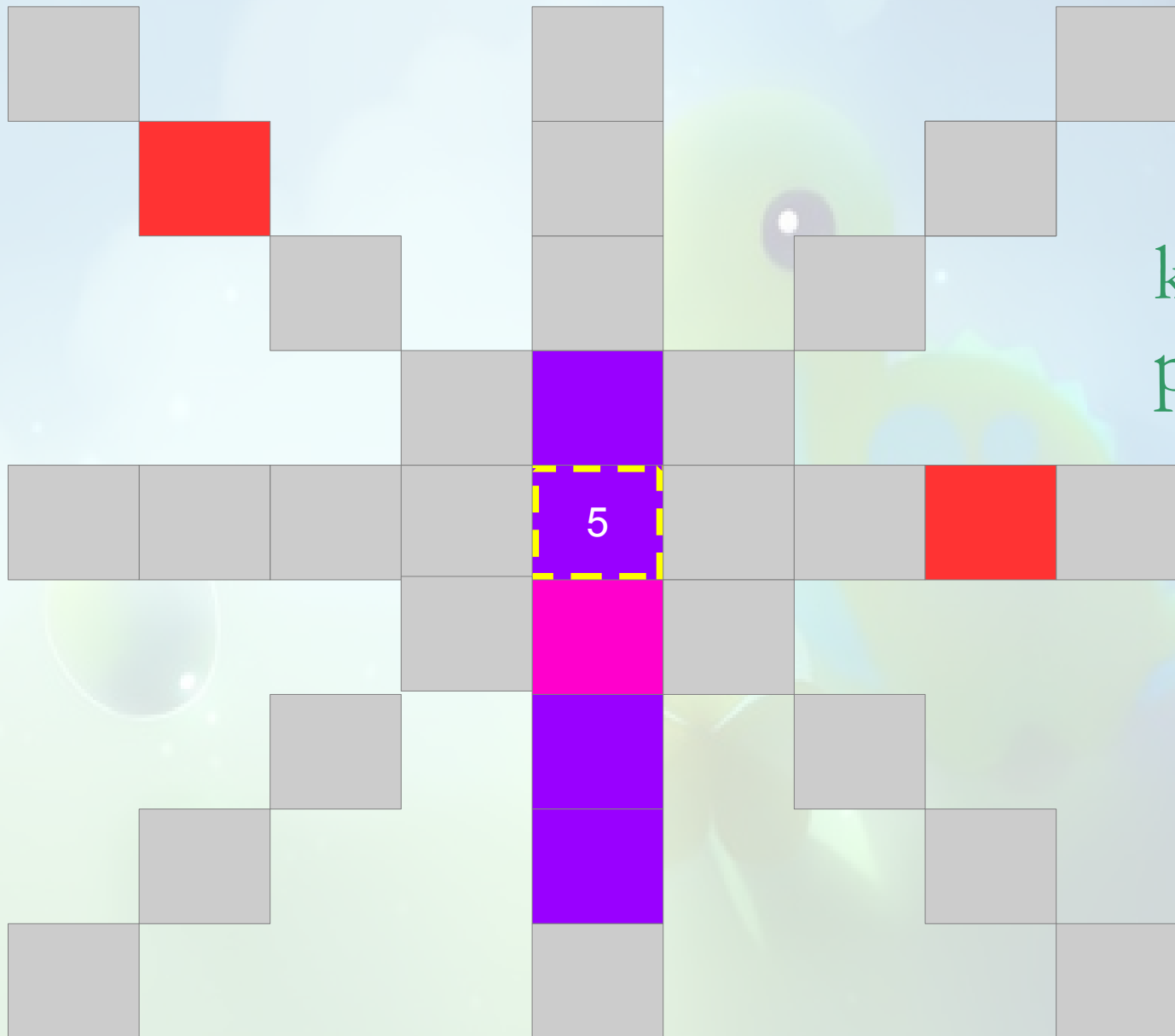
k-connects
possible: 3

Counting Saturated Viable K-Connects



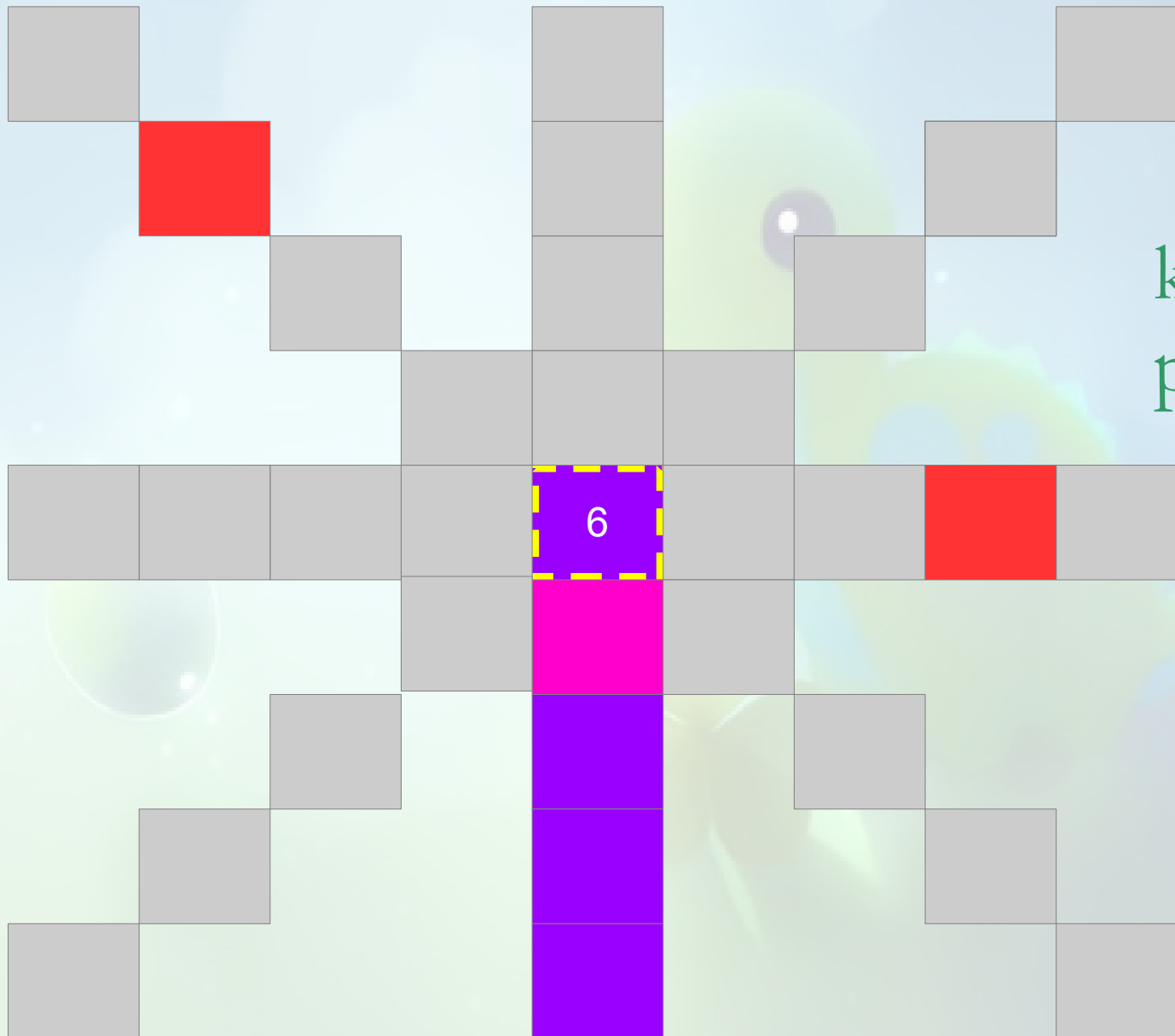
k-connects
possible: 4

Counting Saturated Viable K-Connects



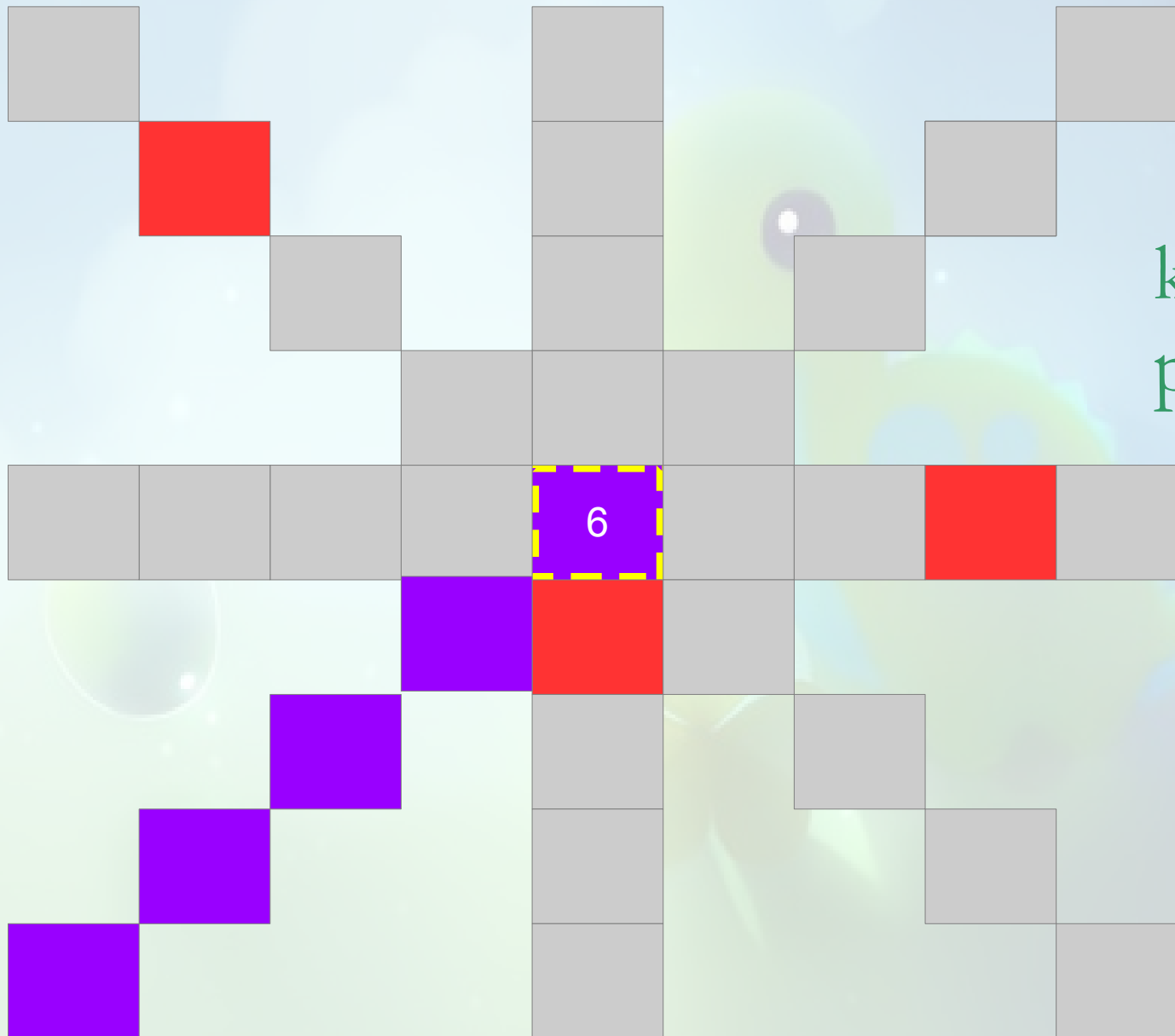
k-connects
possible: 5

Counting Saturated Viable K-Connects



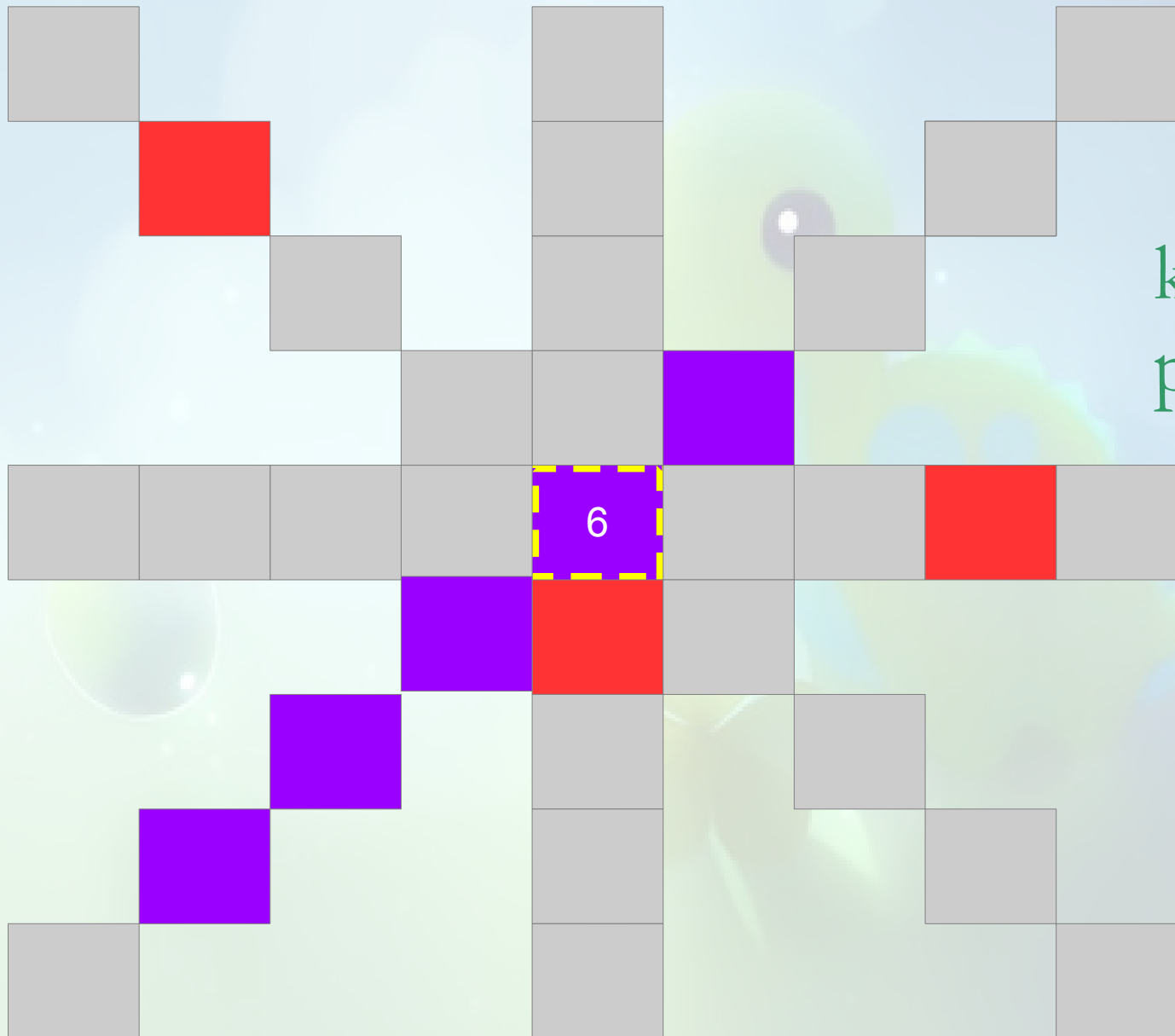
k-connects
possible: 6

Counting Saturated Viable K-Connects



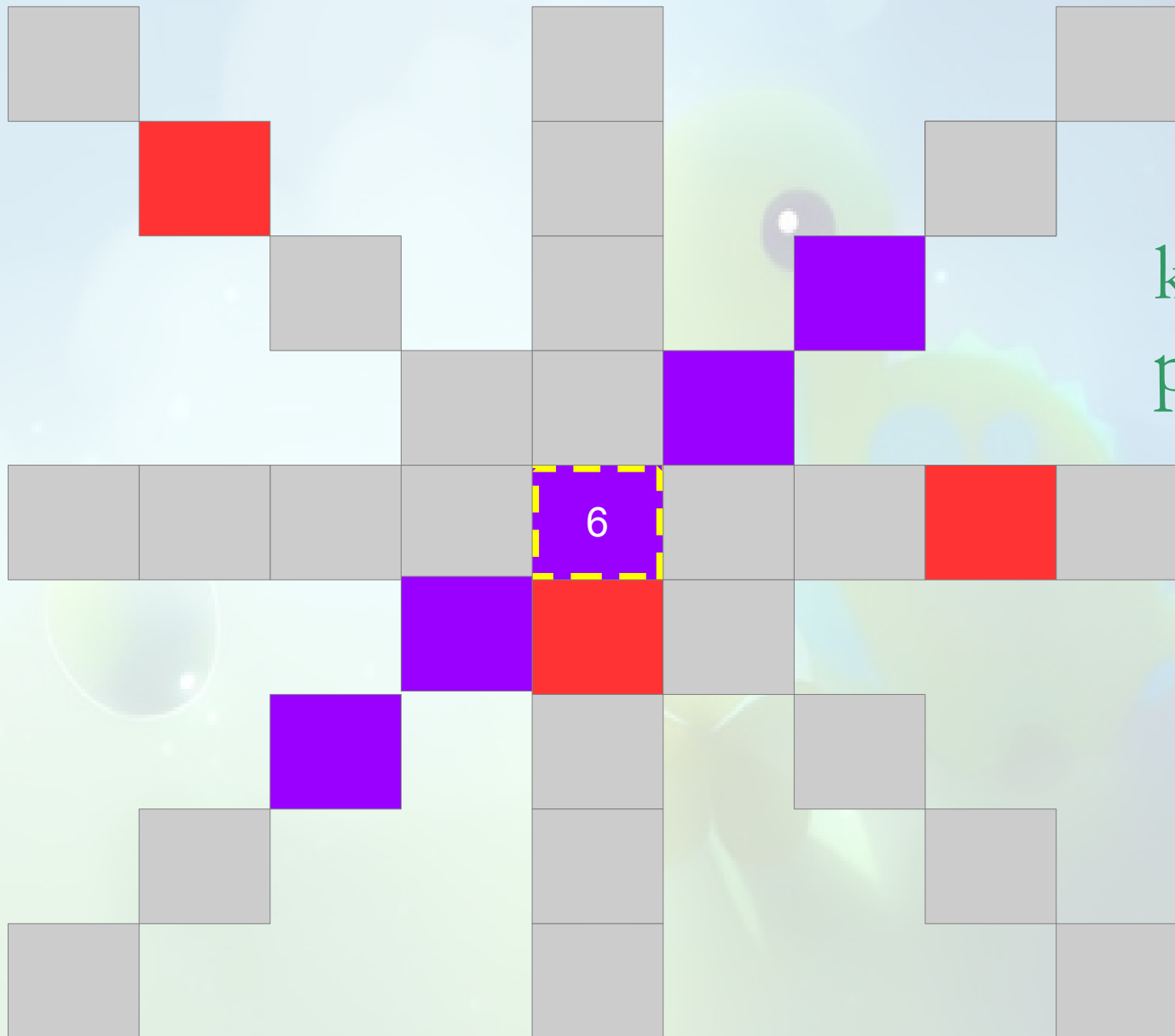
k-connects
possible: 6

Counting Saturated Viable K-Connects



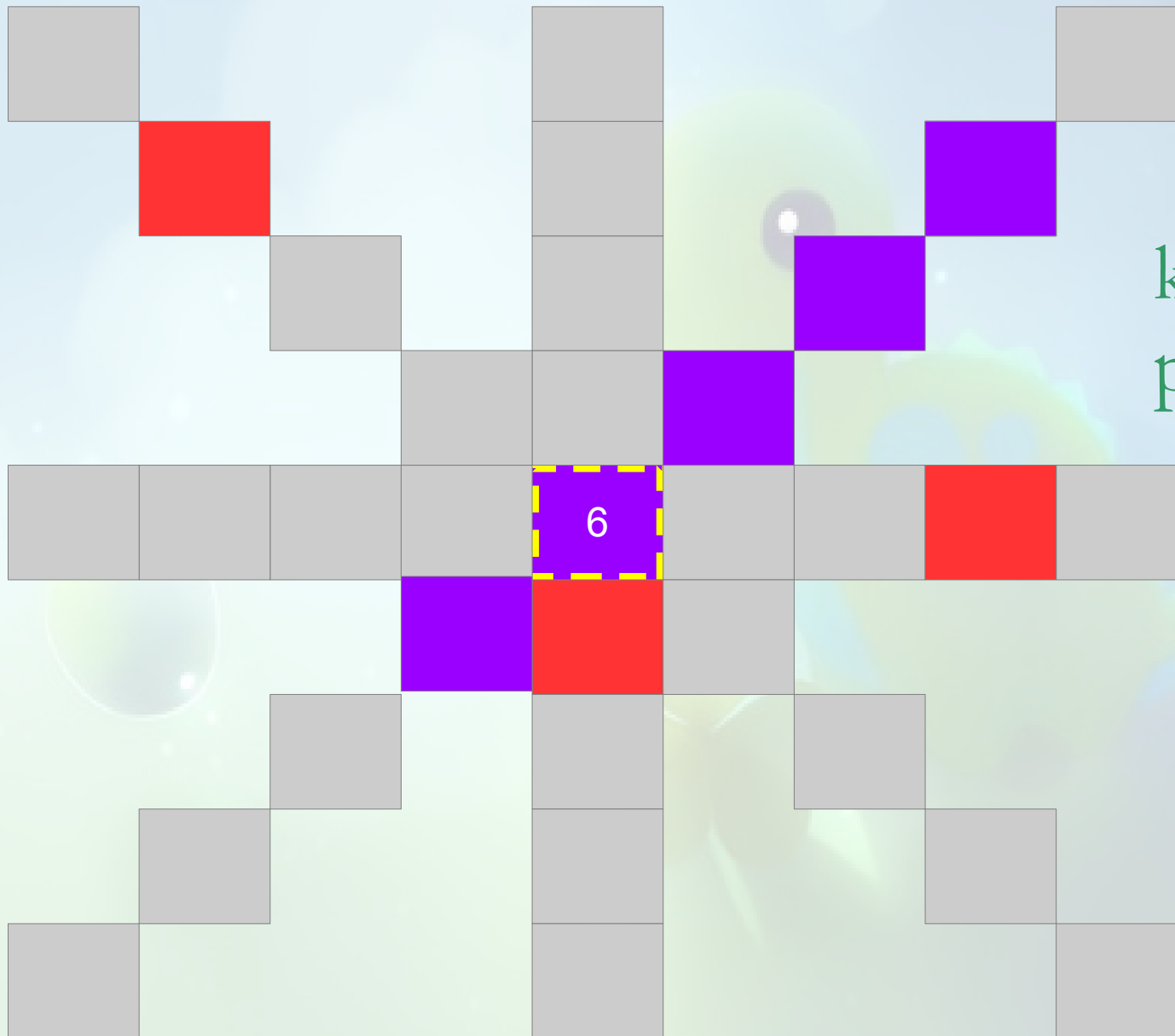
k-connects
possible: 6

Counting Saturated Viable K-Connects



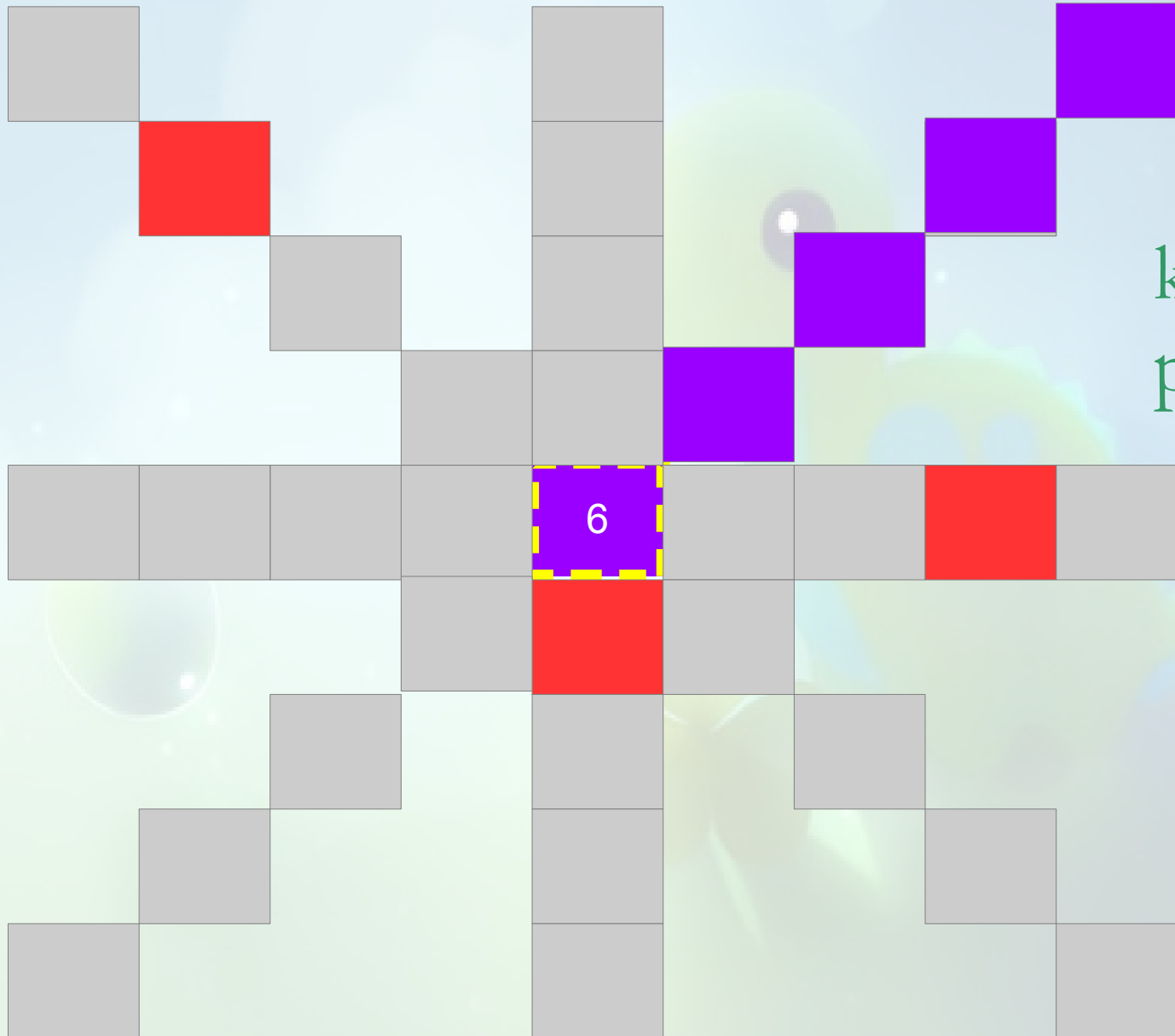
k-connects
possible: 6

Counting Saturated Viable K-Connects



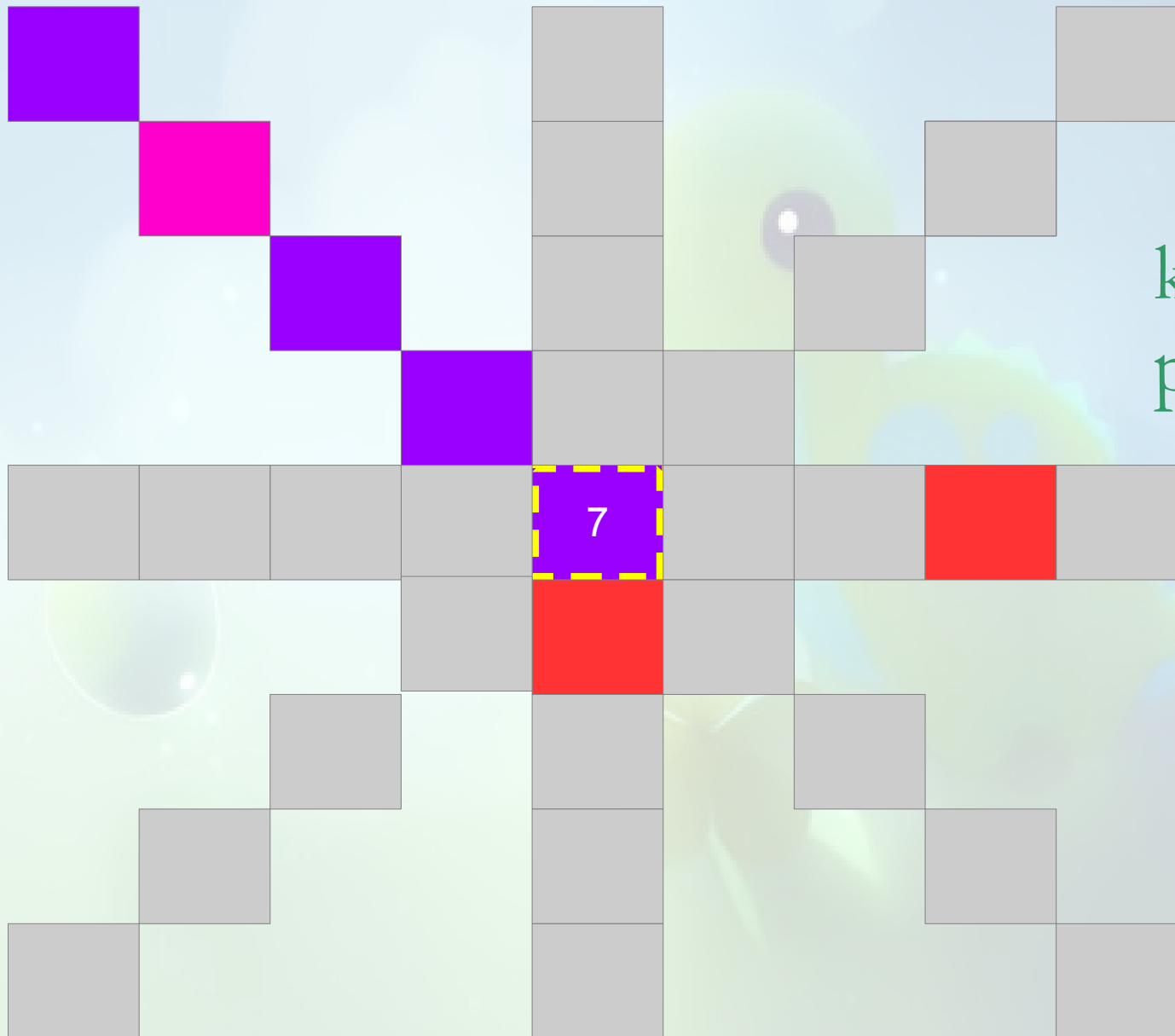
k-connects
possible: 6

Counting Saturated Viable K-Connects



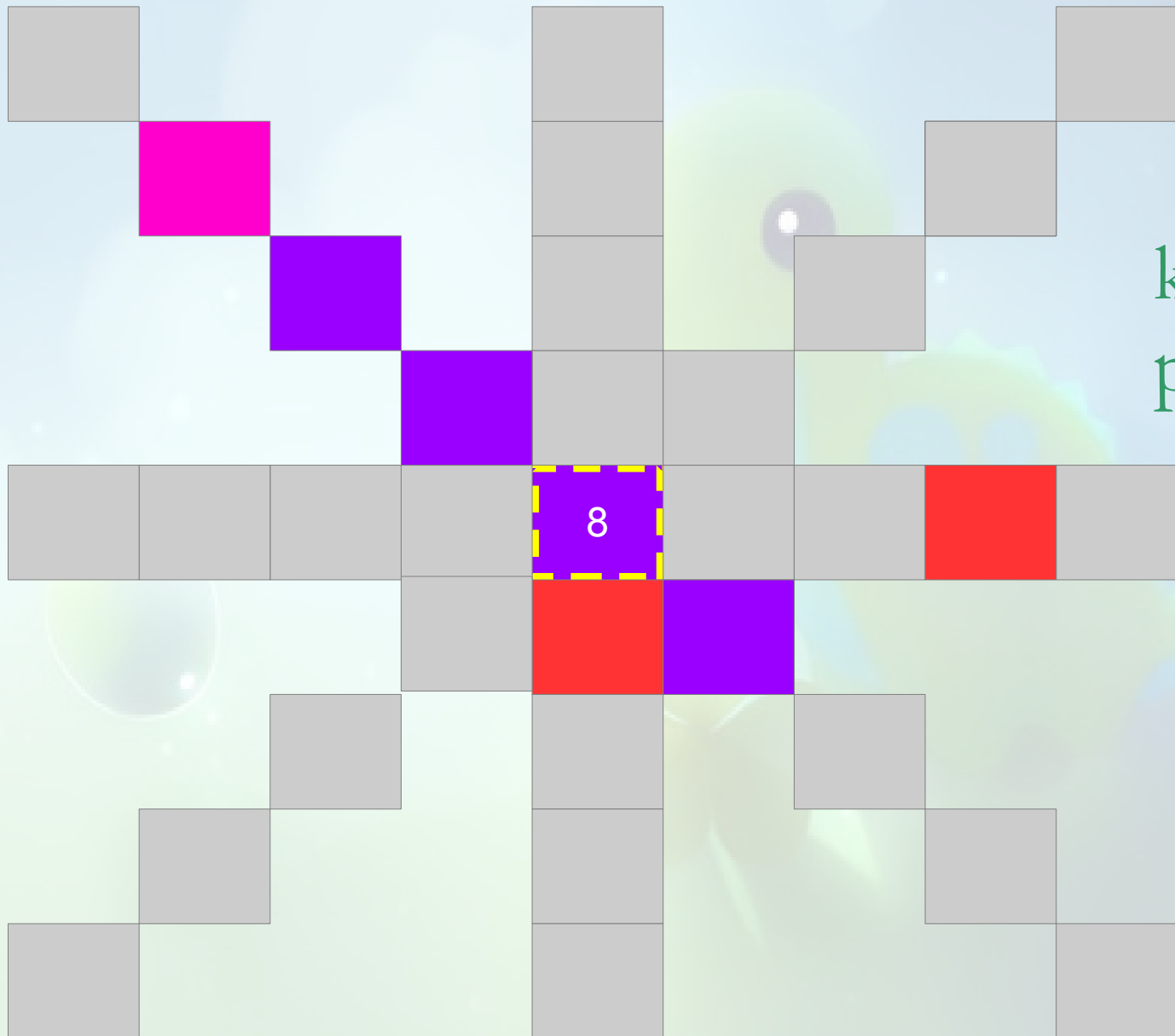
k-connects
possible: 6

Counting Saturated Viable K-Connects



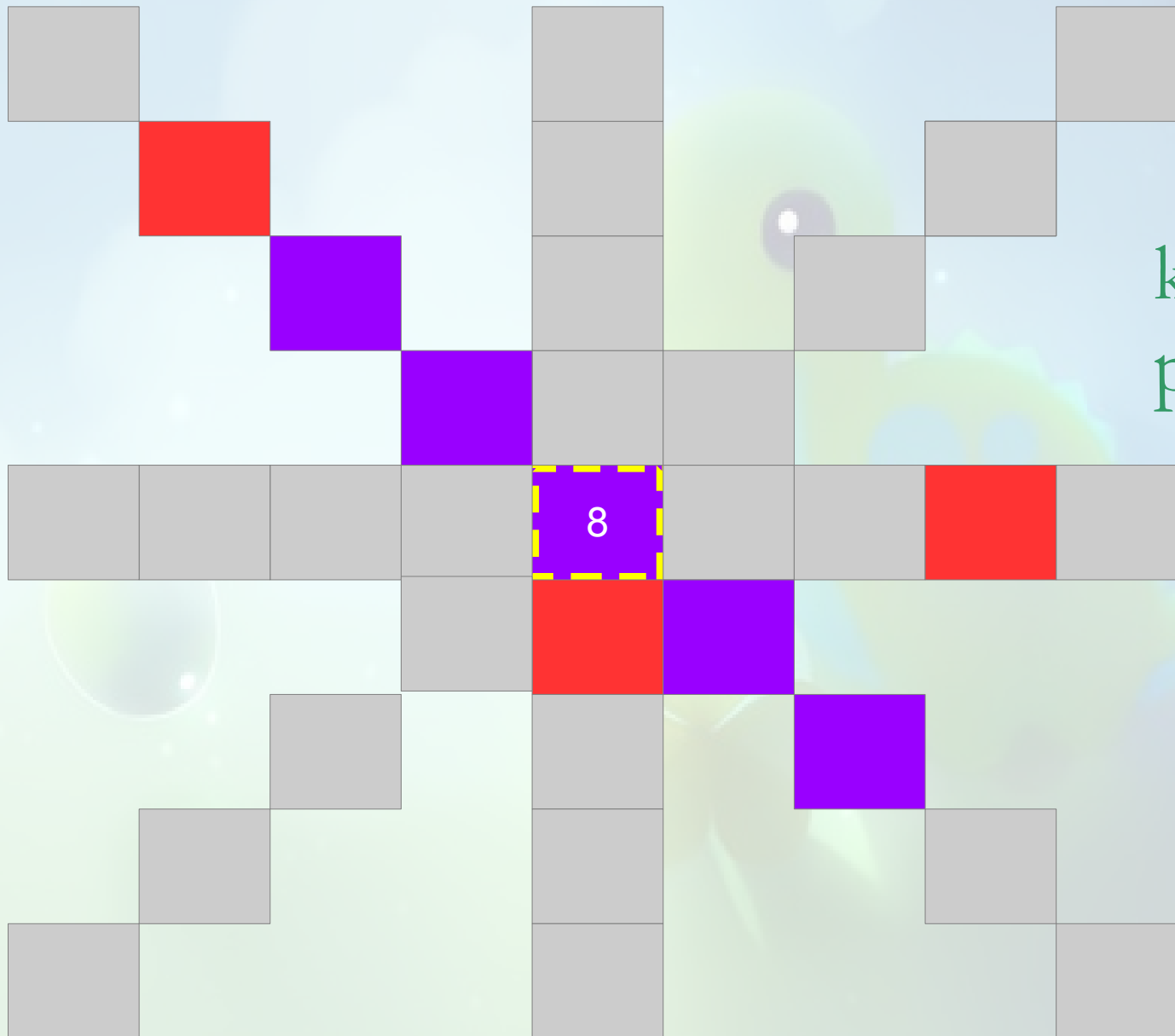
k-connects
possible: 7

Counting Saturated Viable K-Connects



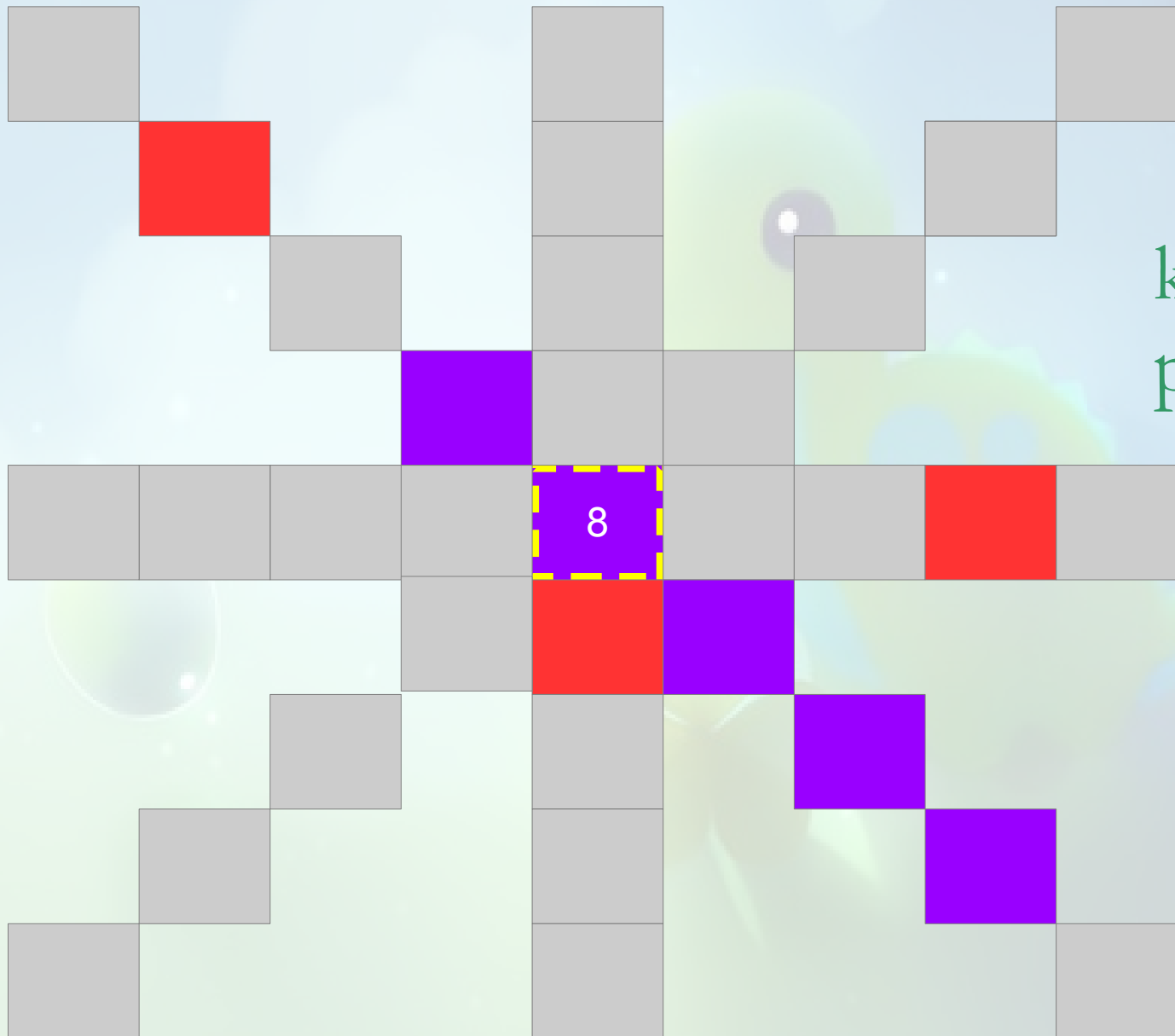
k-connects
possible: 8

Counting Saturated Viable K-Connects



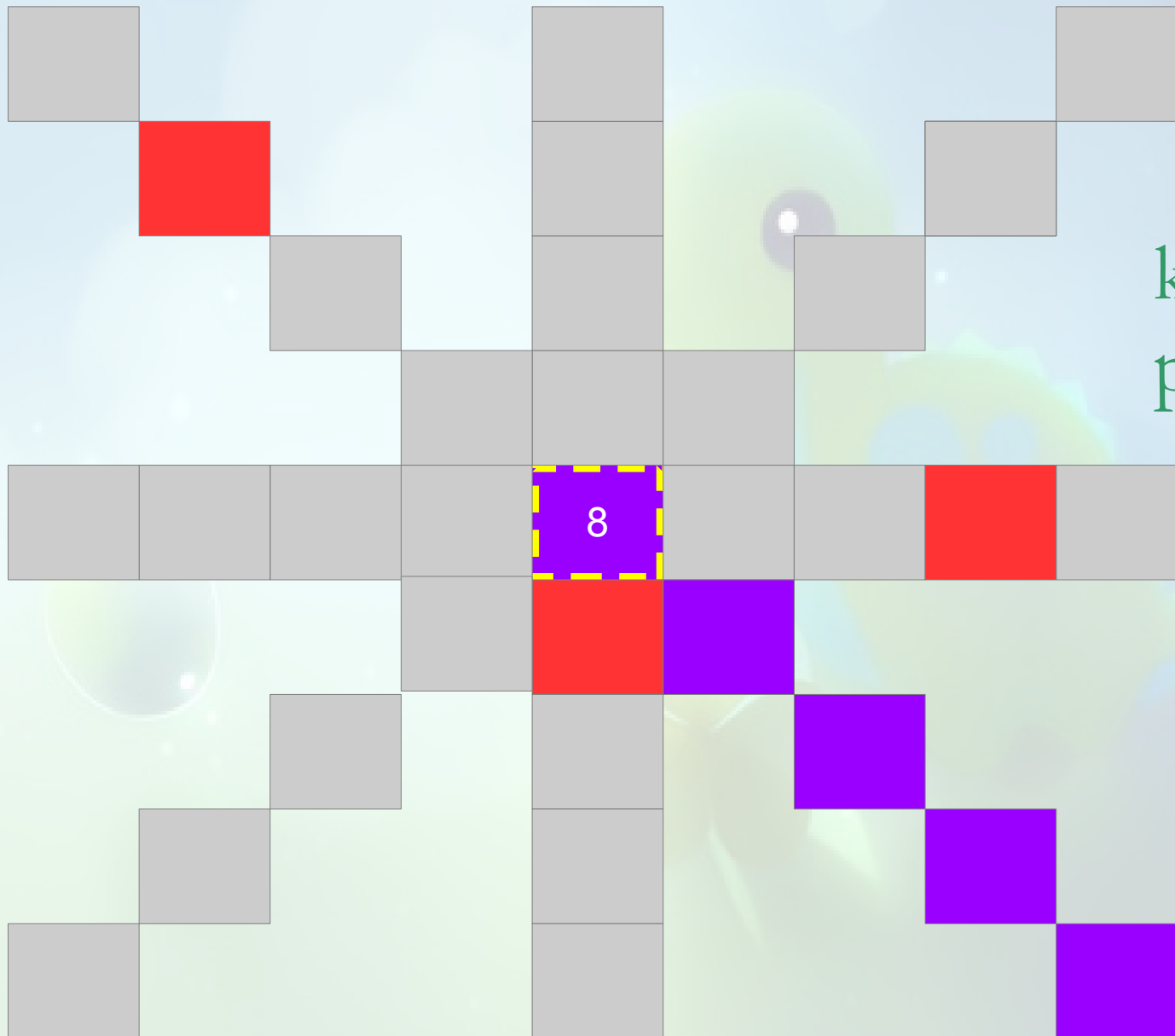
k-connects
possible: 8

Counting Saturated Viable K-Connects



k-connects
possible: 8

Counting Saturated Viable K-Connects



k-connects
possible: 8

Viabale K-connect Unsaturated Opener

Locations with most viable k-connects were chosen

Current opener algorithm

14 k-connects w/viable k-connect function and 5 full legs

versus

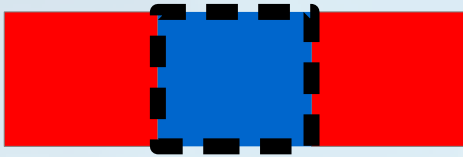
8 k-connects and 2 full legs with common

sense gaming principle
“always take center”

0:6	1:6	2:6	3:6	4:6	5:6	6:6	7:6	8:6
0:5	1:5	2:5	3:5	4:5	5:5	6:5	7:5	8:5
0:4	1:4	2:4	3:4	4:4	5:4	6:4	7:4	8:4
0:3	1:3	2:3	3:3	4:3	5:3	6:3	7:3	8:3
0:2	1:2	2:2	3:2	4:2	5:2	6:2	7:2	8:2
0:1	1:1	2:1	3:1	4:1	5:1	6:1	7:1	8:1
0:0	1:0	2:0	3:0	4:0	5:0	6:0	7:0	8:0

4:2/4:4 is optimal

Defensive Heuristics



AntiConnect

Detects two adjacent tiles

Considered wasted move if overly used

No progression drawback



k-2 chain

2-Stage Preemptive Attack

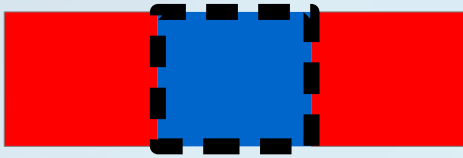
1) Attack one unoccupied end of a k-2 chain
Anticipating the enemy will complete a
k-1 chain next.

2) Seal the other end when ENEMY
plays k-1 chain

Concept designed on paper

High degree of control of outcome

Defensive Heuristics



AntiConnect

Detects two adjacent tiles

Considered wasted move if overly used

No progression drawback



k-1 chain

2-Stage Preemptive Attack

1) Attack one unoccupied end of a k-2 chain

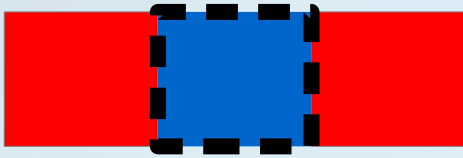
Anticipating the enemy will complete a k-1 chain next.

2) Seal the other end when ENEMY plays k-1 chain

Concept designed on paper

High degree of control of outcome

Defensive Heuristics

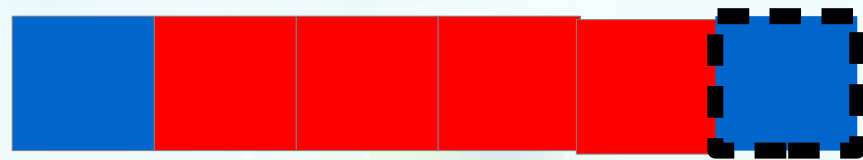


AntiConnect

Detects two adjacent tiles

Considered wasted move if overly used

No progression drawback



forced draw

2-Stage Preemptive Attack

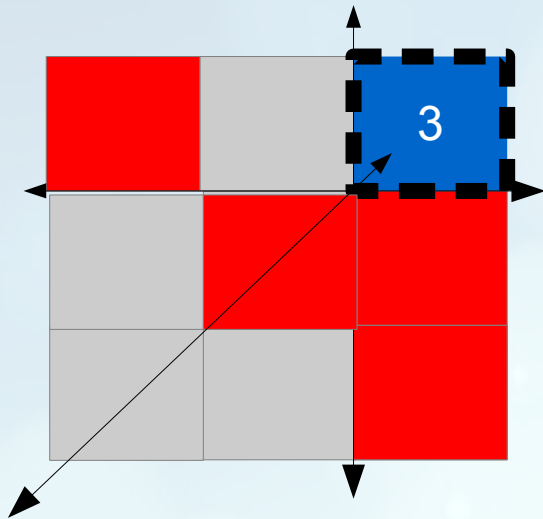
1) Attack one unoccupied end of a $k-2$ chain
Anticipating the enemy will complete a $k-1$ chain next.

2) Seal the other end when ENEMY plays $k-1$ chain

Concept designed on paper

High degree of control of outcome

Defensive Heuristics (cont)



Ambiguous Defensive Attack

Luckasaurus Rex feasted on these spots
Used viable k-connect saturated function to discover them

Optimistic that two or more moves
supplanted by one actual move

Attack Heuristics

IWin

Detects k-length complete saturated chain

Highest weights

Eval Function added bonus based
on depth of IDS:

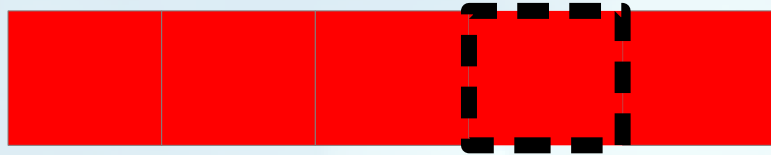
```
iwin=IWin()
```

```
w=highest weight score
```

```
//score=base + bonus scores
```

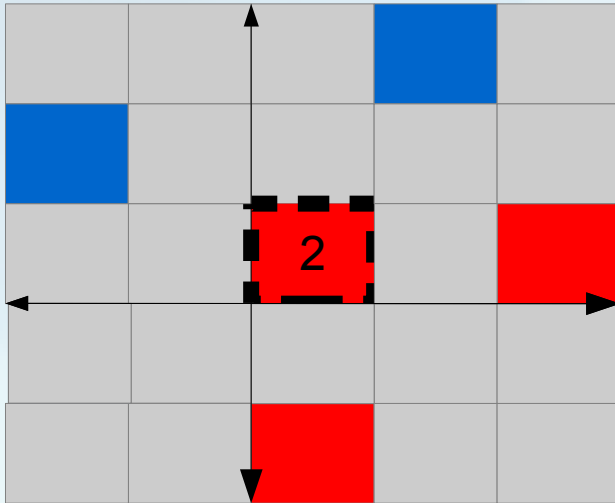
```
score=w*iwin+w/depth*iwin
```

Bonus helps converge towards winning
with the fewest moves (e.g. 4 move win
is more valuable than 6 move win)



Attack Heuristics (cont)

k=5



Maximize k-connect saturated

Awards points to the number of possible saturated k-connects.

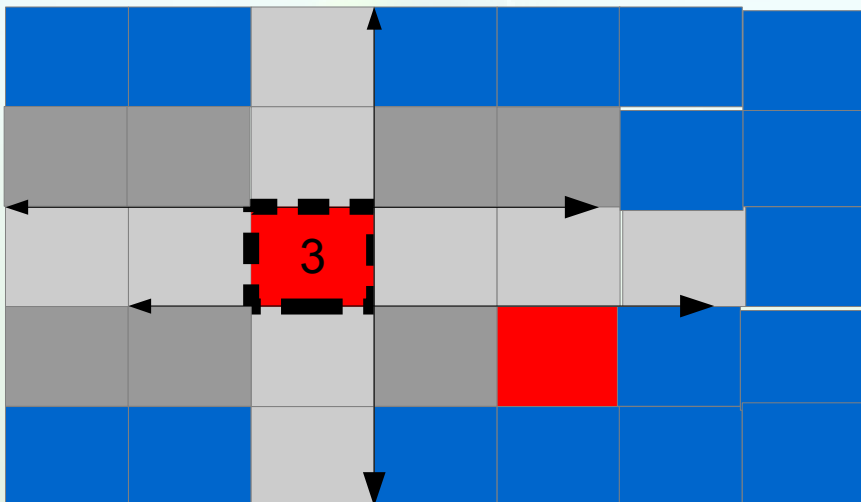
Primary progressive attack heuristic

Moderate weight value

Biased towards played area

Same as *Ambiguous Attack* except searches Lucasaurus Rex tiles

k=5



Maximize k-connect unsaturated

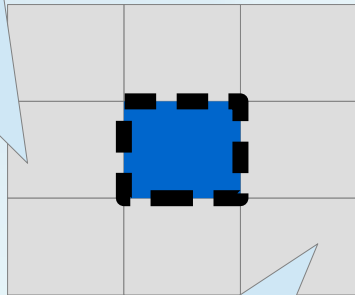
Secondary progressive attack heuristic

Lower weight value since doesn't progress towards win.

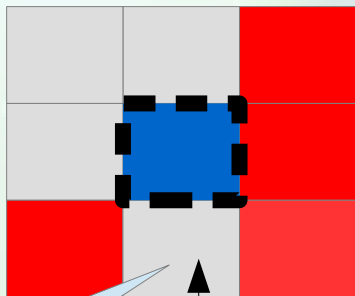
Biased towards edge of the board
drawback

Foresight Analysis

IsKConnectViableForEnemy?



IsKConnectViableForEnemy?



IsWell?

Used for quiescent analysis and negative feedback scoring
Checked existence of k-connect surrounding the target tile.

Current build limited to 1 step Ahead

Wanted to detect 2 steps ahead but no time

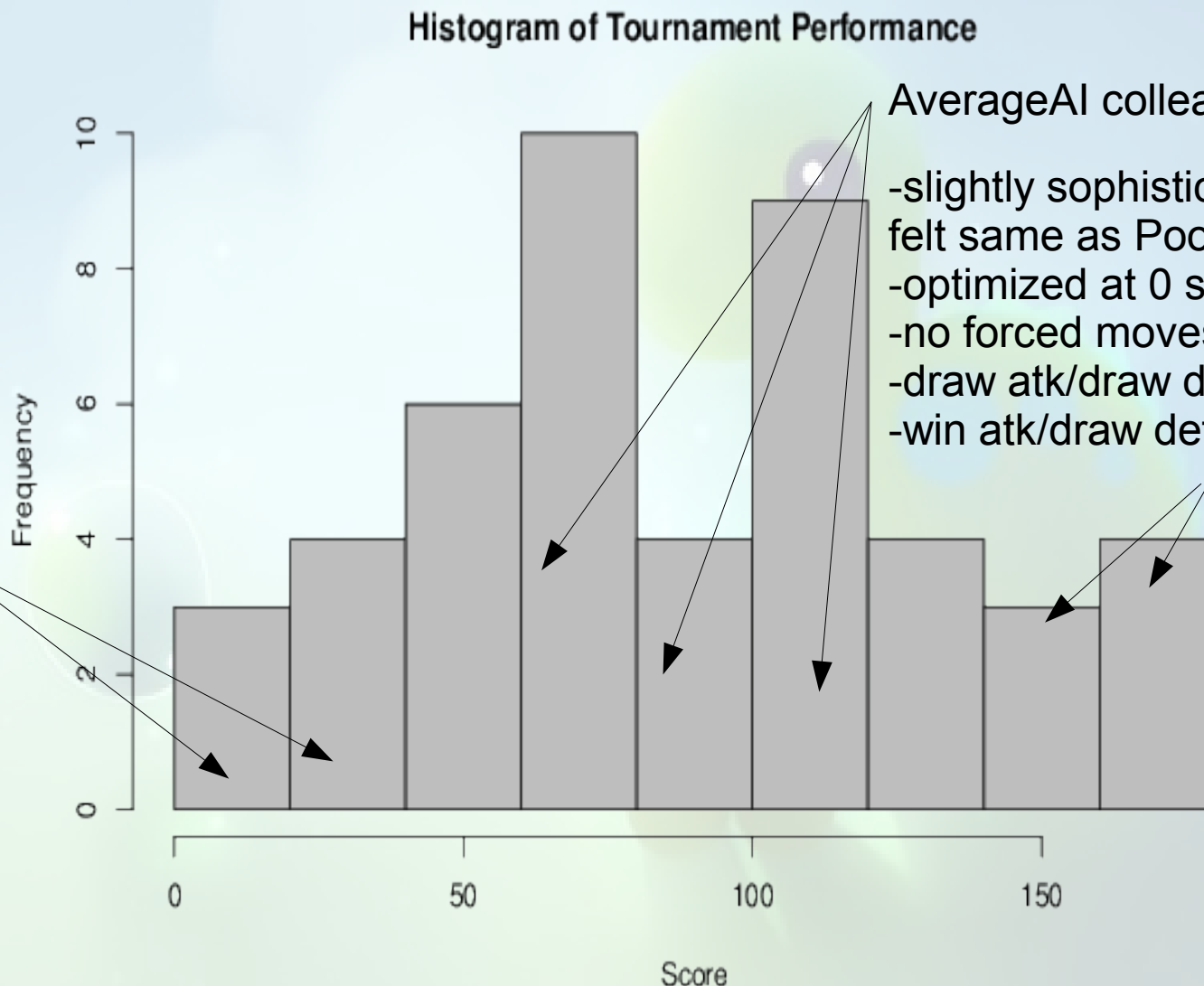
Flaw that it doesn't preform global analysis

Zero scores for wells in gravity

Guesstimating My Competition

- Time spent the most
- Worried about immediate results
- Mandatory win atk/win def

PoorAI Colleagues



AverageAI colleagues

- slightly sophisticated heuristics but felt same as PoorAI
- optimized at 0 steps ahead
- no forced moves
- draw atk/draw def acceptable
- win atk/draw def ideal

GoodAI colleagues

- Likely forced moves
- optimized for ≥ 1 step ahead
- draw atk/draw def acceptable
- considered draw=win

The tournament motivates selecting algorithms to defeat 80% instead of top 20% (e.g. avoid wedge defensive attack). Questions? Send to orsonteodoro@yahoo.com