

Computing Win Probabilities in Niedermaier's Dart Game

Jay Pantone
Marquette University

joint work with

*Christian Bean, Keele University
Émile Nadeau, RUV*

Andy Niedermaier's Dart Game

At Permutation Patterns 2022, Andy Niedermaier introduced the following dart game.

m players line up in front of a dart board

Andy Niedermaier's Dart Game

At Permutation Patterns 2022, Andy Niedermaier introduced the following dart game.

m players line up in front of a dart board

Player 1 throws, moves to the back of the line.

Andy Niedermaier's Dart Game

At Permutation Patterns 2022, Andy Niedermaier introduced the following dart game.

m players line up in front of a dart board

Player 1 throws, moves to the back of the line.

Player 2 throws. If it's closer to the bullseye than Player 1 (a “hit”), they stay in the game and move to the back of the line. If it's further (a “miss”), they are eliminated.

Andy Niedermaier's Dart Game

At Permutation Patterns 2022, Andy Niedermaier introduced the following dart game.

m players line up in front of a dart board

Player 1 throws, moves to the back of the line.

Player 2 throws. If it's closer to the bullseye than Player 1 (a “hit”), they stay in the game and move to the back of the line. If it's further (a “miss”), they are eliminated.

The player in front throws, and if their throw is closer to the bullseye than *all previous throws*, they stay in and move to the back. Otherwise they are eliminated.

Andy Niedermaier's Dart Game

At Permutation Patterns 2022, Andy Niedermaier introduced the following dart game.

m players line up in front of a dart board

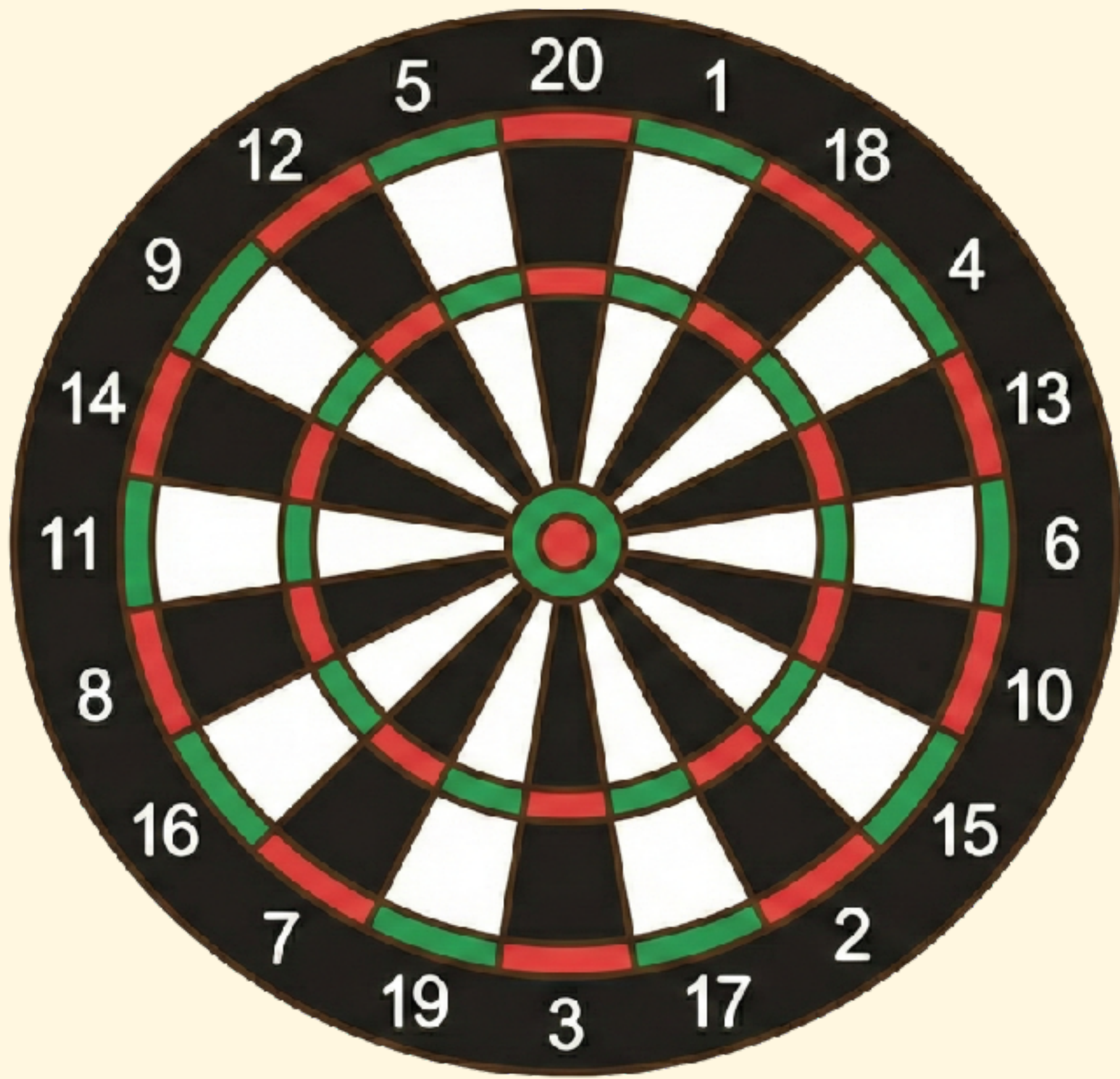
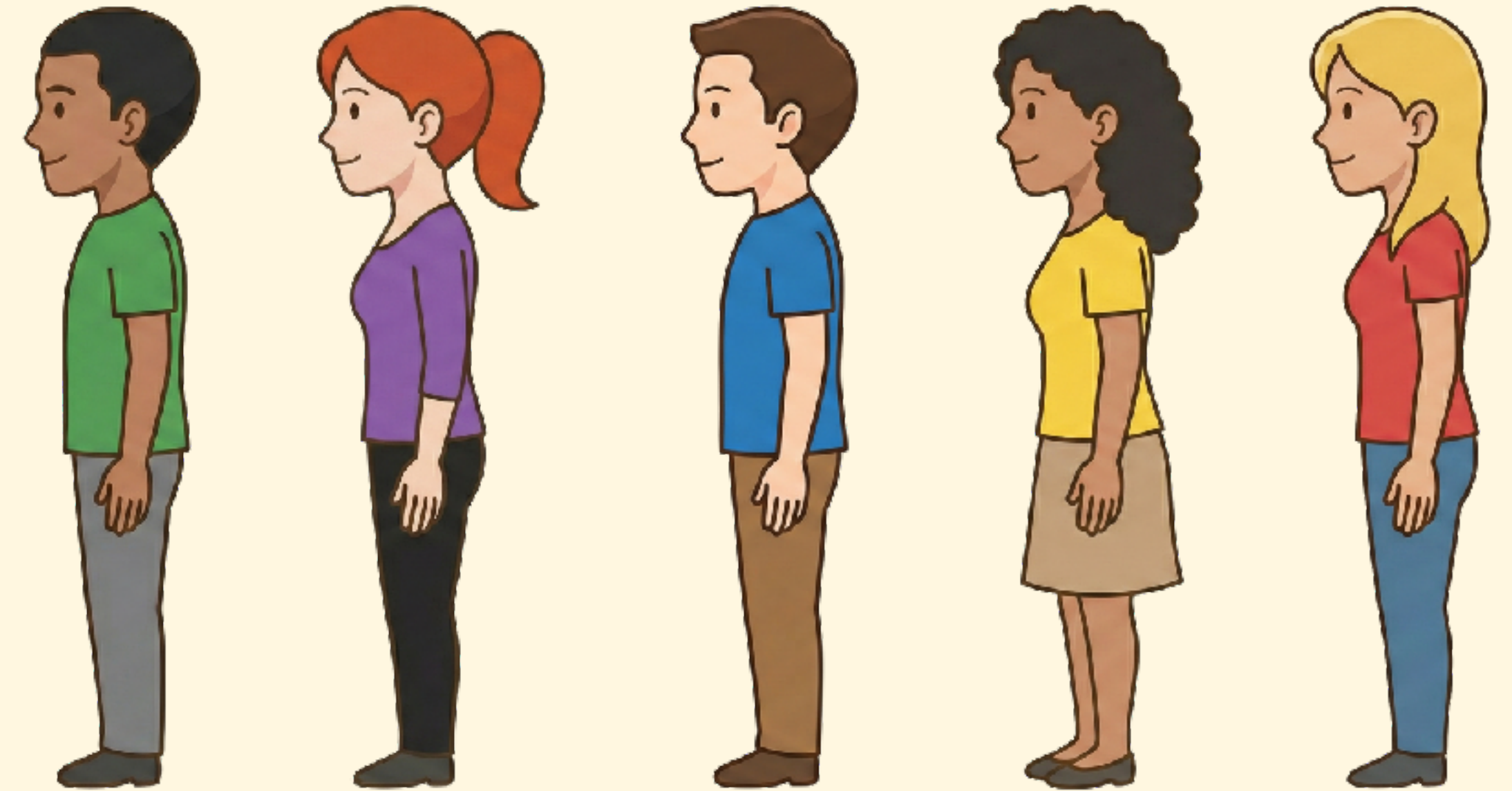
Player 1 throws, moves to the back of the line.

Player 2 throws. If it's closer to the bullseye than Player 1 (a “hit”), they stay in the game and move to the back of the line. If it's further (a “miss”), they are eliminated.

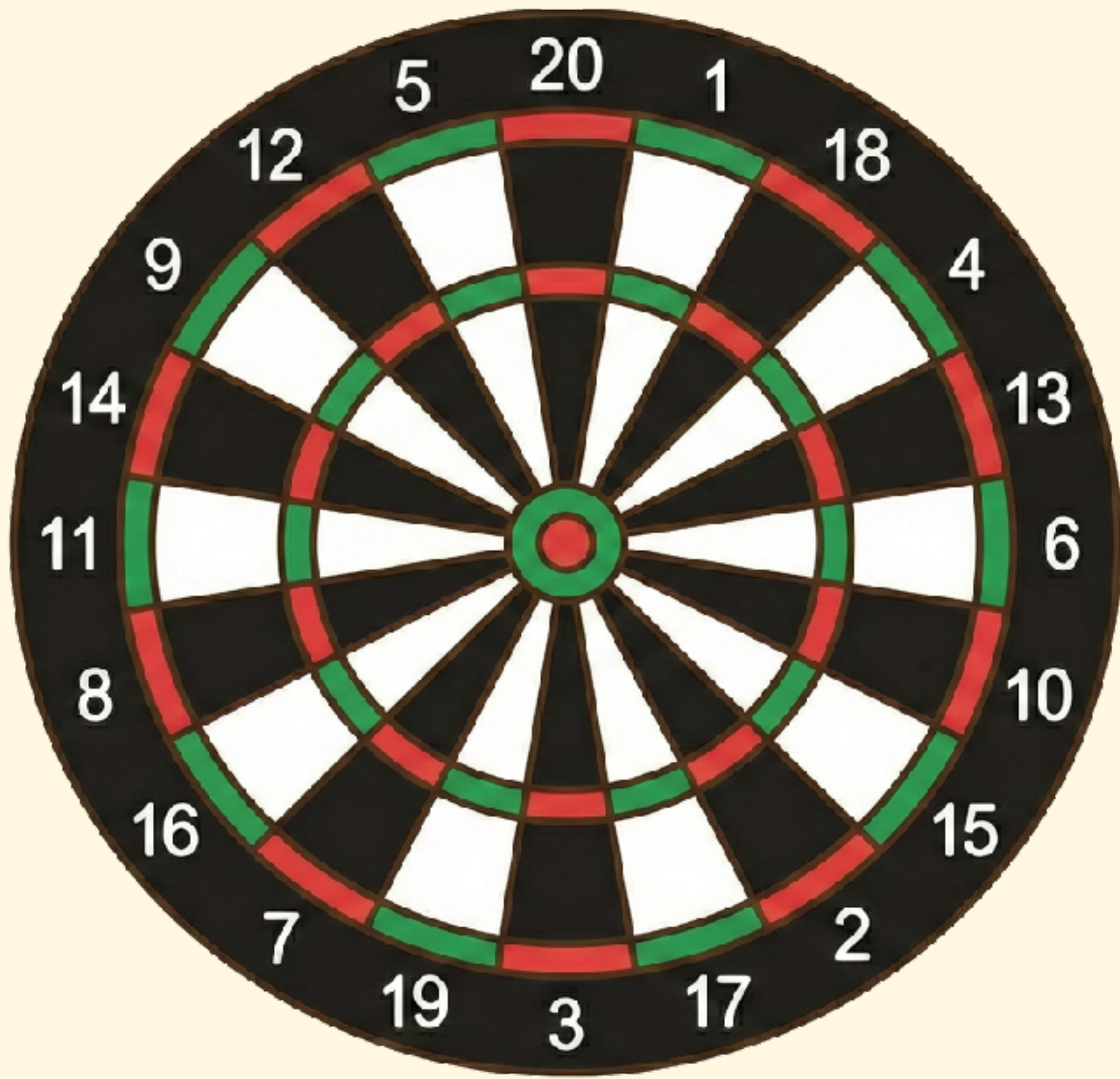
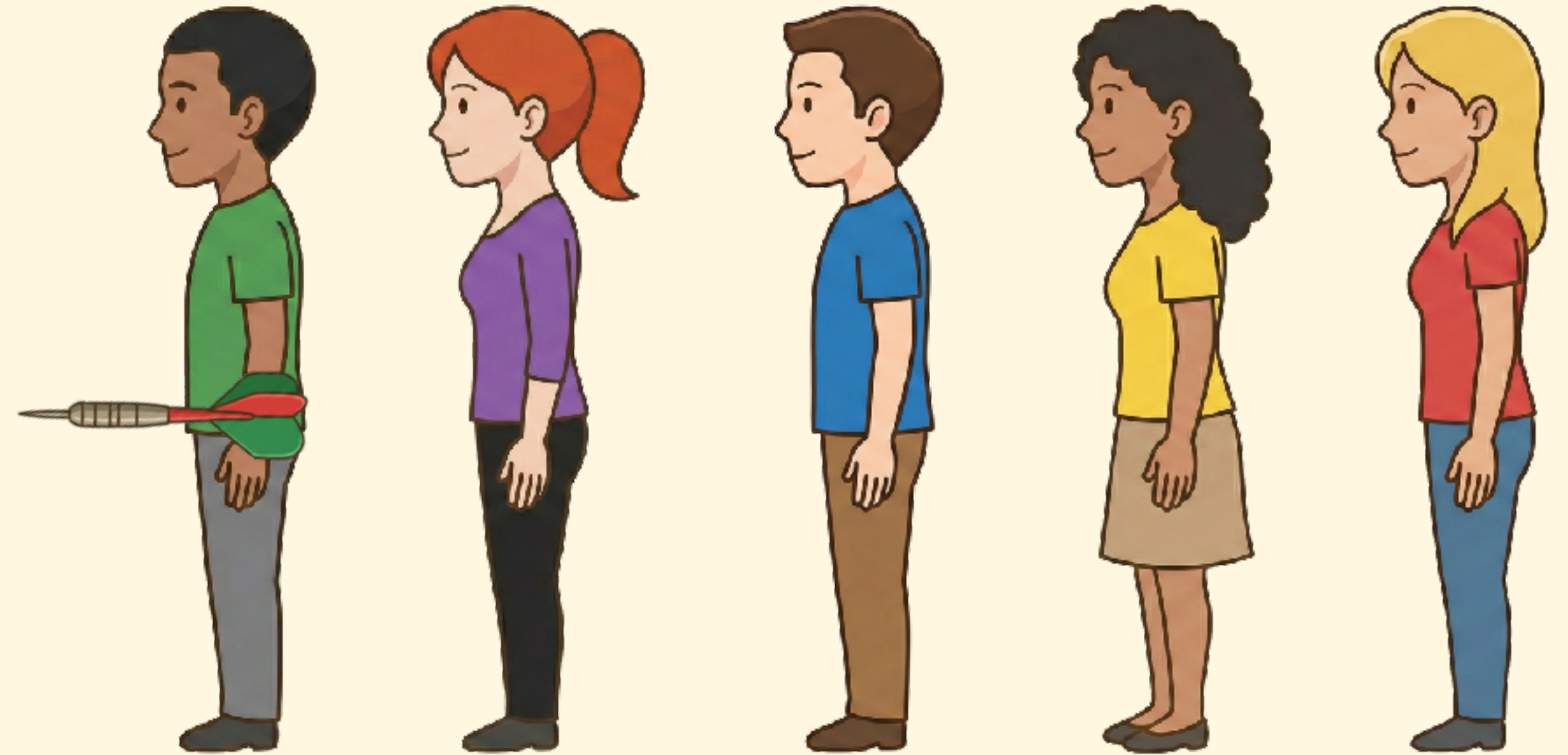
The player in front throws, and if their throw is closer to the bullseye than *all previous throws*, they stay in and move to the back. Otherwise they are eliminated.

The last player remaining is the winner.

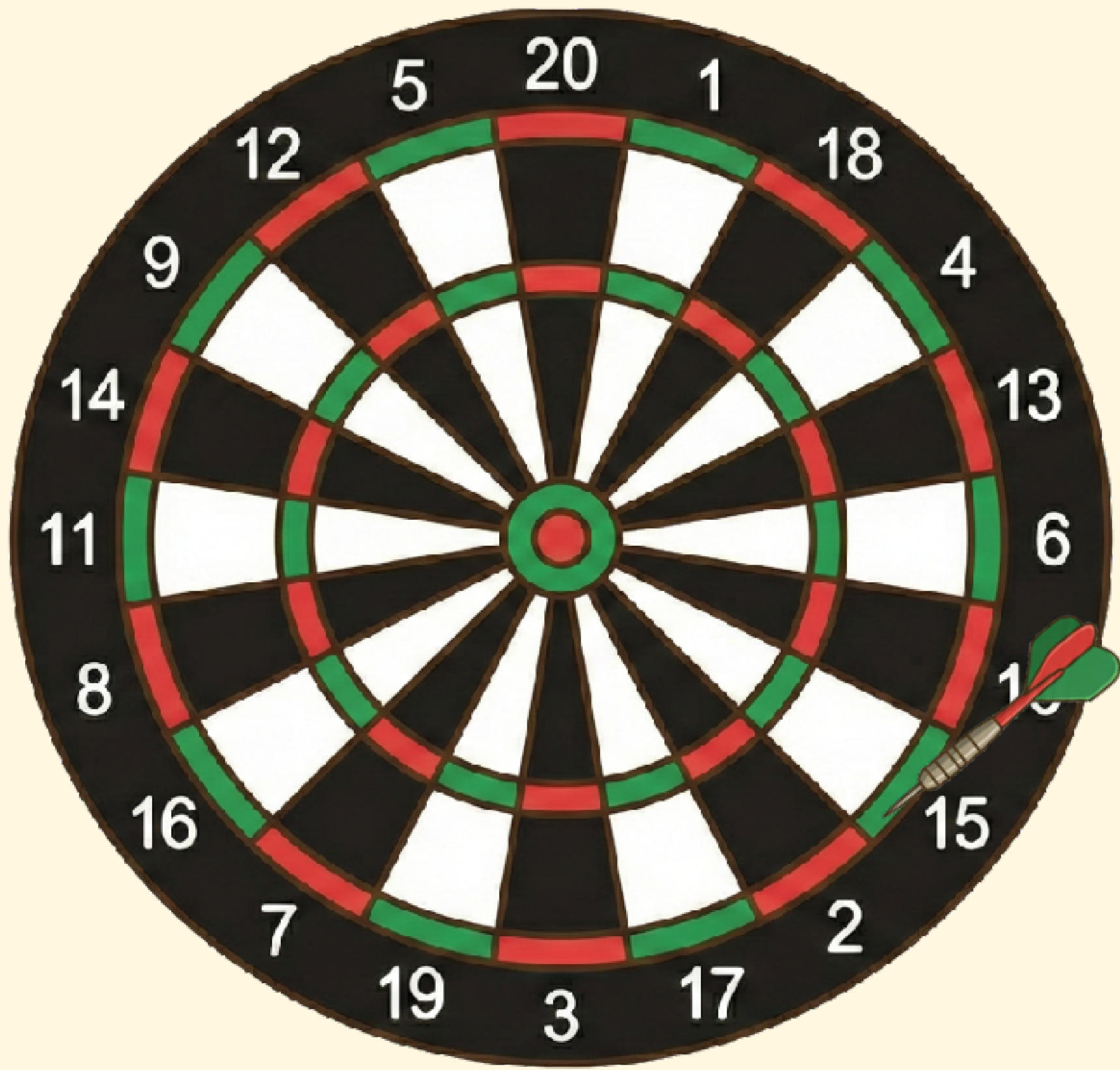
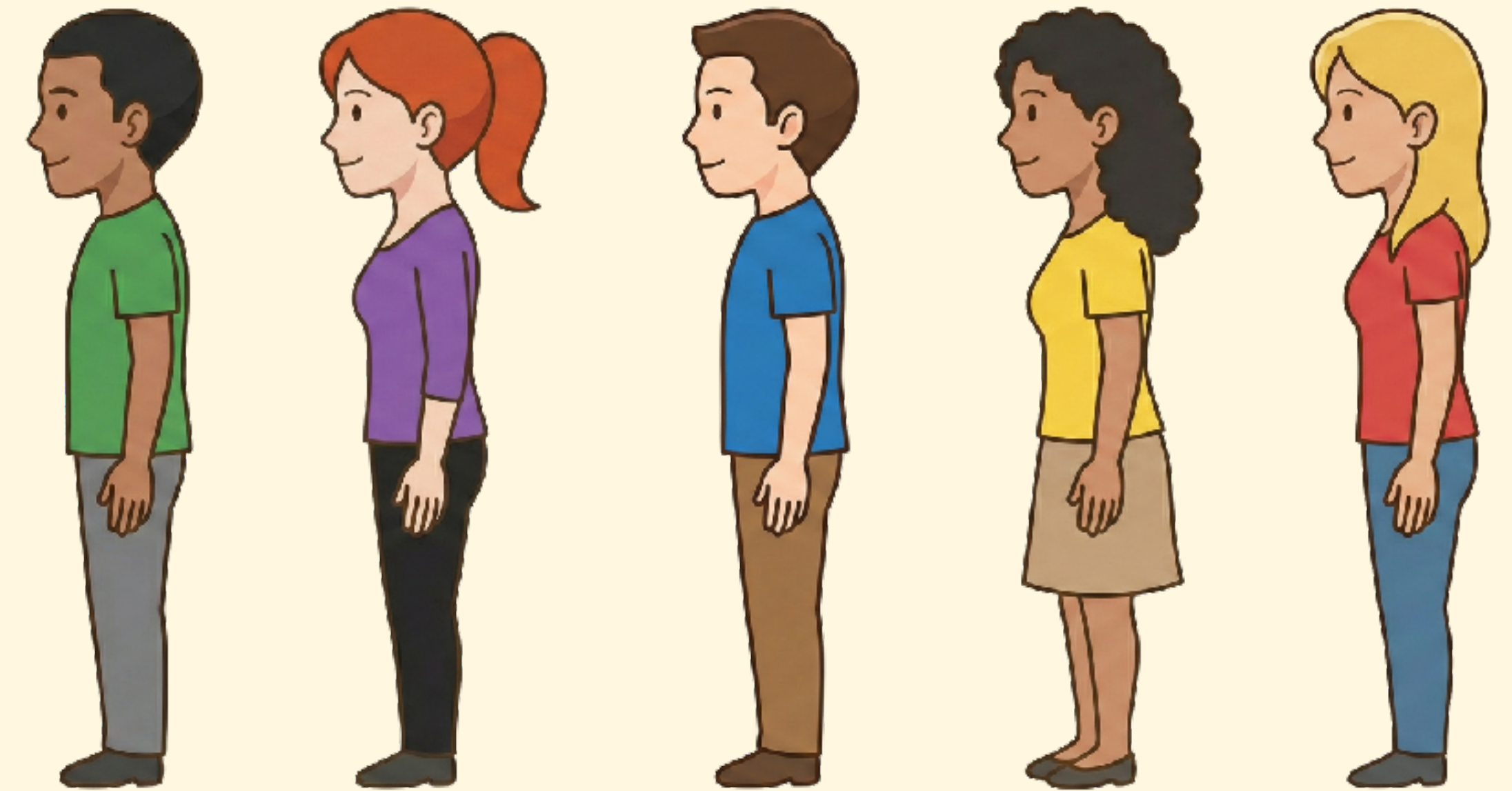
Andy Niedermaier's Dart Game



Andy Niedermaier's Dart Game

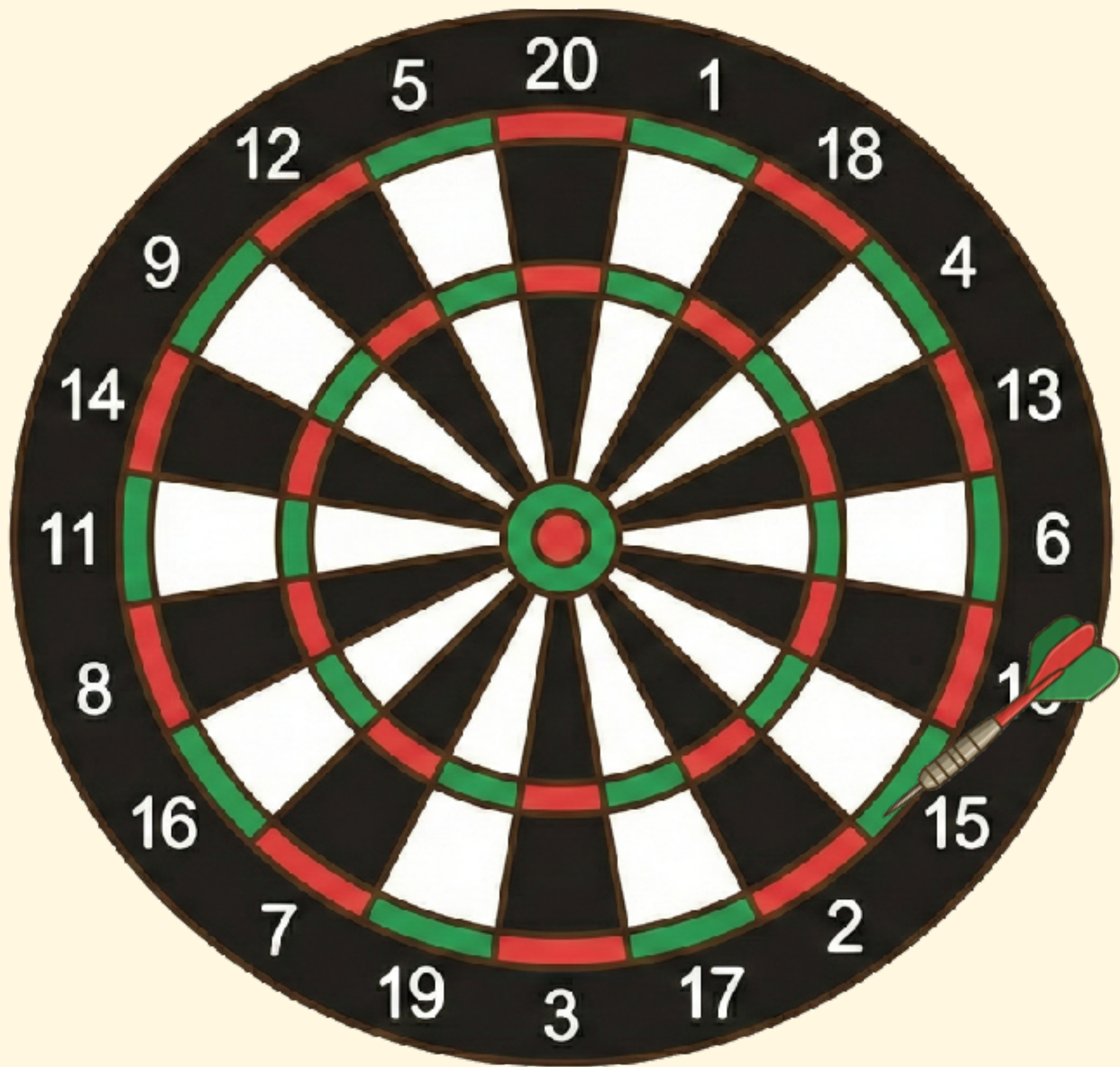
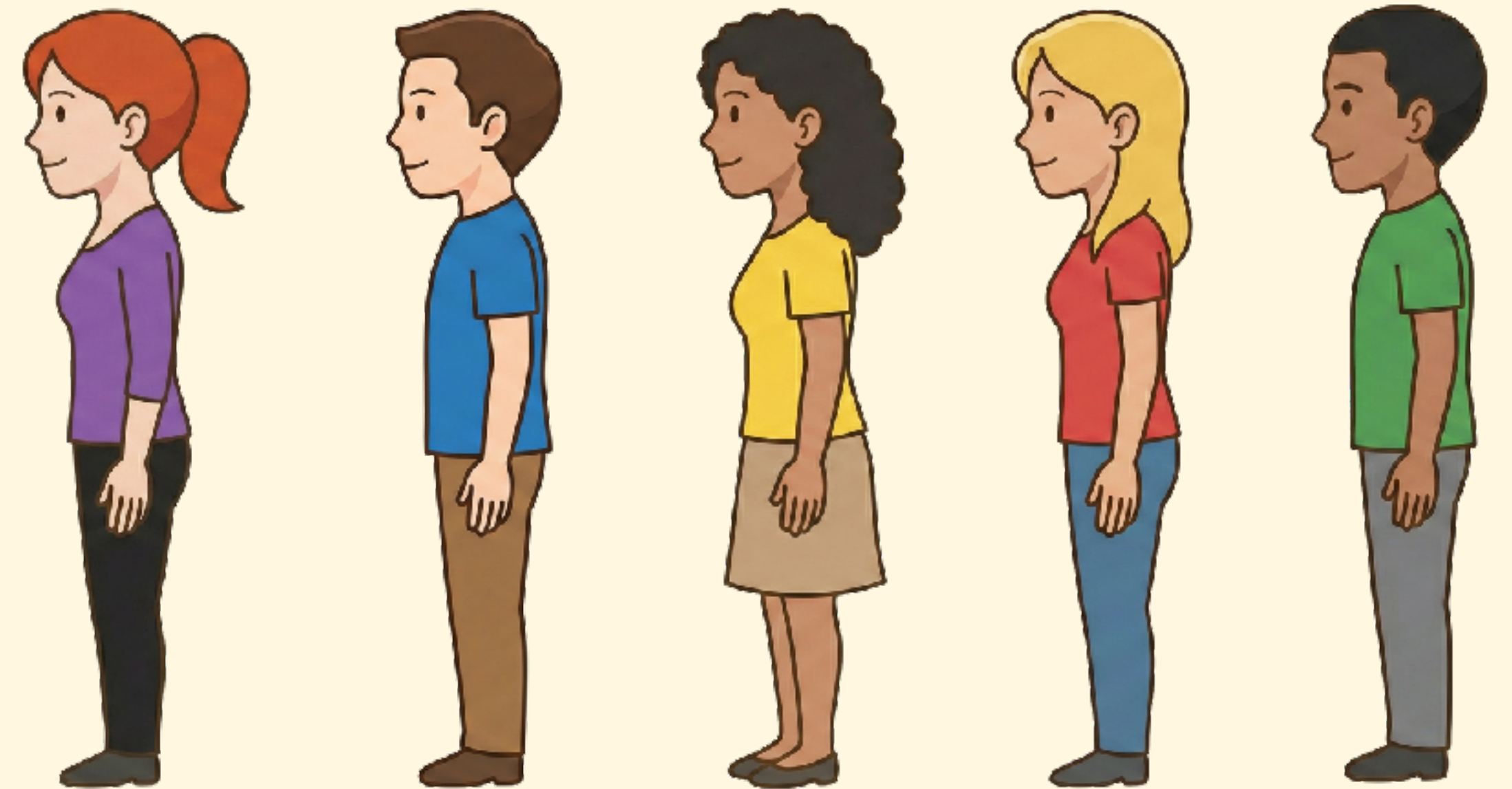


Andy Niedermaier's Dart Game



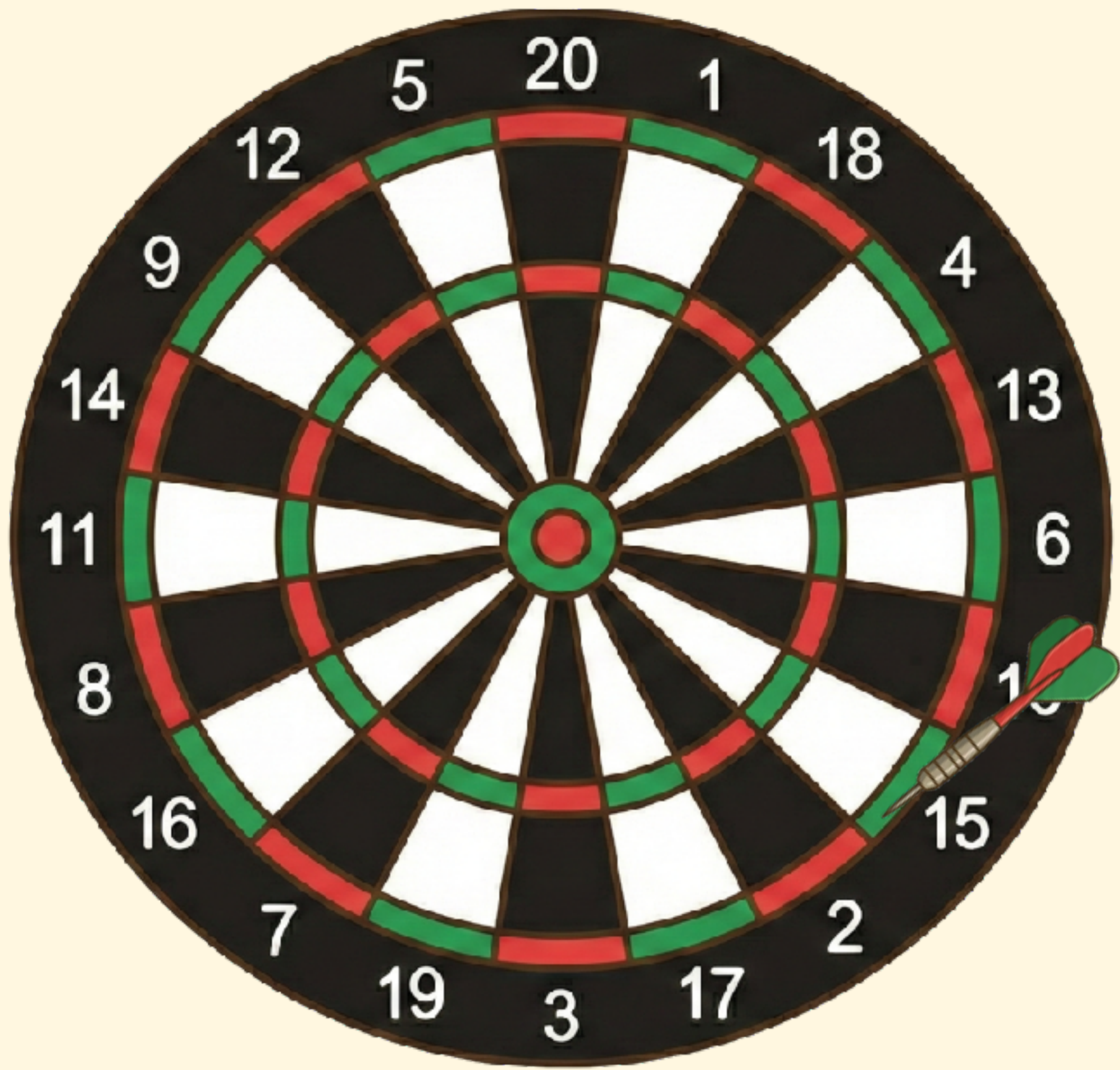
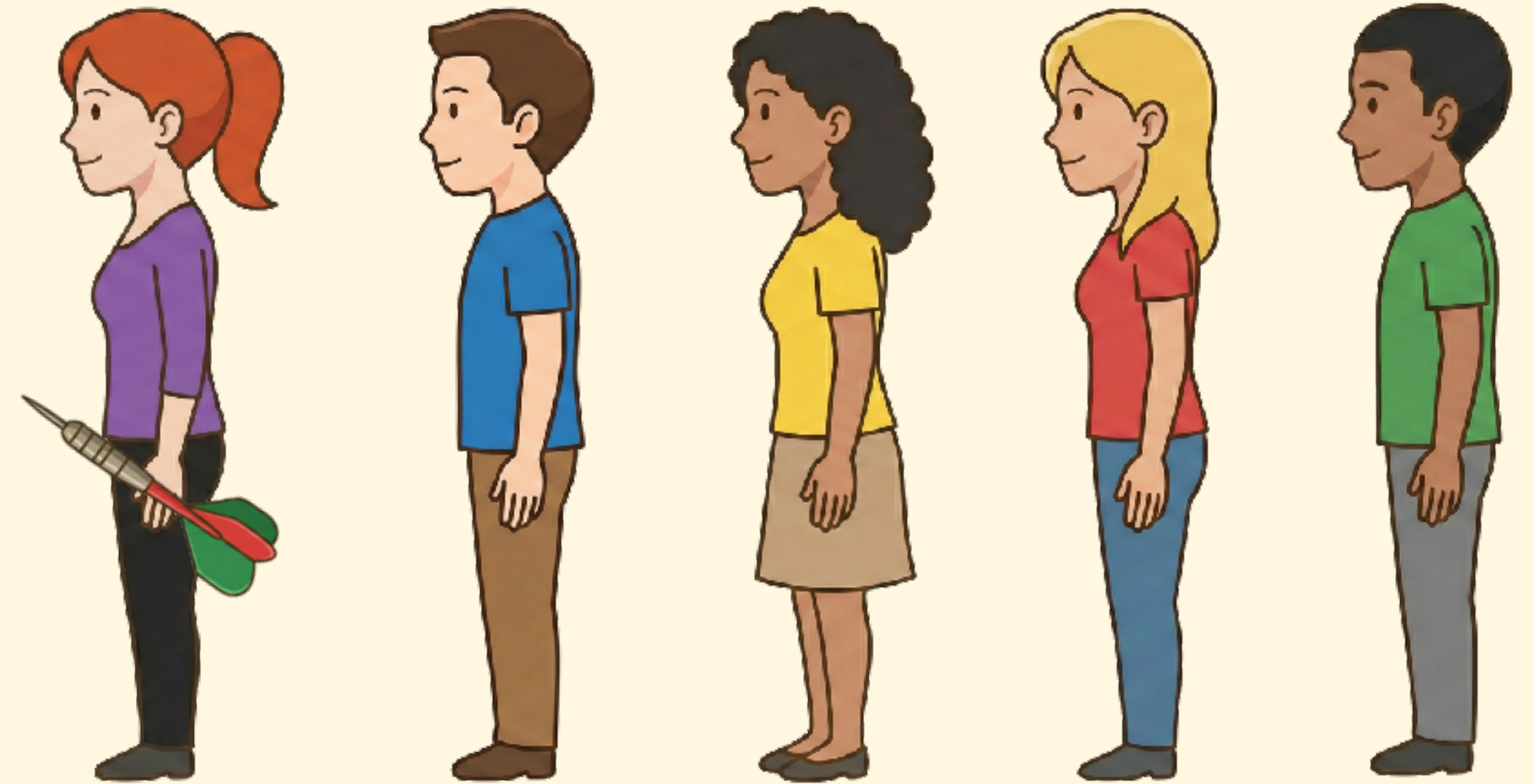
Player 1 hit.

Andy Niedermaier's Dart Game

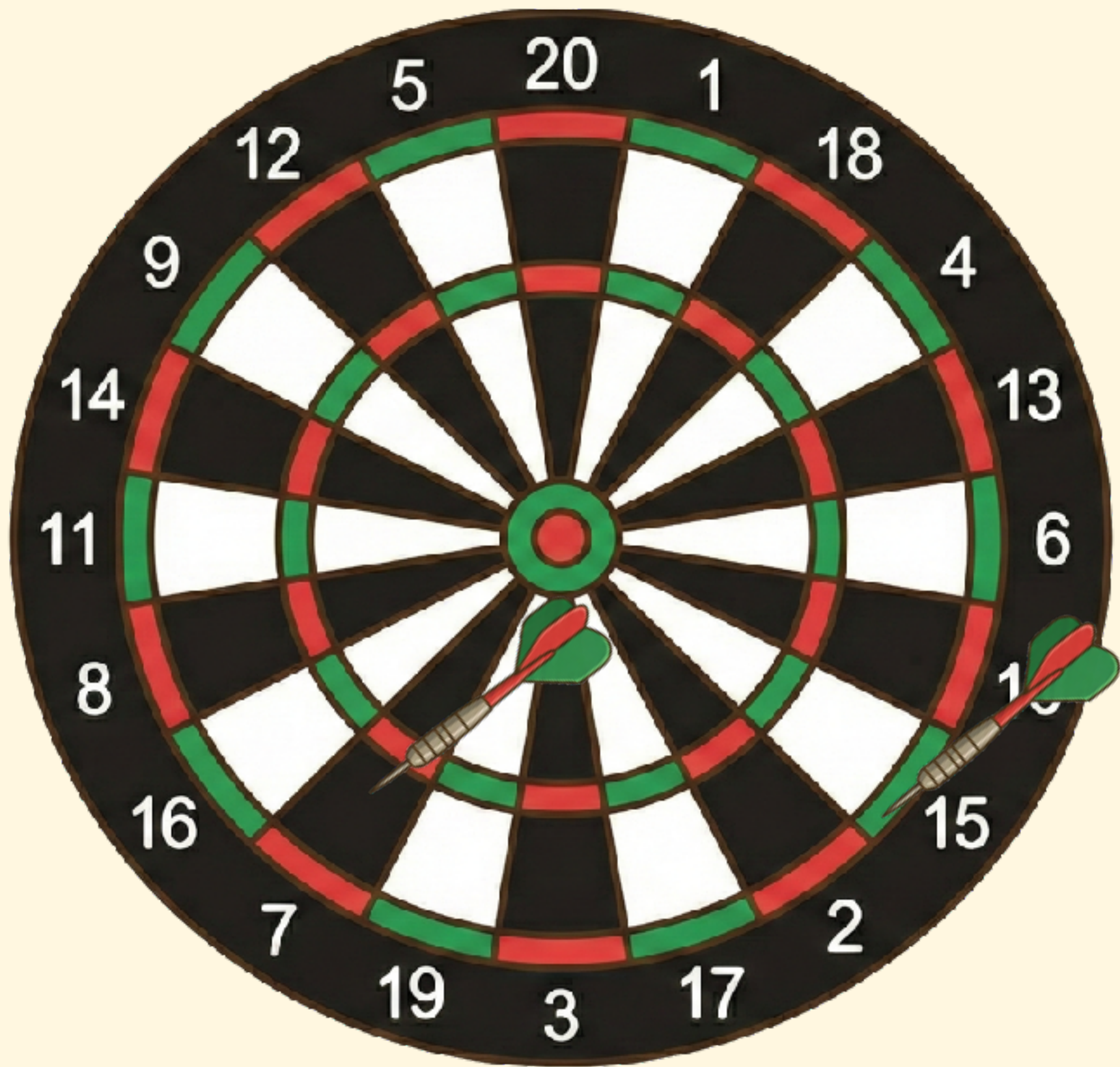
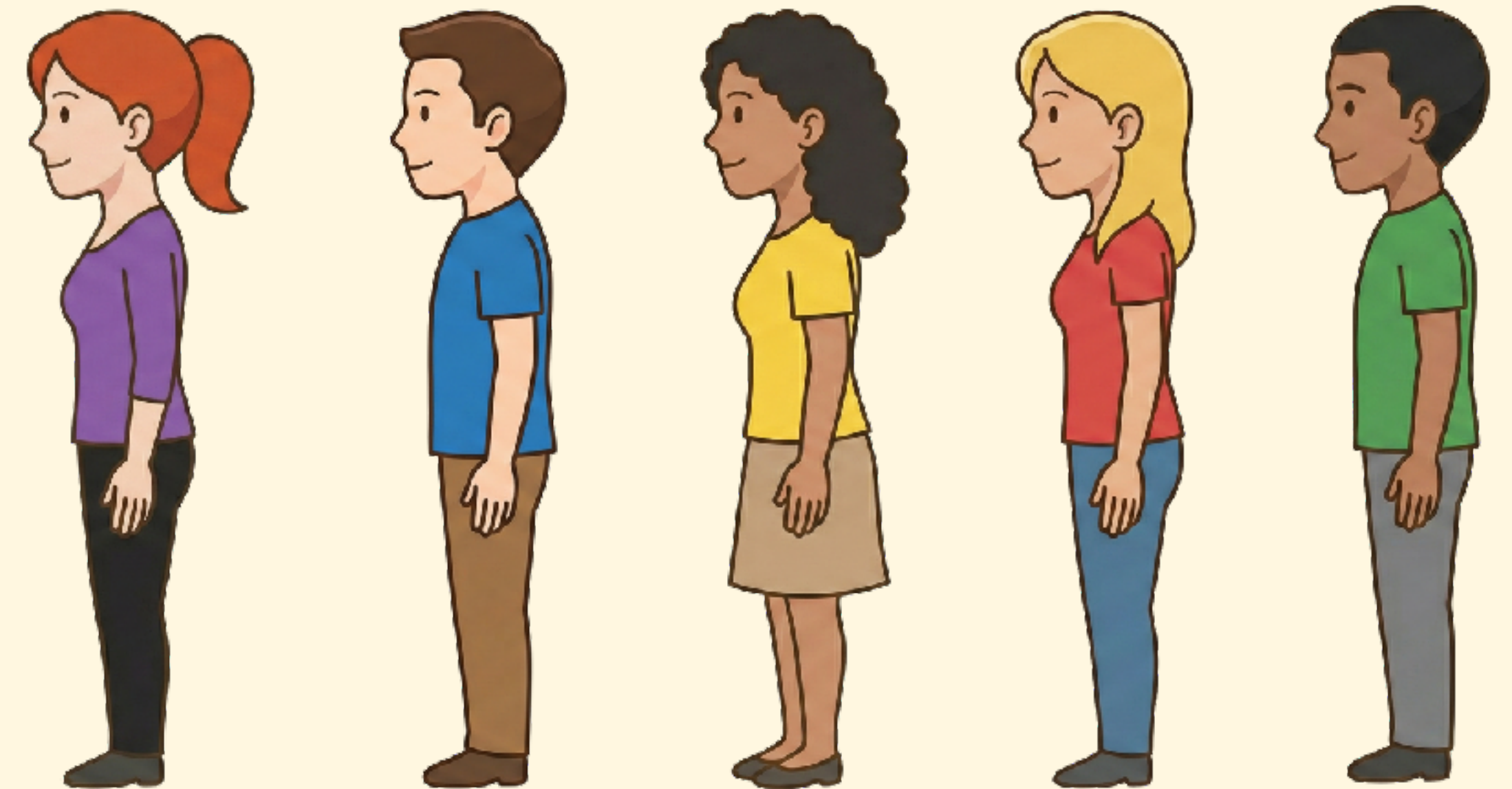


Player 1 hit.

Andy Niedermaier's Dart Game

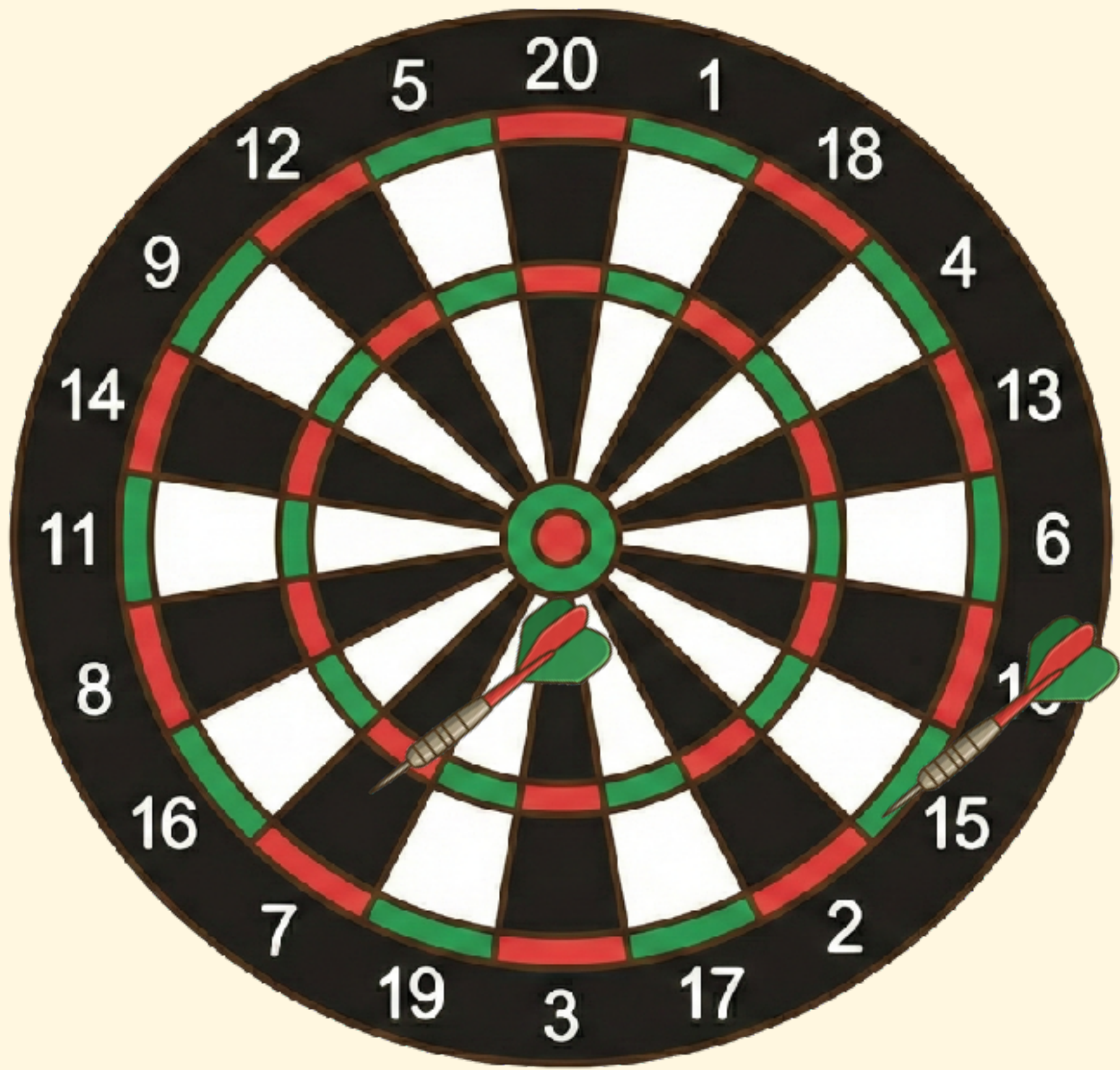
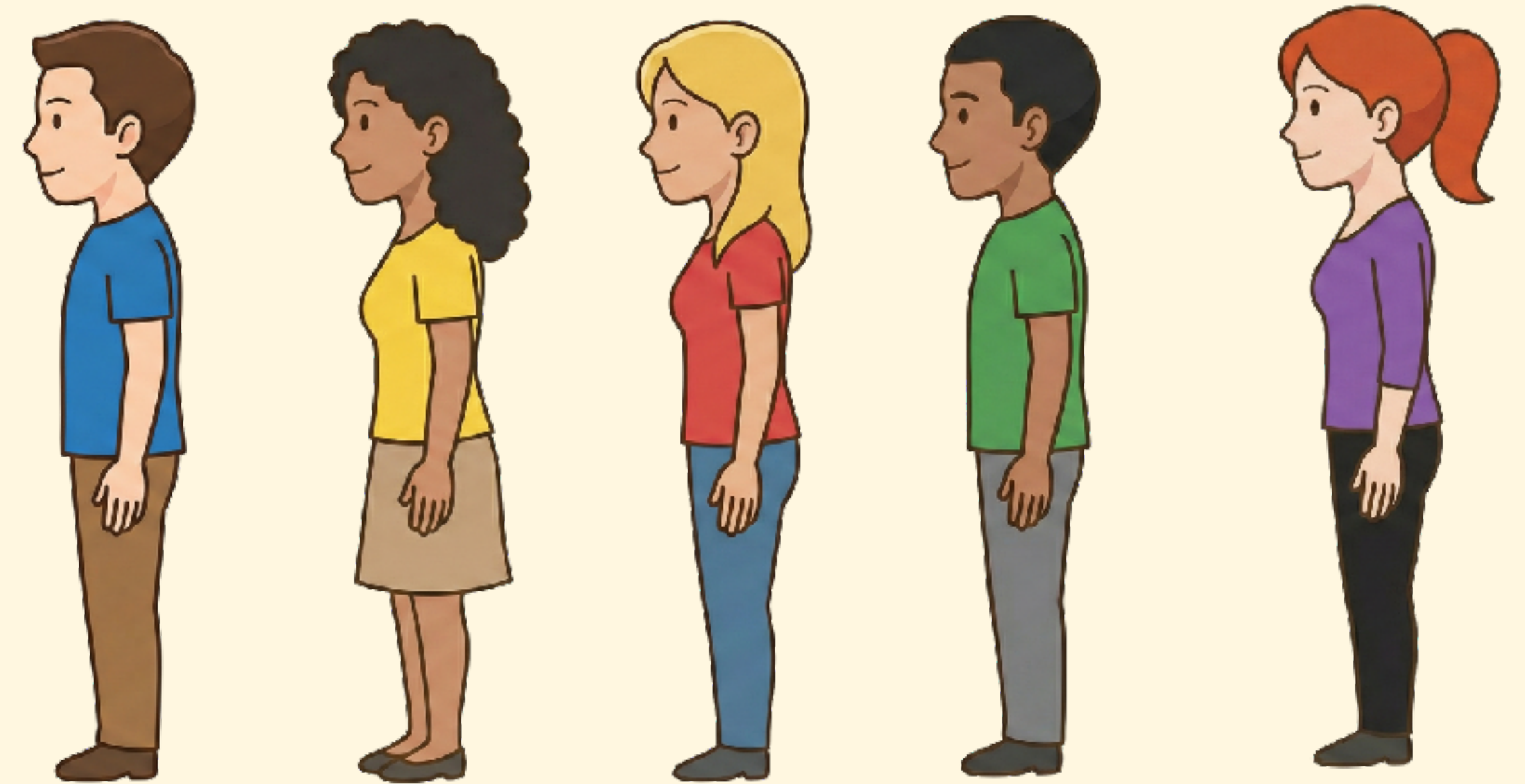


Andy Niedermaier's Dart Game



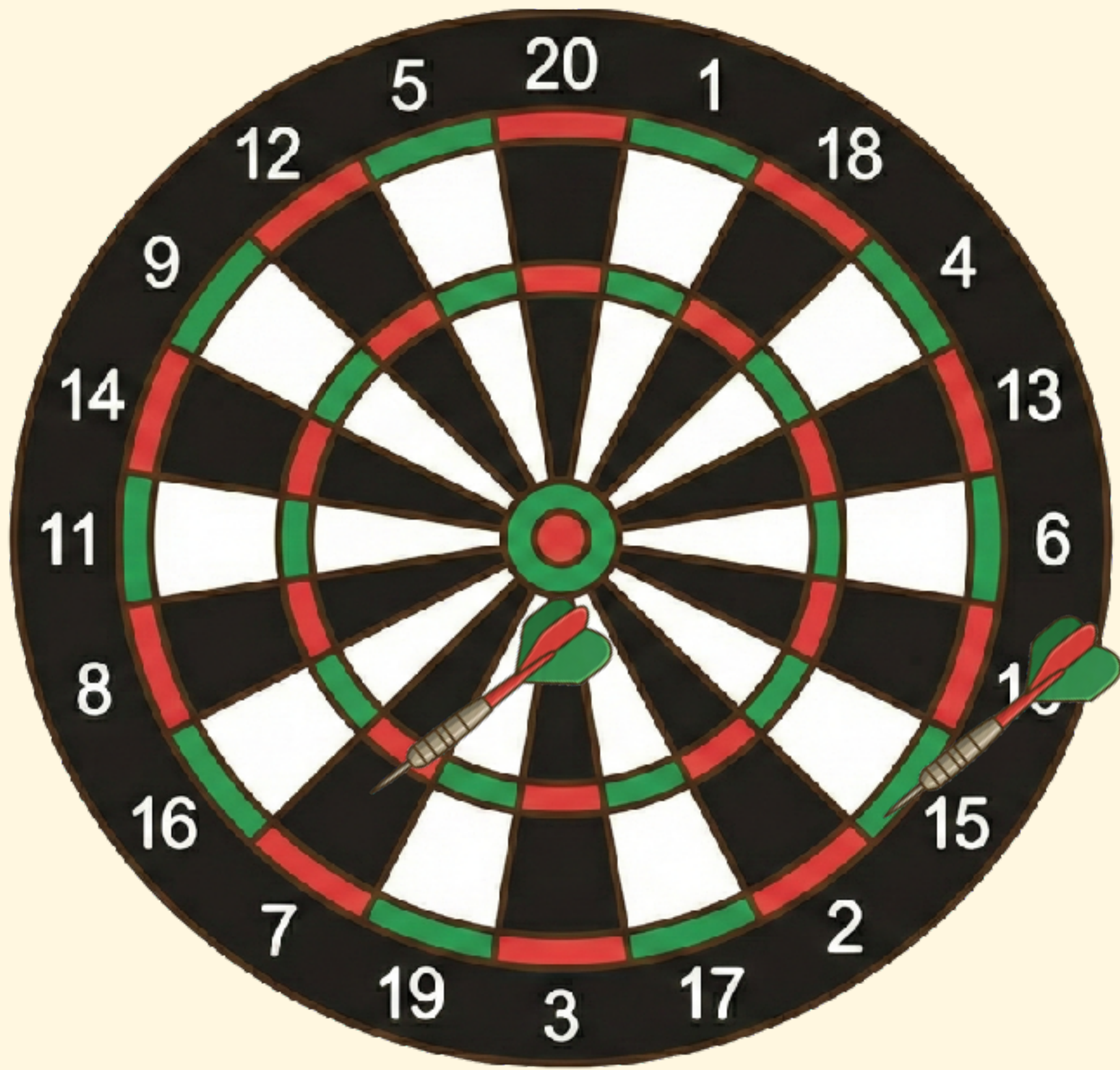
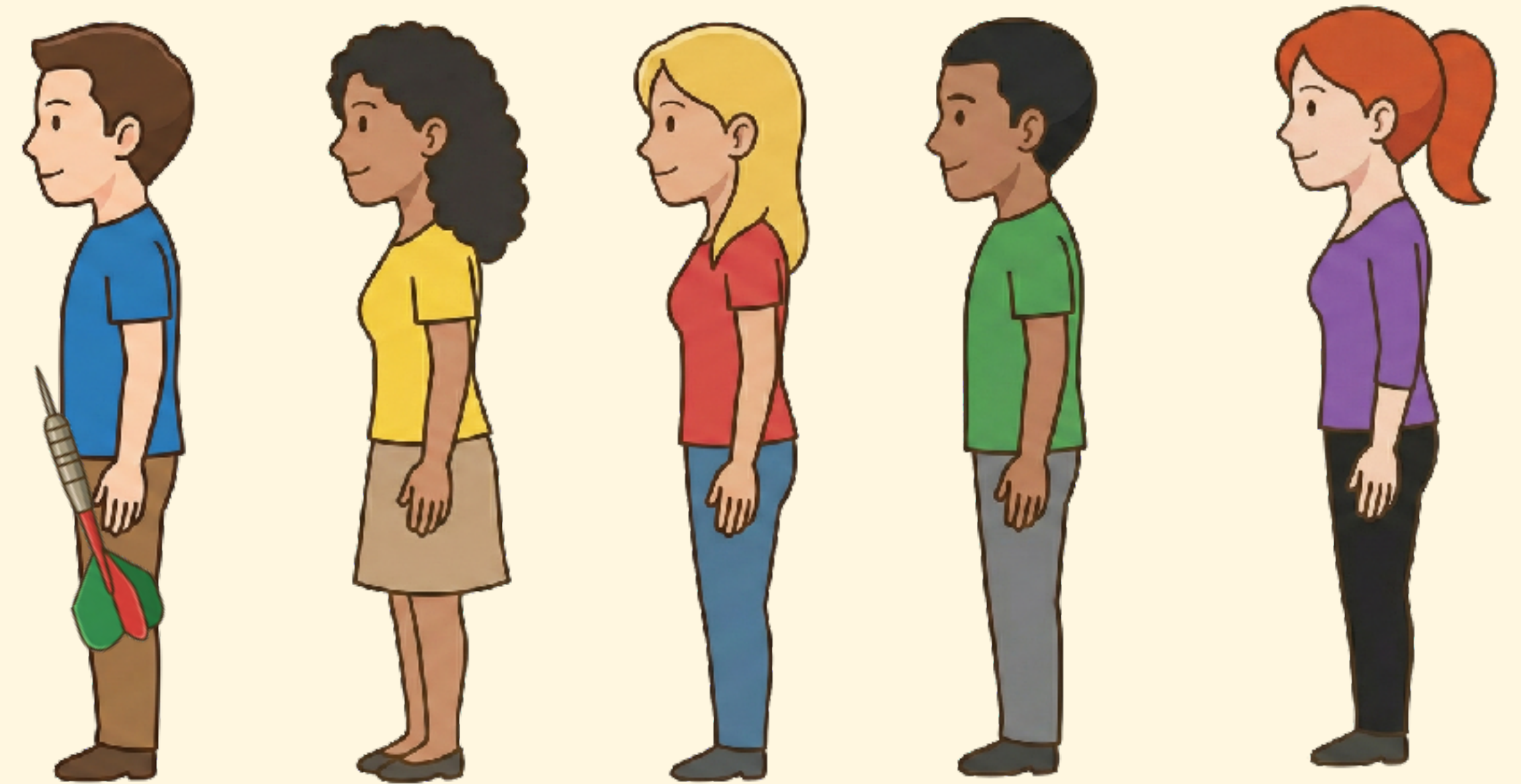
Player 2 hit.

Andy Niedermaier's Dart Game

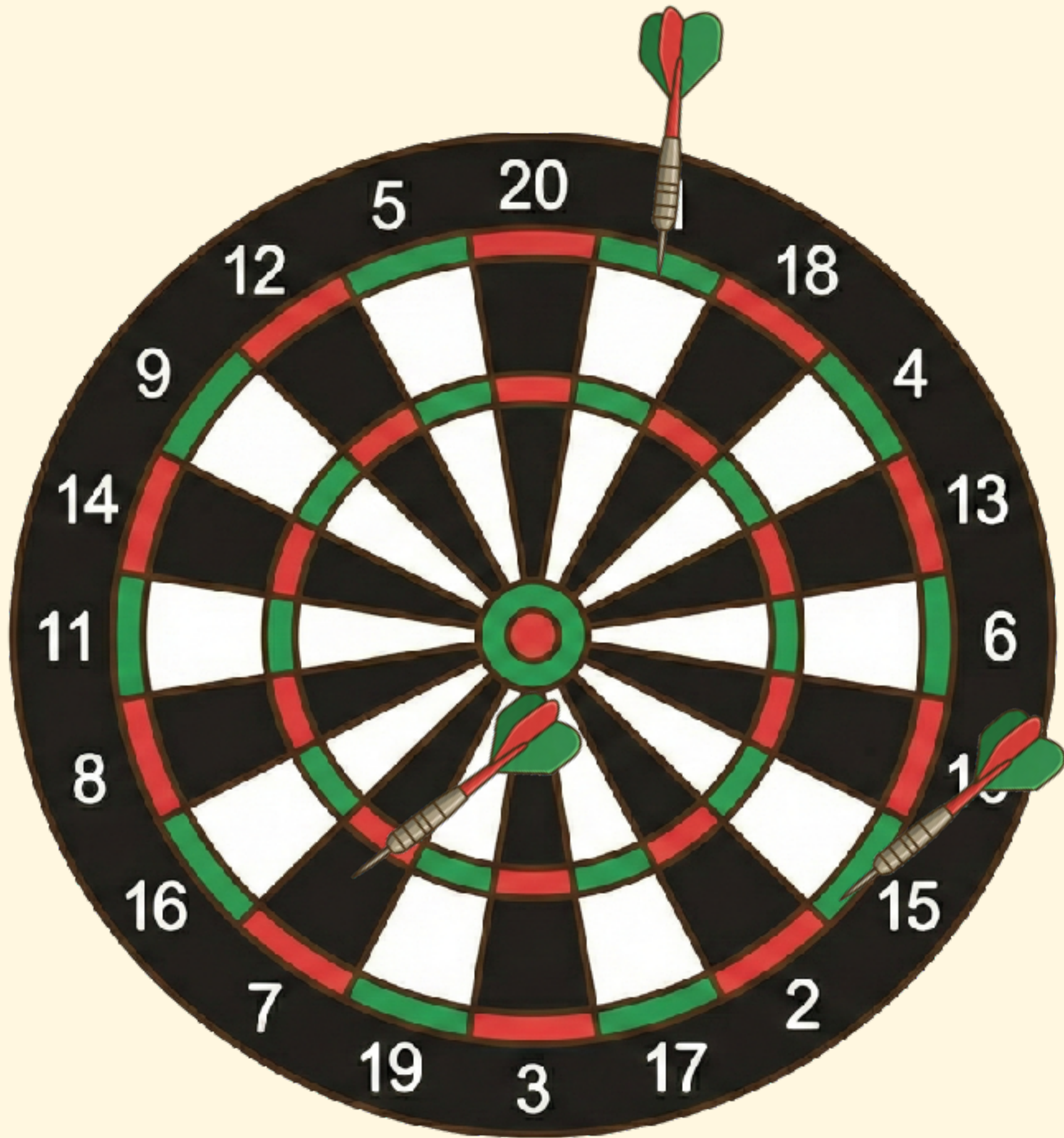
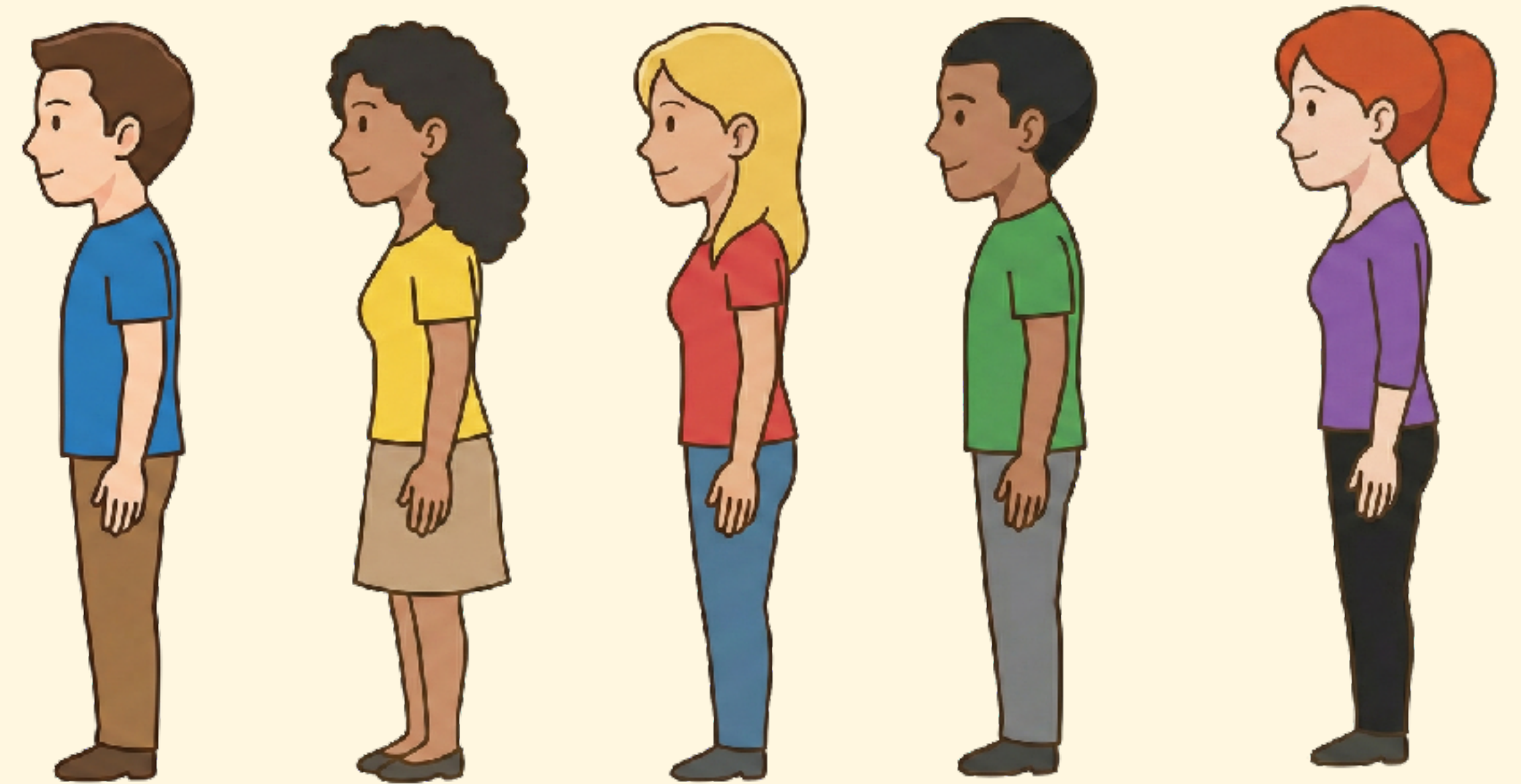


Player 2 hit.

Andy Niedermaier's Dart Game

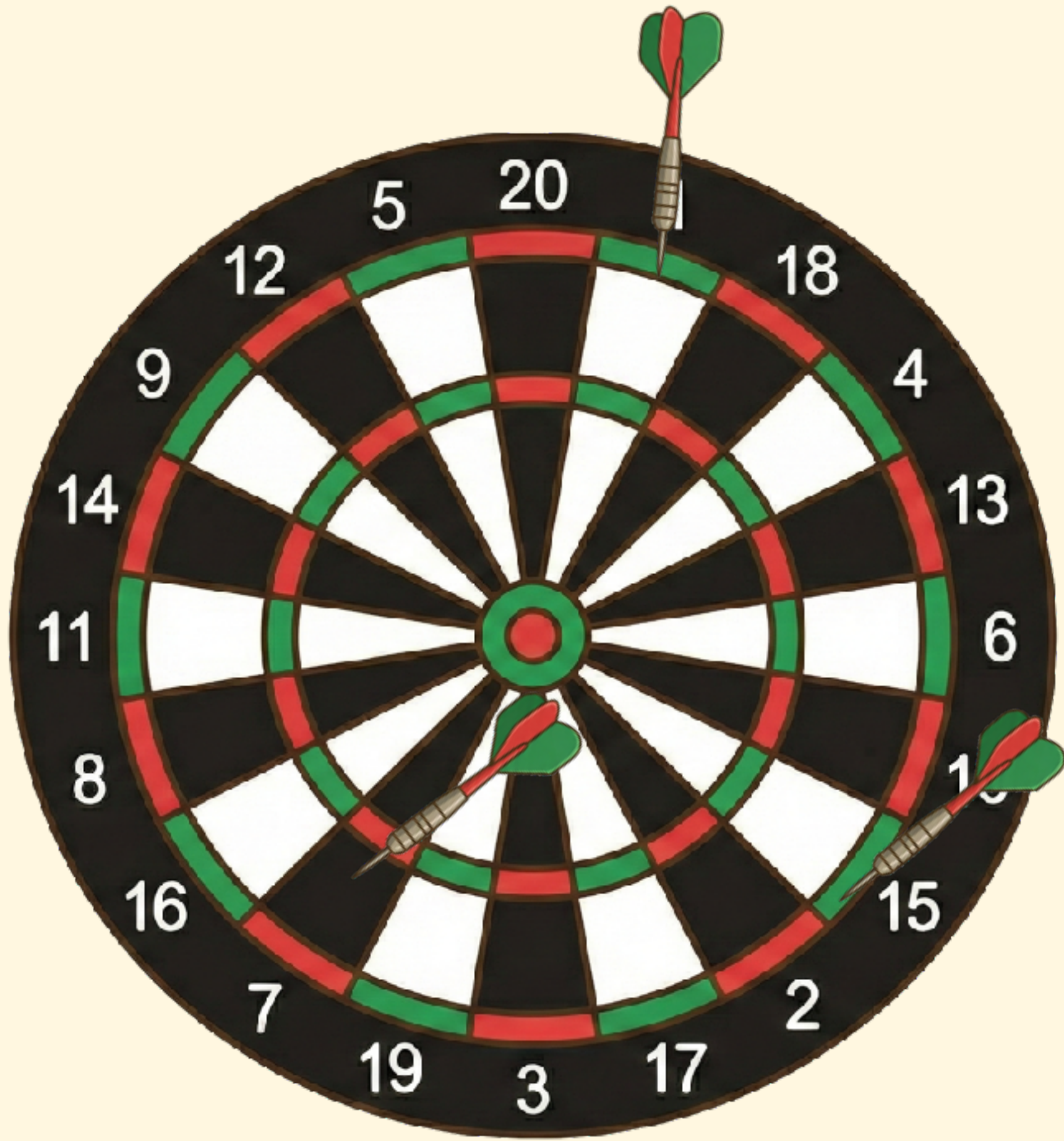
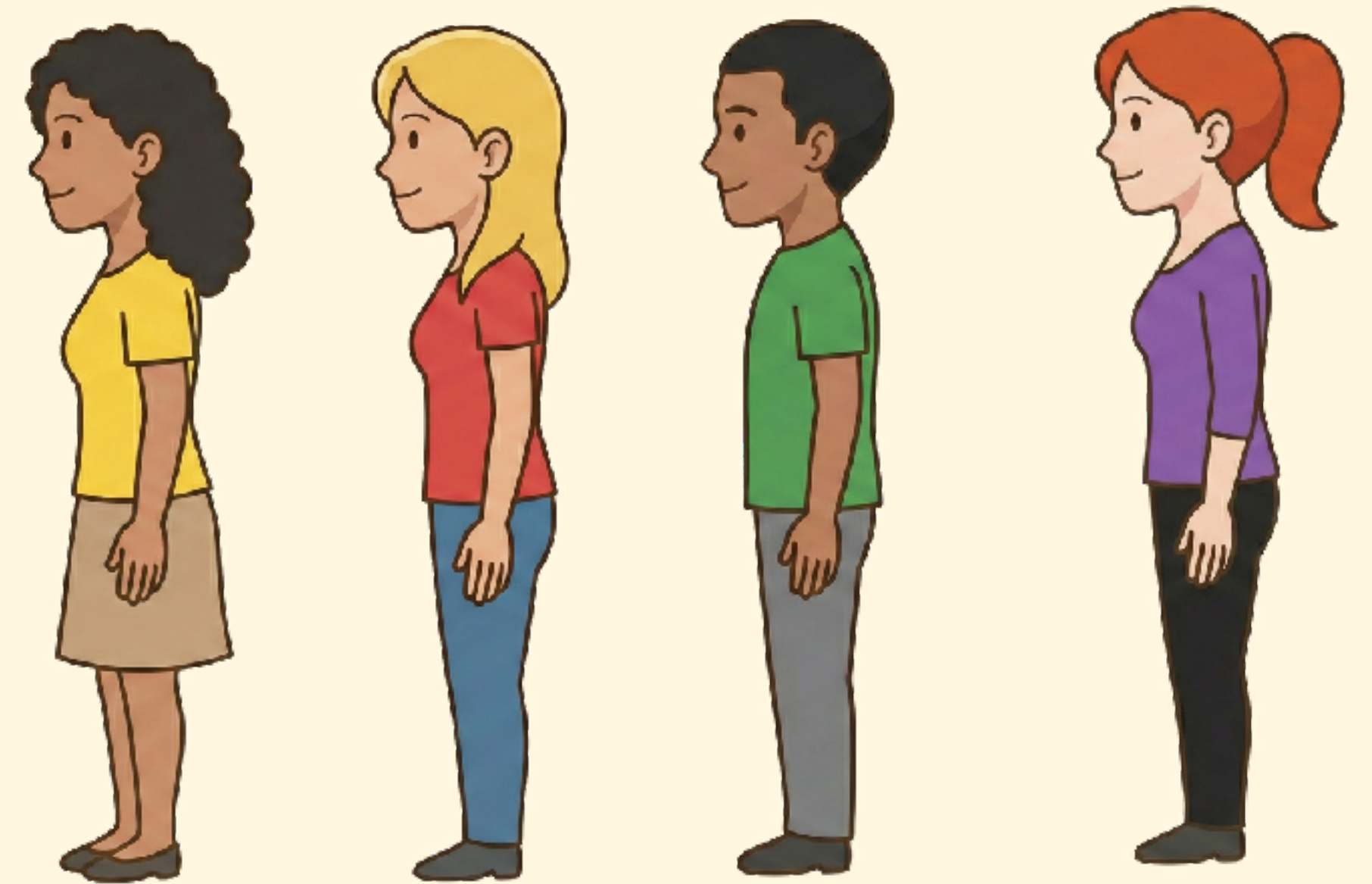


Andy Niedermaier's Dart Game



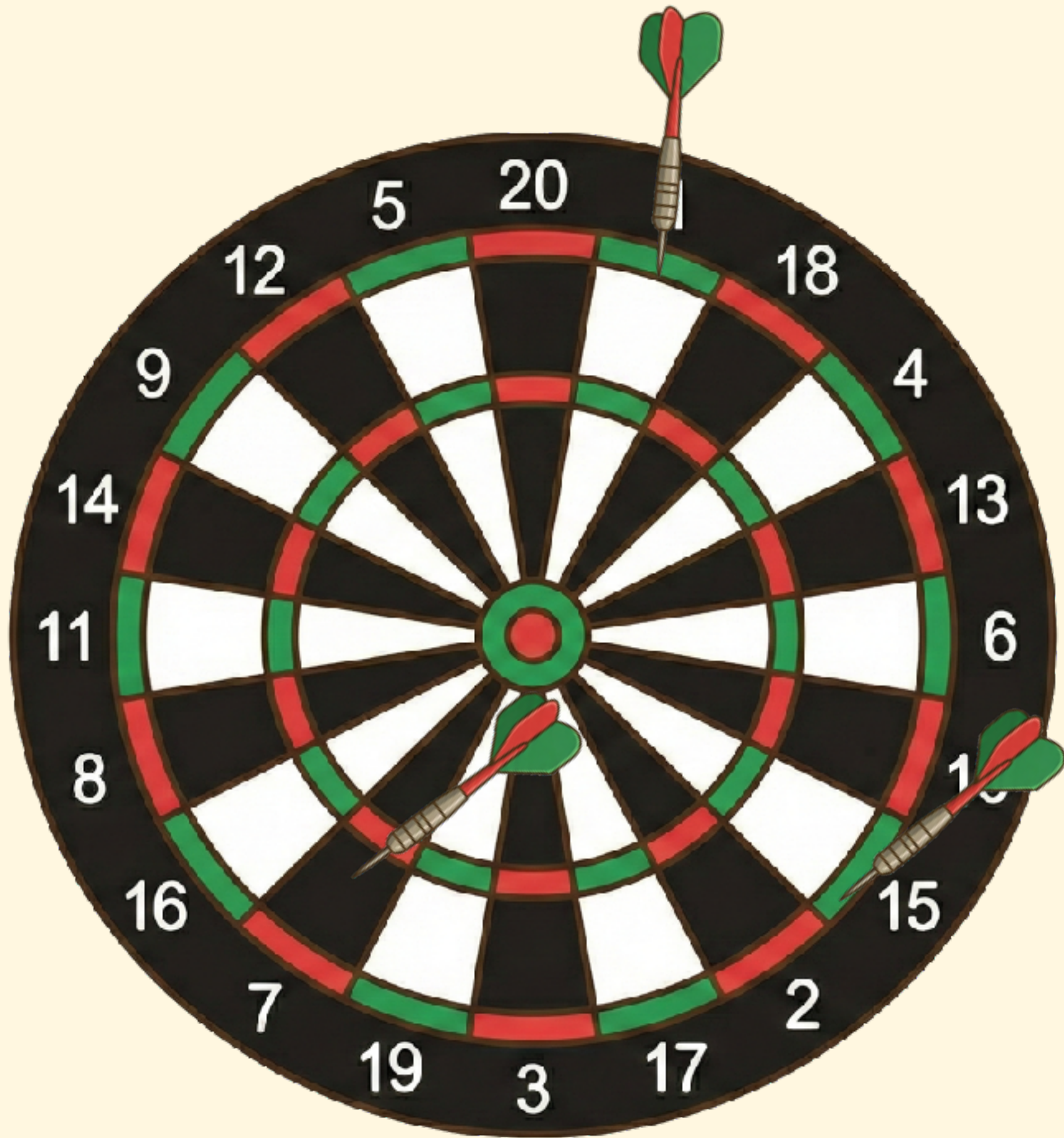
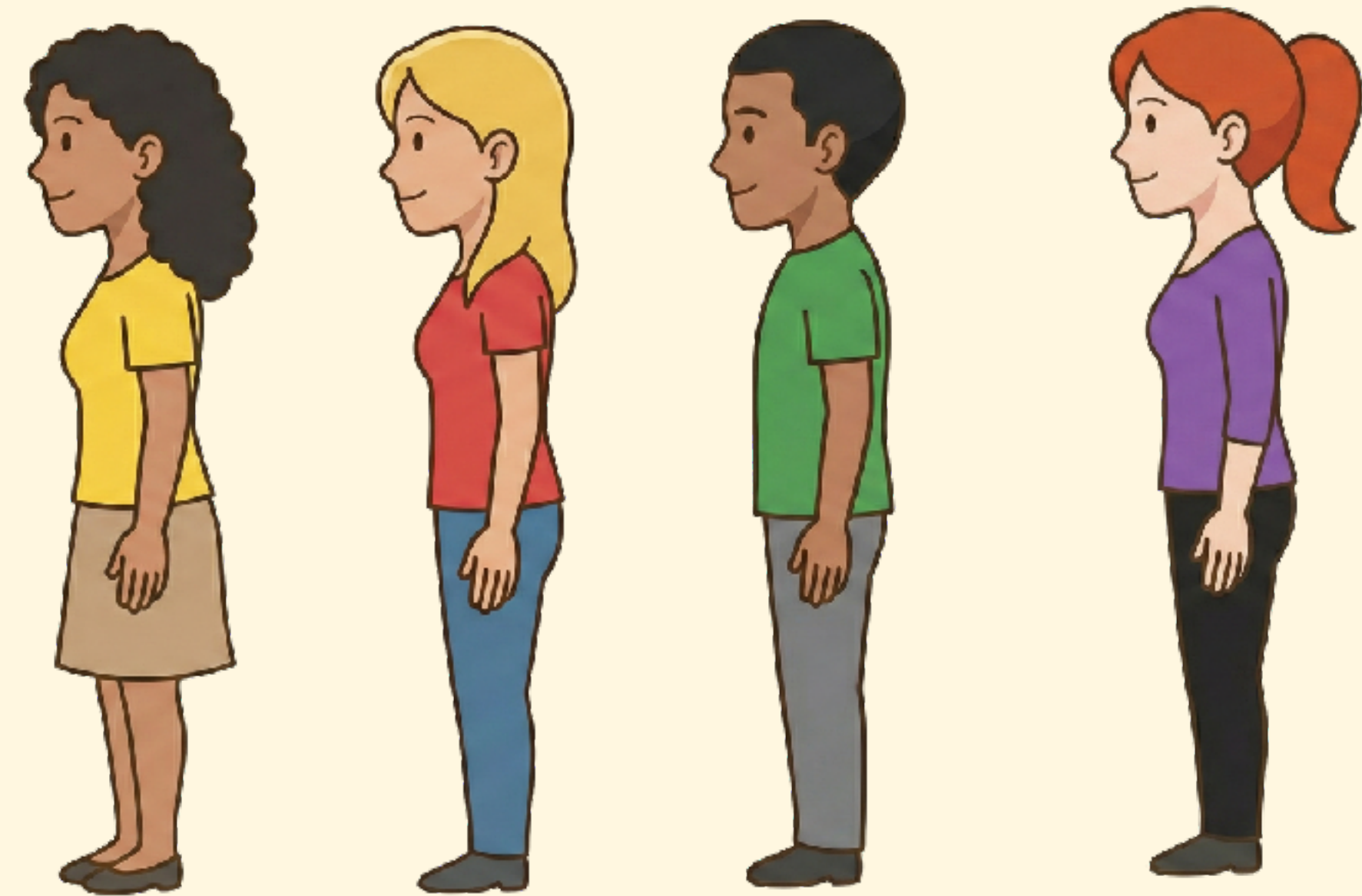
Player 3 miss.

Andy Niedermaier's Dart Game



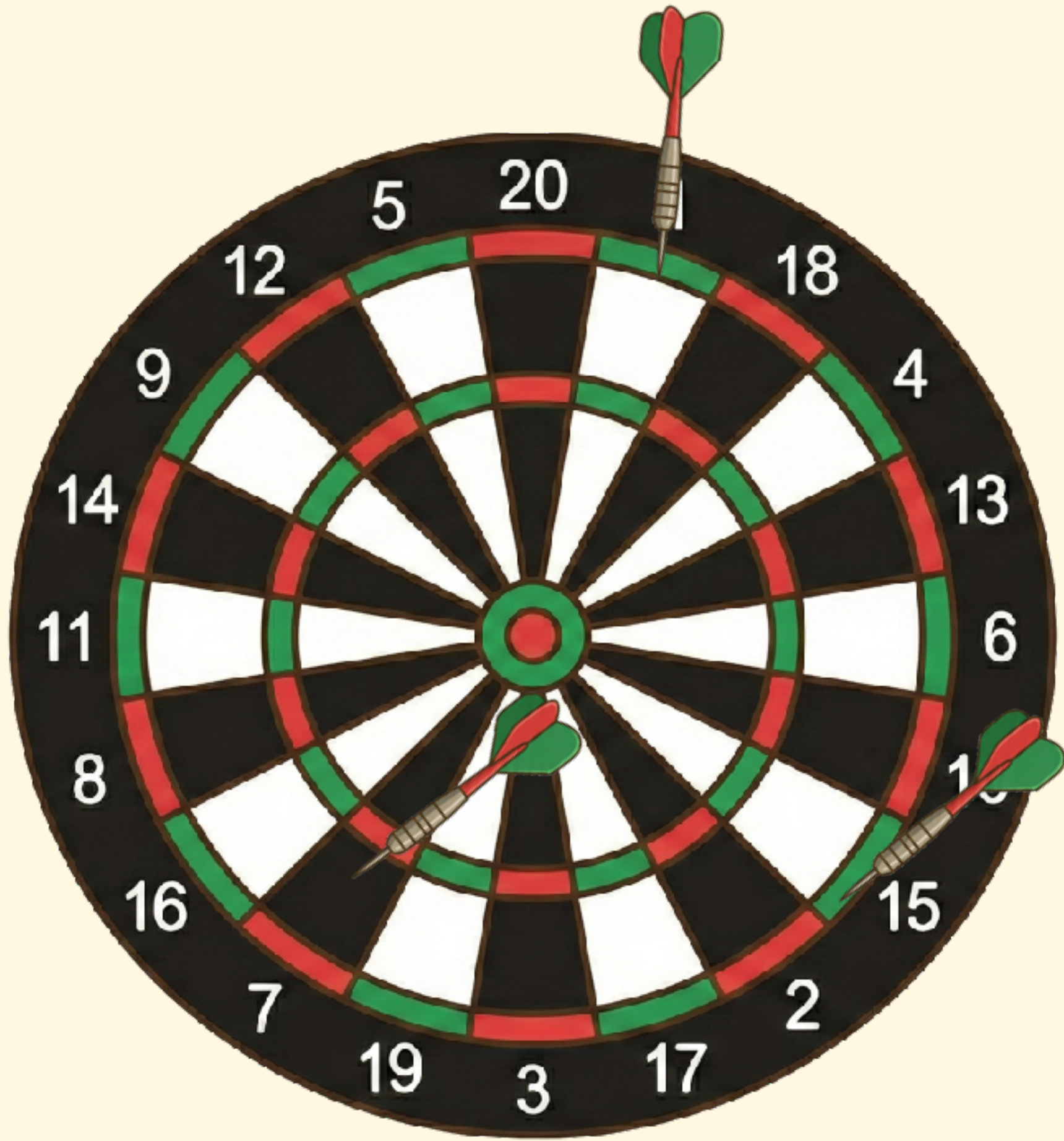
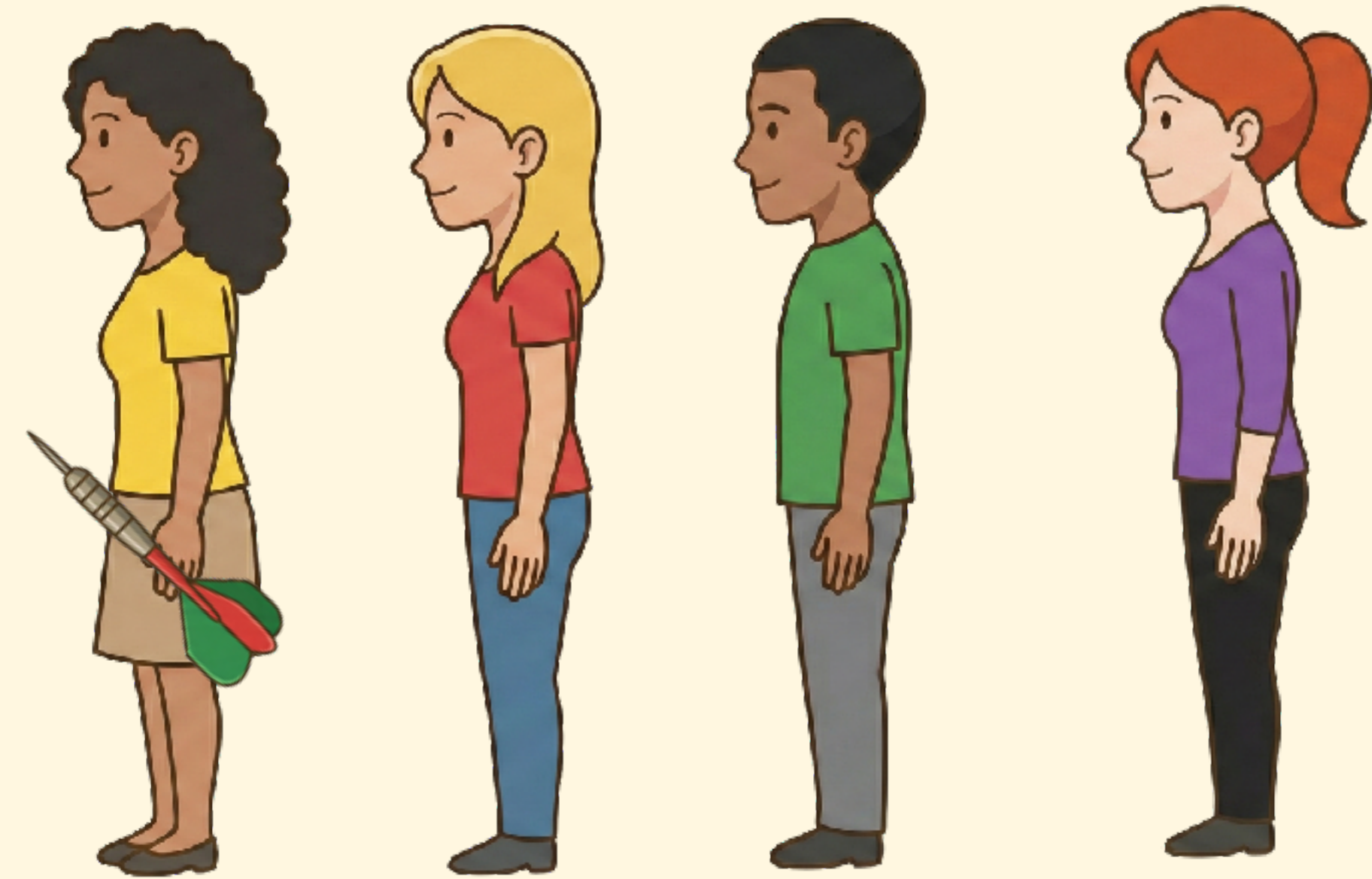
Player 3 miss.

Andy Niedermaier's Dart Game

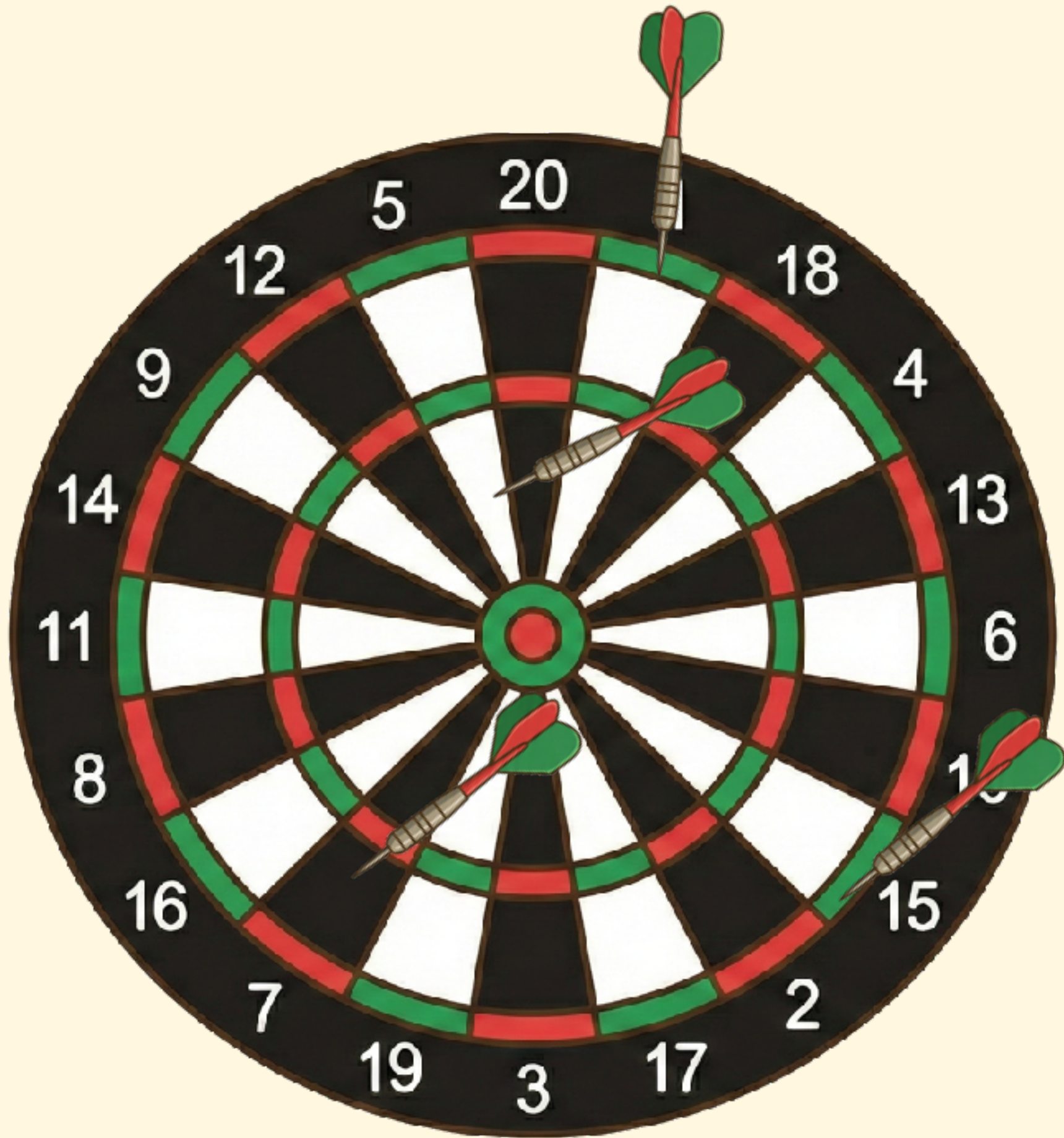
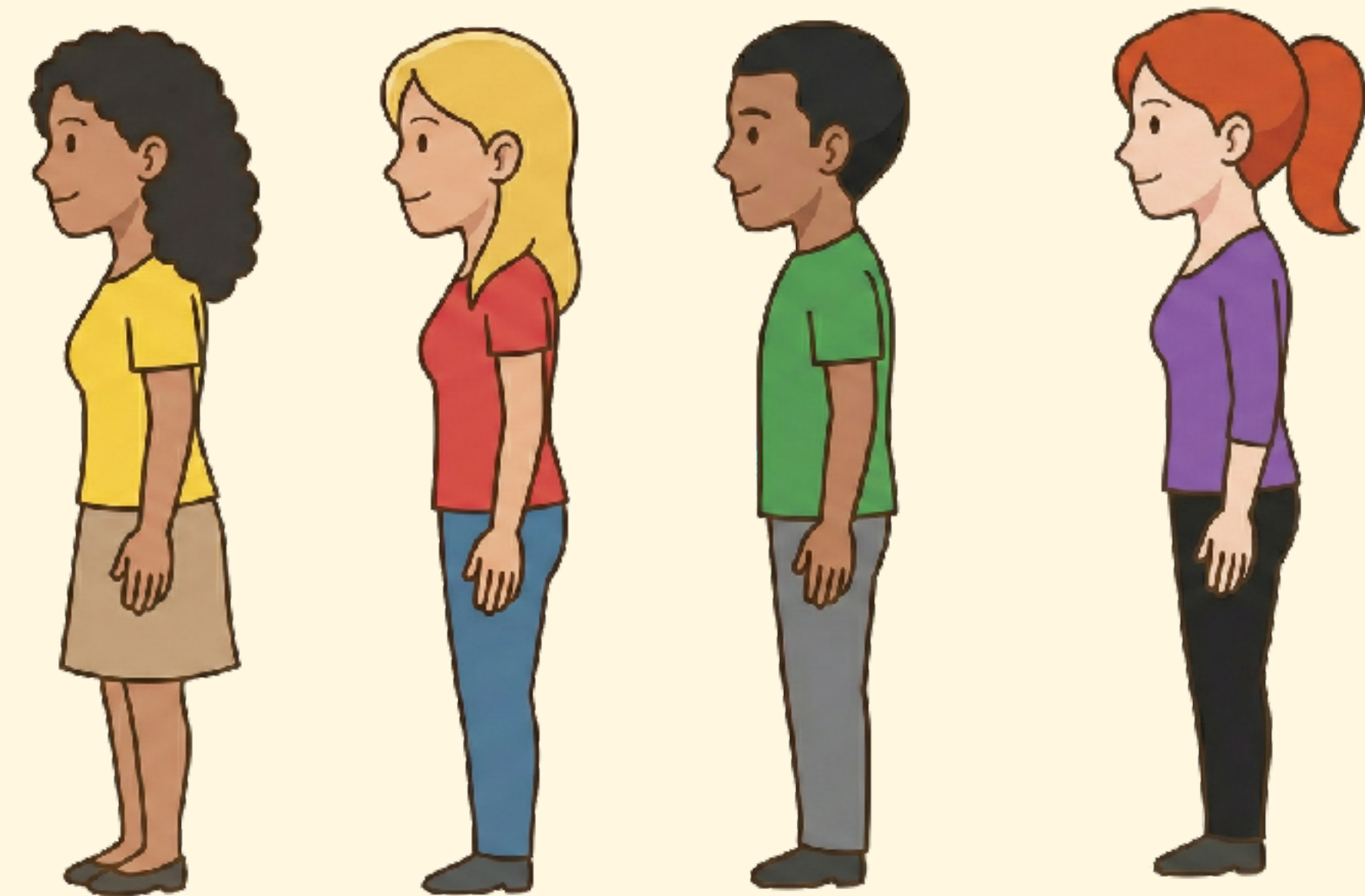


Player 3 miss.

Andy Niedermaier's Dart Game

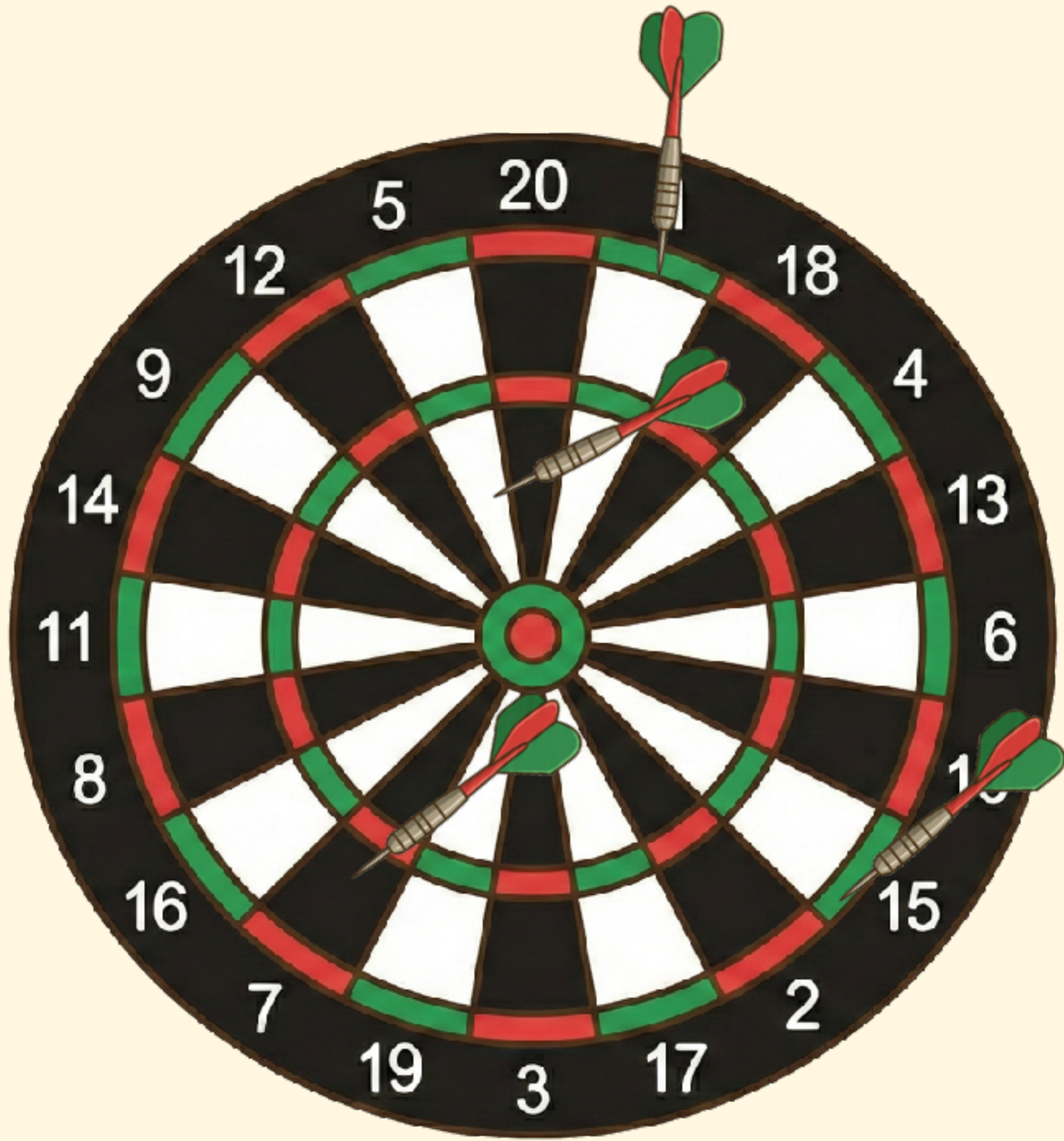
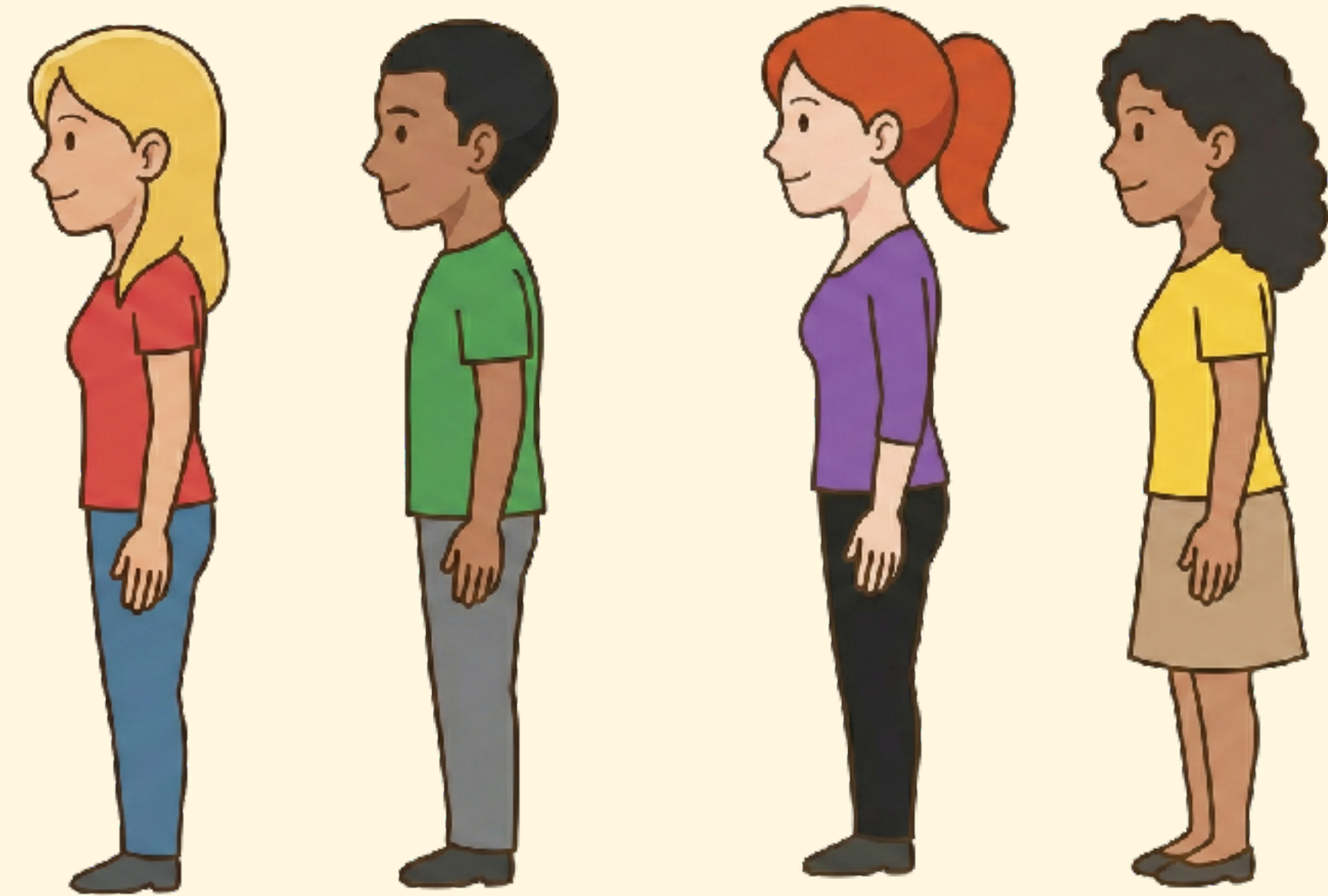


Andy Niedermaier's Dart Game



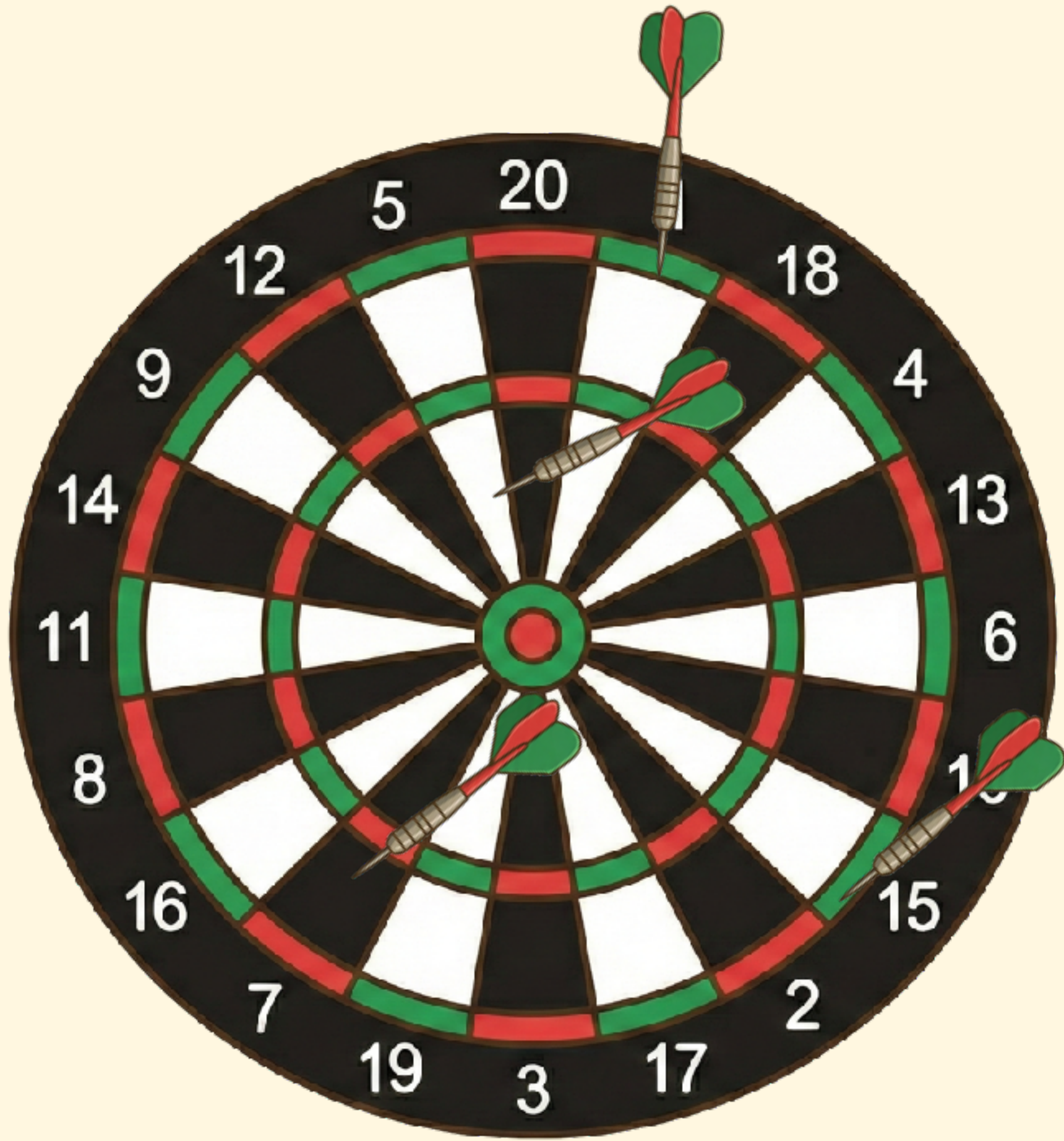
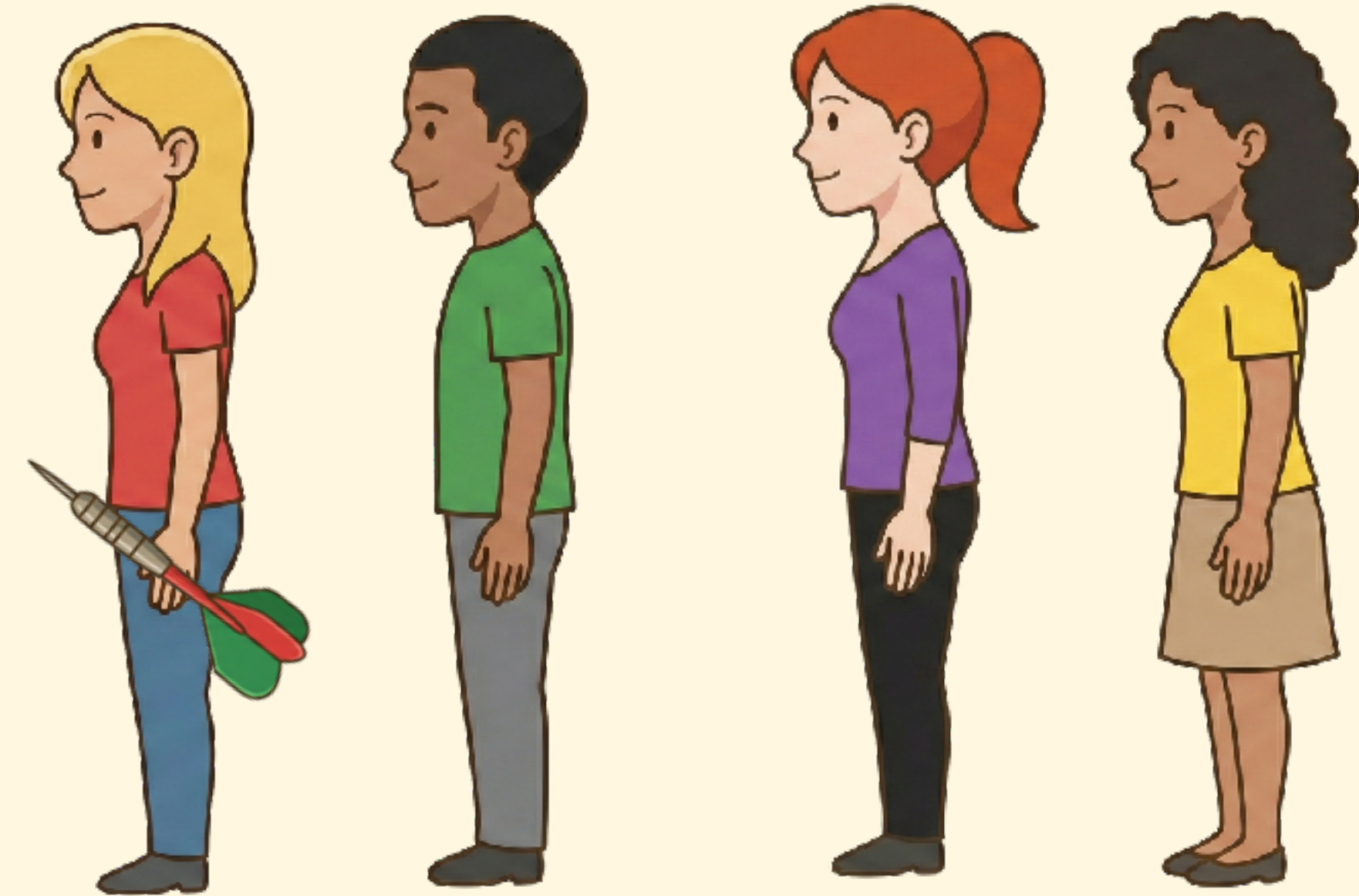
Player 4 hit.

Andy Niedermaier's Dart Game

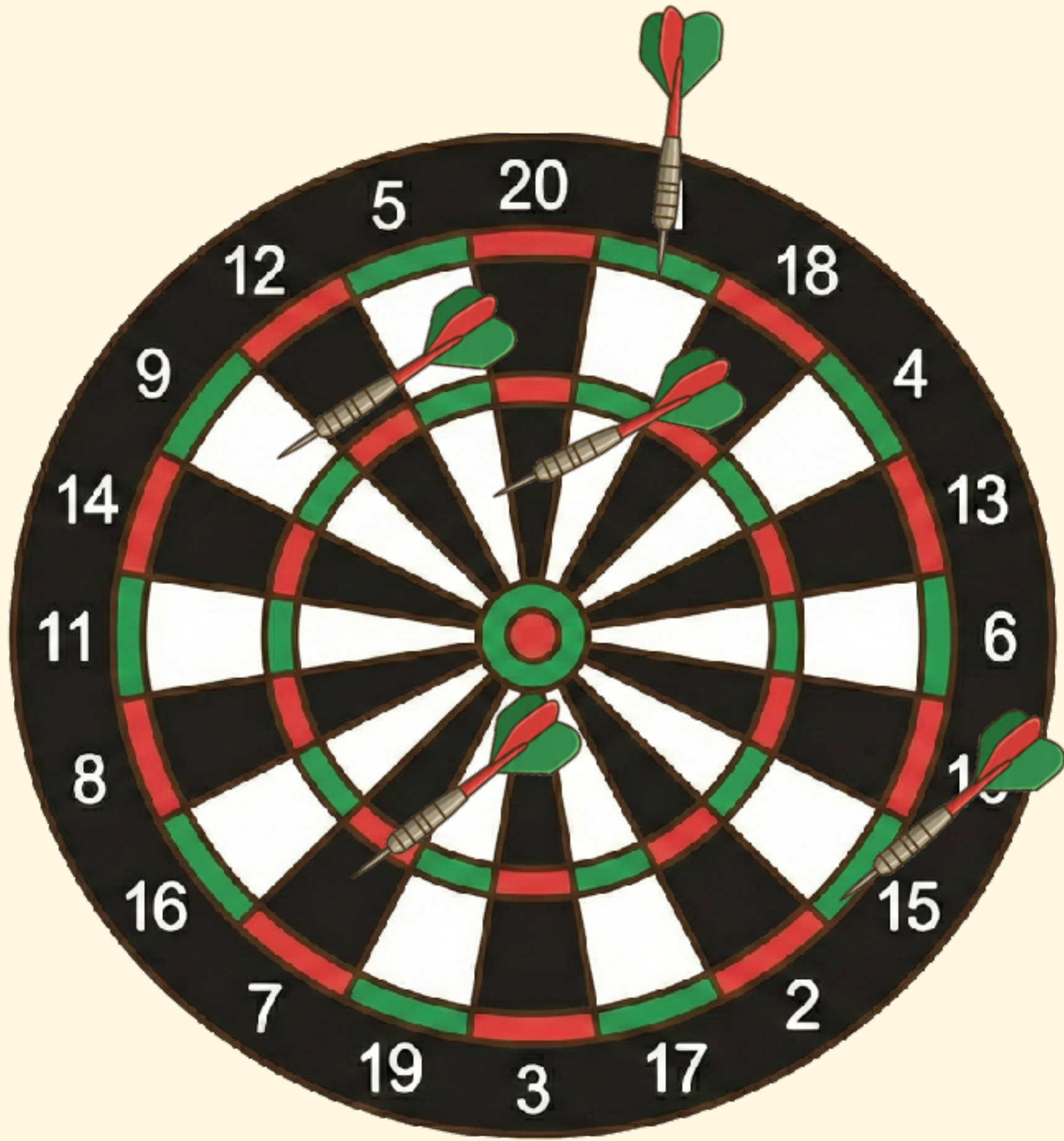
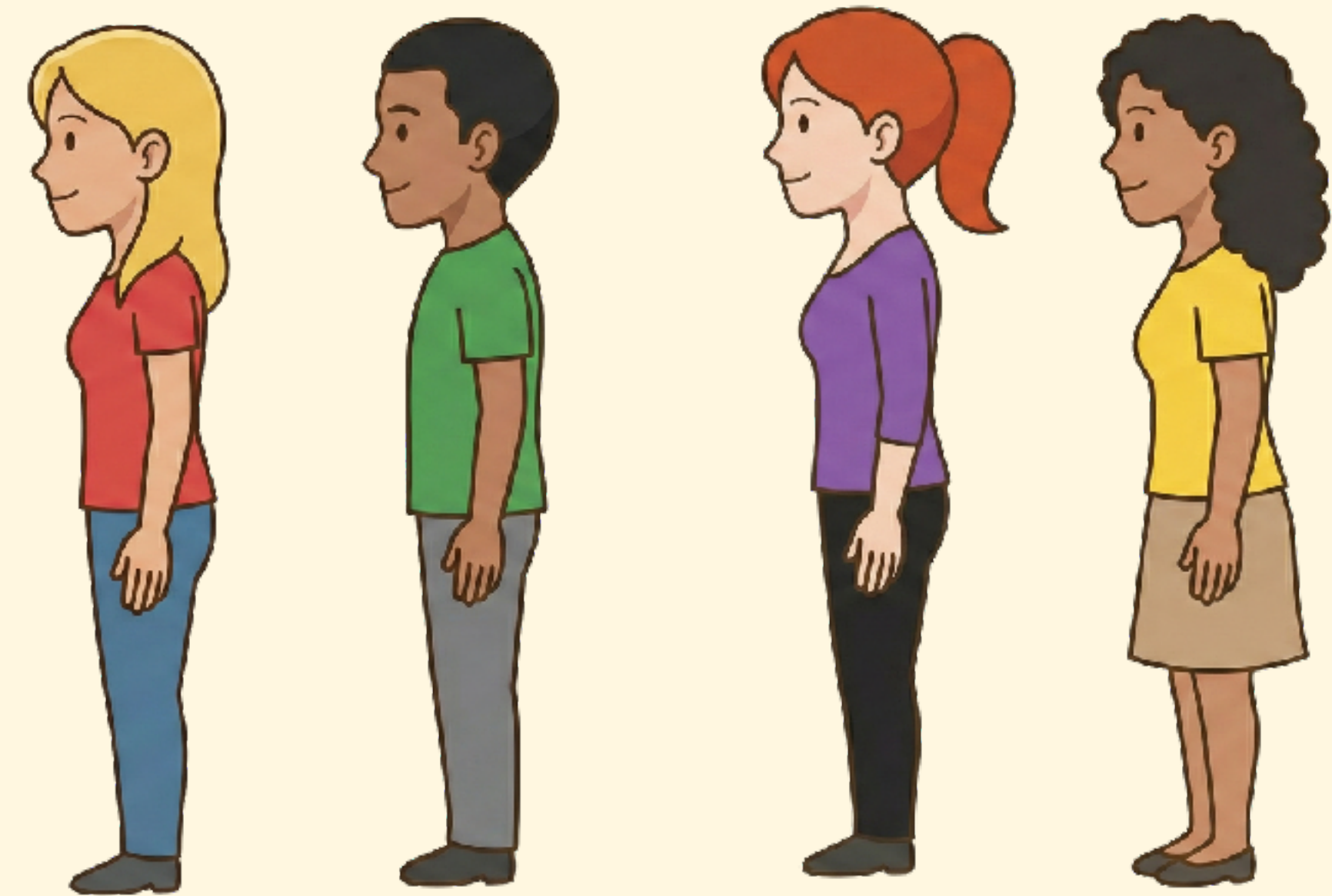


Player 4 hit.

Andy Niedermaier's Dart Game

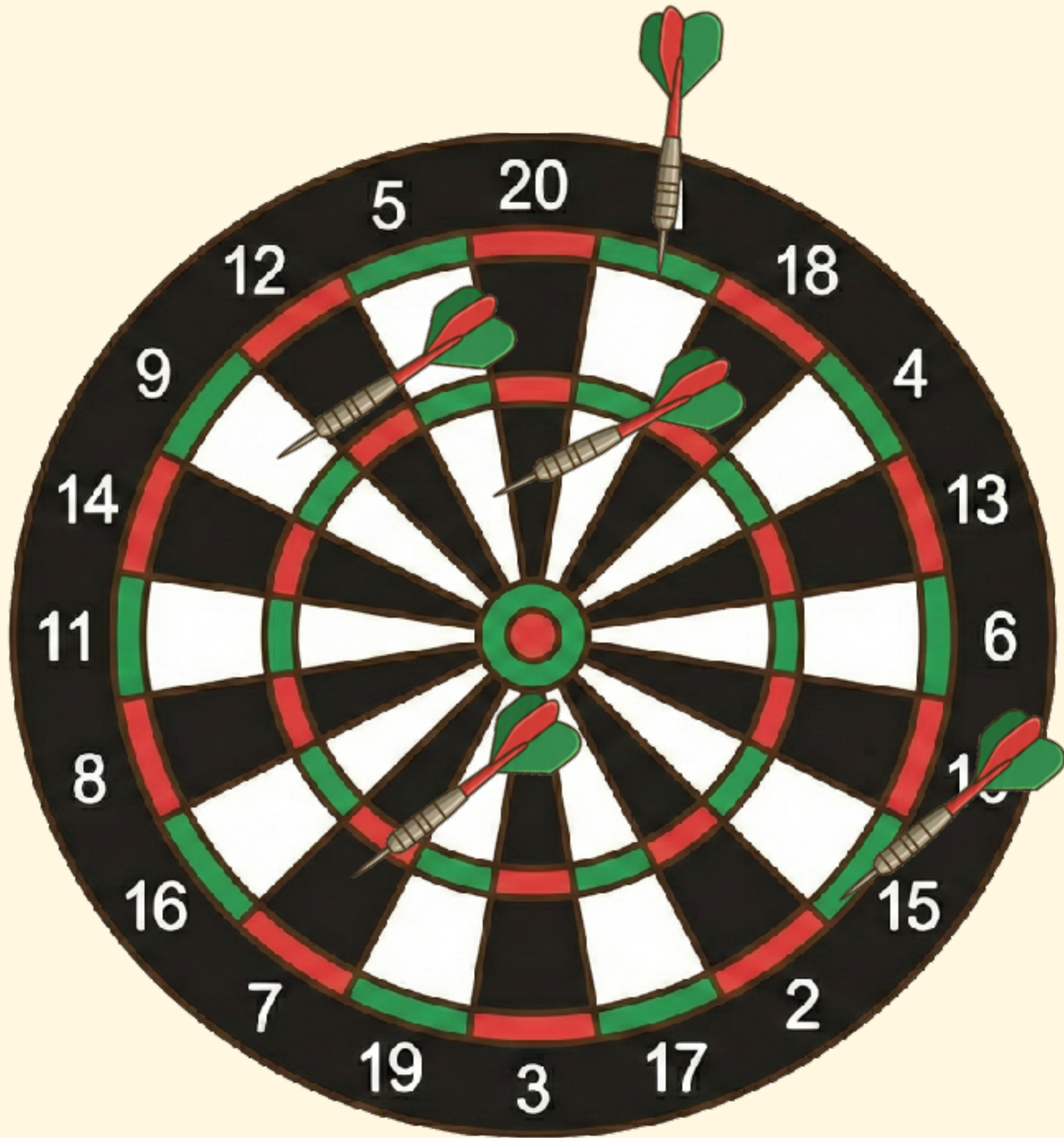
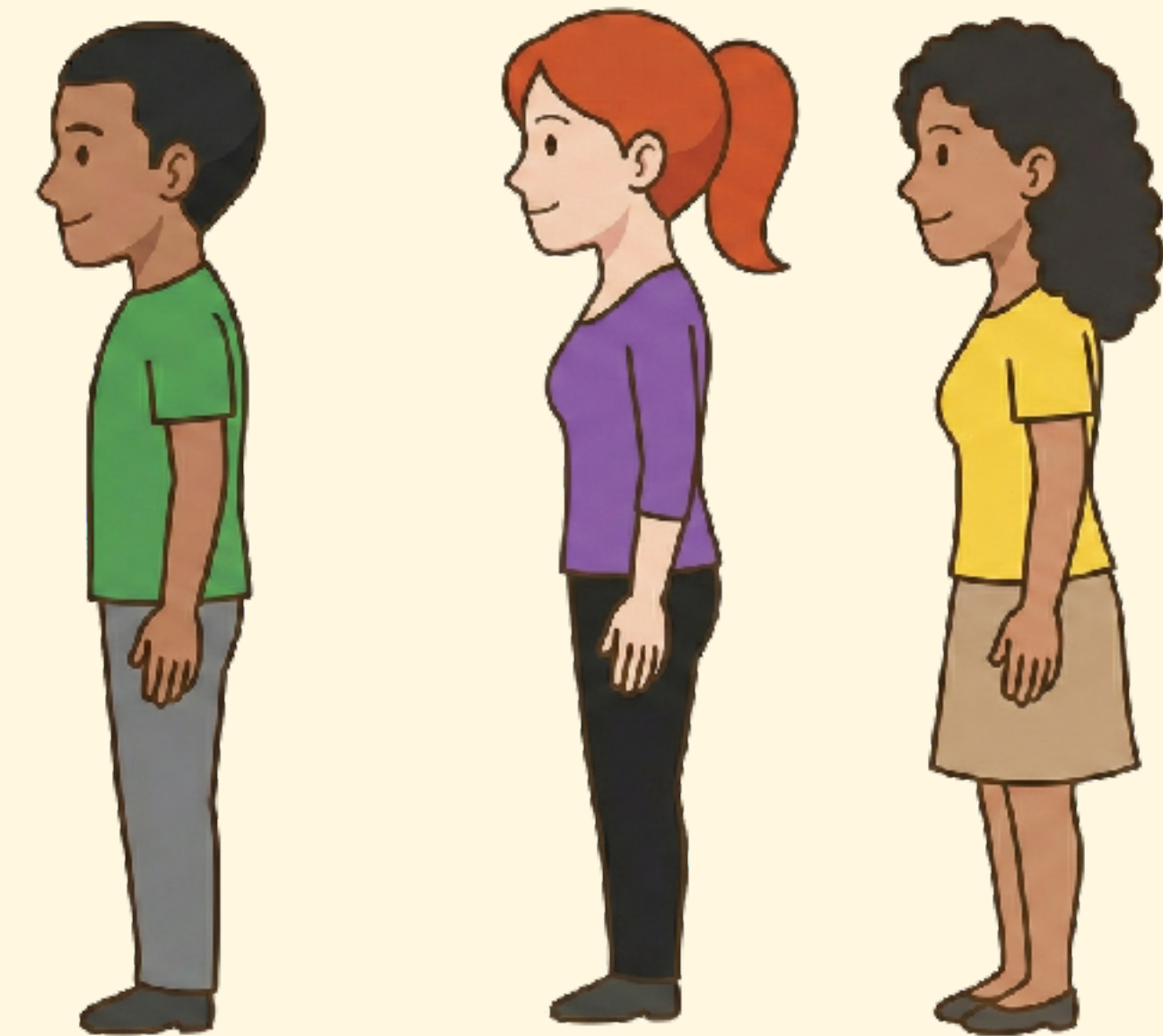


Andy Niedermaier's Dart Game



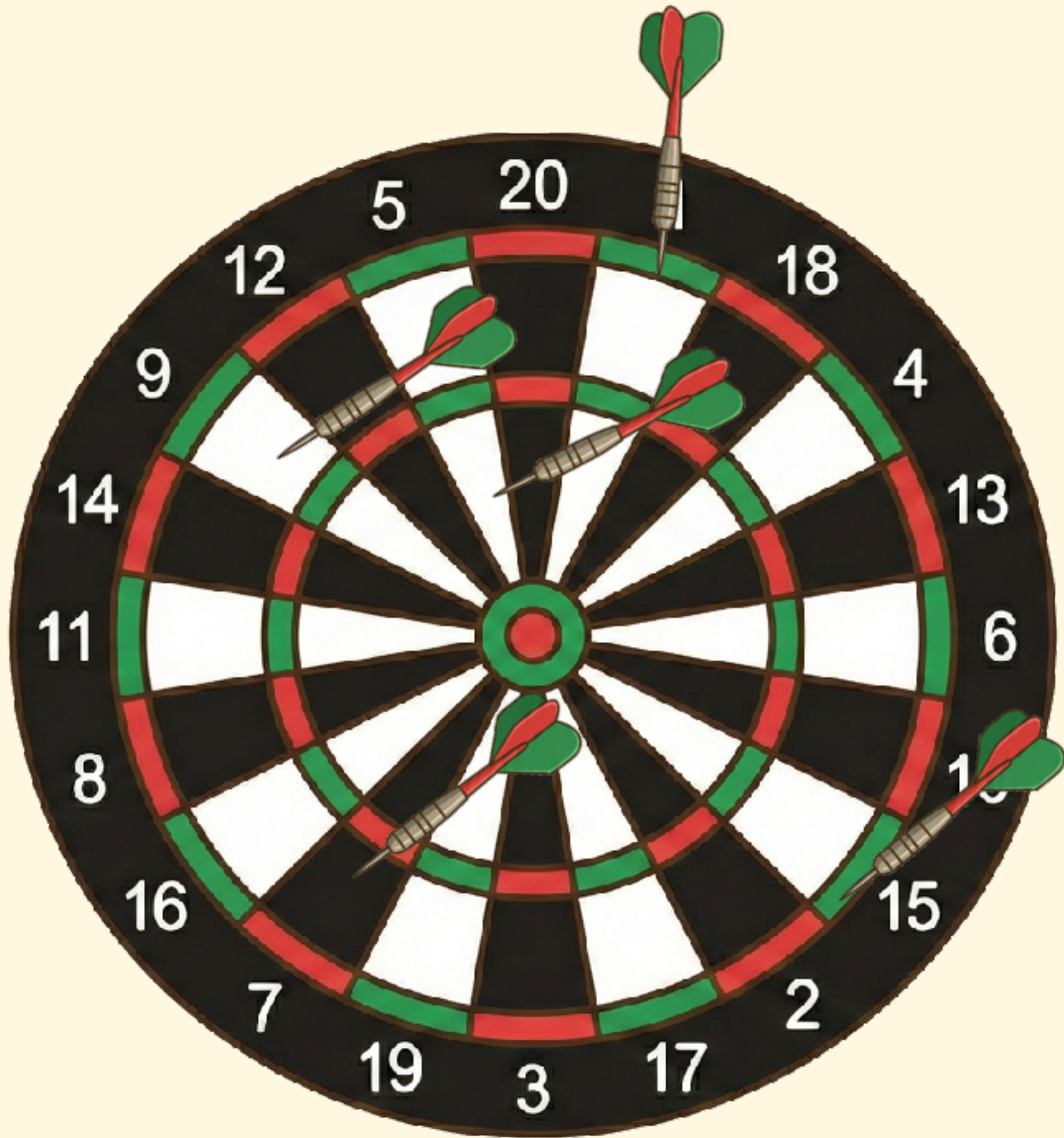
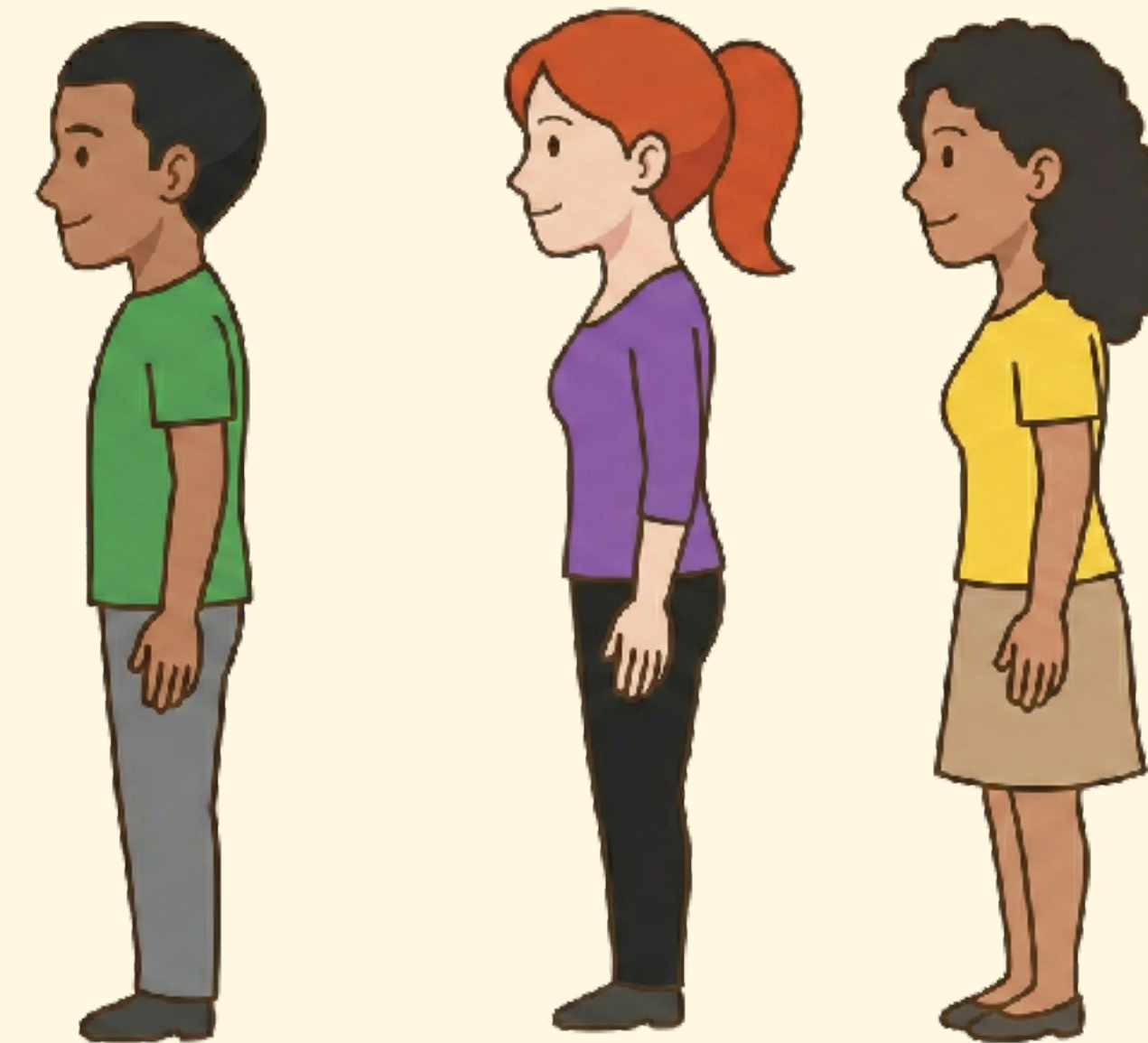
Player 5 miss.

Andy Niedermaier's Dart Game



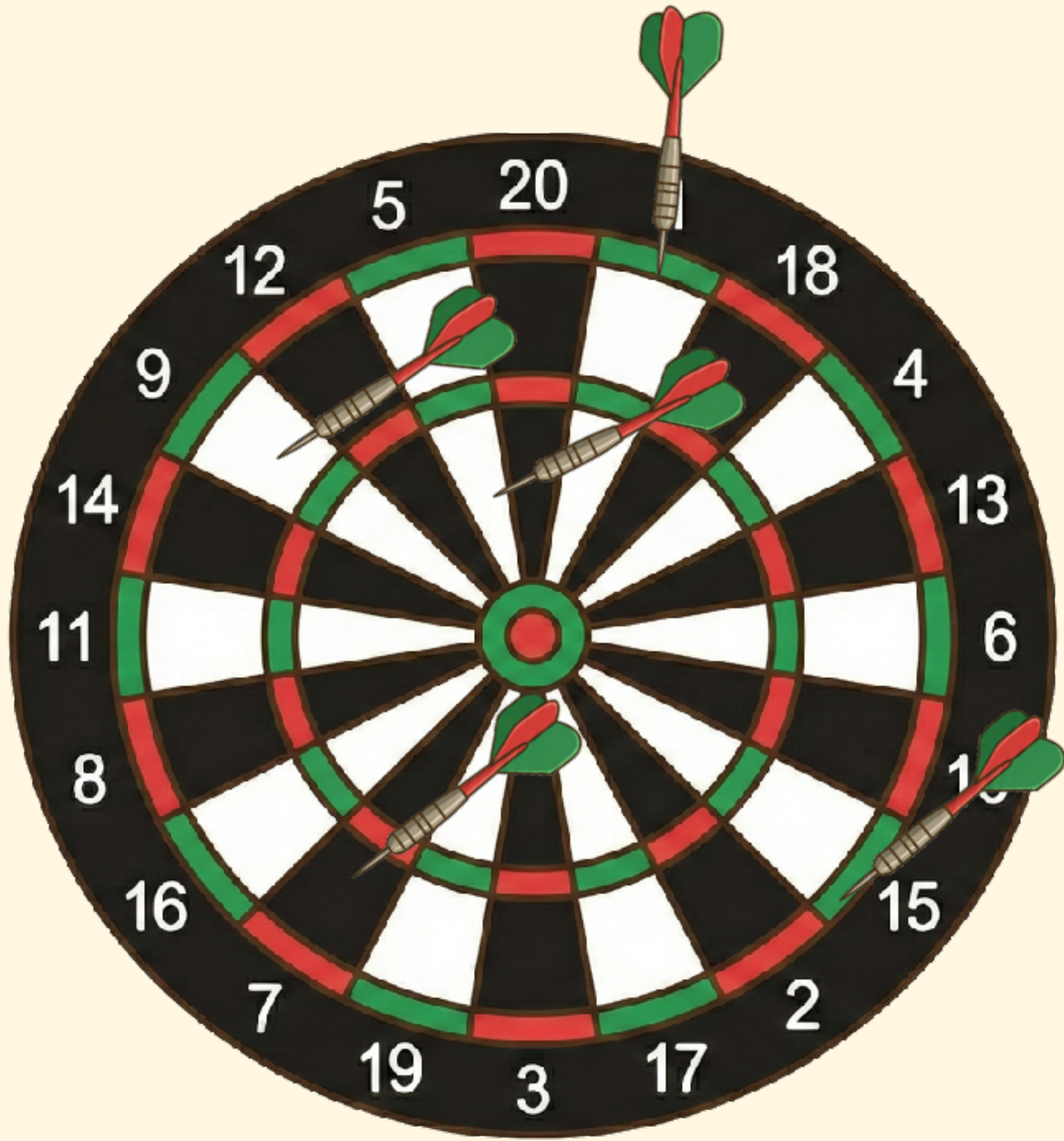
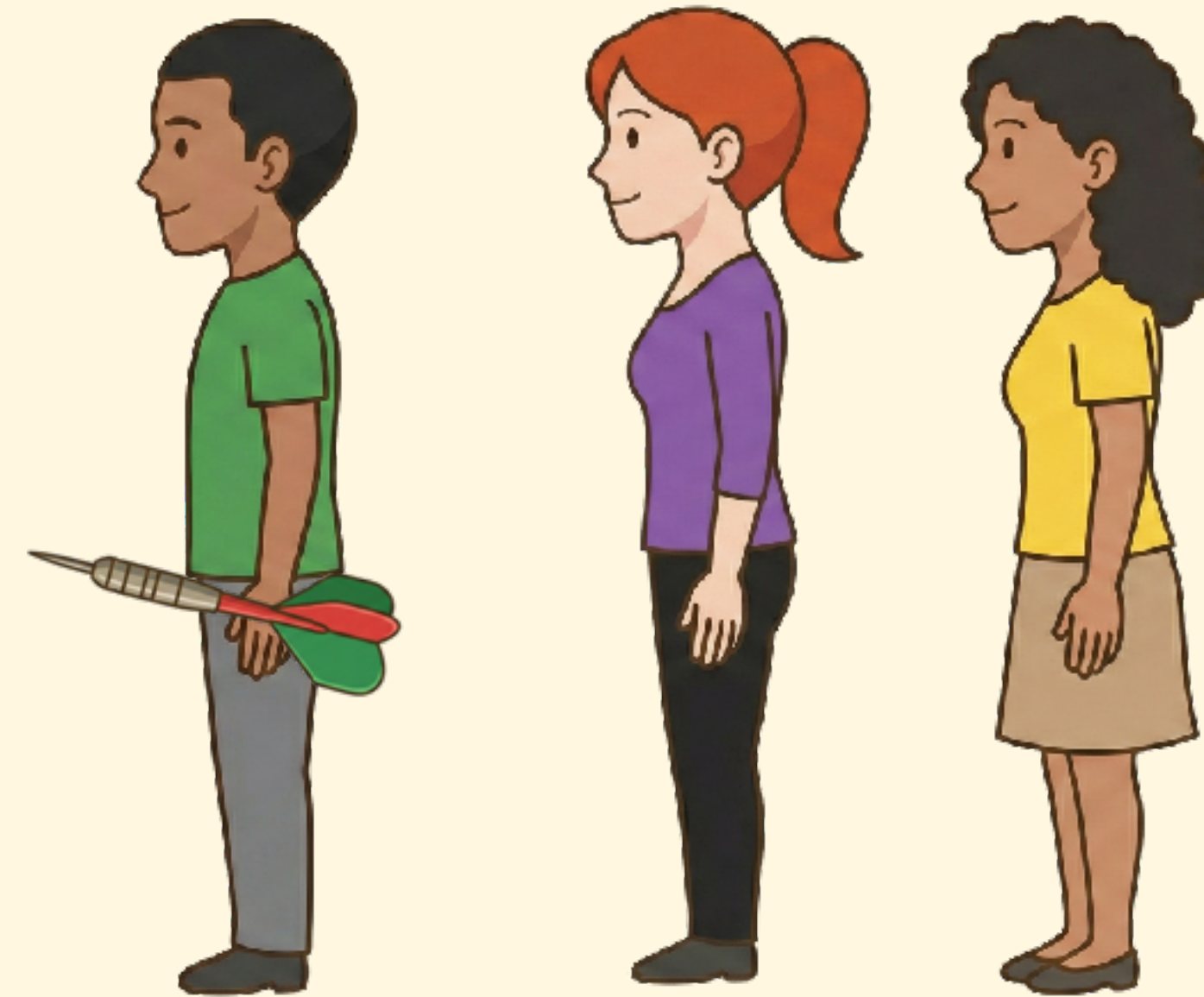
Player 5 miss.

Andy Niedermaier's Dart Game

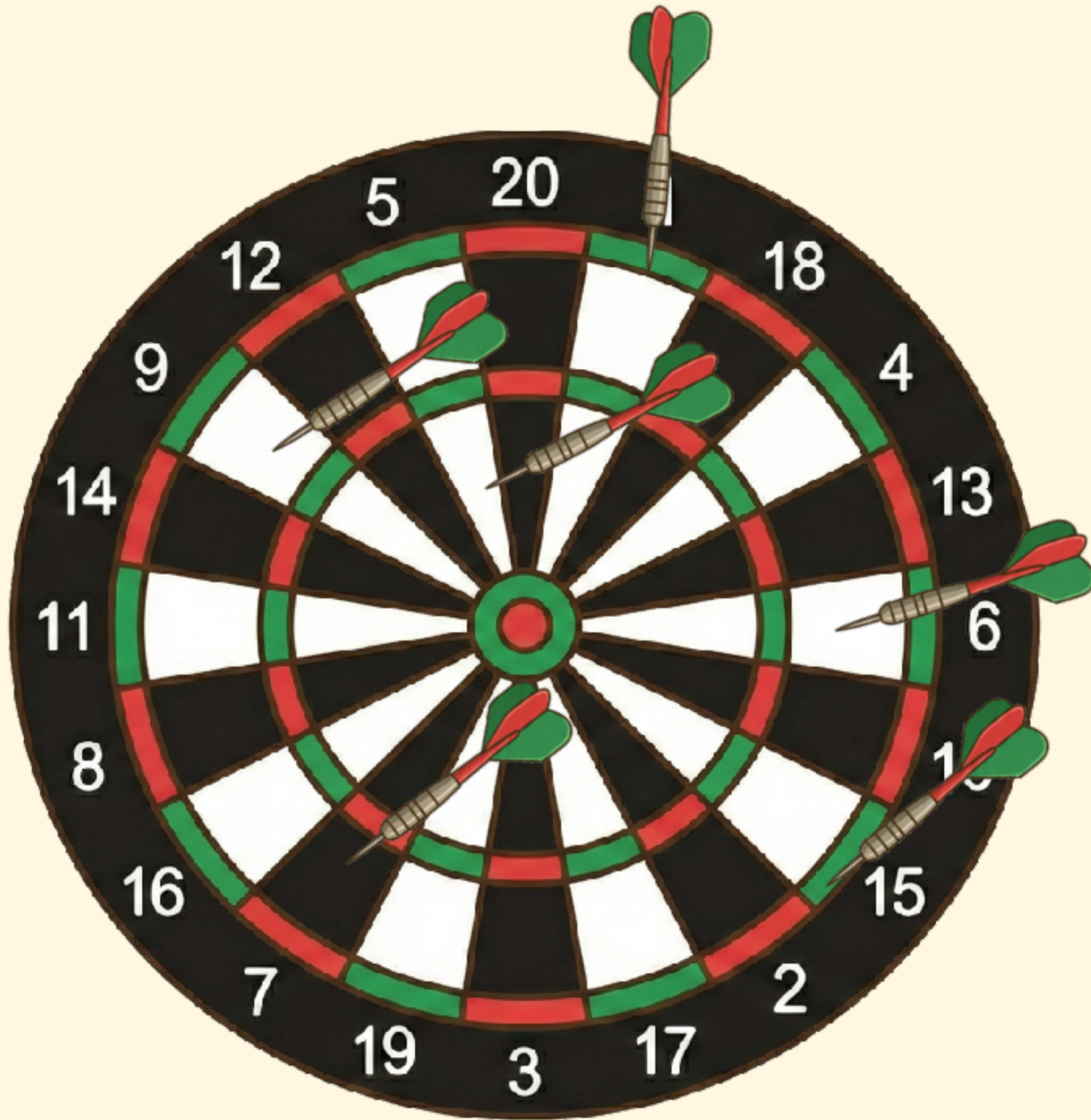
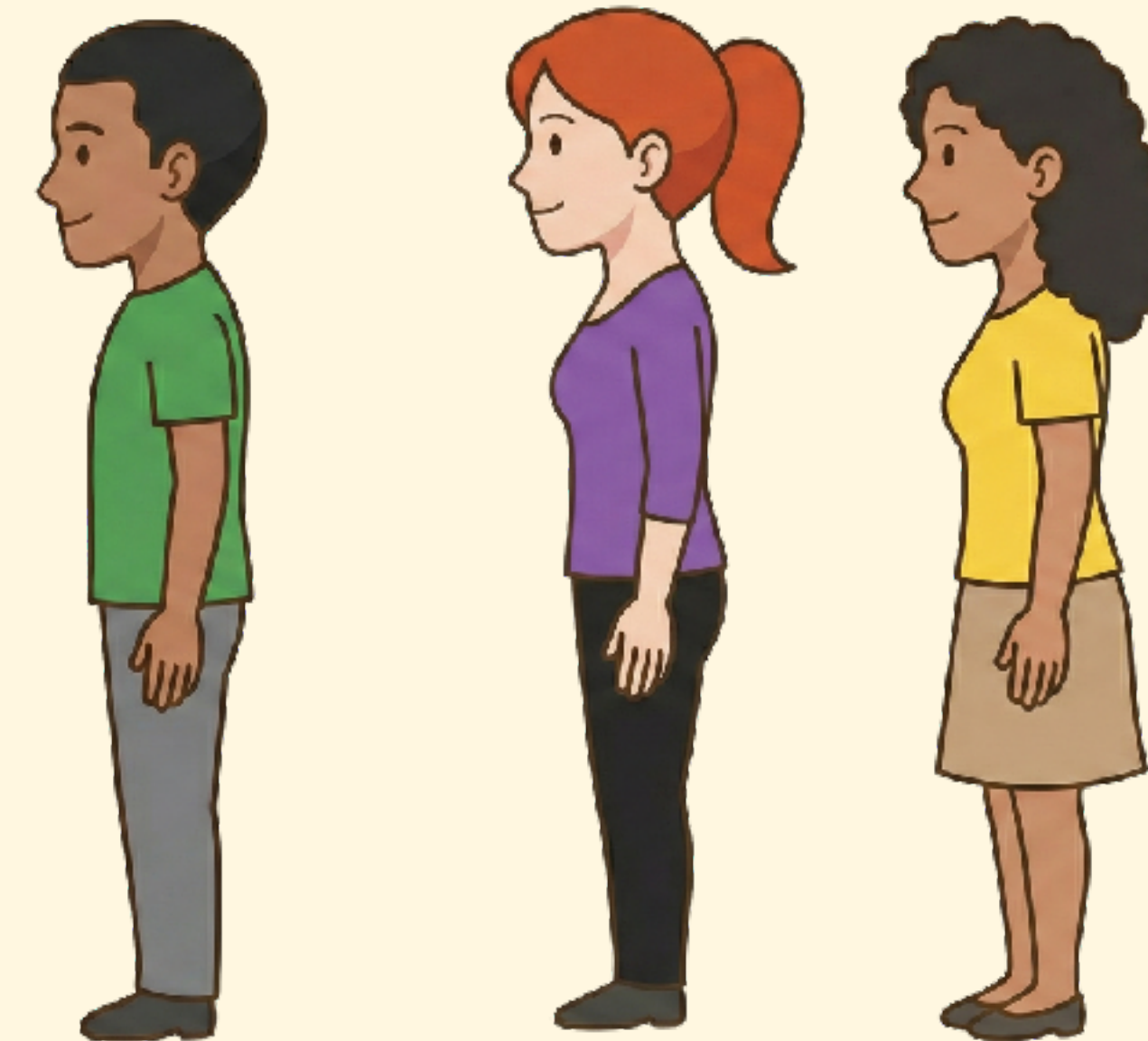


Player 5 miss.

Andy Niedermaier's Dart Game

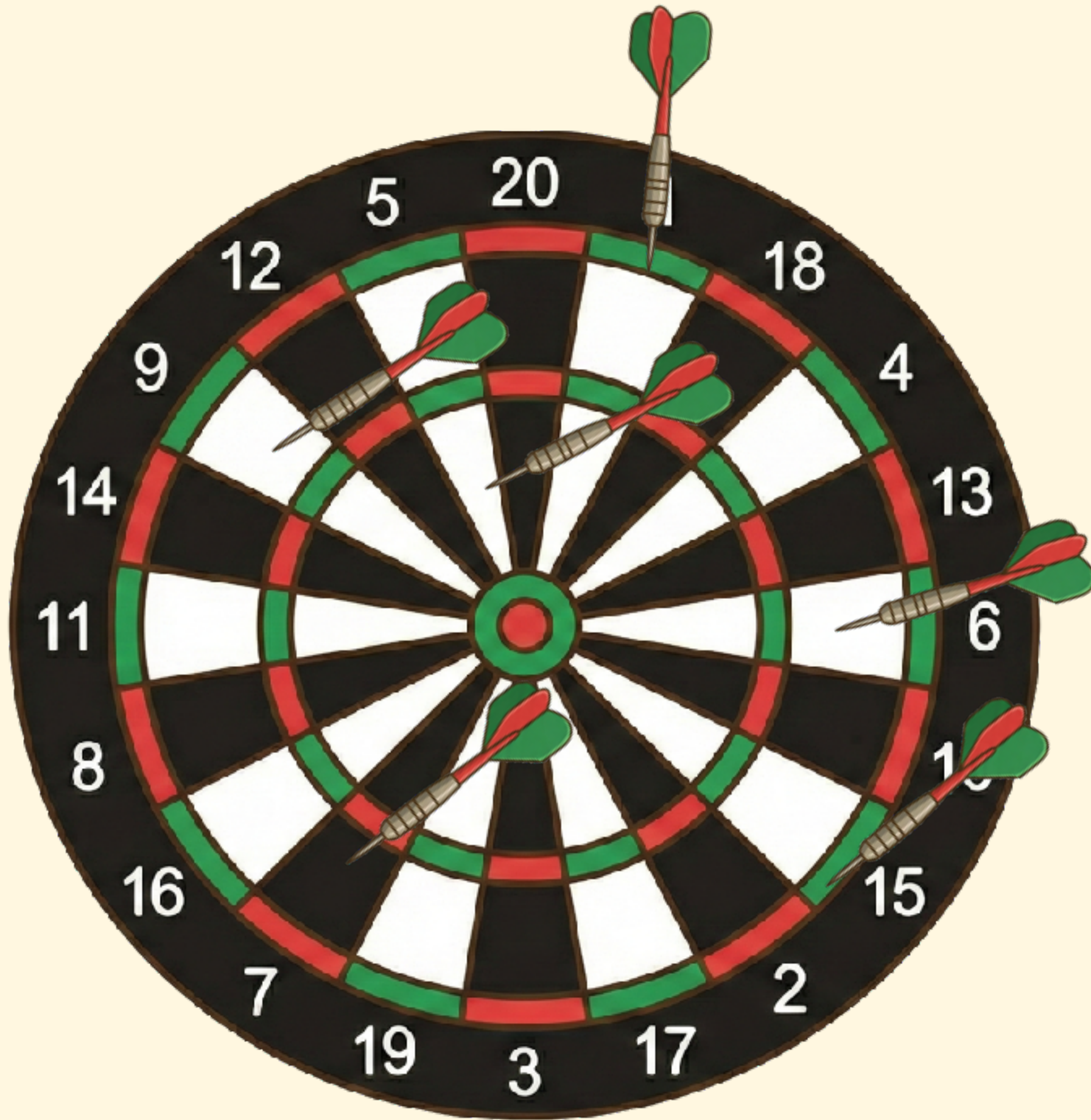


Andy Niedermaier's Dart Game



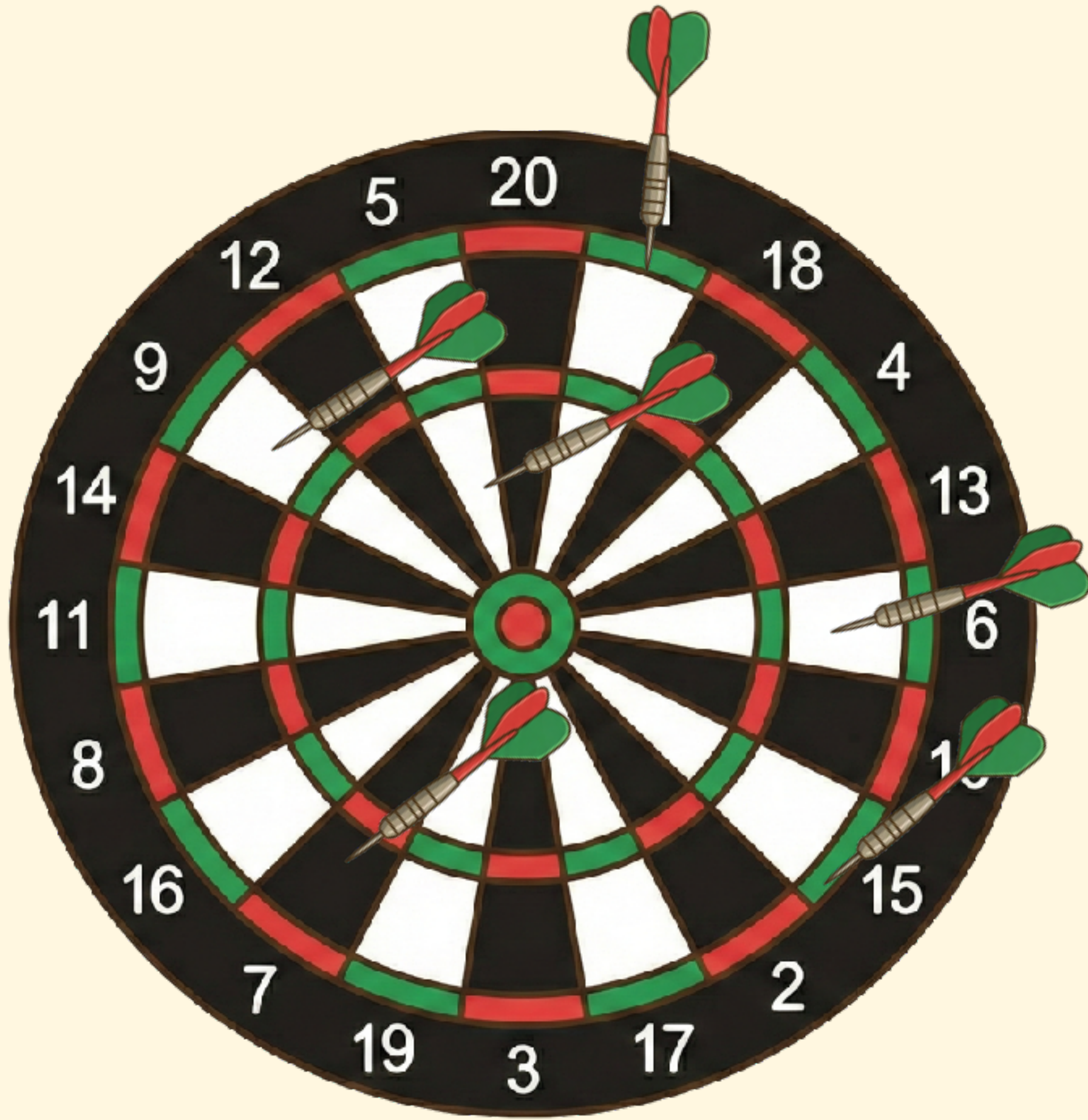
Player 1 miss.

Andy Niedermaier's Dart Game

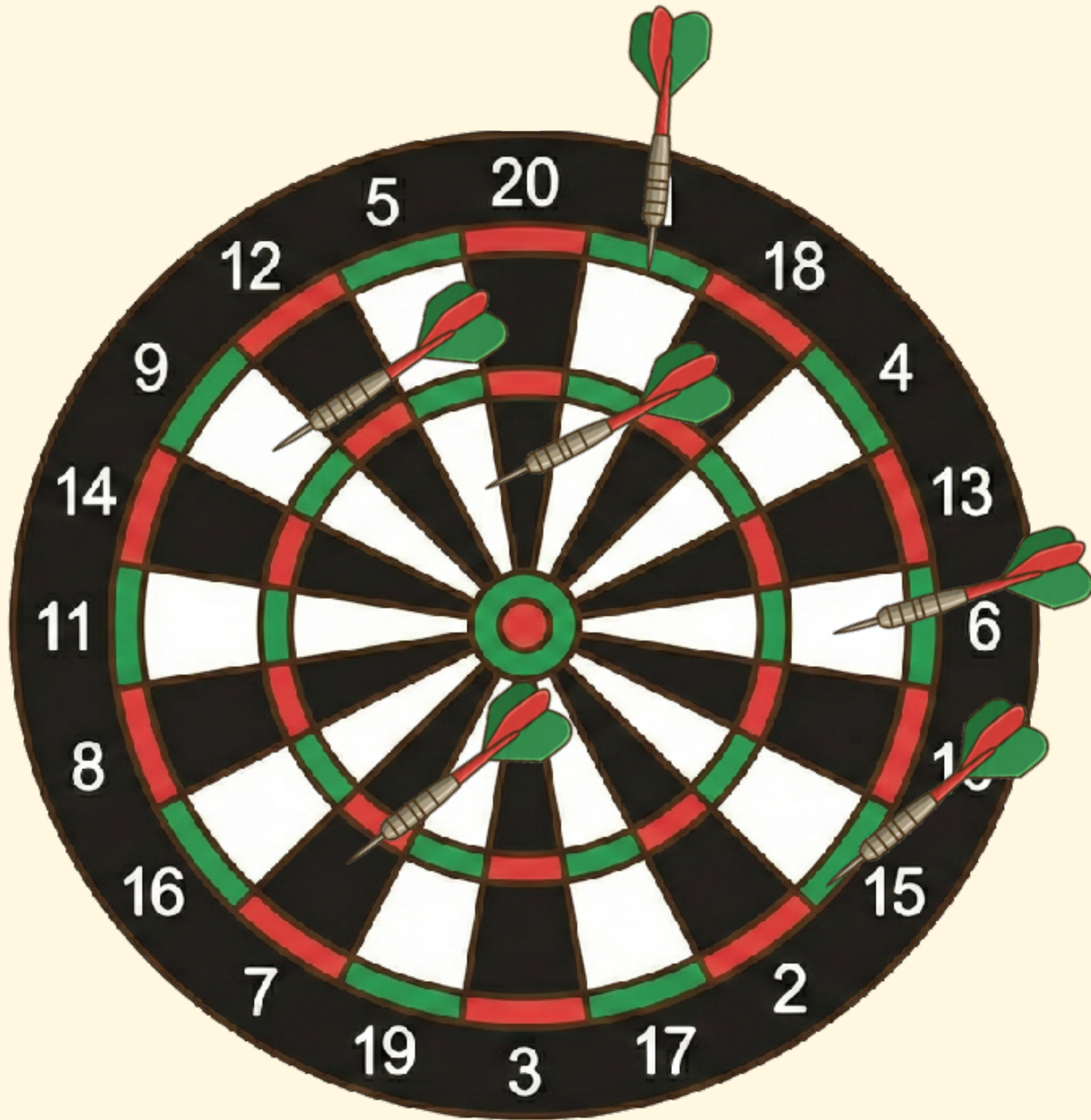


Player 1 miss.

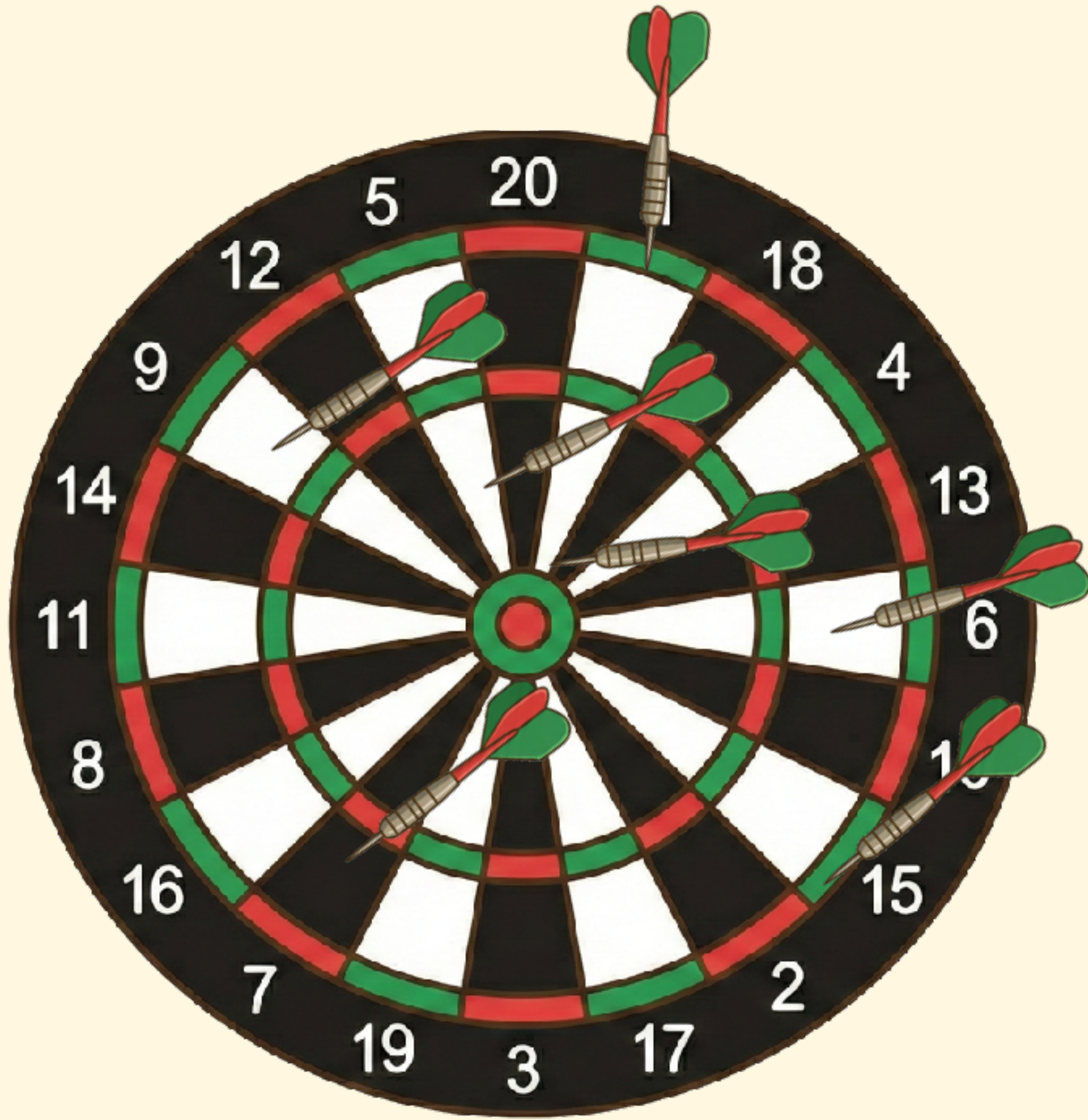
Andy Niedermaier's Dart Game



Andy Niedermaier's Dart Game

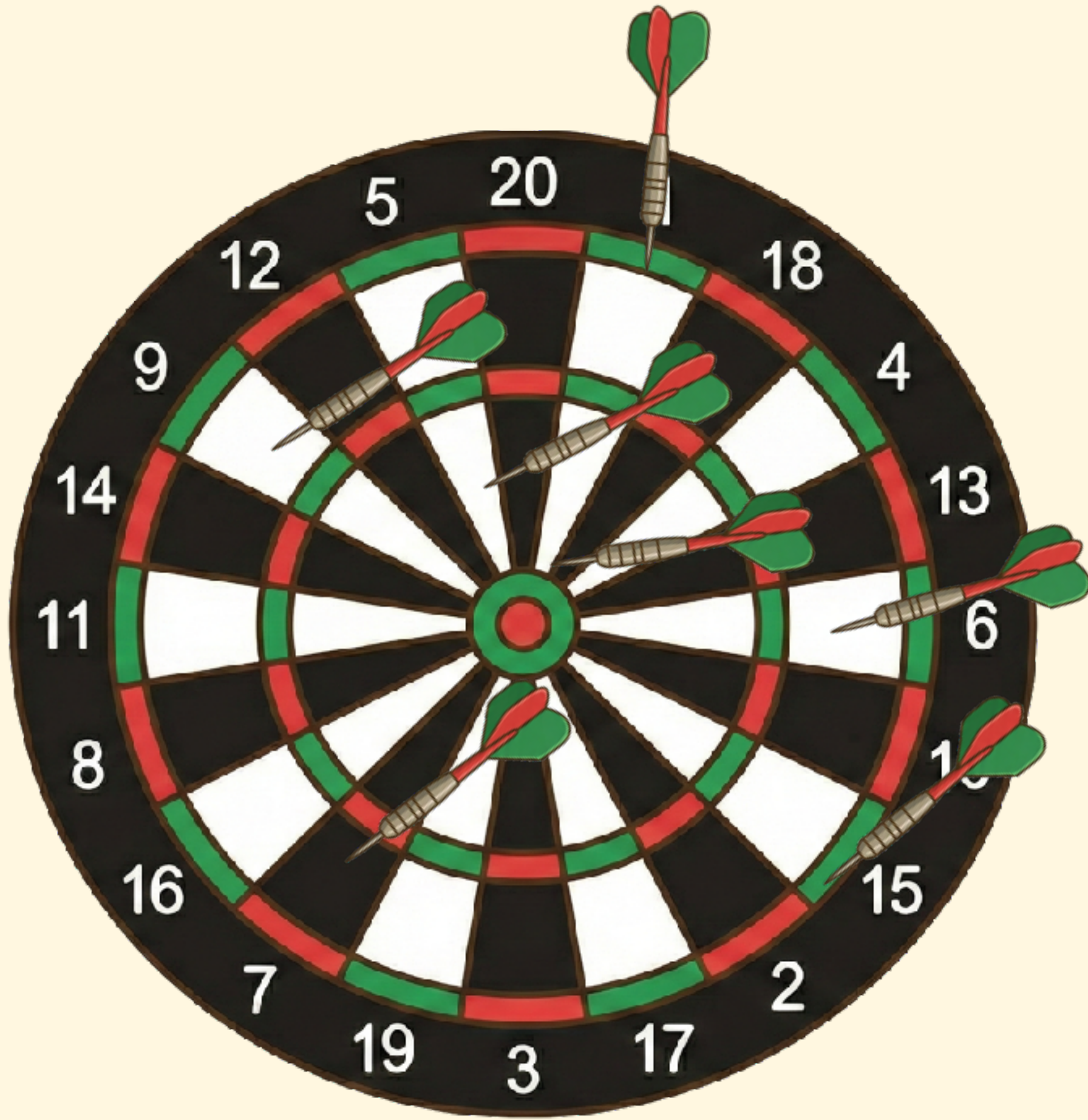
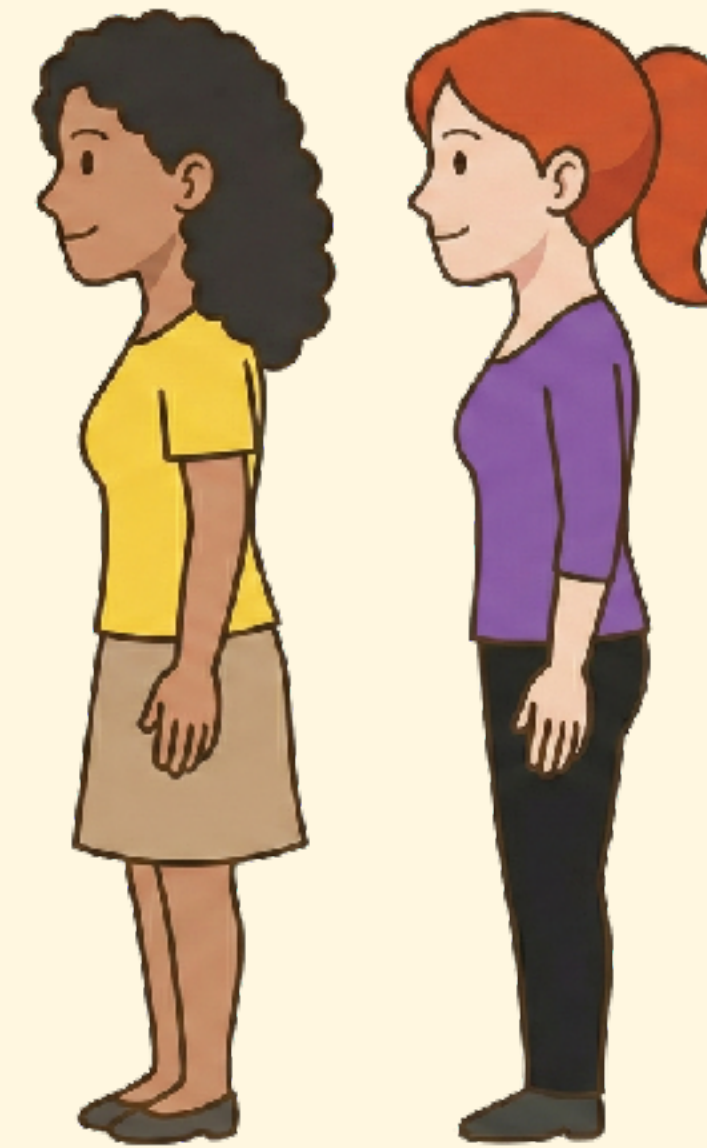


Andy Niedermaier's Dart Game

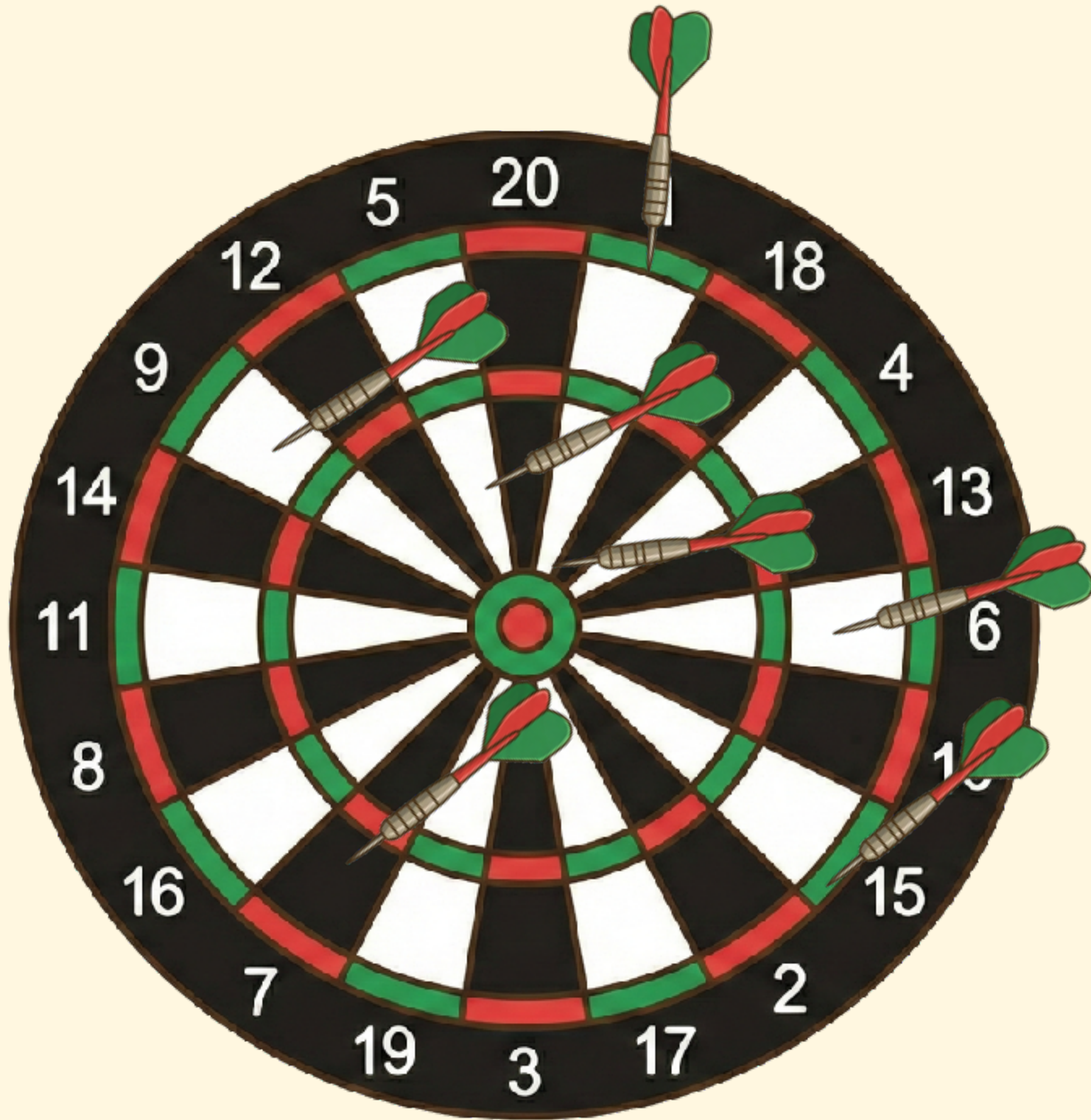


Player 2 hit.

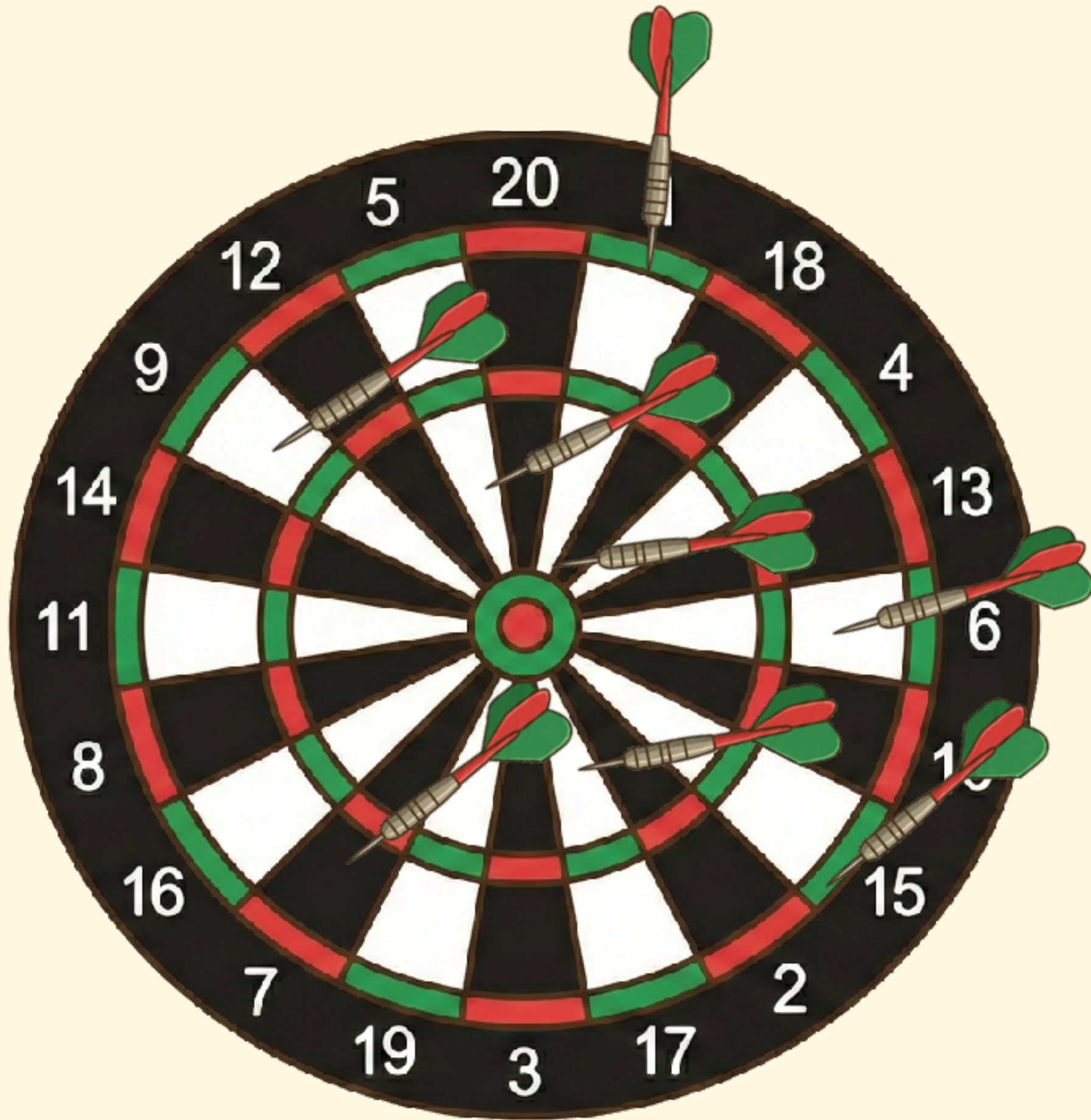
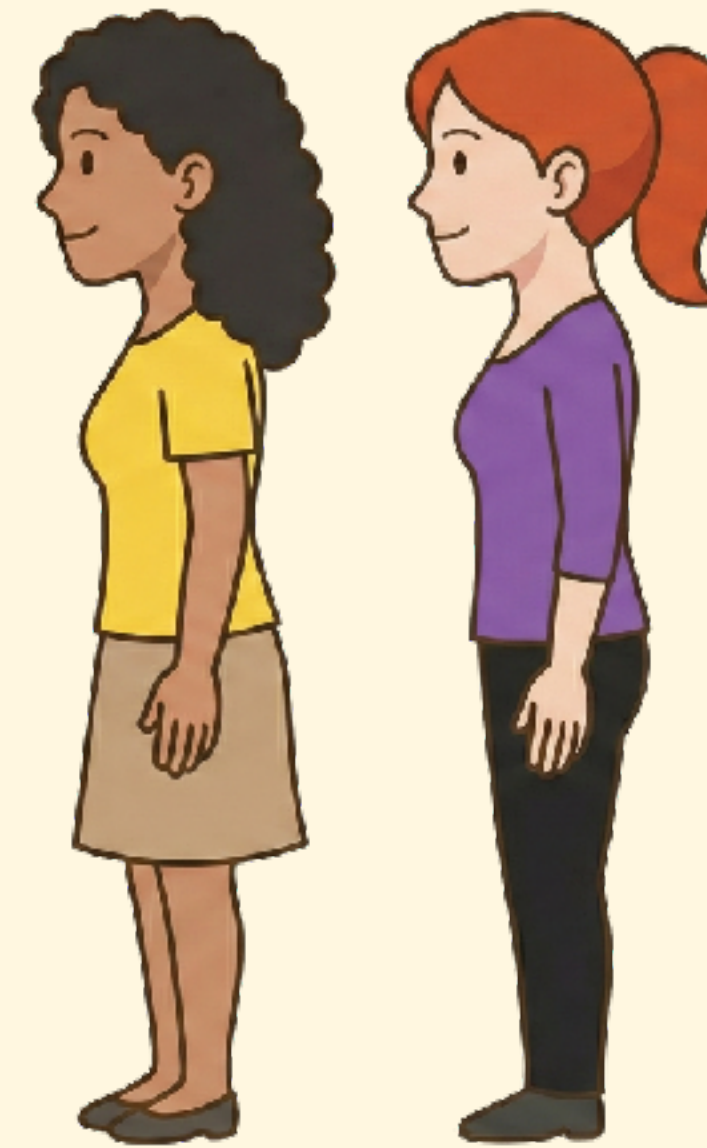
Andy Niedermaier's Dart Game



Andy Niedermaier's Dart Game

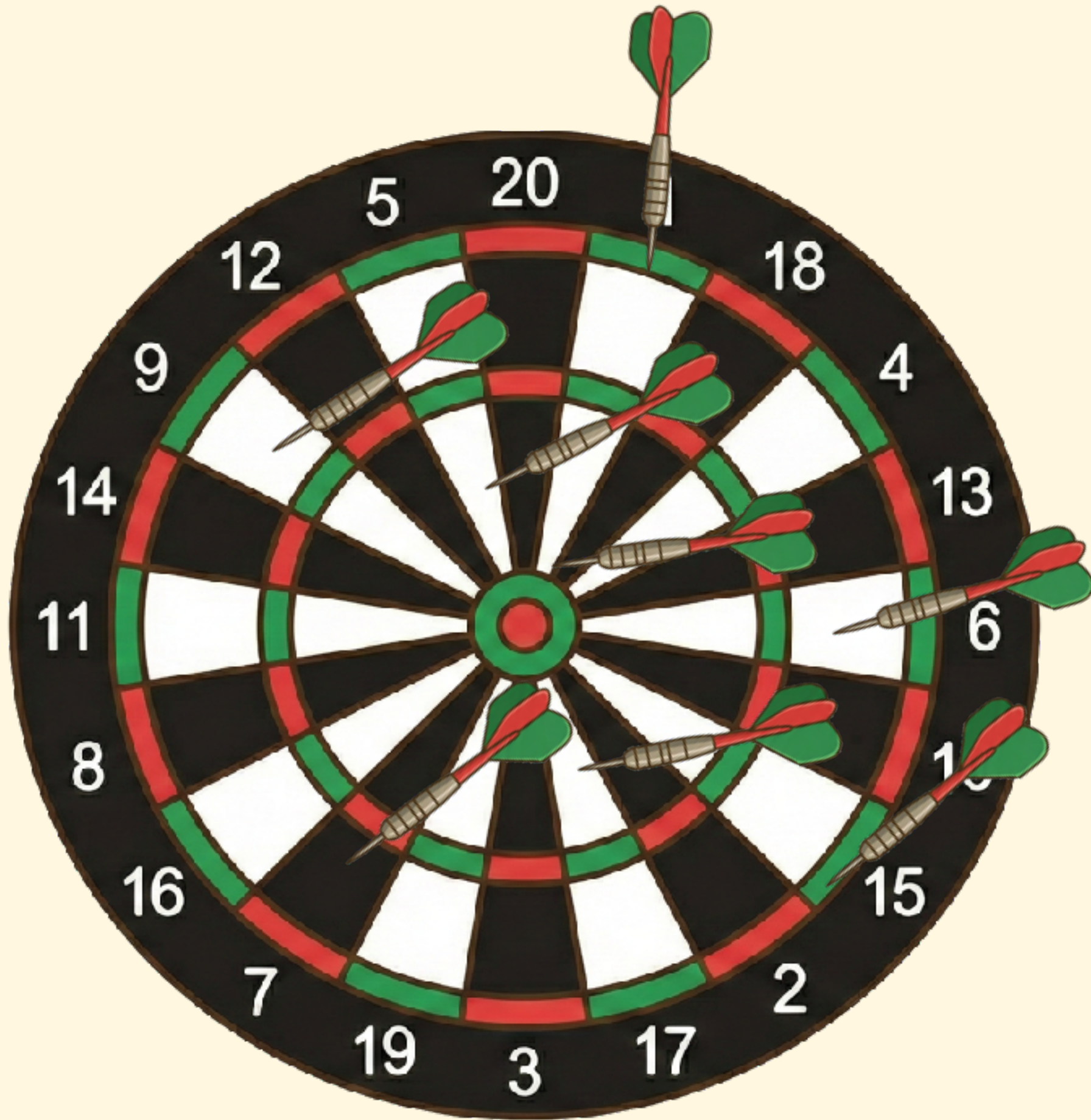


Andy Niedermaier's Dart Game



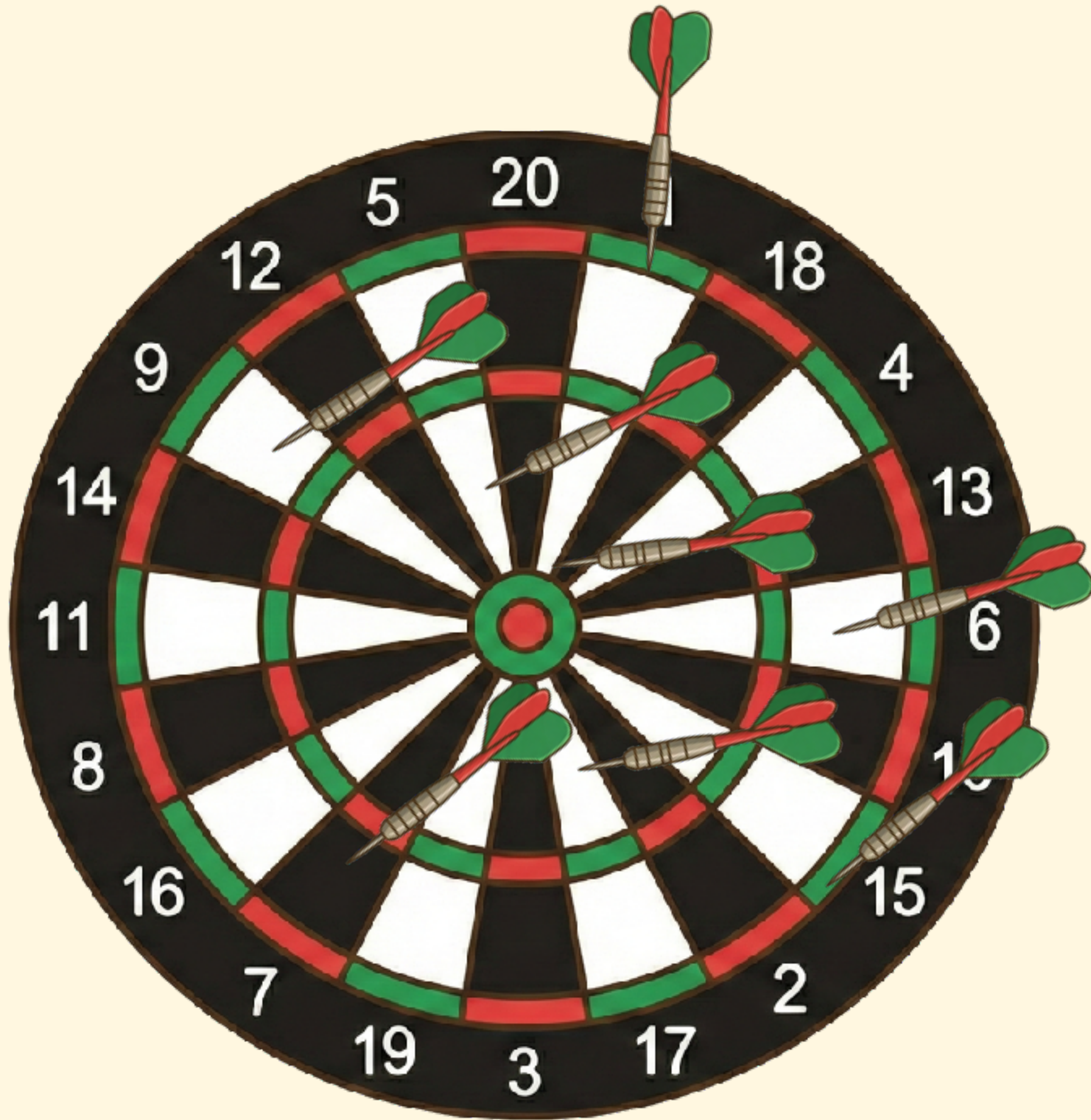
Player 4 miss.

Andy Niedermaier's Dart Game



Player 4 miss.

Andy Niedermaier's Dart Game



Player 2 has won the game!

Andy Niedermaier's Dart Game

We assume:

- the players never miss the dart board
- the locations of their throws are IID random variables
- the CDF is continuous (ties occur with probability 0)

Andy Niedermaier's Dart Game

We assume:

- the players never miss the dart board
- the locations of their throws are IID random variables
- the CDF is continuous (ties occur with probability 0)

Main question: In a game with m players, what is the probability that the player who starts k^{th} in line wins? We will call this quantity $P_{m,k}$.

Andy Niedermaier's Dart Game

In his talk, Andy found the win probabilities for 2-player, 3-player, and 4-player games using a method I'll describe later.

Andy Niedermaier's Dart Game

In his talk, Andy found the win probabilities for 2-player, 3-player, and 4-player games using a method I'll describe later.

$$P_{2,1} = 1 - \frac{1}{e} \approx 0.63212$$

$$P_{2,2} = \frac{1}{e} \approx 0.36788$$

Andy Niedermaier's Dart Game

In his talk, Andy found the win probabilities for 2-player, 3-player, and 4-player games using a method I'll describe later.

$$P_{3,1} = 1 - \frac{2}{\sqrt{3}e} \sin\left(\frac{\sqrt{3}}{2}\right) \approx 0.46649$$

$$P_{3,2} = -\frac{1}{e} + \frac{1}{\sqrt{3}e} \sin\left(\frac{\sqrt{3}}{2}\right) + \frac{1}{\sqrt{e}} \cos\left(\frac{\sqrt{3}}{2}\right) \approx 0.29182$$

$$P_{3,3} = \frac{1}{e} + \frac{1}{\sqrt{3}e} \sin\left(\frac{\sqrt{3}}{2}\right) - \frac{1}{\sqrt{e}} \cos\left(\frac{\sqrt{3}}{2}\right) \approx 0.24167$$

Andy Niedermaier's Dart Game

In his talk, Andy found the win probabilities for 2-player, 3-player, and 4-player games using a method I'll describe later.

$$P_{4,1} = 1 + \frac{2}{\sqrt{3}e} \sin\left(\frac{\sqrt{3}}{2}\right) - \frac{2}{\sqrt{e}} \cos\left(\frac{\sqrt{3}}{2}\right) - \frac{5}{4} \sin(1) + \frac{5}{4} \cos(1) \approx 0.37115$$

$$P_{4,2} = \frac{1}{e} - \frac{4}{\sqrt{3}e} \sin\left(\frac{\sqrt{3}}{2}\right) - \frac{2}{\sqrt{3}} \cos\left(\frac{\sqrt{3}}{2}\right) + \frac{5}{4} \sin(1) + \frac{5}{4} \cos(1) \approx 0.24219$$

$$P_{4,3} = -\frac{3}{2e} + \frac{\sqrt{3}}{\sqrt{e}} \sin\left(\frac{\sqrt{3}}{2}\right) + \frac{3}{\sqrt{e}} \cos\left(\frac{\sqrt{3}}{2}\right) + \frac{5}{4} \sin(1) - \frac{5}{4} \cos(1) \approx 0.20322$$

$$P_{4,4} = \frac{1}{2e} + \frac{5}{\sqrt{3}e} \sin\left(\frac{\sqrt{3}}{2}\right) + \frac{1}{\sqrt{e}} \cos\left(\frac{\sqrt{3}}{2}\right) - \frac{5}{4} \sin(1) - \frac{5}{4} \cos(1) \approx 0.18344$$

Andy Niedermaier's Dart Game

Andy's method doesn't work for 5 players because the computation (solving a system of differential equations) is too difficult.

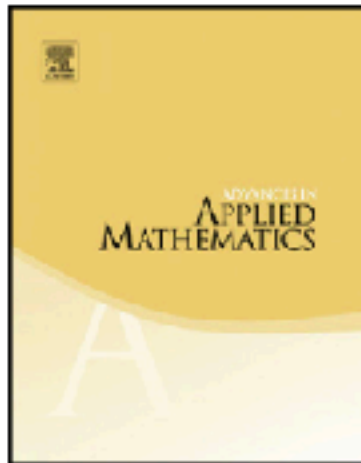
Sergi Elizalde's contributions

A few months after Andy's talk, Sergi wrote a paper studying Andy's dart game further.



ELSEVIER

Contents lists available at [ScienceDirect](#)



Advances in Applied Mathematics

journal homepage: www.elsevier.com/locate/yaama

On a dart game of Niedermaier



Sergi Elizalde

Department of Mathematics, Dartmouth College, Hanover, NH 03755, USA

ARTICLE INFO

ABSTRACT

Article history:
Received 15 September 2022
Received in revised form 1 January 2023
Accepted 2 January 2023
Available online 12 January 2023

MSC:
05A05

We analyze a game introduced by Andy Niedermaier, where m players take turns throwing a dart at a dartboard. A player is eliminated unless his dart lands closer to the center than all previously thrown darts, in which case he goes to the back of the line, until only one player remains. Using generating functions, we determine the distribution of the number of throws in the game, and we obtain a recursive formula to compute the probability that each player wins.

Sergi Elizalde's contributions

A few months after Andy's talk, Sergi wrote a paper studying Andy's dart game further.

Among many other results, he found a generating function recurrence for functions that give the win probabilities.

Theorem 4.4. *For $m \geq 2$, the generating functions $R_m(x, t, z)$ satisfy the recurrence*

$$R_m(x, t, z) = \frac{t^{m+1} - t}{m} \sum_{j=0}^{m-1} \frac{\tilde{R}_m(x, \zeta_m^j, z)}{t - \zeta_m^j}, \quad (4)$$

where $\zeta_m = e^{2\pi i/m}$ is a primitive m th root of unity, and

$$\tilde{R}_m(x, t, z) = \left(z^{m-1} + tz \int_0^x \frac{\partial}{\partial v} [(1-v)R_{m-1}(v, t, z)] e^{-vtz} dv \right) e^{xtz}, \quad (5)$$

with initial condition $R_1(x, t, z) = t$.

The win probability
of player k in an m -
player game is

$$[t^k]R_m(1, t, 1)$$

Sergi Elizalde's contributions

Sergi uses this recurrence to compute the 5-player win probabilities.

$$\begin{aligned} P_{5,1} = & 1 - \frac{25}{2} \cos(1) - \frac{\sqrt{10 - 2\sqrt{5}} (3\sqrt{5} + 2)}{10} e^{\frac{\sqrt{5}-1}{4}} \sin \left(\frac{\sqrt{10 - 2\sqrt{5}} (\sqrt{5} + 1)}{8} \right) \\ & + \frac{3\sqrt{5} + 7}{2} e^{\frac{\sqrt{5}-1}{4}} \cos \left(\frac{\sqrt{10 - 2\sqrt{5}} (\sqrt{5} + 1)}{8} \right) - \frac{3\sqrt{5} - 7}{2} e^{-\frac{\sqrt{5}-1}{4}} \cos \left(\frac{\sqrt{10 - 2\sqrt{5}}}{4} \right) \\ & - \frac{\sqrt{10 - 2\sqrt{5}} (\sqrt{5} + 13)}{20} e^{-\frac{\sqrt{5}-1}{4}} \sin \left(\frac{\sqrt{10 - 2\sqrt{5}}}{4} \right) \\ & + 5 e^{-\frac{1}{2}} \left(\cos \left(\frac{\sqrt{3}}{2} \right) + \frac{1}{\sqrt{3}} \sin \left(\frac{\sqrt{3}}{2} \right) \right). \end{aligned}$$

I couldn't use it to compute 6-player win probabilities, and Sergi tells me that he couldn't either.

Sergi Elizalde's contributions

Nicer form that I'll justify later:

$$1 - \frac{5}{3} \sum_{\Phi_3(\alpha)=0} e^{-\alpha}(\alpha - 1) - \frac{25}{4} \sum_{\Phi_4(\alpha)=0} e^{-\alpha} - \frac{1}{5} \sum_{\Phi_5(\alpha)=0} e^{-\alpha}(4\alpha^3 + 5\alpha^2 - 6\alpha - 8)$$

$$\Phi_3(x) = 1 + x + x^2$$

$$\Phi_4(x) = 1 + x^2$$

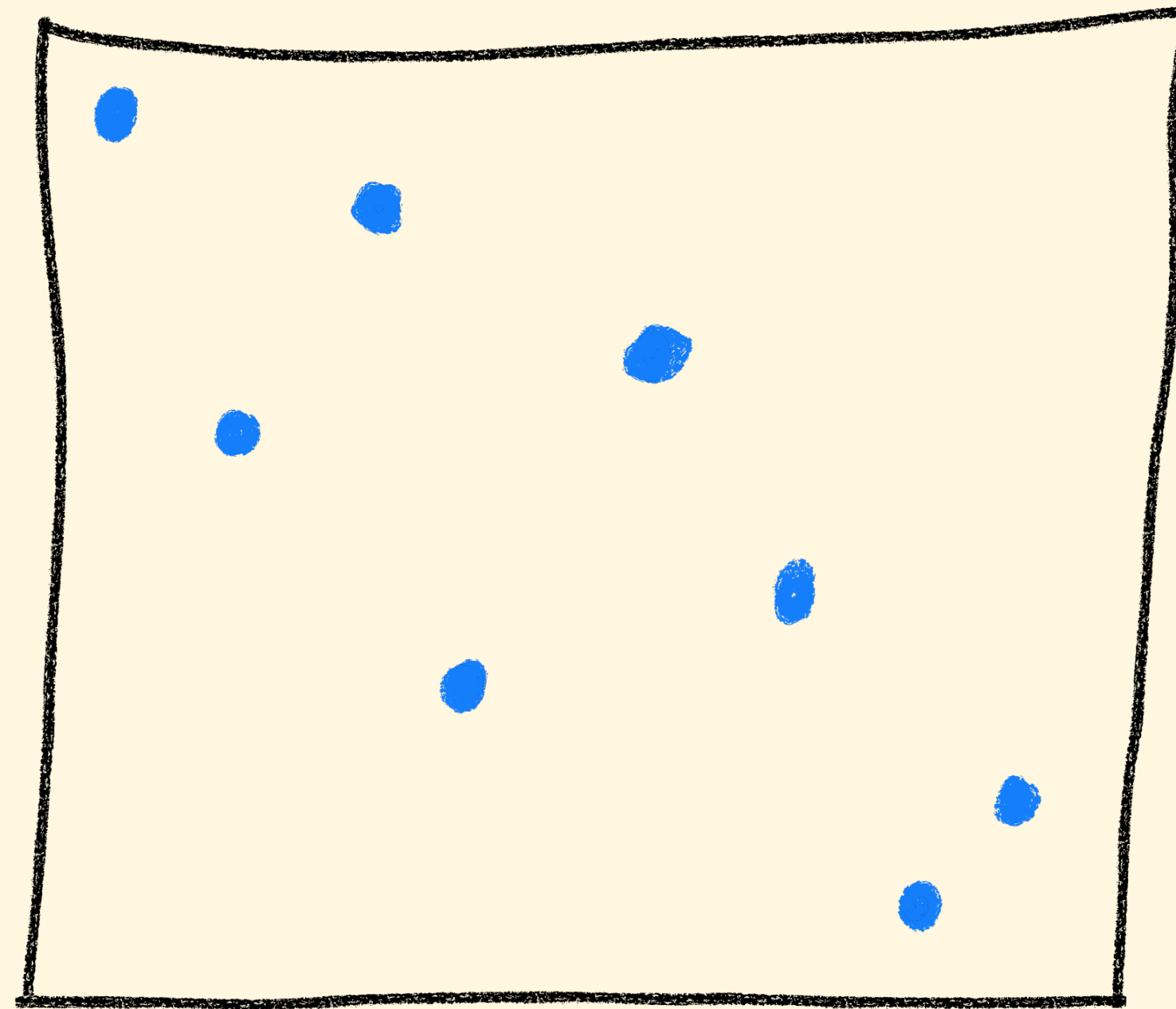
$$\Phi_5(x) = 1 + x + x^2 + x^3 + x^4$$

Two new approaches in our work

- 1) Generating function enumeration of the number of m -player games in which player k wins.
- 2) Solving Andy Niedermaier's differential systems using guess-and-check.

Dart games as permutations

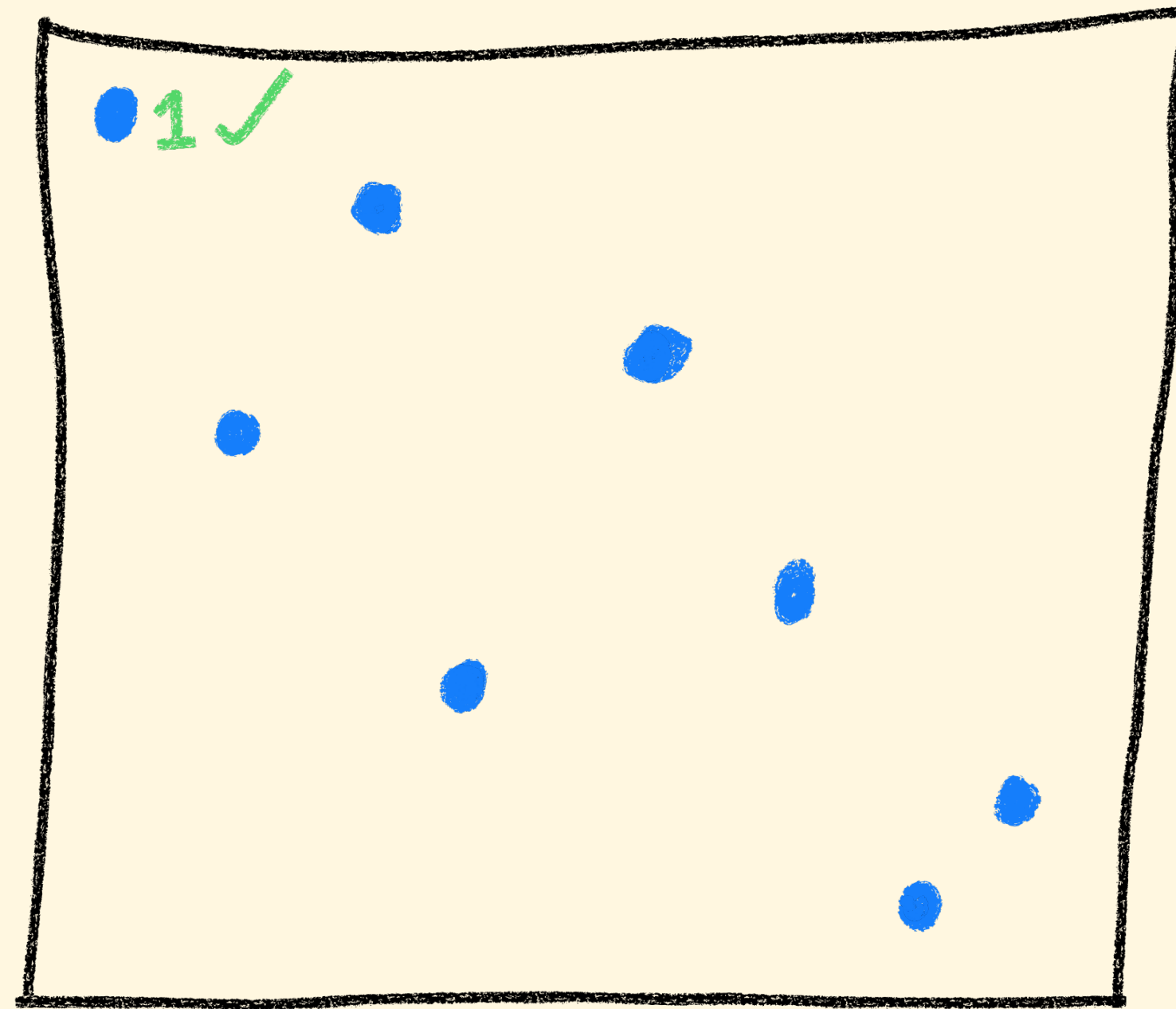
Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



8 5 7 3 6 4 1 2

Dart games as permutations

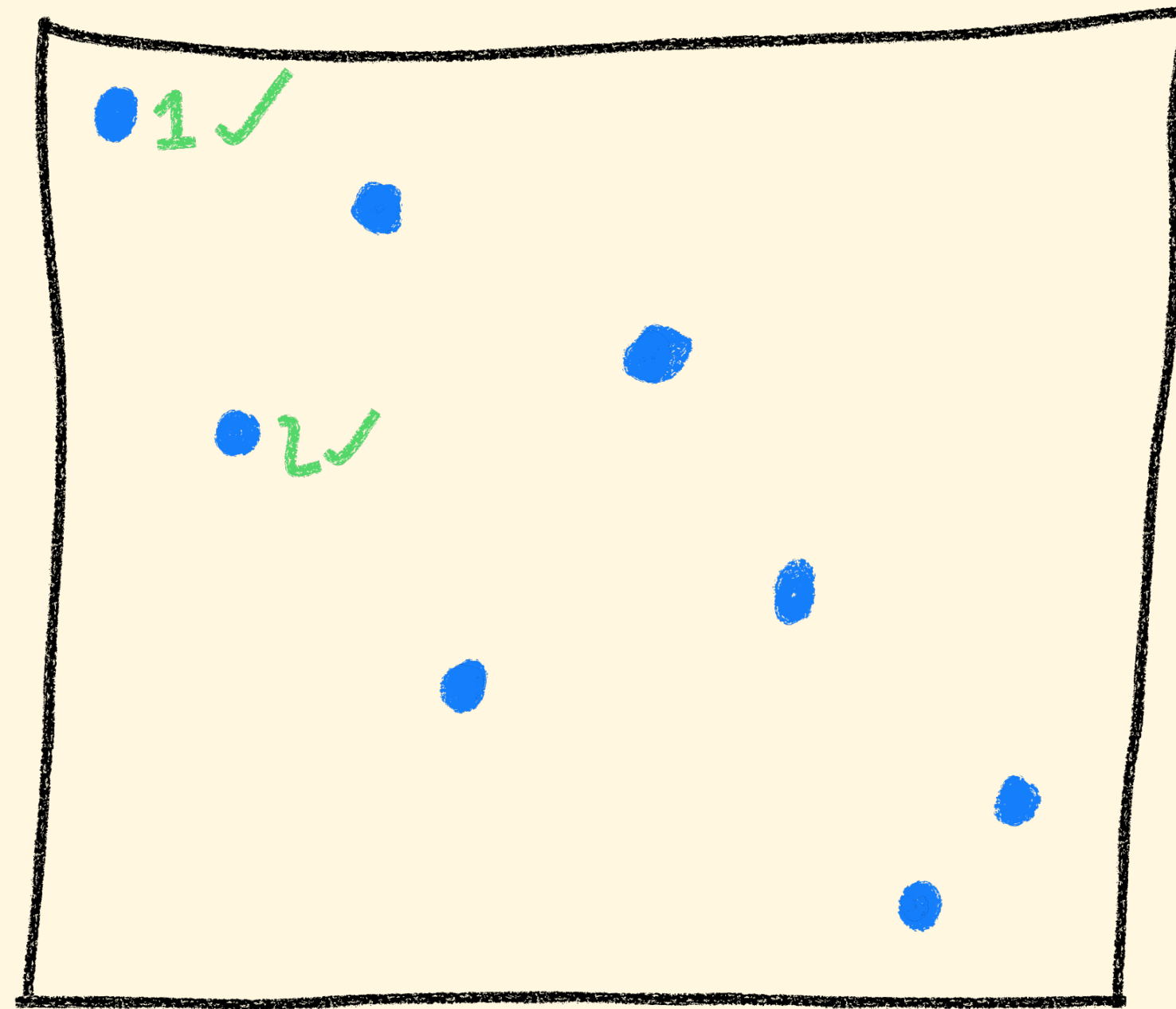
Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



8 5 7 3 6 4 1 2

Dart games as permutations

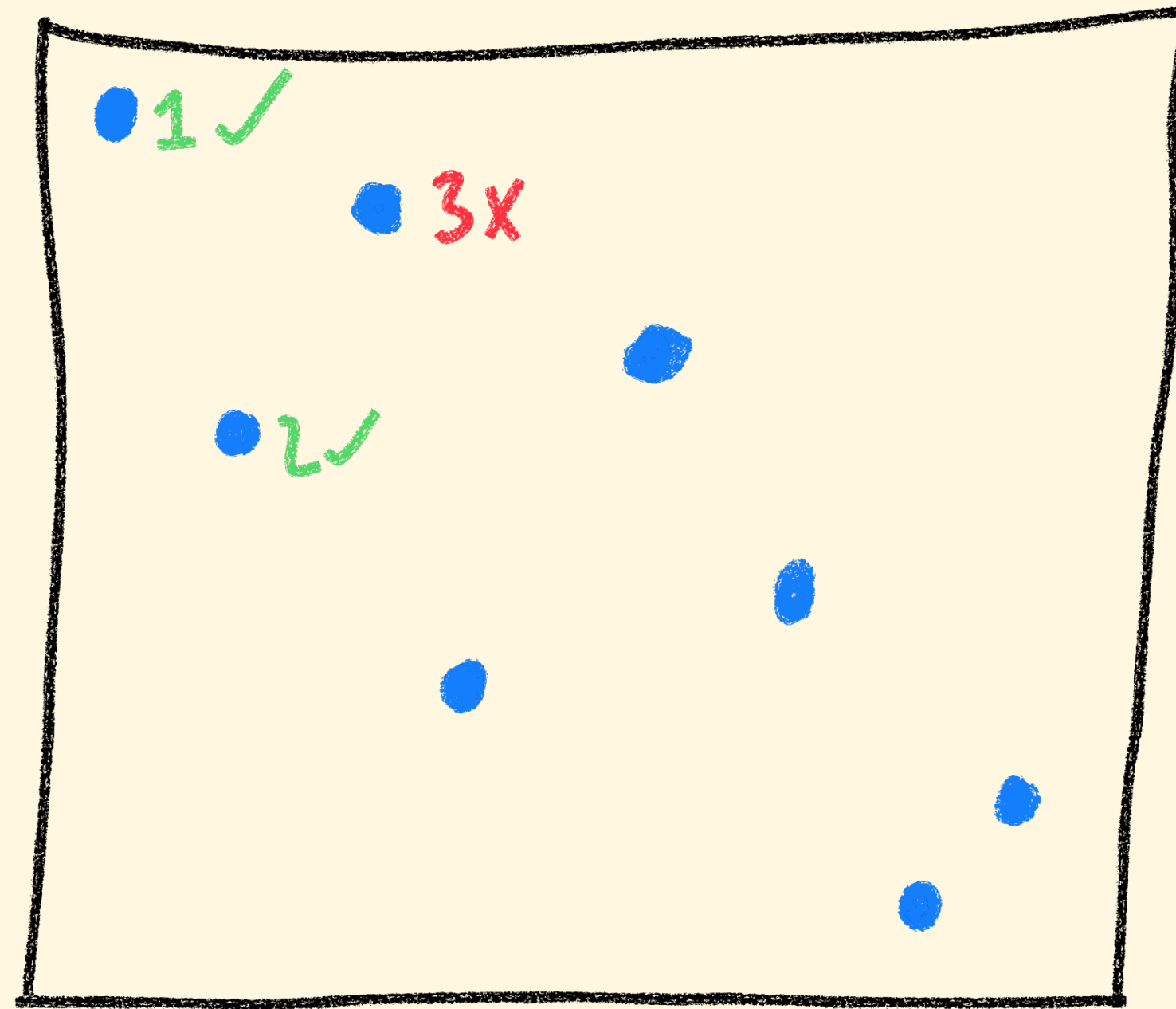
Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



8 5 7 3 6 4 1 2

Dart games as permutations

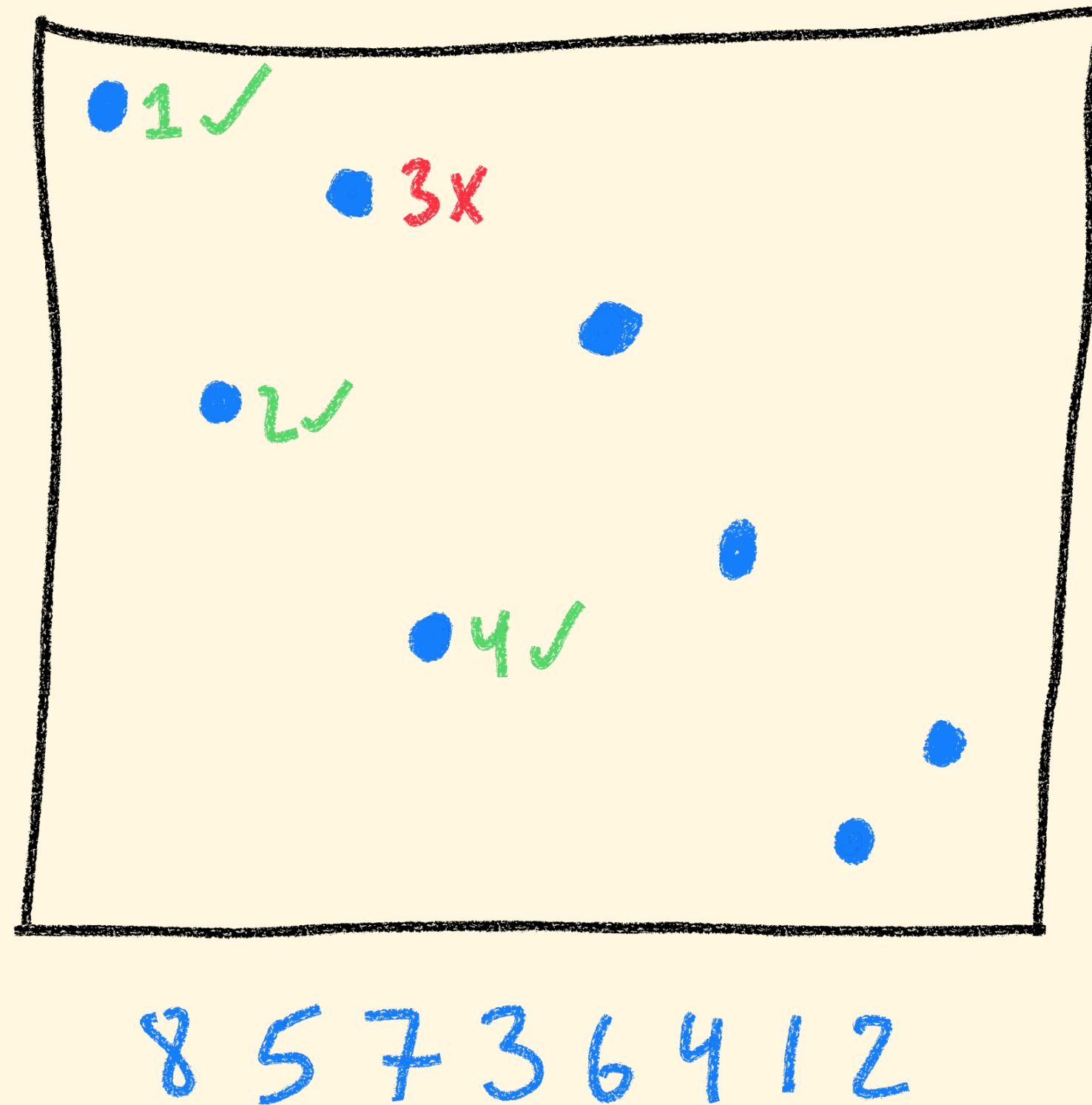
Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



8 5 7 3 6 4 1 2

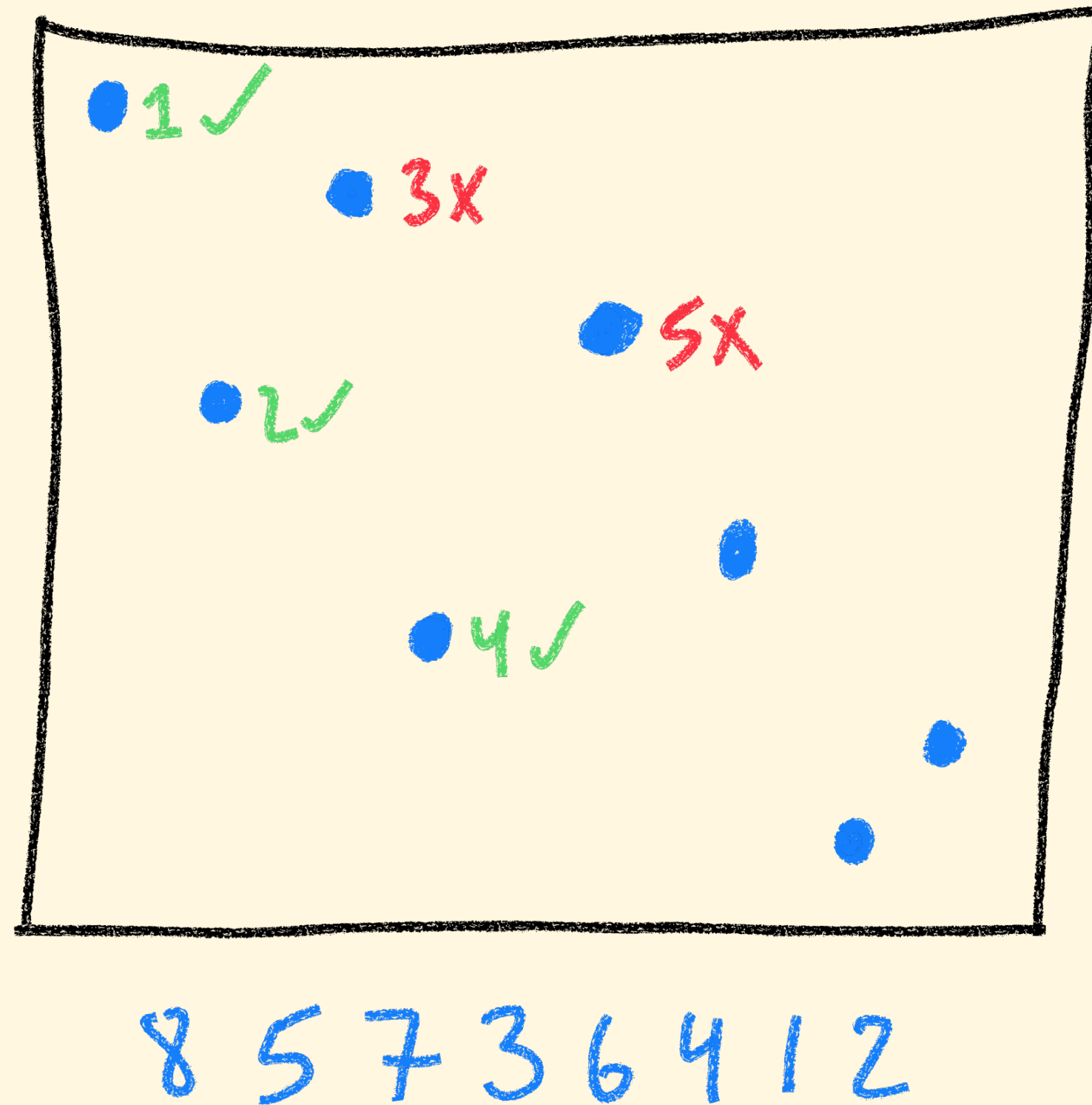
Dart games as permutations

Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



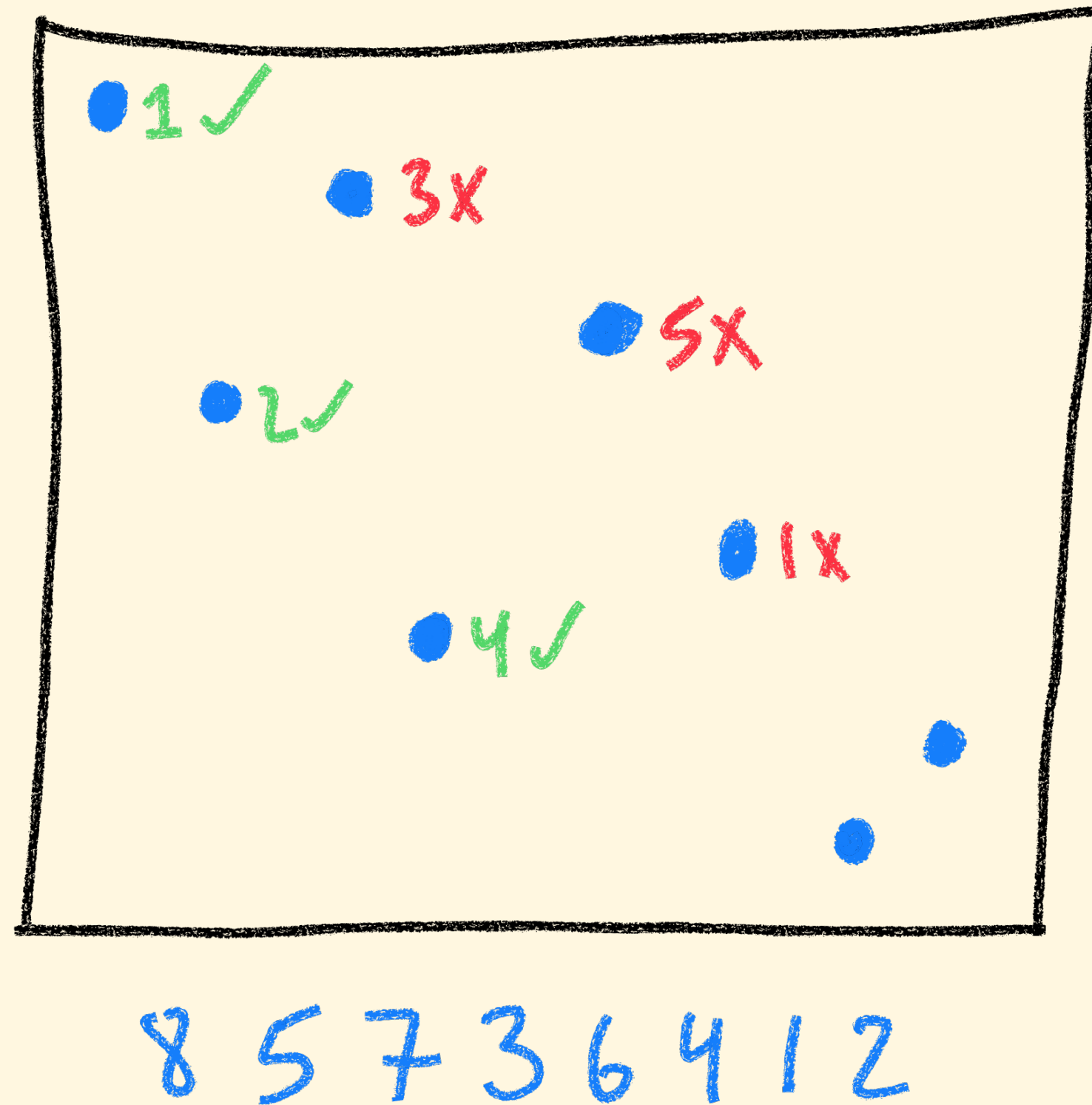
Dart games as permutations

Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



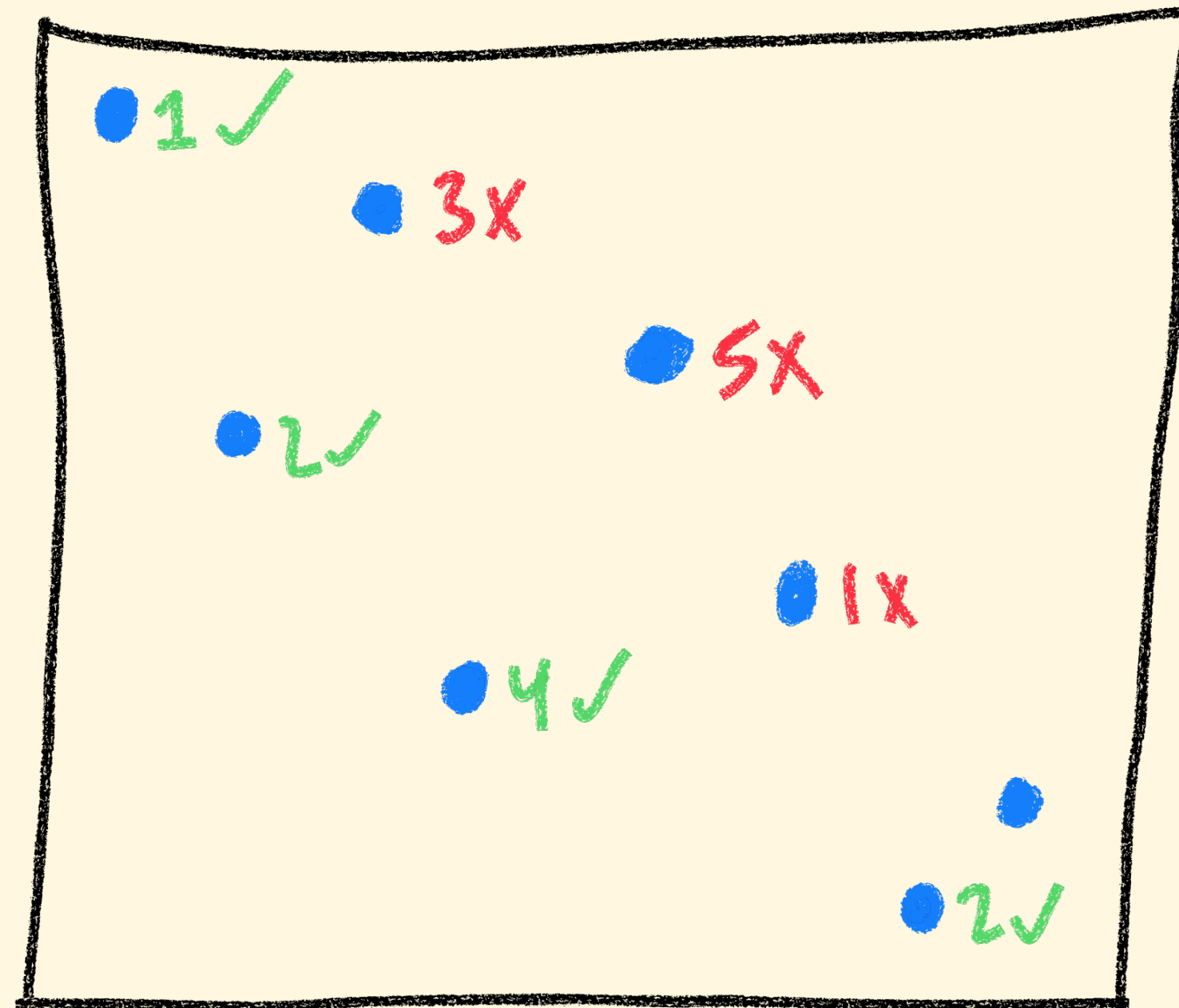
Dart games as permutations

Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



Dart games as permutations

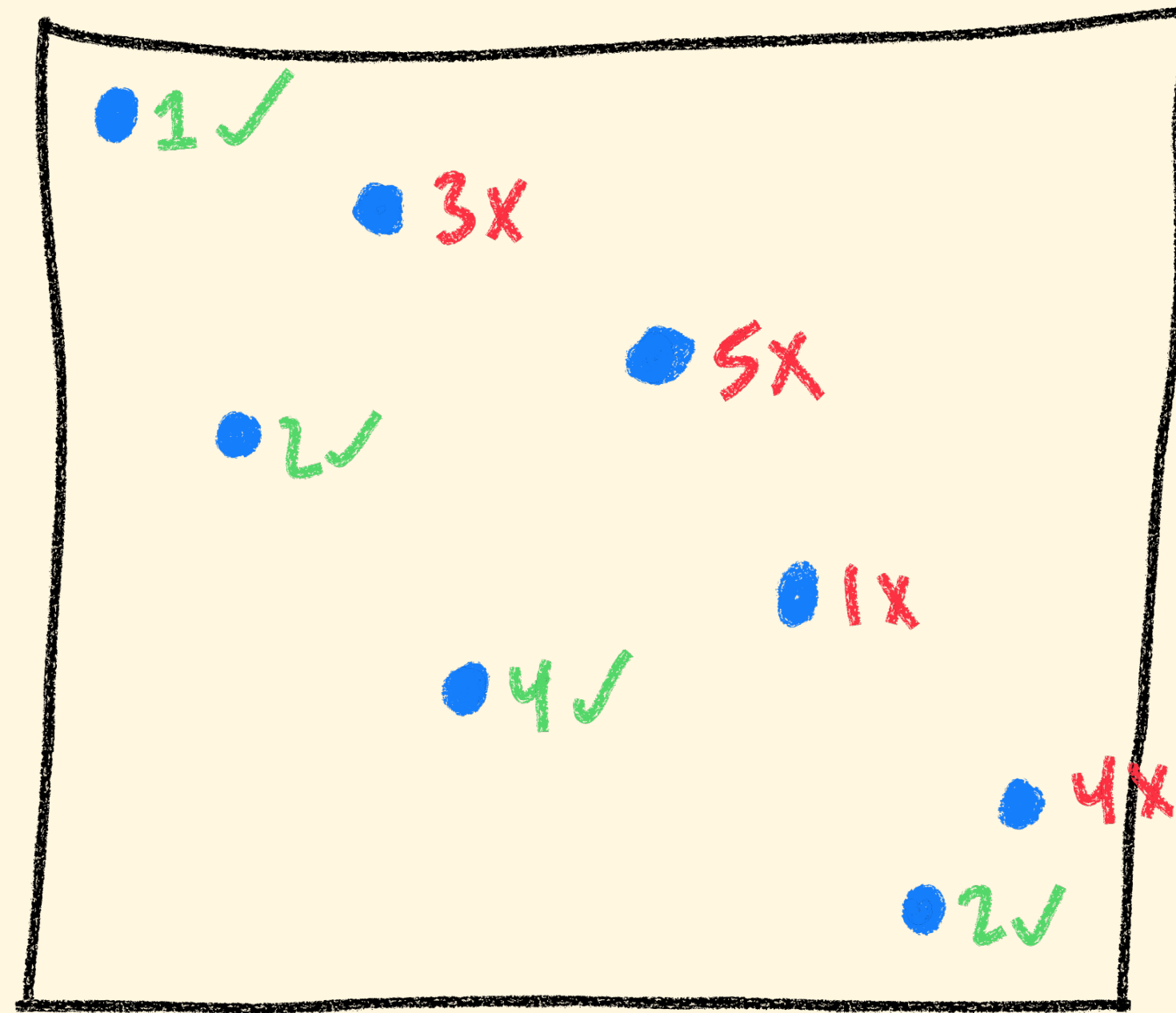
Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



8 5 7 3 6 4 1 2

Dart games as permutations

Since there are no ties, each dart game can be encoded as a permutation based on the distances of the throws to the center.



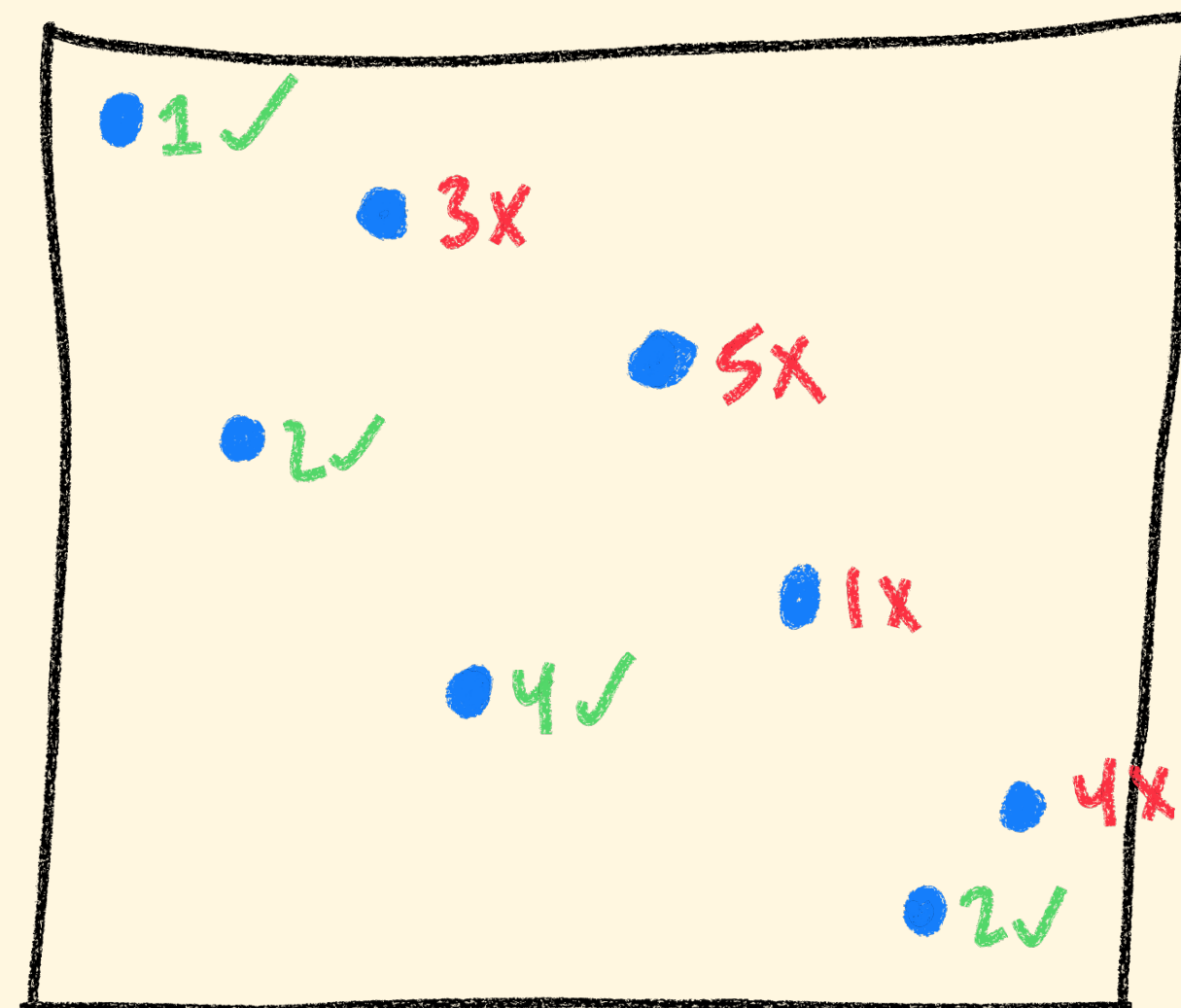
8 5 7 3 6 4 1 2

Dart games as permutations

The left-to-right minima of the permutation are the successful throws (“hits”).

The non-left-to-right minima of the permutation are the eliminating throws (“misses”).

A valid, complete game with m players corresponds to a permutation with $m - 1$ non-left-to-right minima that does not end with the entry 1.



Dart games as permutations

Define $w_{n,m,k}$ to be the number of complete games with n total throws, m players, in which player k wins.

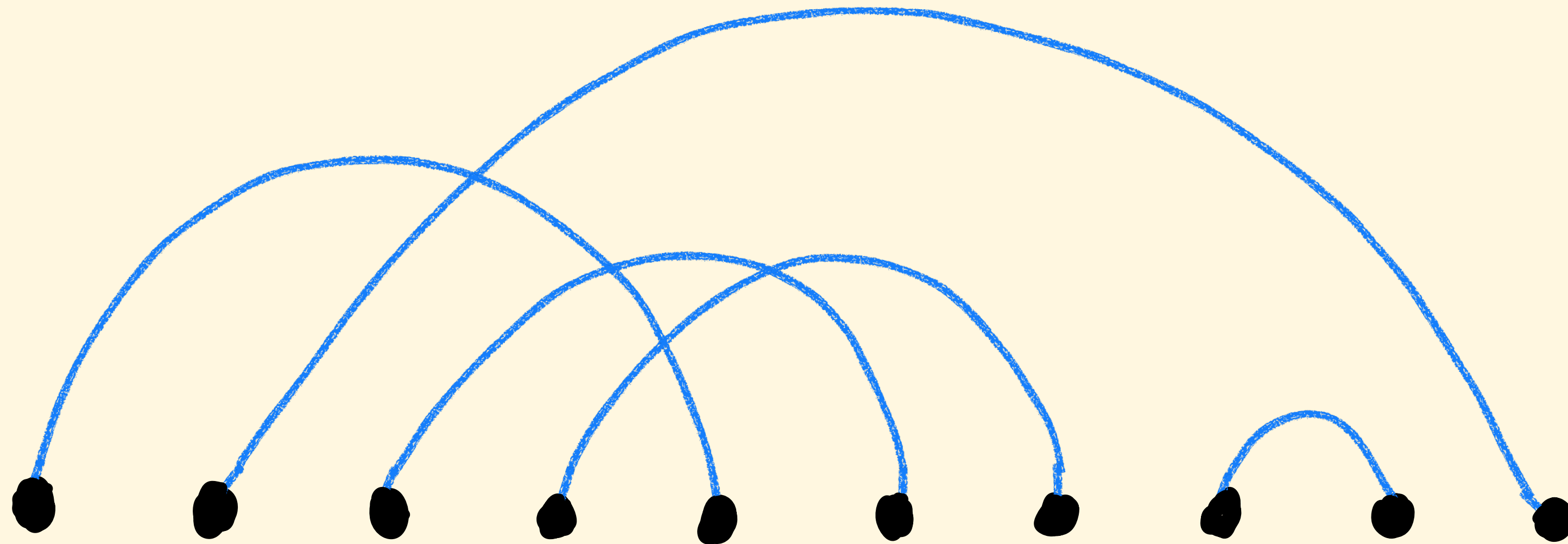
If n darts are thrown in sequence (valid game or not) each permutation of length n is equally likely to occur.

So the win probabilities are $P_{m,k} = \sum_{n \geq 0} \frac{w_{n,m,k}}{n!}$.

Dart games as permutations

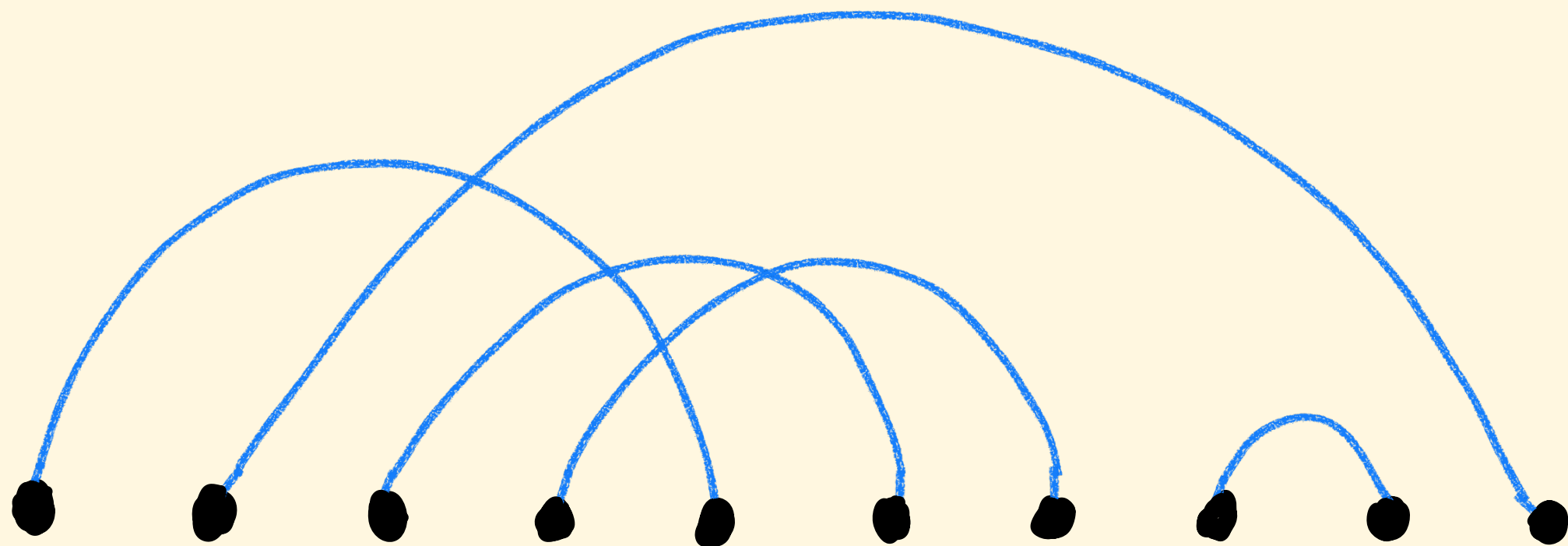
For now, forget about which player wins, and let's compute $w_{n,m}$, the number of m -player games with n throws.

We are going to model games with perfect matchings.



Dart games as matchings

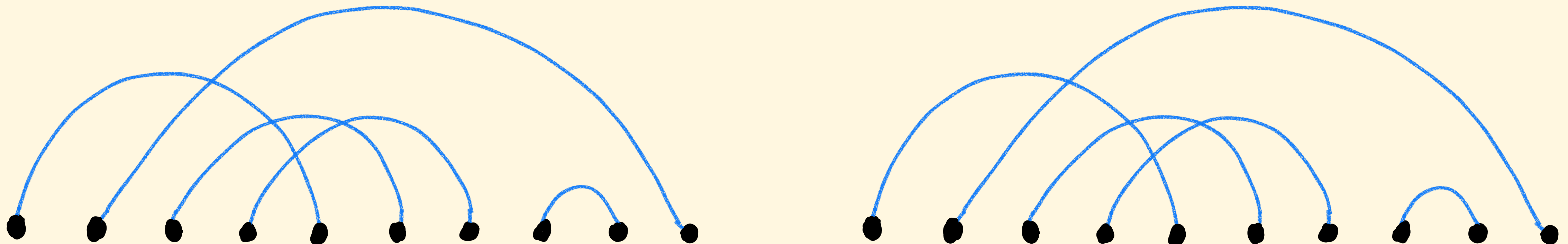
Define a *greedy matching* to be one in which you never have an opening followed by the closing of a different chord.



By the way, is there an existing name for these? I couldn't find one.

Dart games as matchings

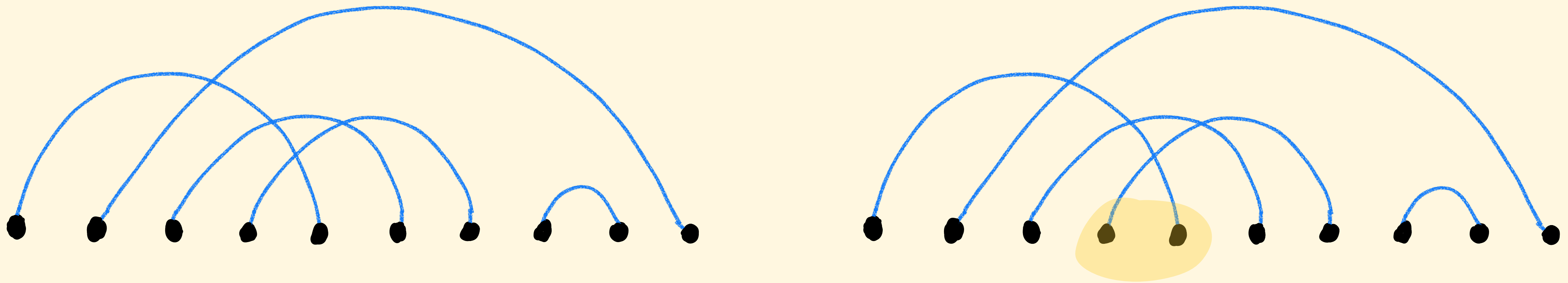
Define a *greedy matching* to be one in which you never have an opening followed by the closing of a different chord.



By the way, is there an existing name for these? I couldn't find one.

Dart games as matchings

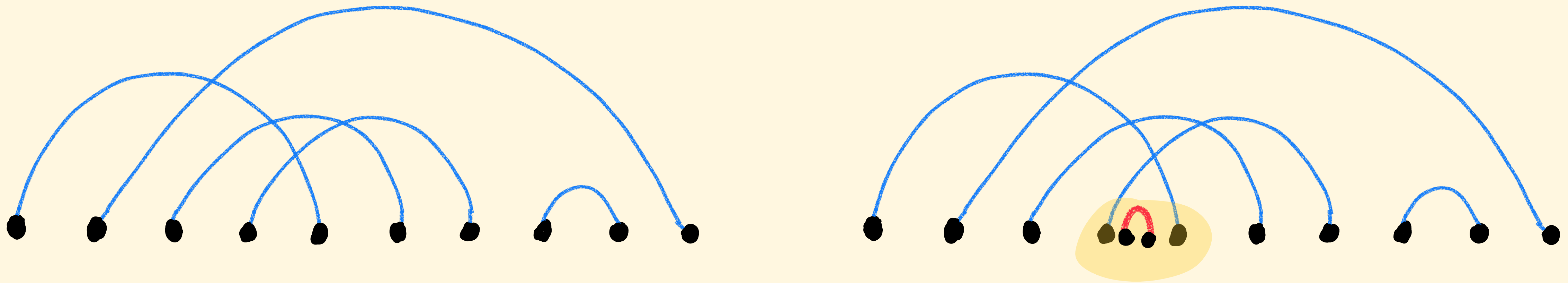
Define a *greedy matching* to be one in which you never have an opening followed by the closing of a different chord.



By the way, is there an existing name for these? I couldn't find one.

Dart games as matchings

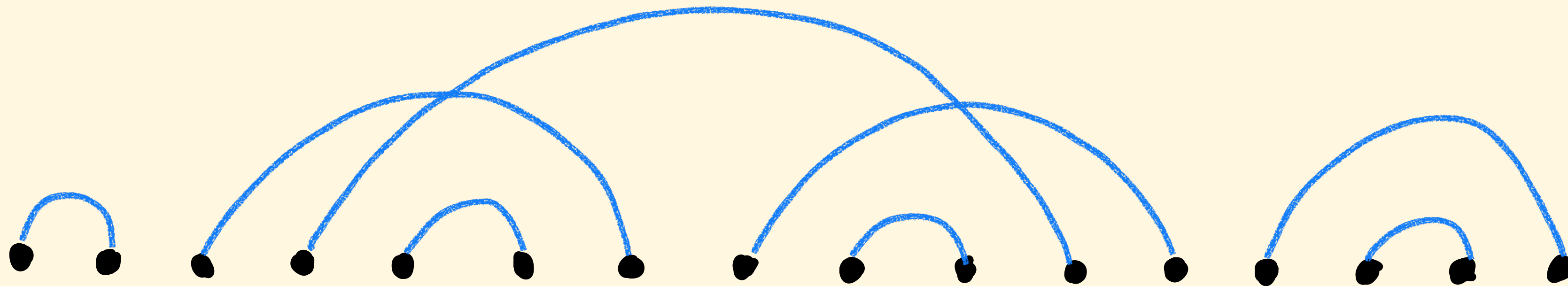
Define a *greedy matching* to be one in which you never have an opening followed by the closing of a different chord.



By the way, is there an existing name for these? I couldn't find one.

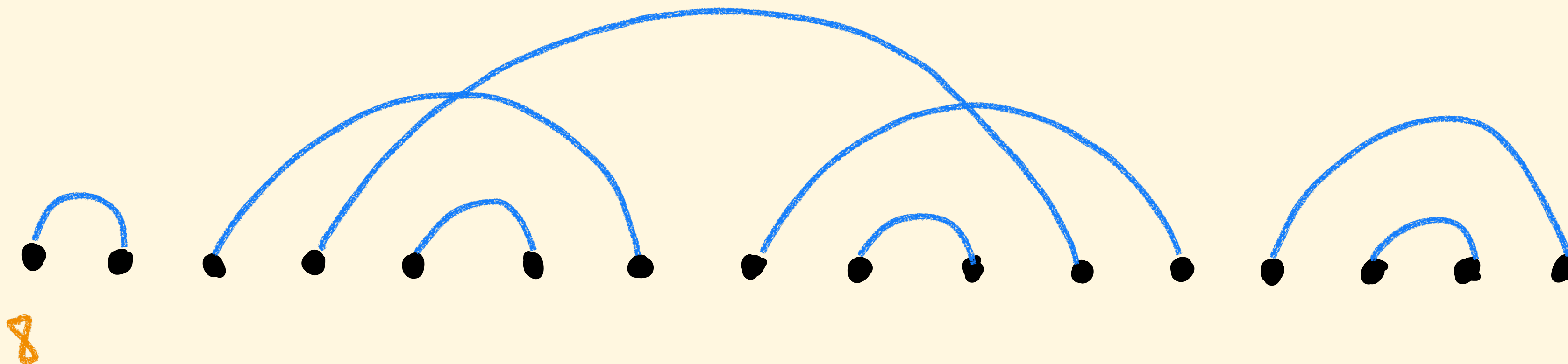
Dart games as matchings

Greedy matchings are in bijection with permutations.



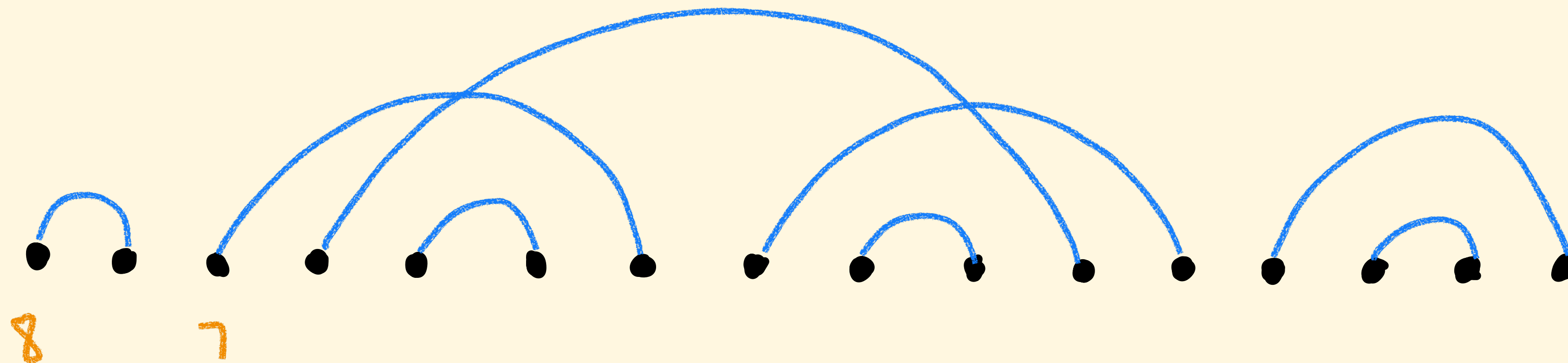
Dart games as matchings

Greedy matchings are in bijection with permutations.



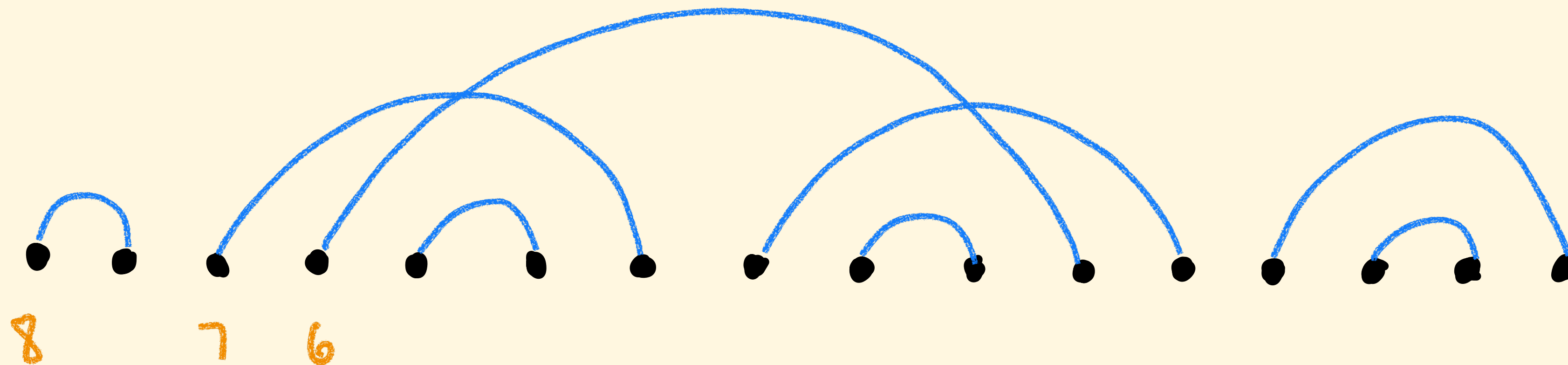
Dart games as matchings

Greedy matchings are in bijection with permutations.



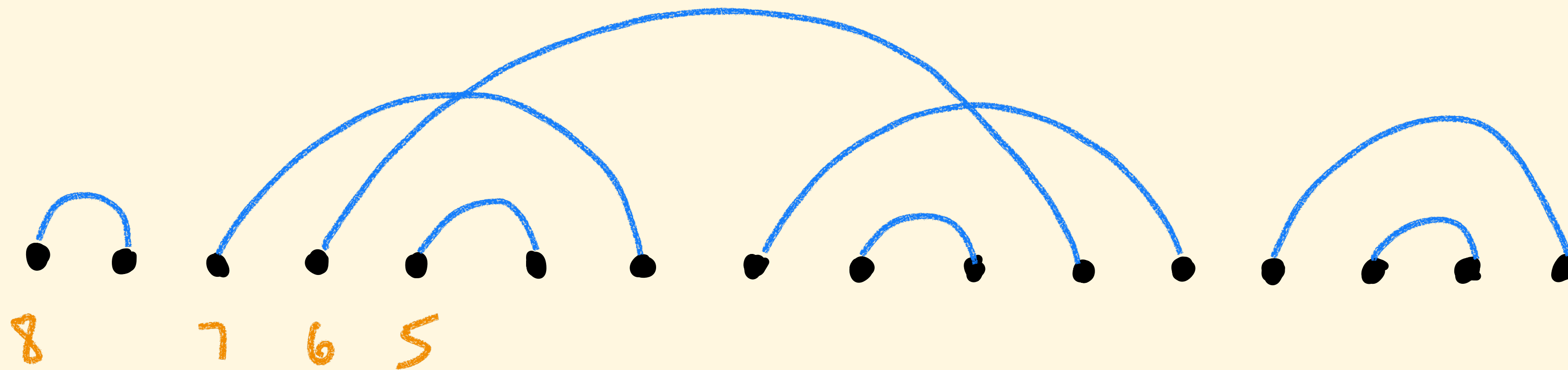
Dart games as matchings

Greedy matchings are in bijection with permutations.



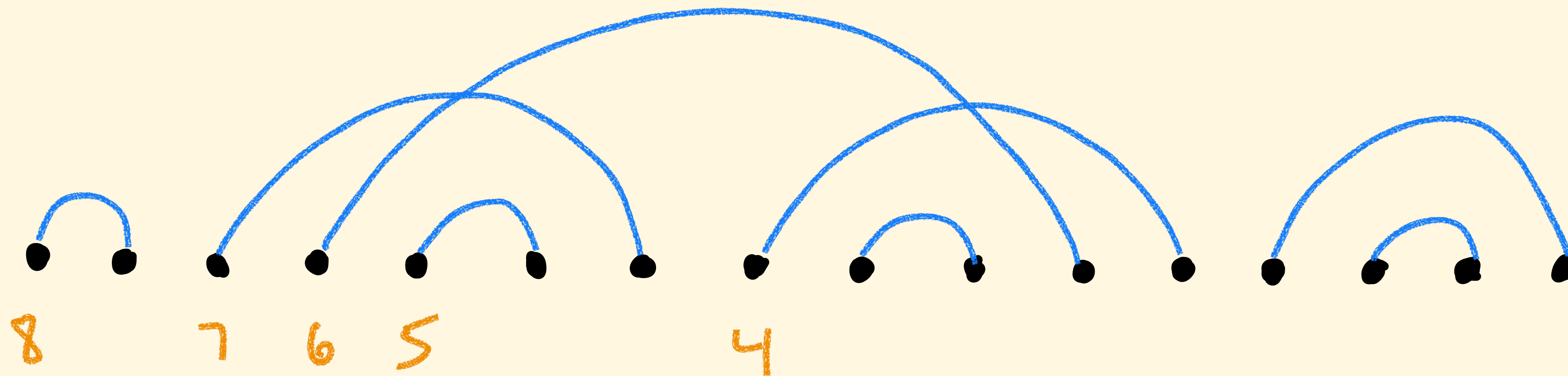
Dart games as matchings

Greedy matchings are in bijection with permutations.



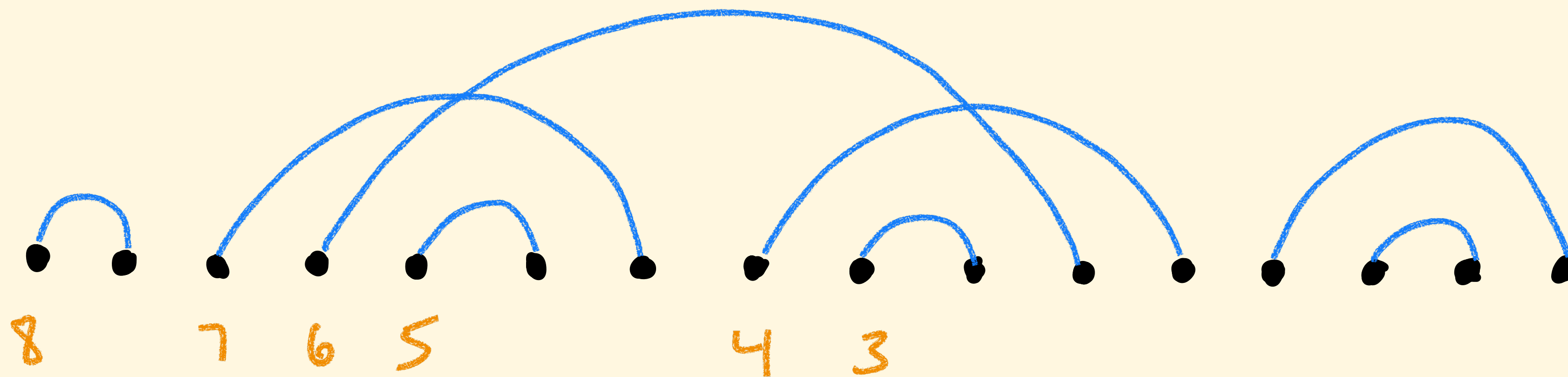
Dart games as matchings

Greedy matchings are in bijection with permutations.



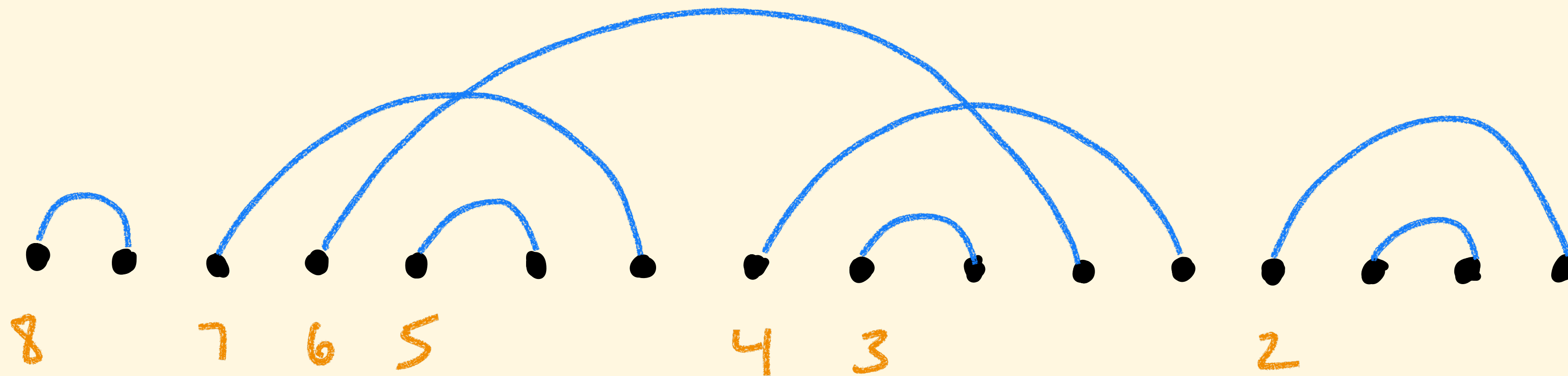
Dart games as matchings

Greedy matchings are in bijection with permutations.



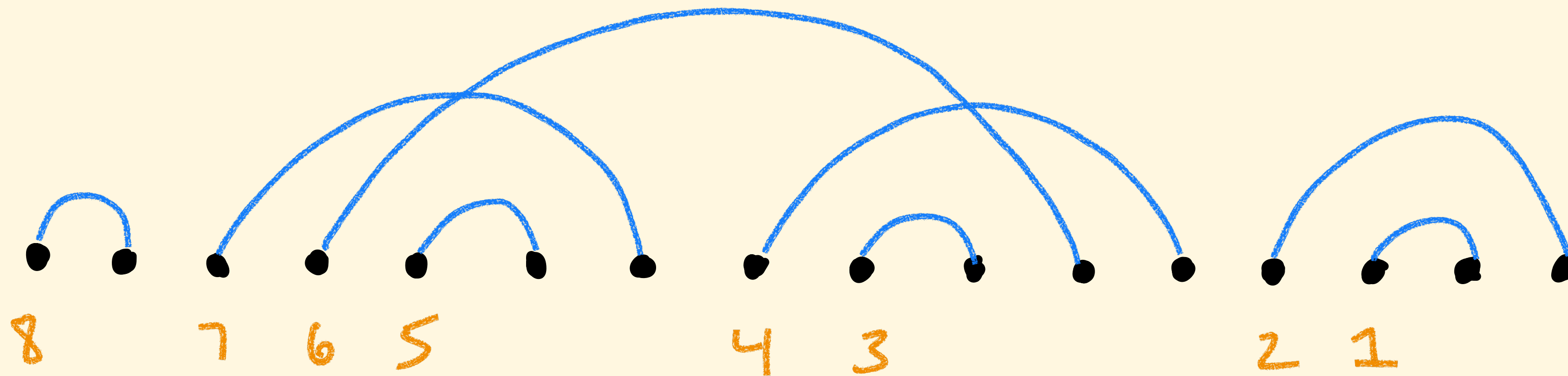
Dart games as matchings

Greedy matchings are in bijection with permutations.



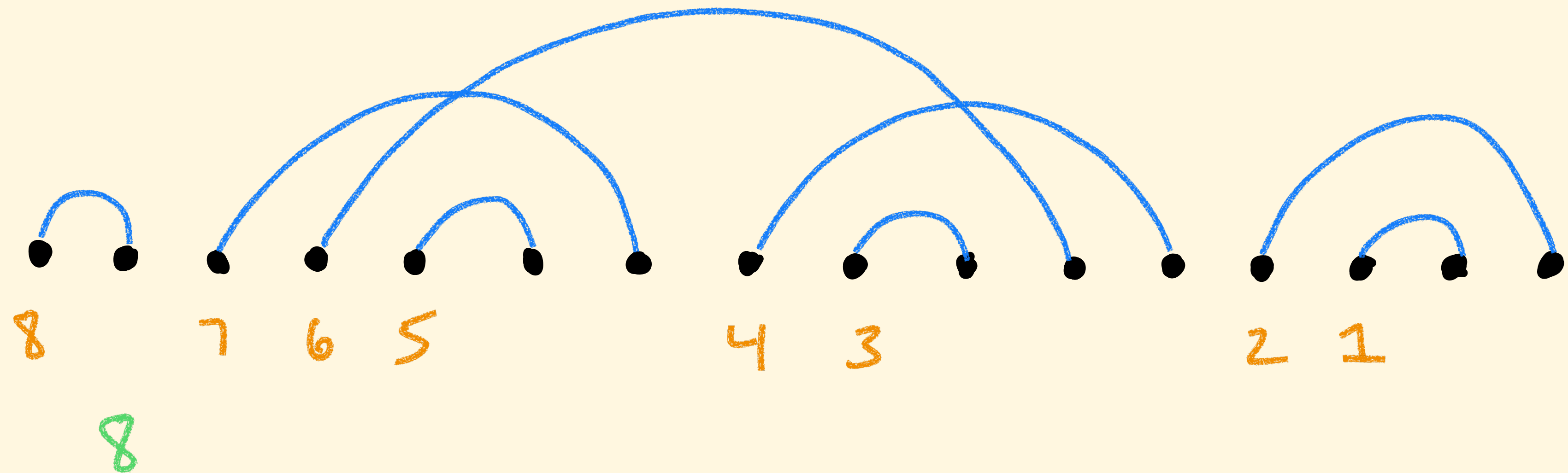
Dart games as matchings

Greedy matchings are in bijection with permutations.



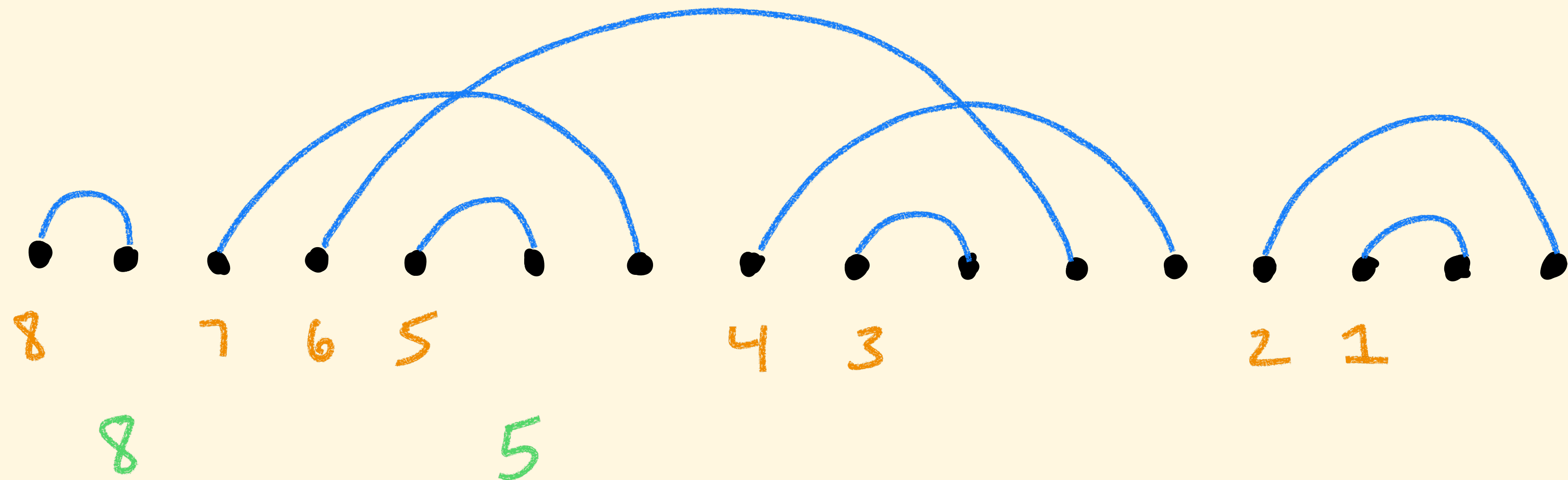
Dart games as matchings

Greedy matchings are in bijection with permutations.



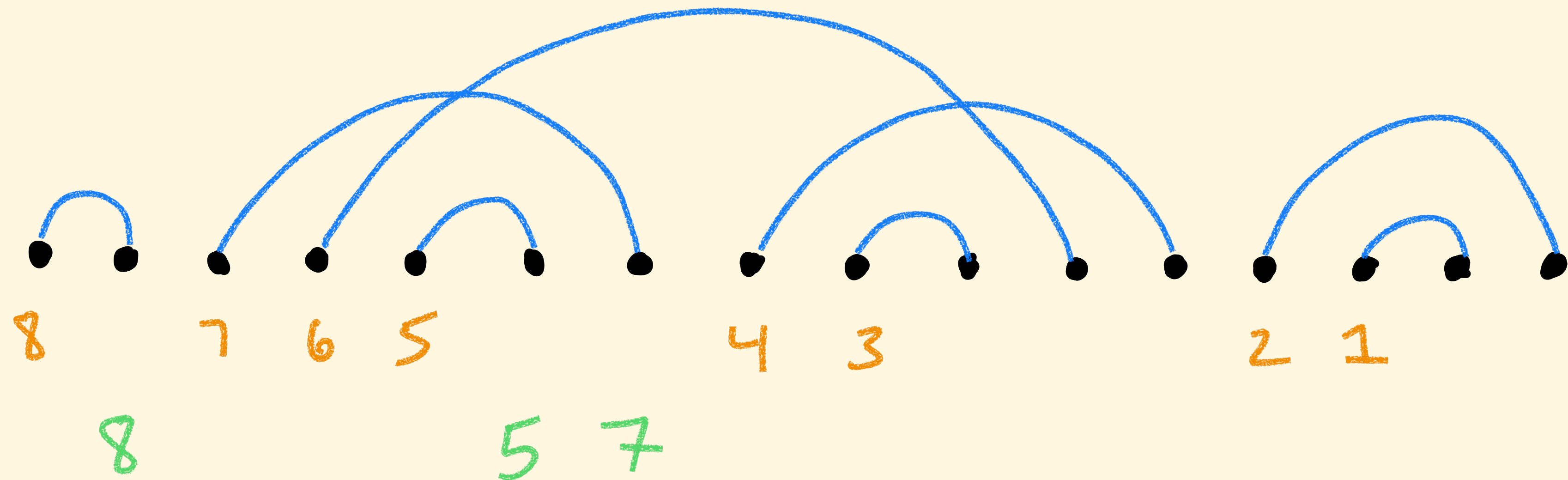
Dart games as matchings

Greedy matchings are in bijection with permutations.



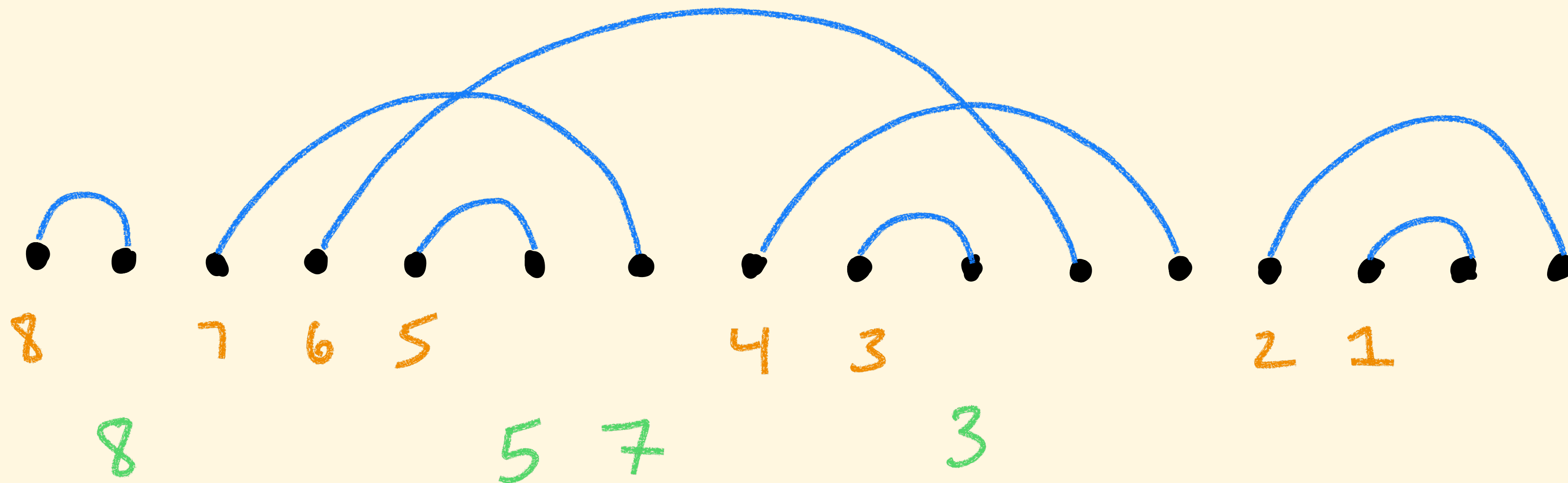
Dart games as matchings

Greedy matchings are in bijection with permutations.



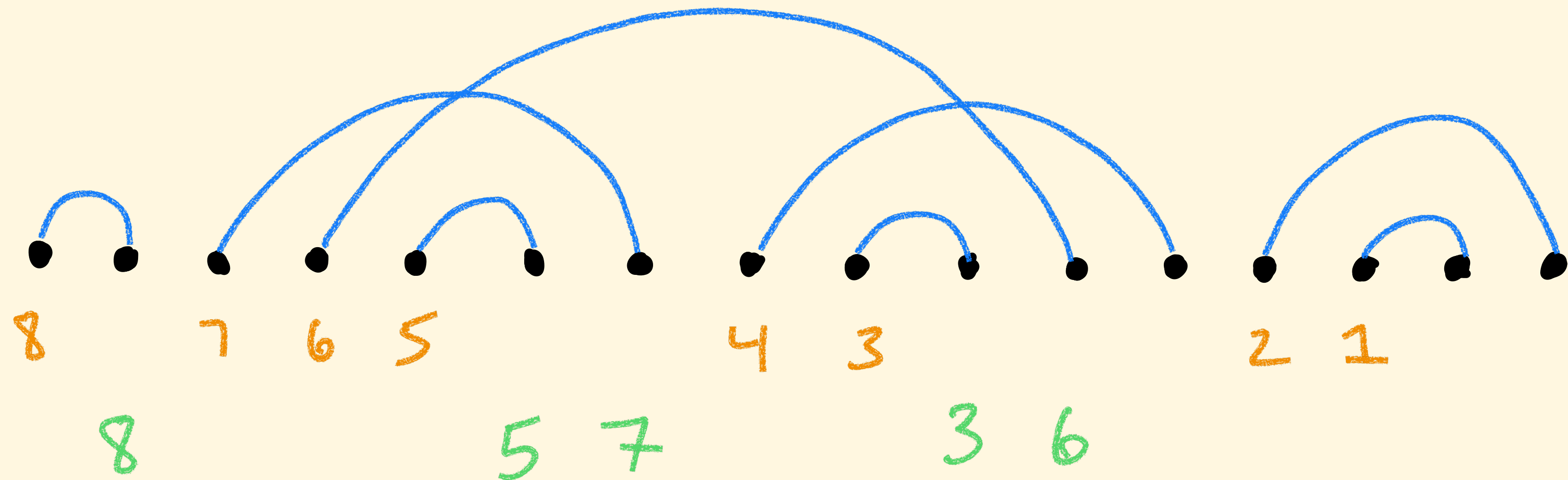
Dart games as matchings

Greedy matchings are in bijection with permutations.



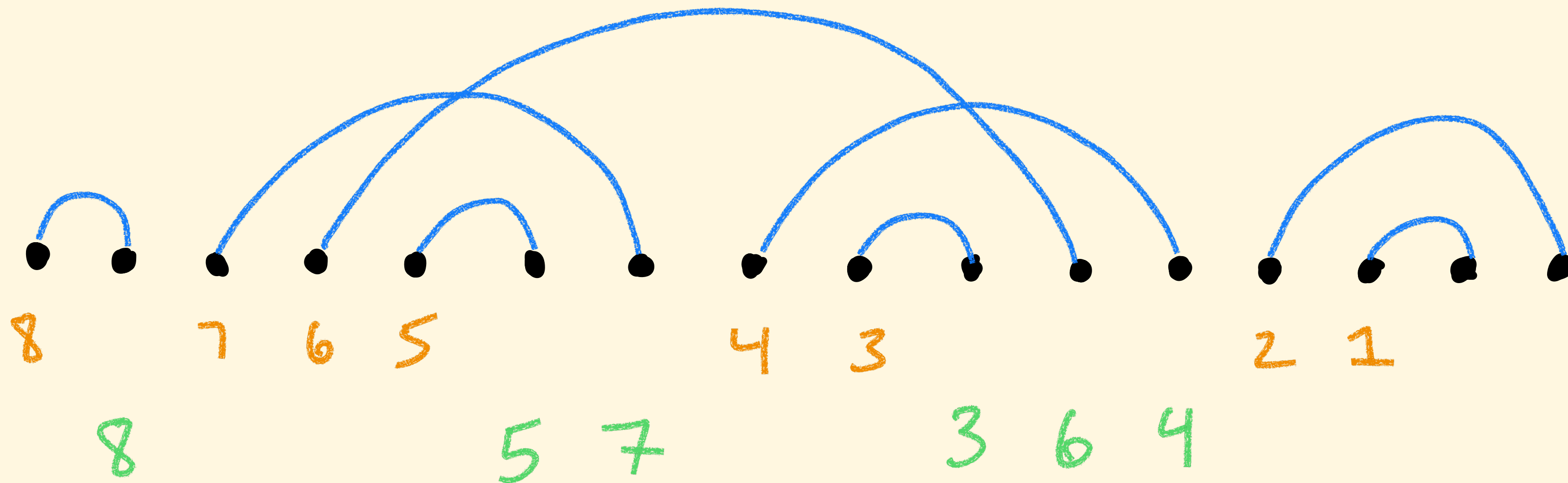
Dart games as matchings

Greedy matchings are in bijection with permutations.



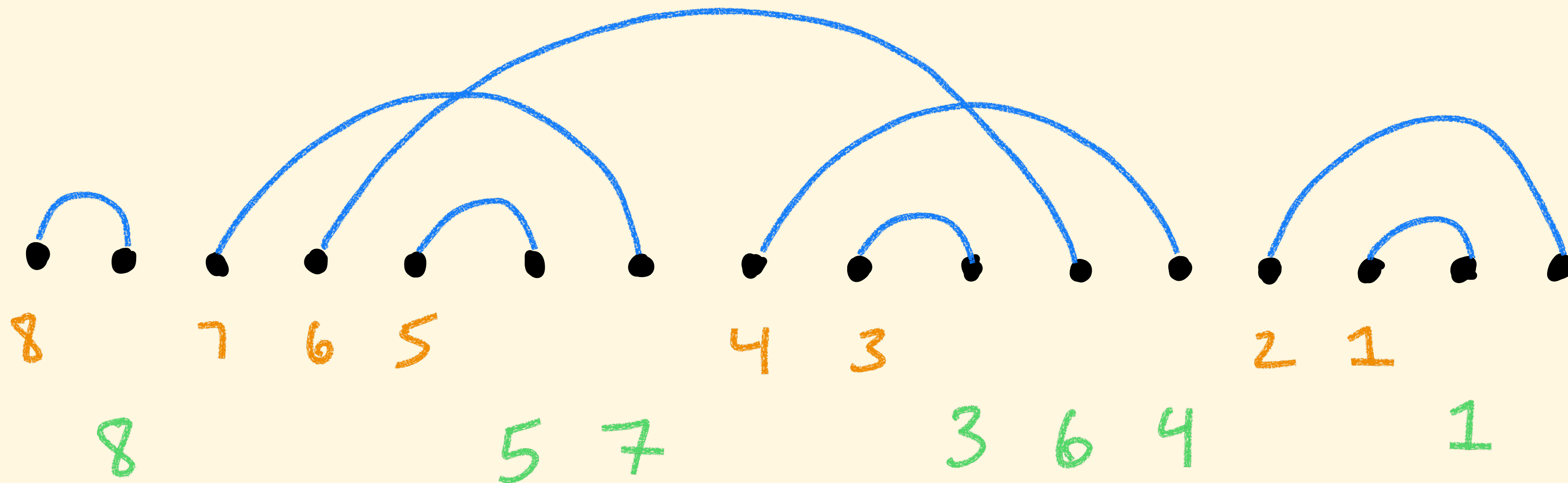
Dart games as matchings

Greedy matchings are in bijection with permutations.



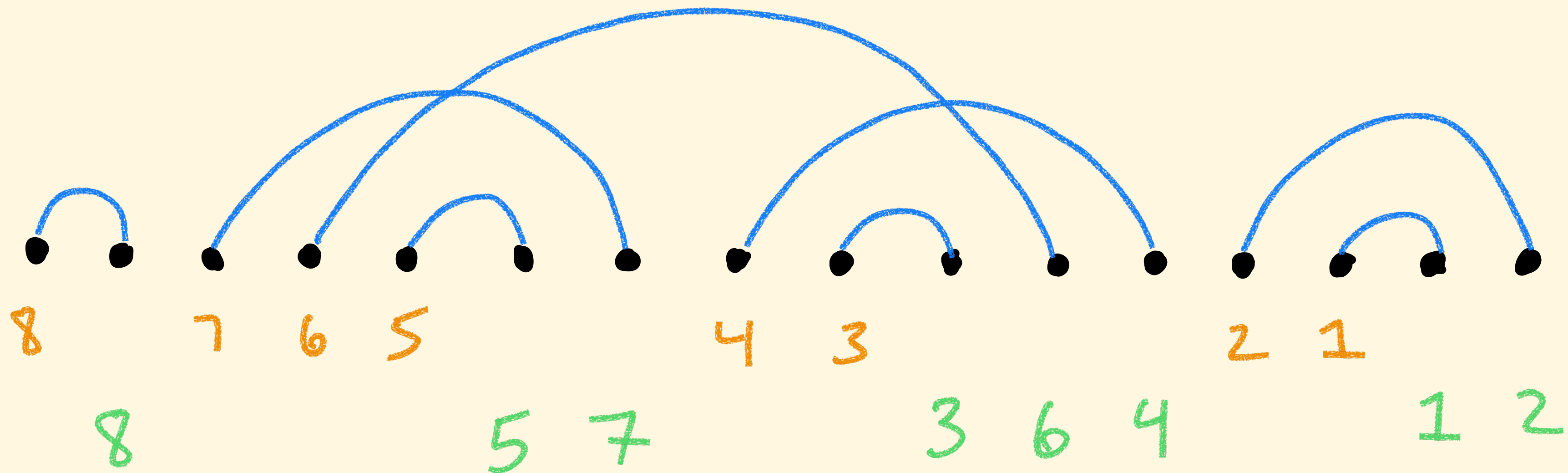
Dart games as matchings

Greedy matchings are in bijection with permutations.



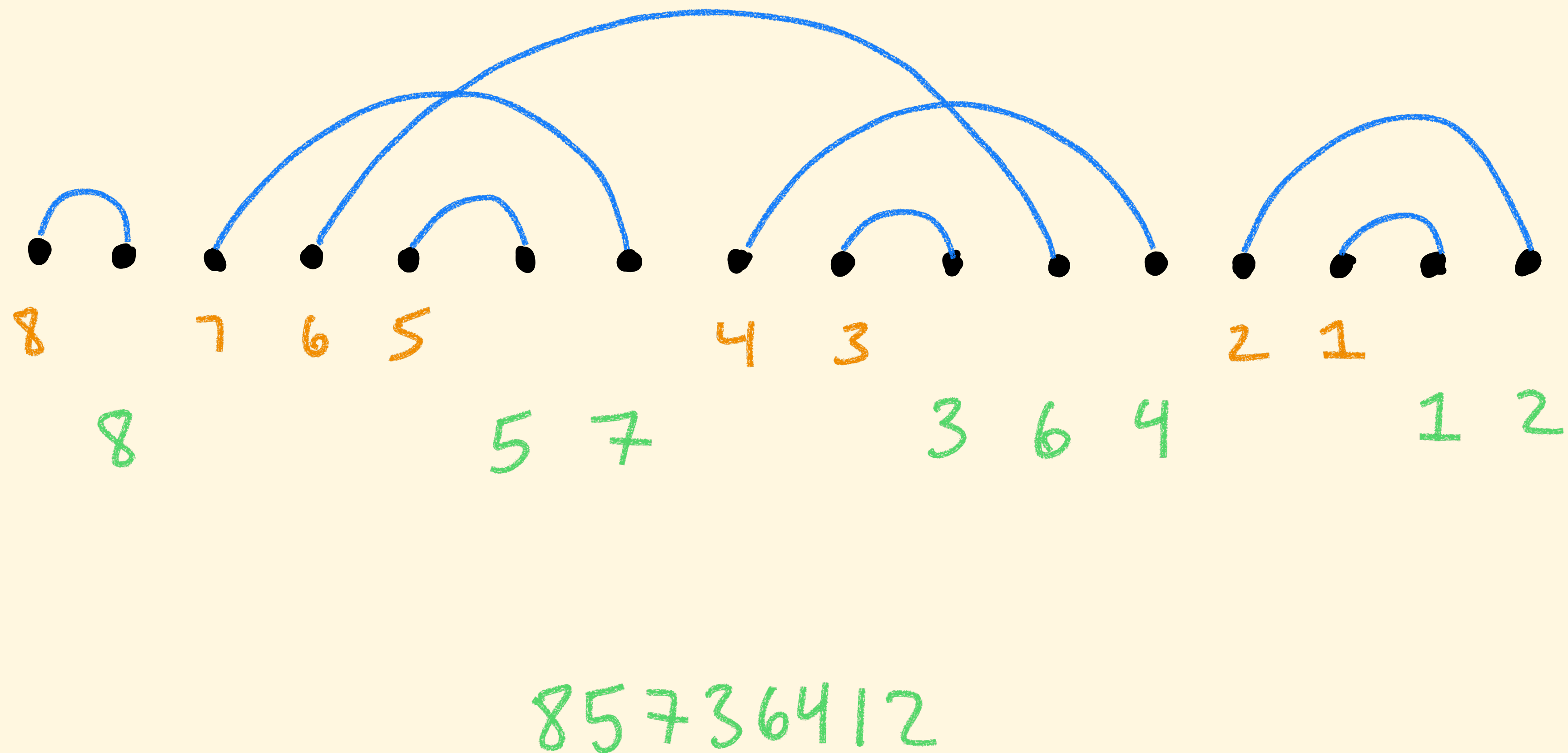
Dart games as matchings

Greedy matchings are in bijection with permutations.



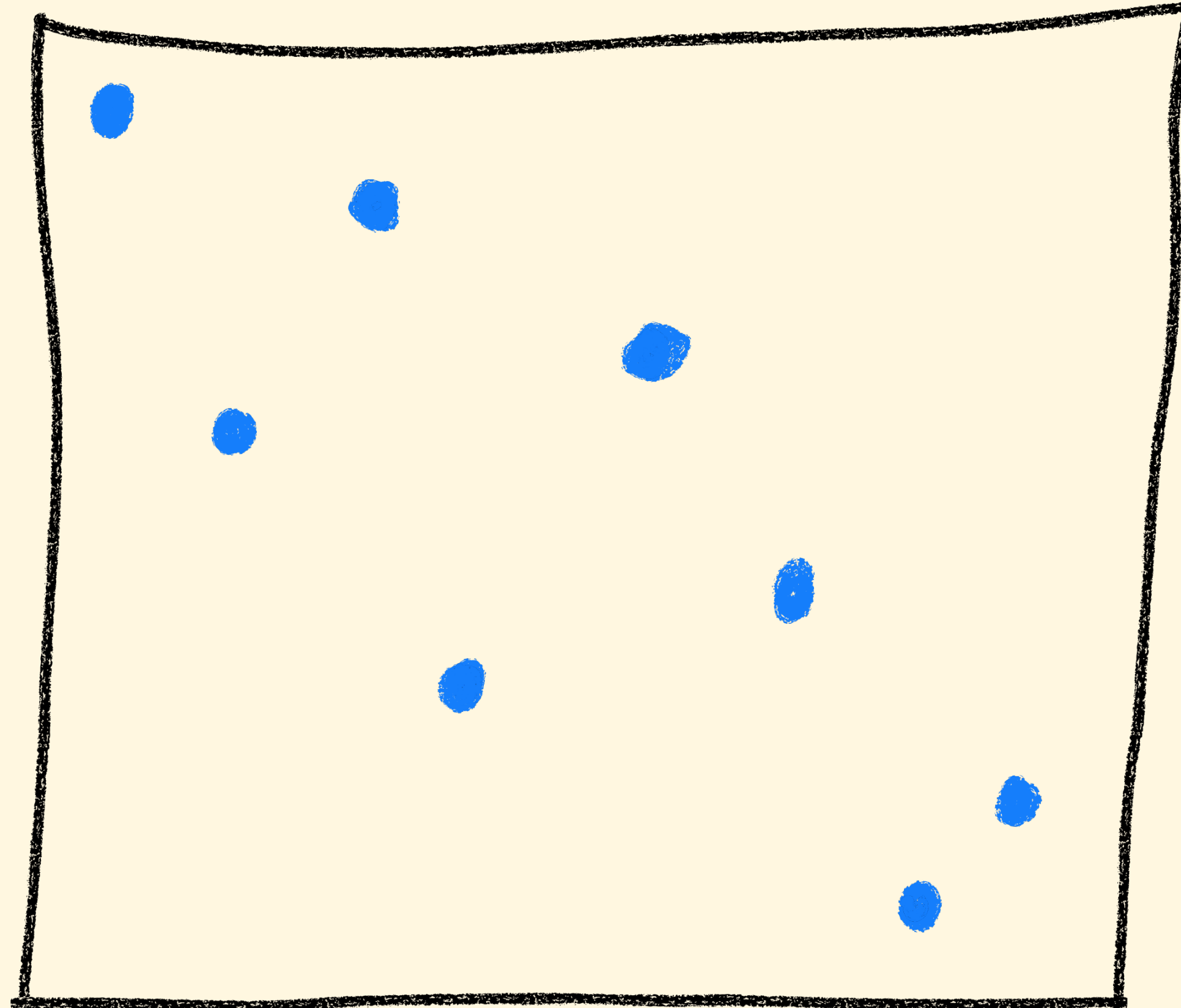
Dart games as matchings

Greedy matchings are in bijection with permutations.



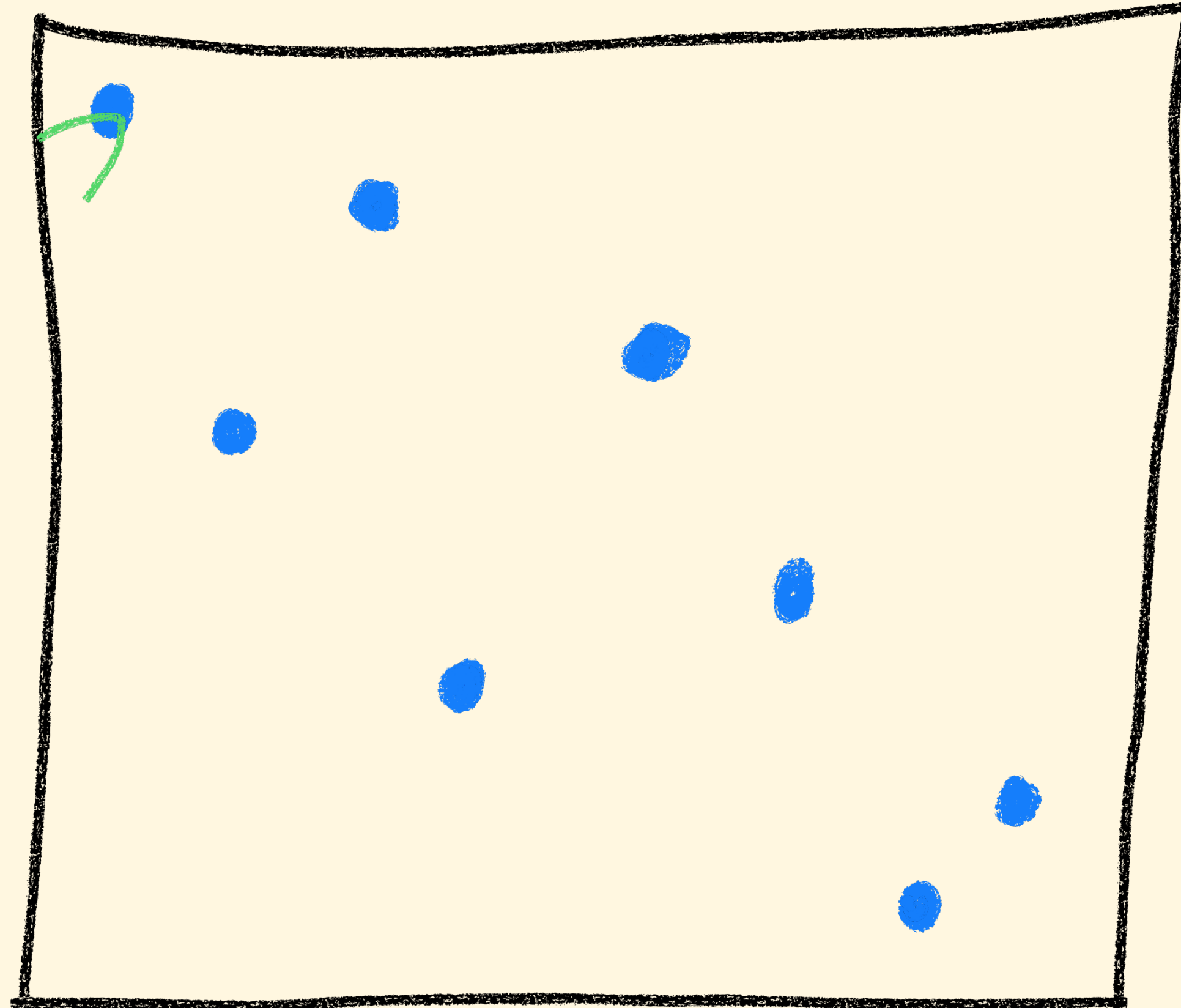
Dart games as matchings

Greedy matchings are in bijection with permutations.



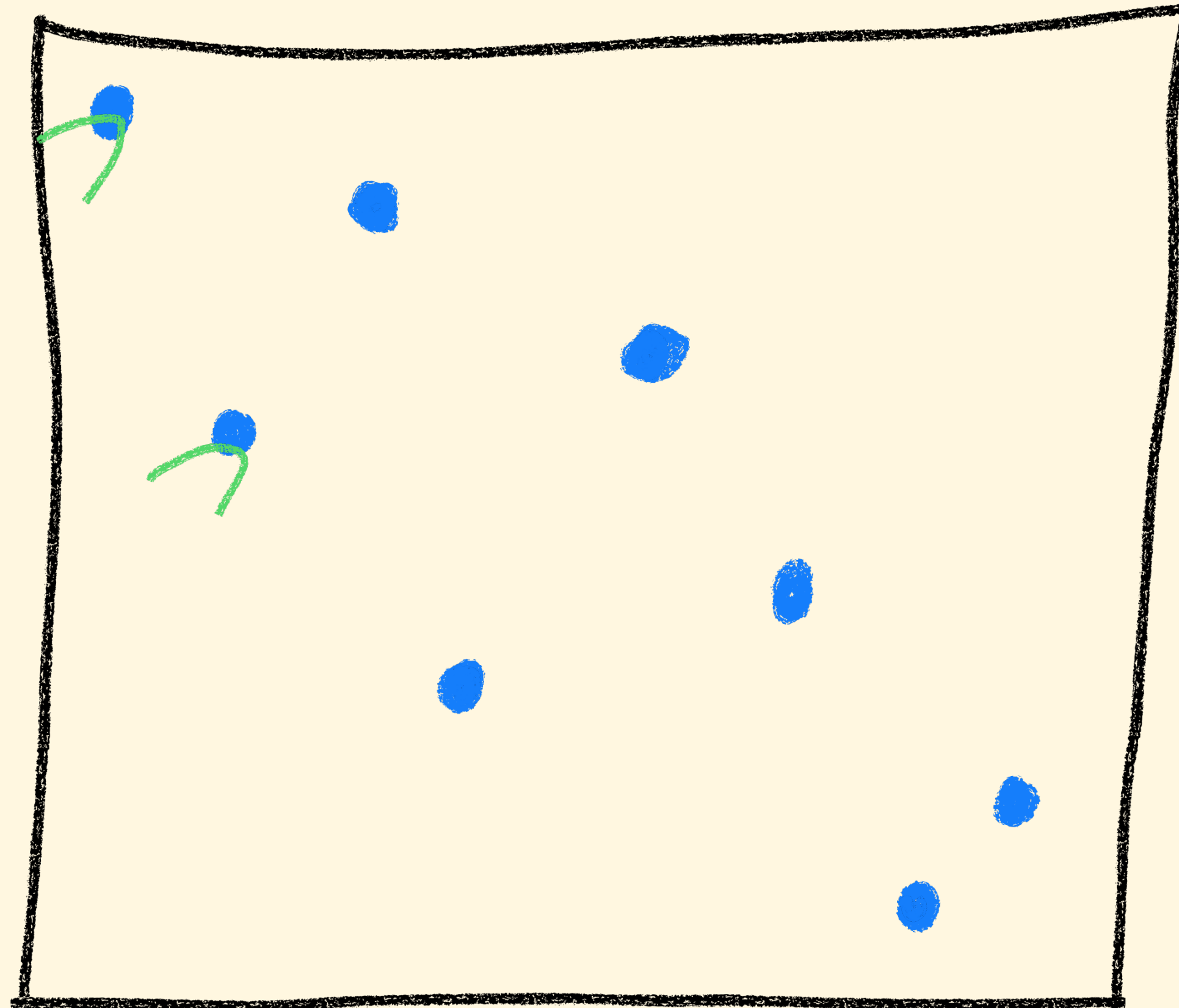
Dart games as matchings

Greedy matchings are in bijection with permutations.



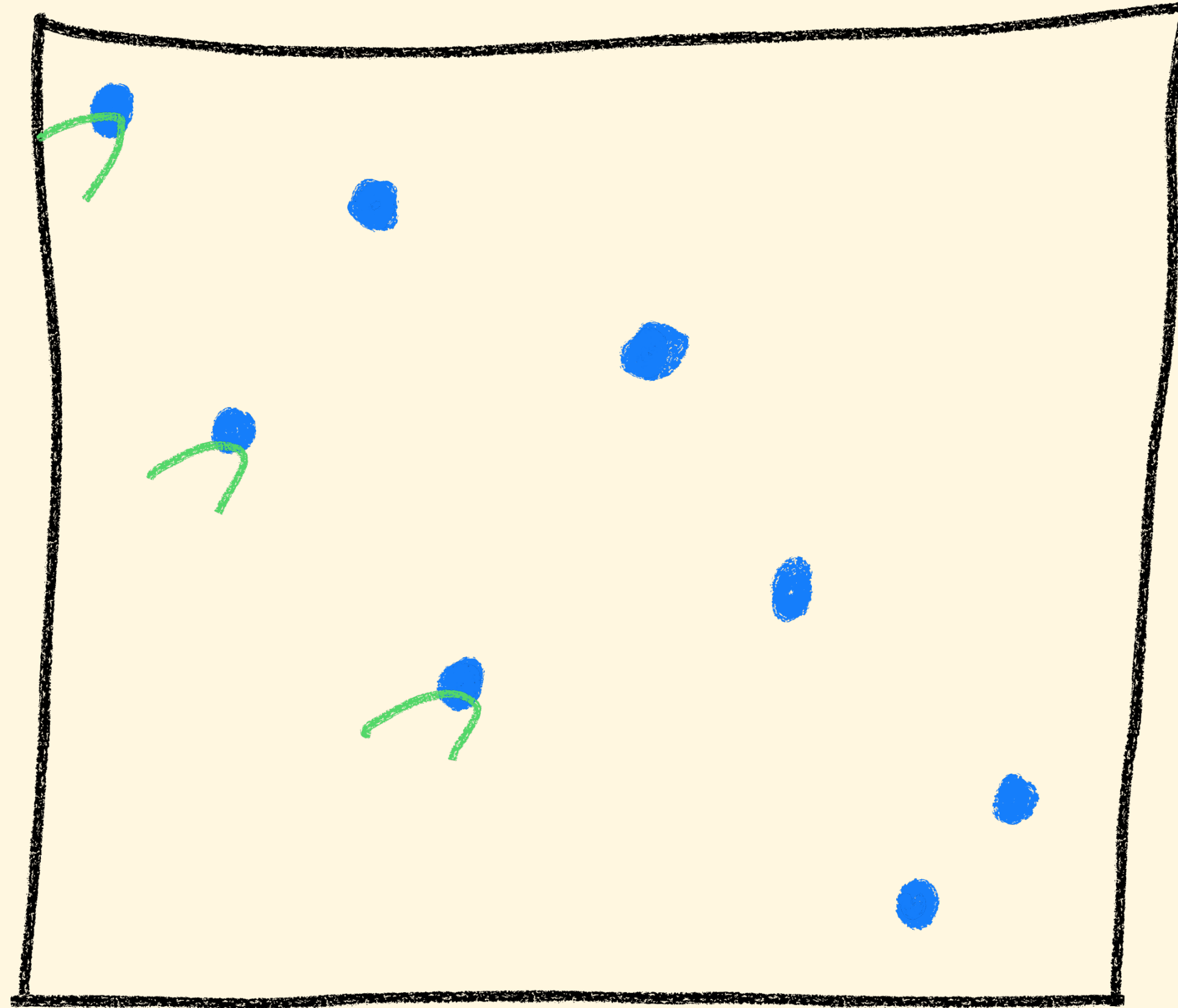
Dart games as matchings

Greedy matchings are in bijection with permutations.



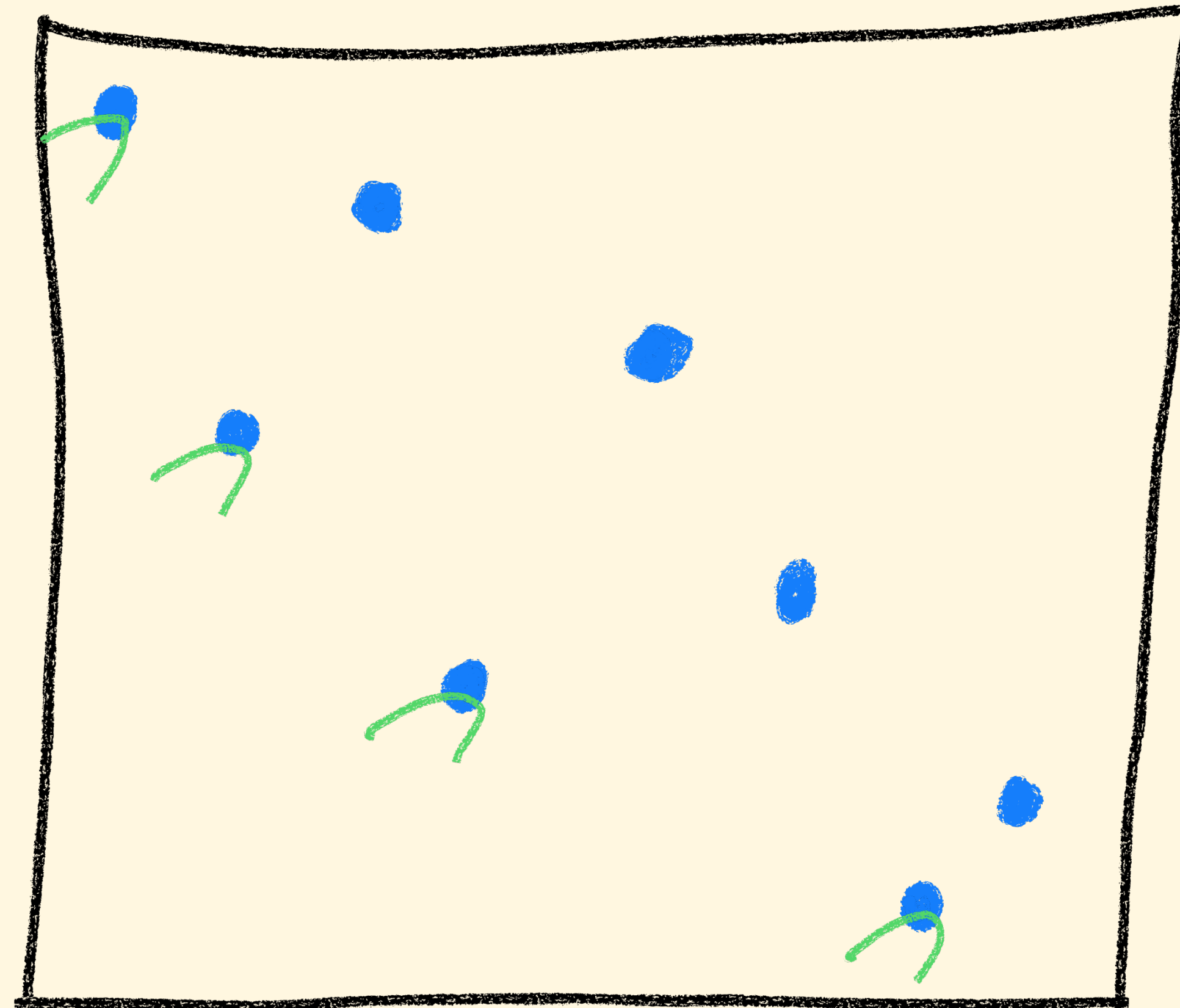
Dart games as matchings

Greedy matchings are in bijection with permutations.



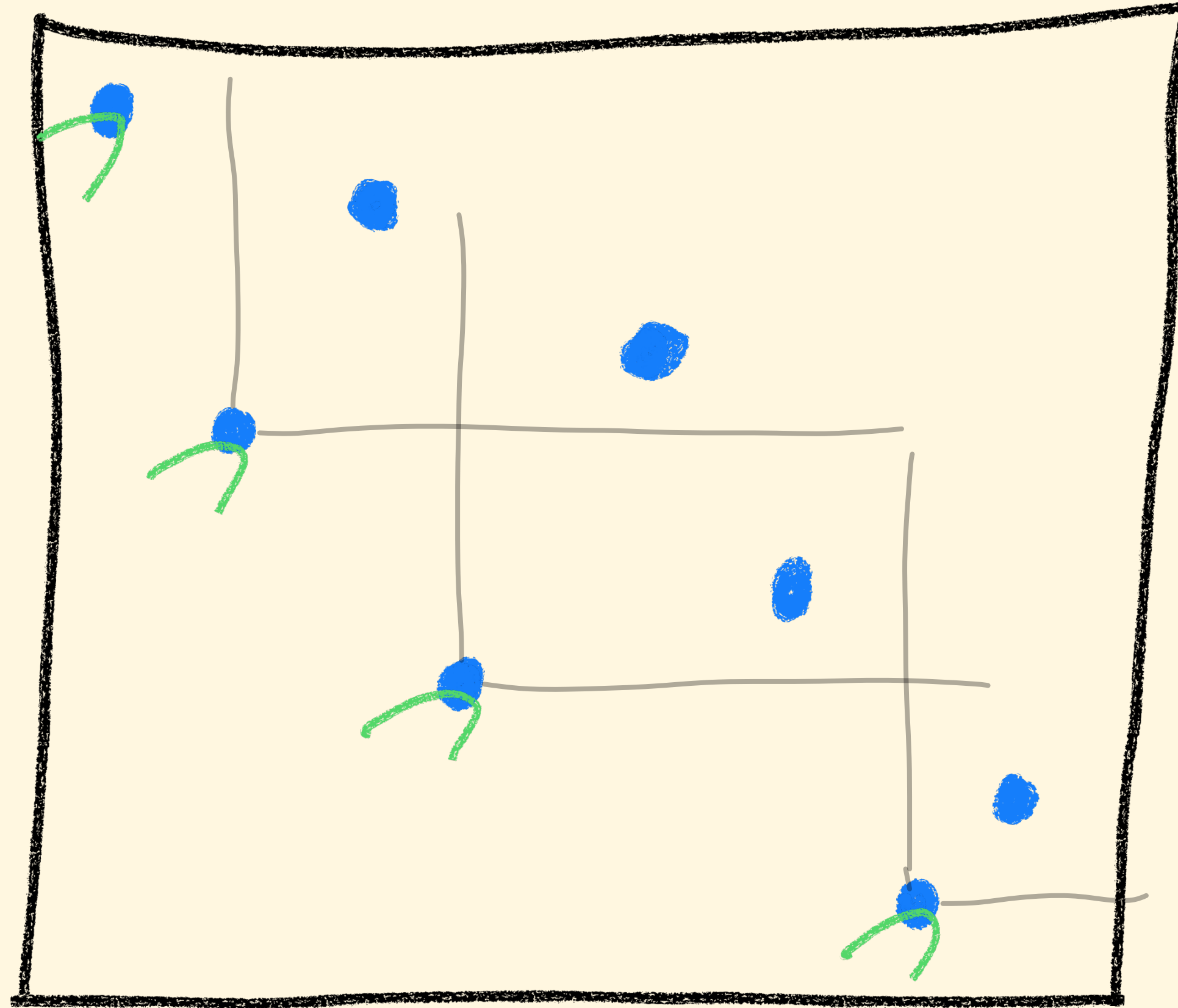
Dart games as matchings

Greedy matchings are in bijection with permutations.



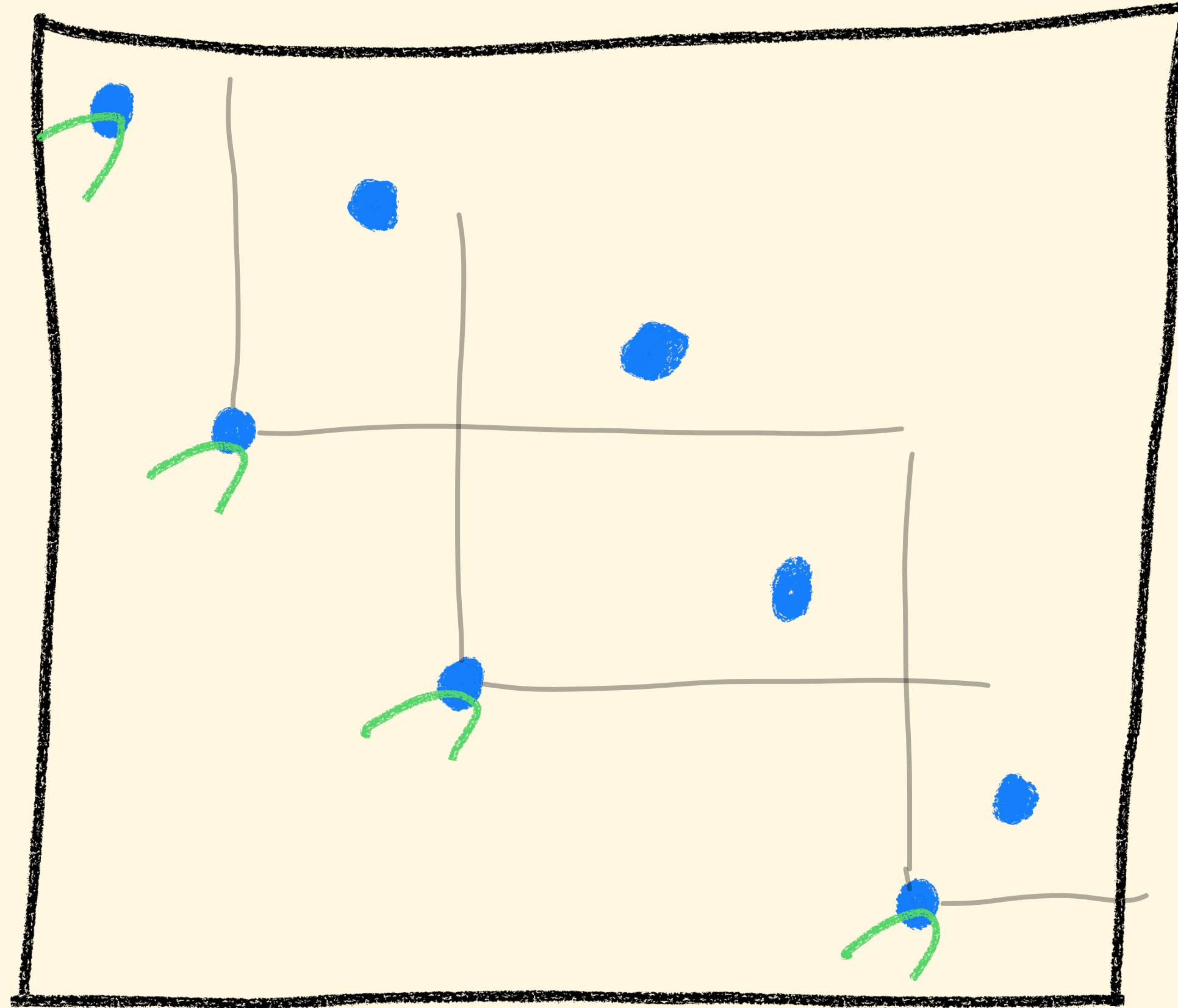
Dart games as matchings

Greedy matchings are in bijection with permutations.



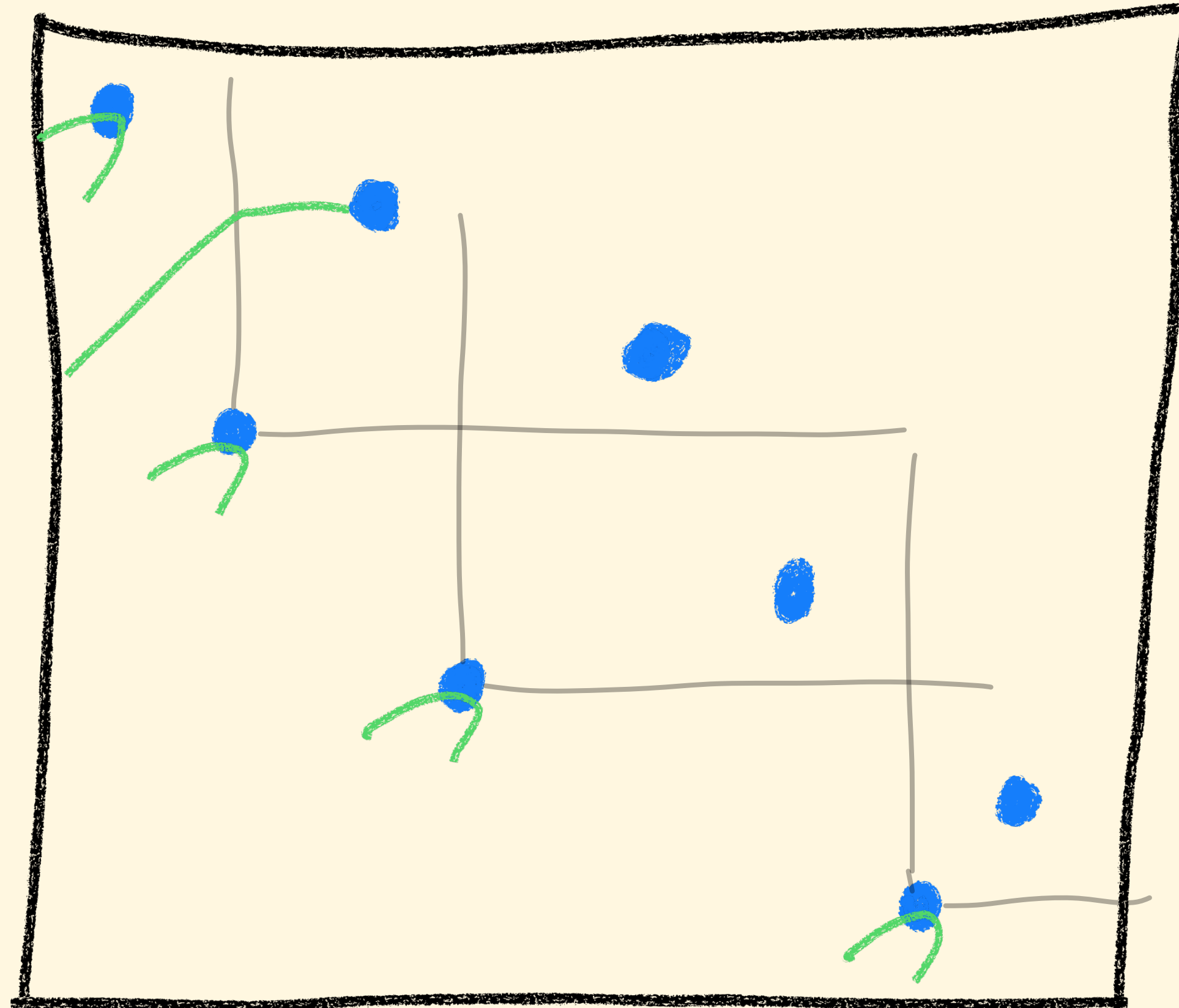
Dart games as matchings

Greedy matchings are in bijection with permutations.



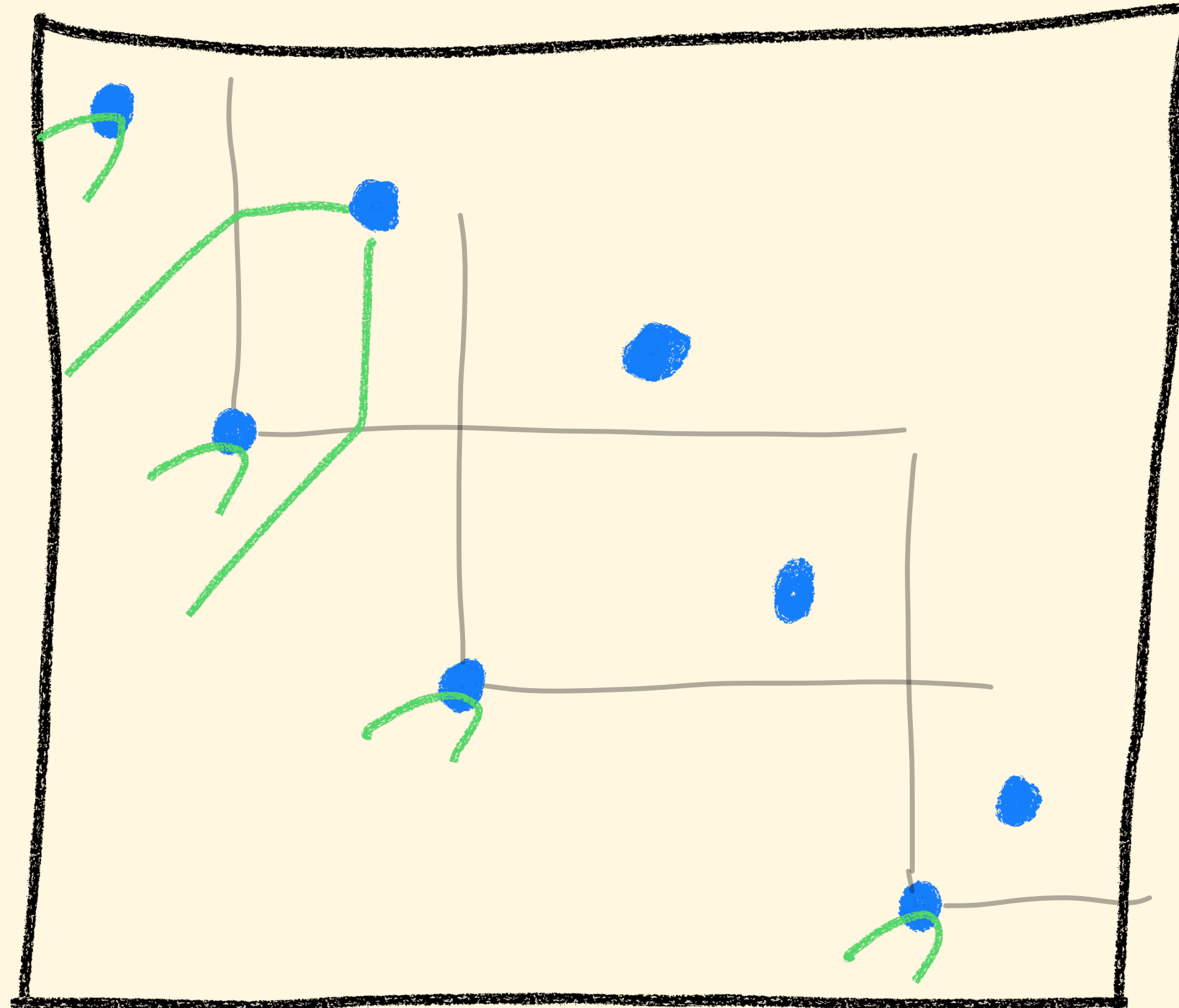
Dart games as matchings

Greedy matchings are in bijection with permutations.



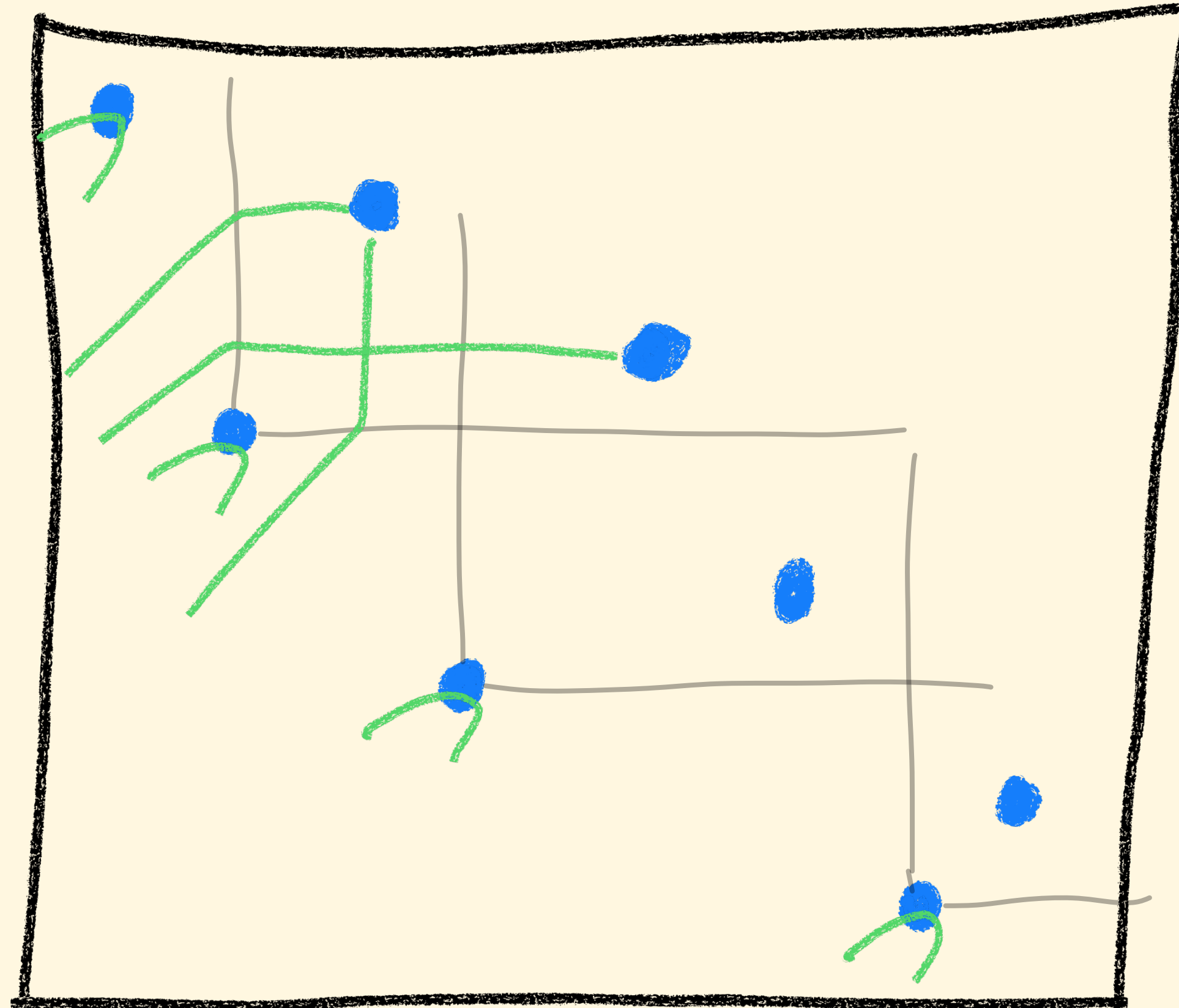
Dart games as matchings

Greedy matchings are in bijection with permutations.



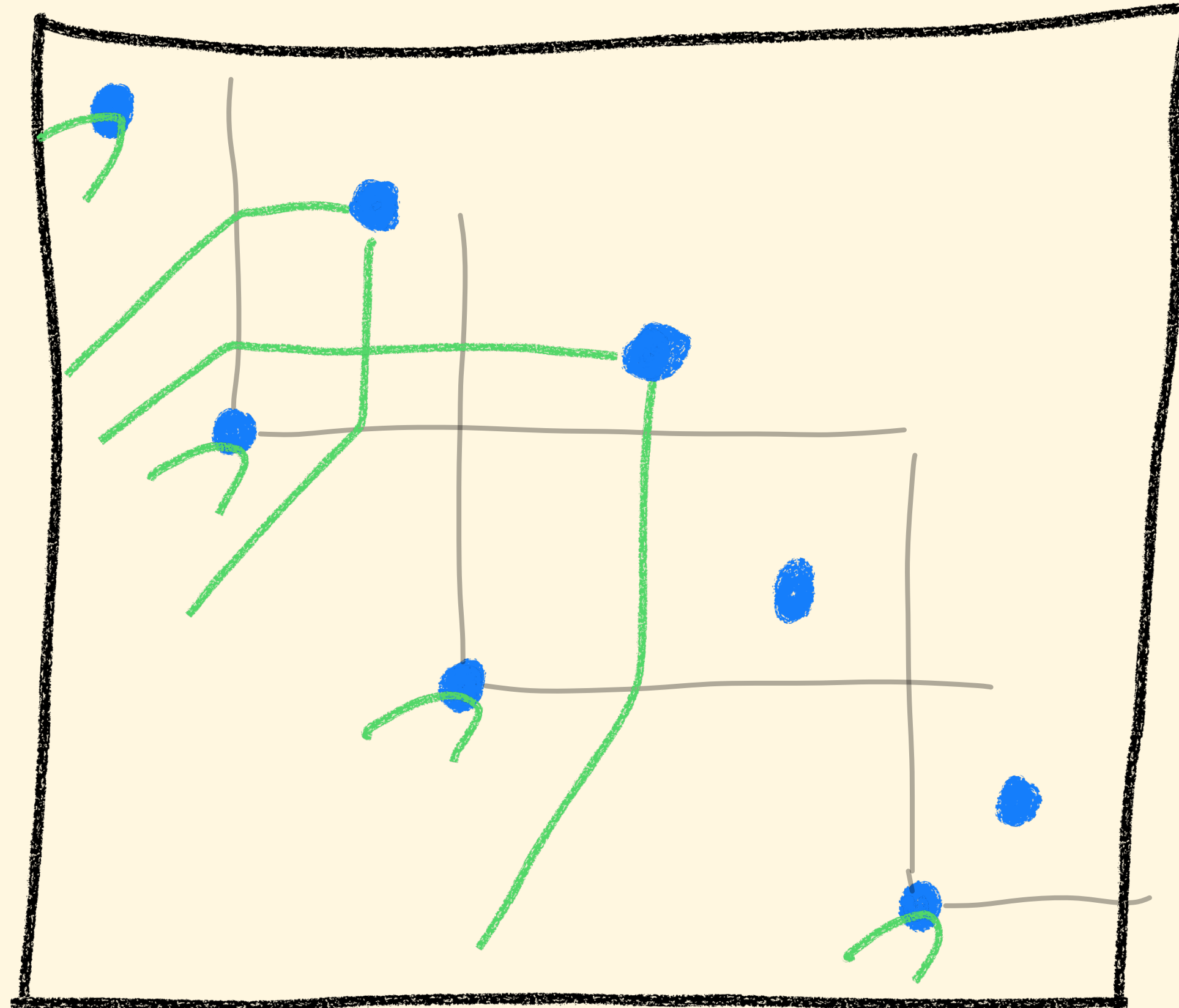
Dart games as matchings

Greedy matchings are in bijection with permutations.



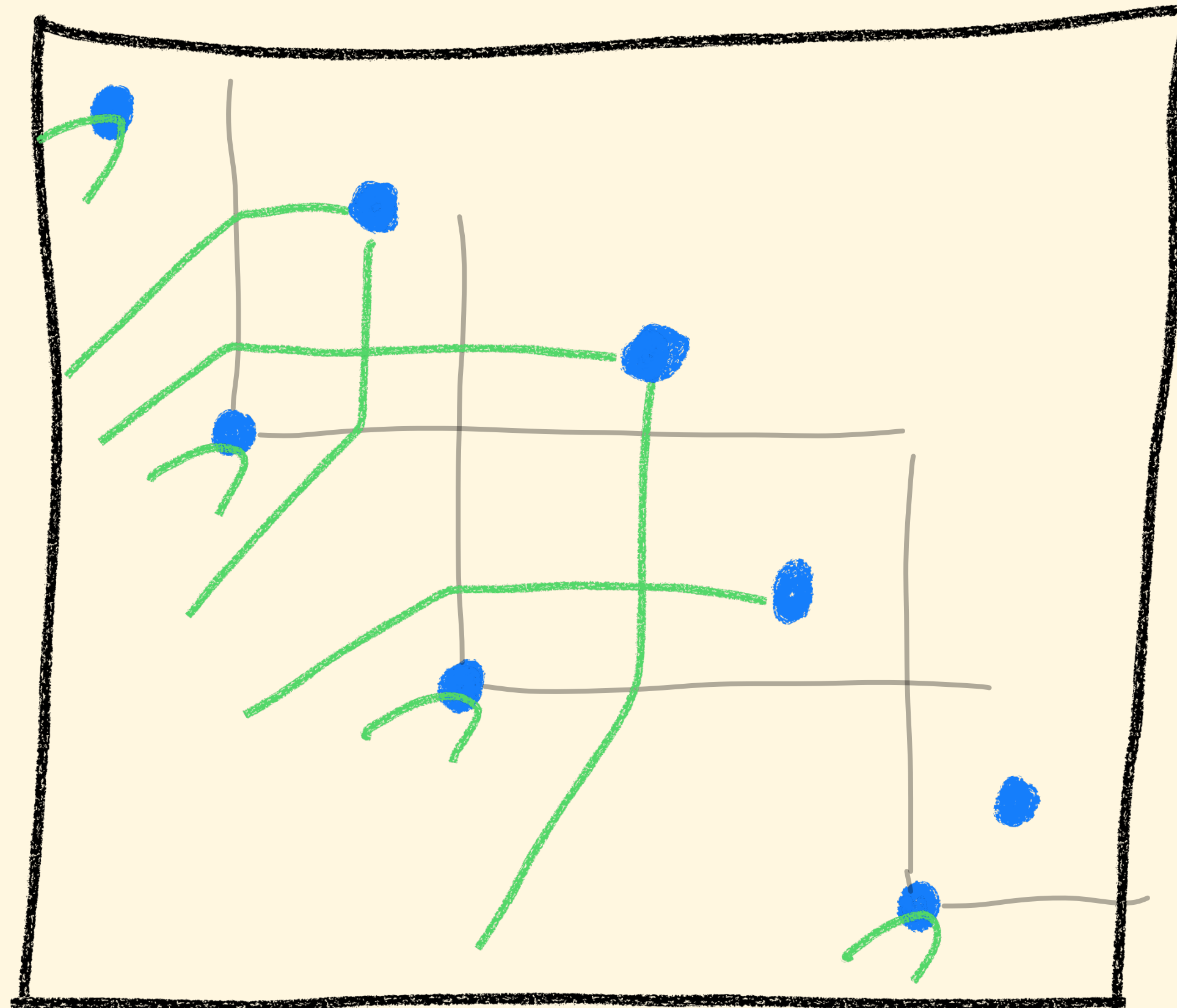
Dart games as matchings

Greedy matchings are in bijection with permutations.



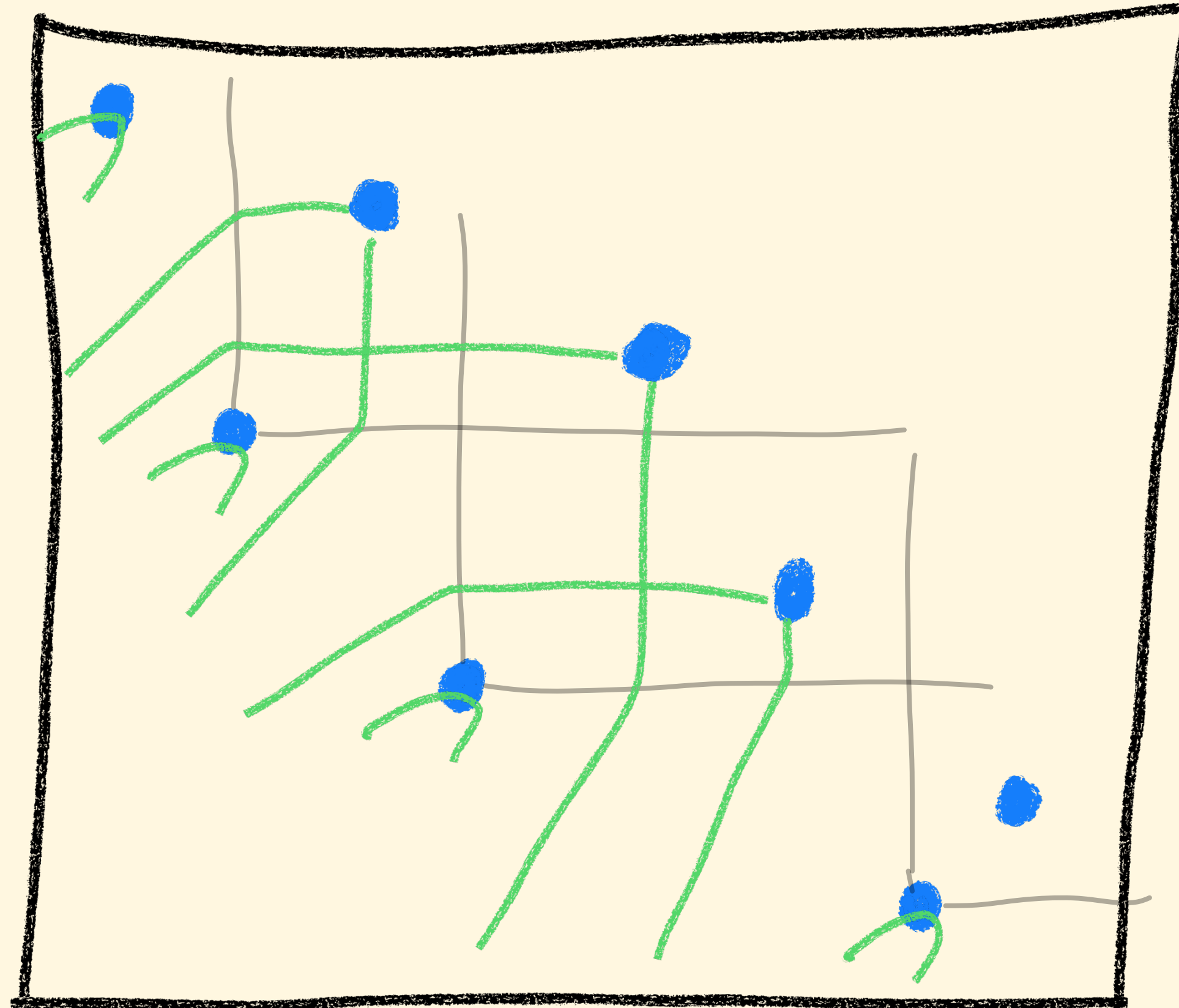
Dart games as matchings

Greedy matchings are in bijection with permutations.



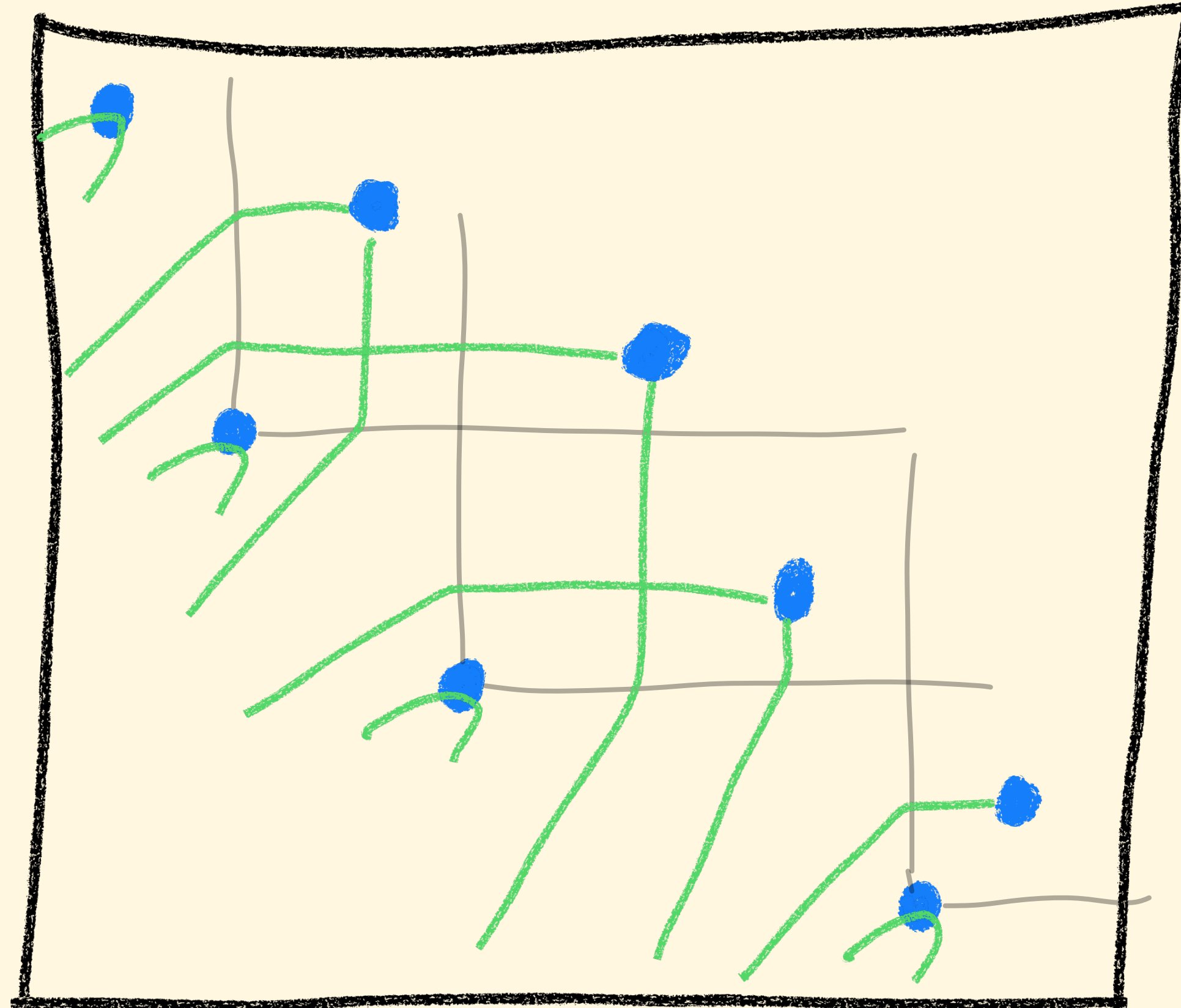
Dart games as matchings

Greedy matchings are in bijection with permutations.



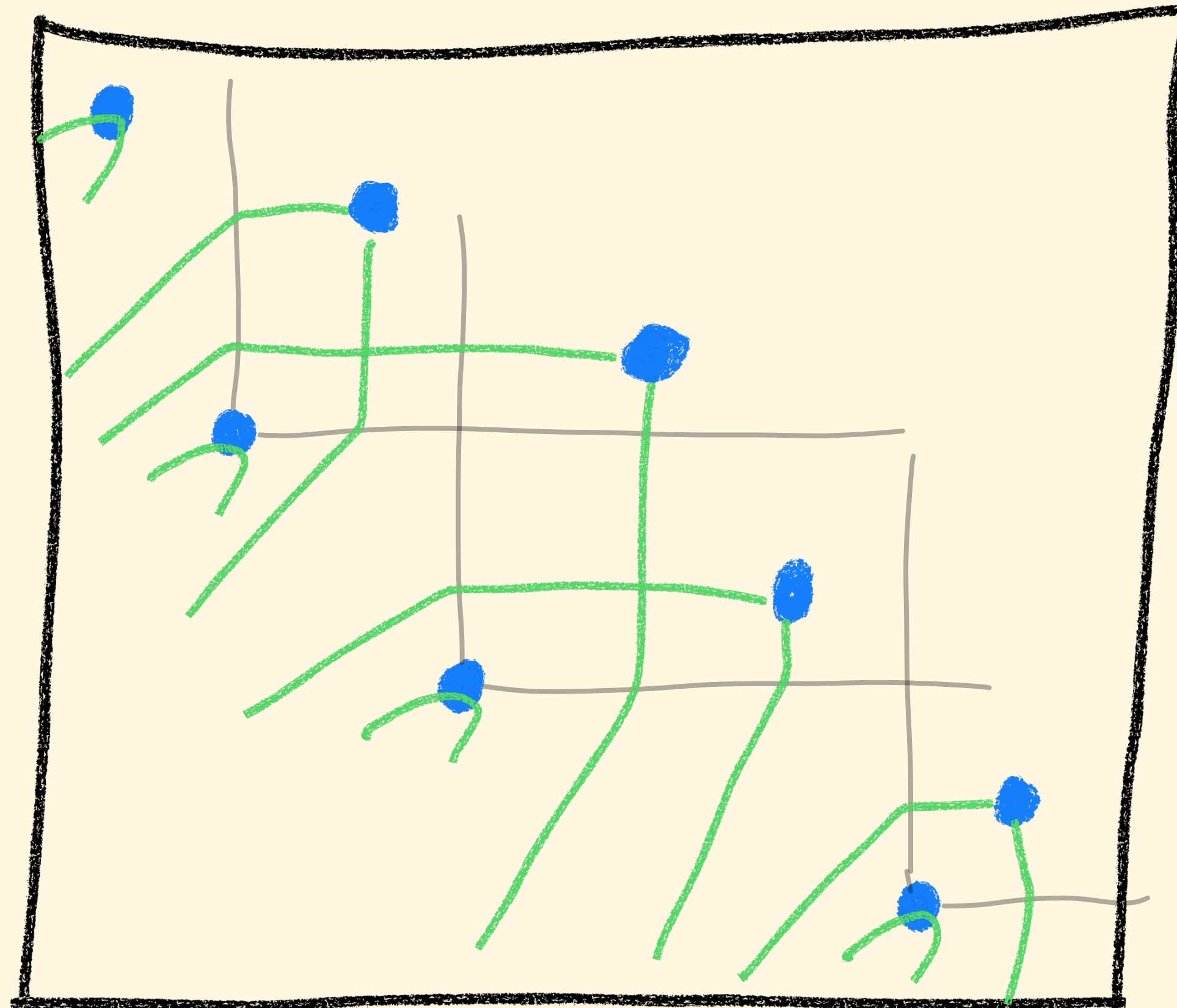
Dart games as matchings

Greedy matchings are in bijection with permutations.



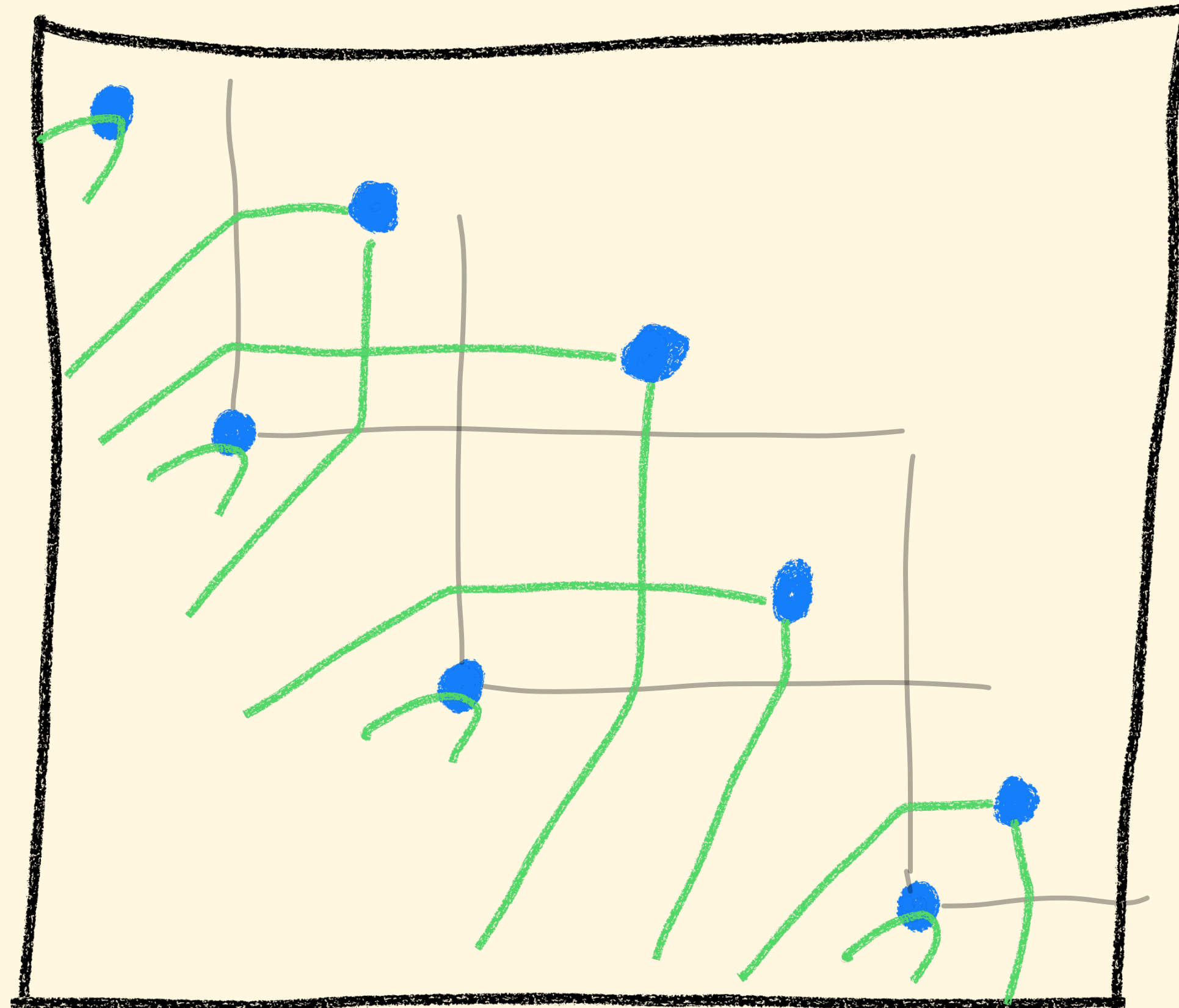
Dart games as matchings

Greedy matchings are in bijection with permutations.



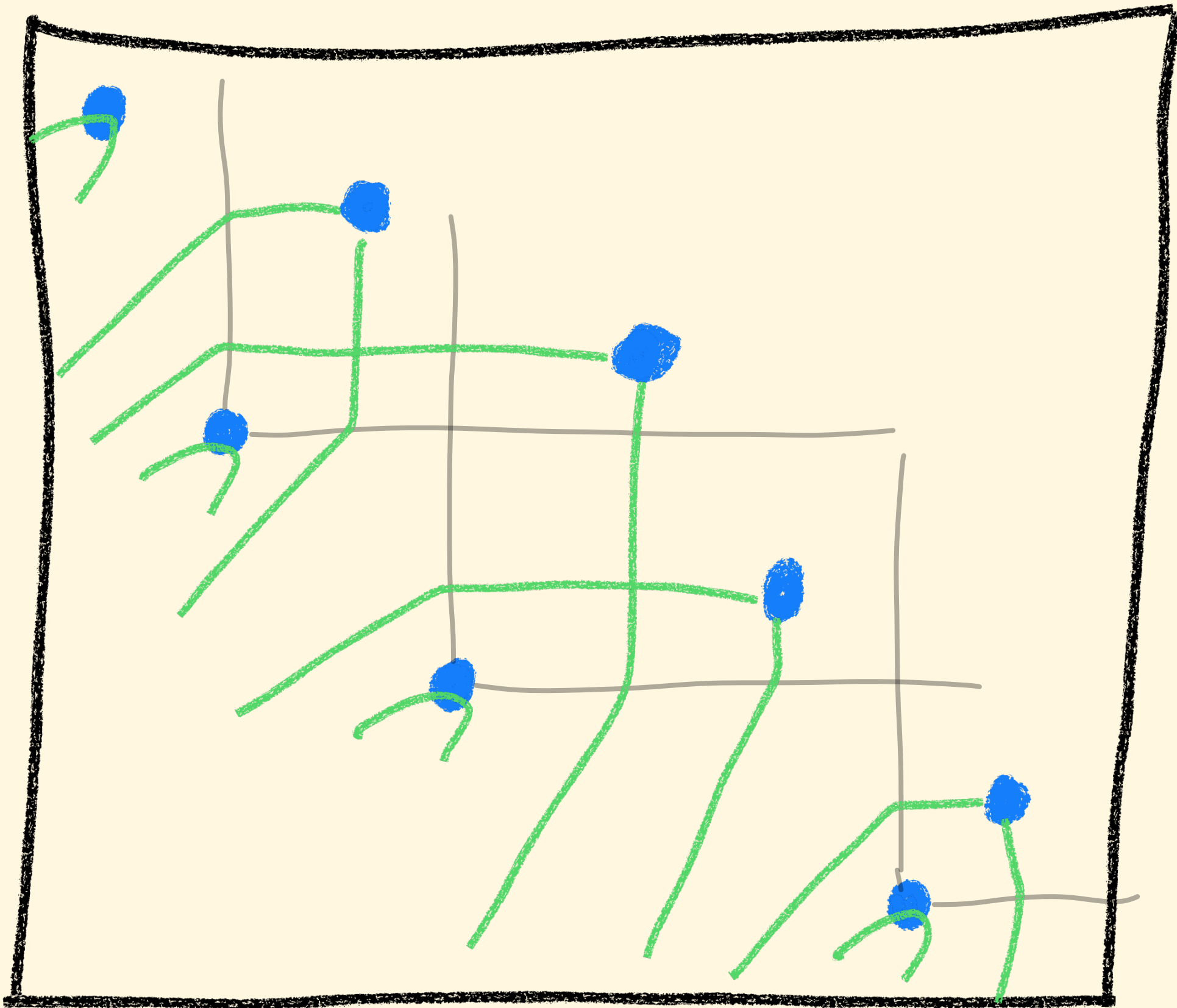
Dart games as matchings

Greedy matchings are in bijection with permutations.



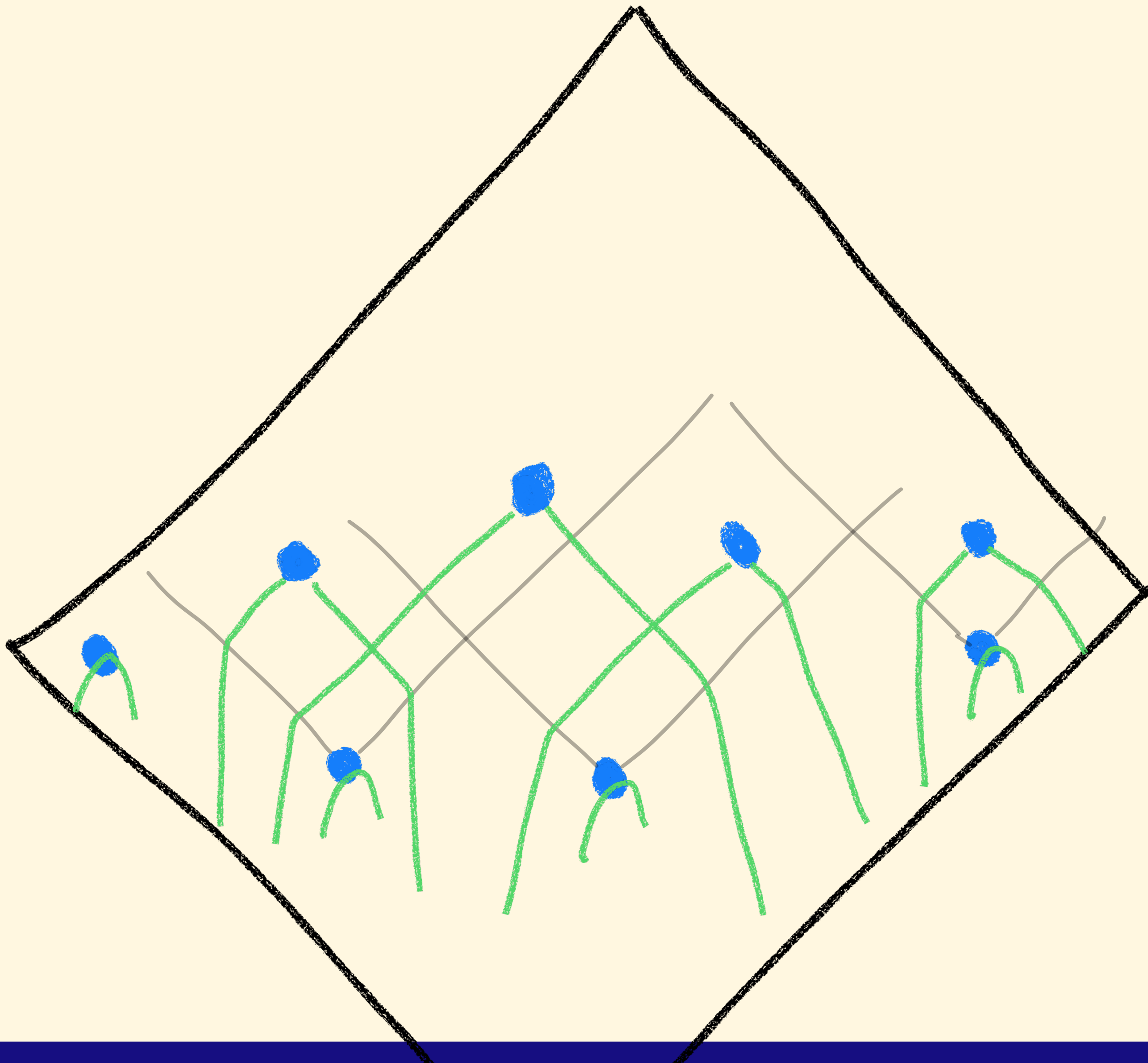
Dart games as matchings

Greedy matchings are in bijection with permutations.



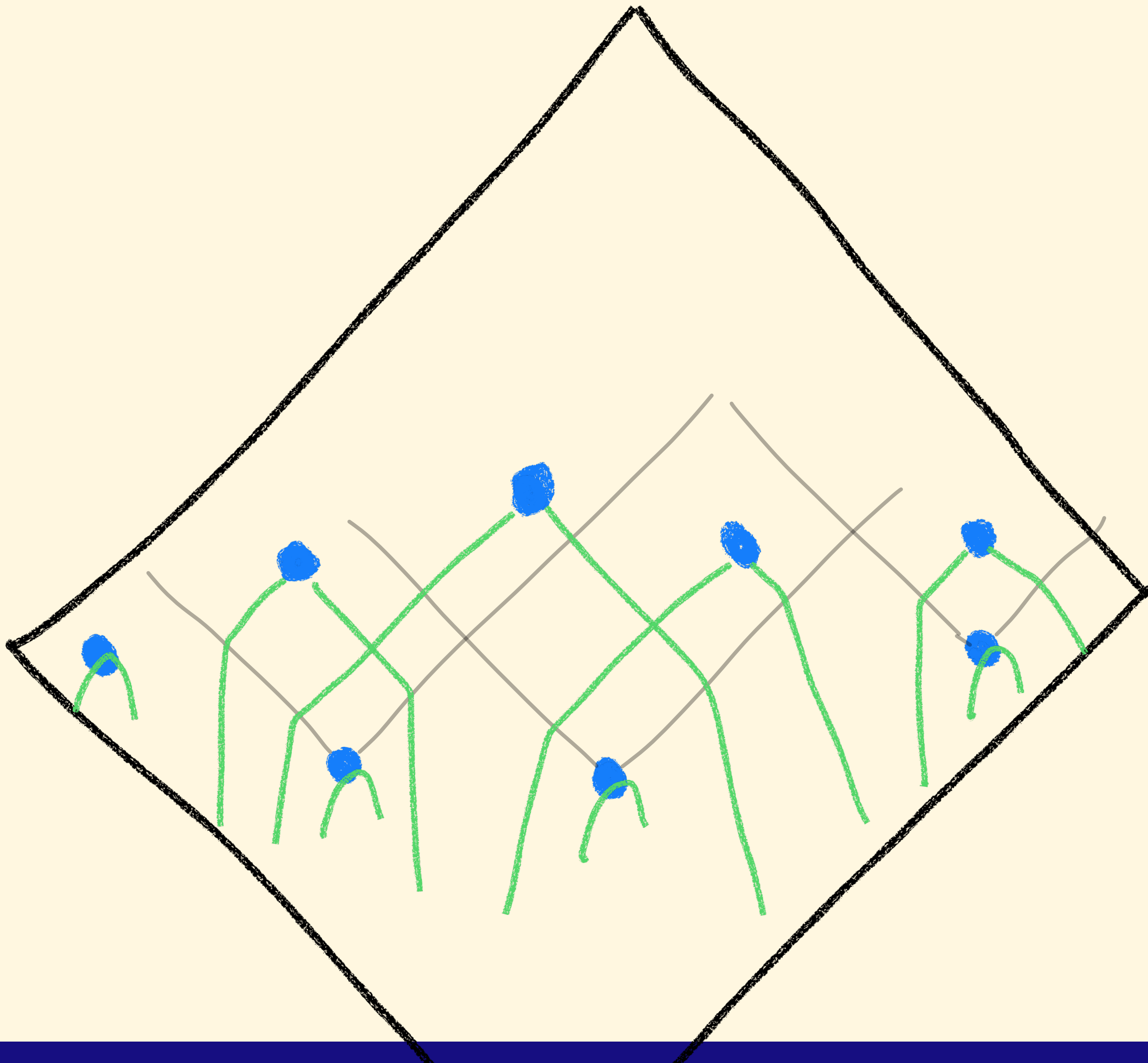
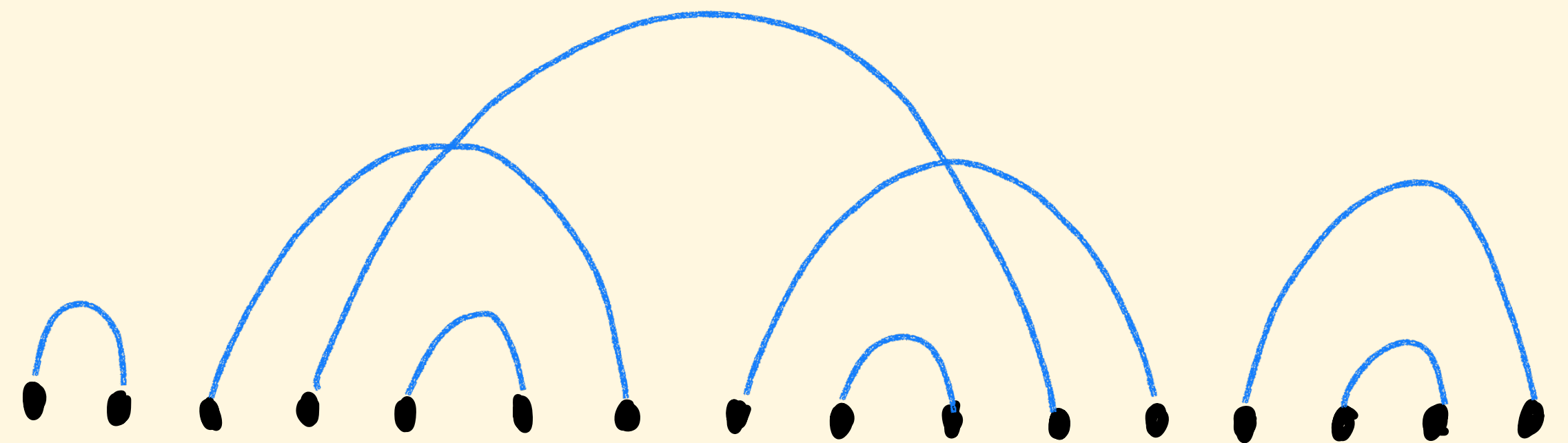
Dart games as matchings

Greedy matchings are in bijection with permutations.



Dart games as matchings

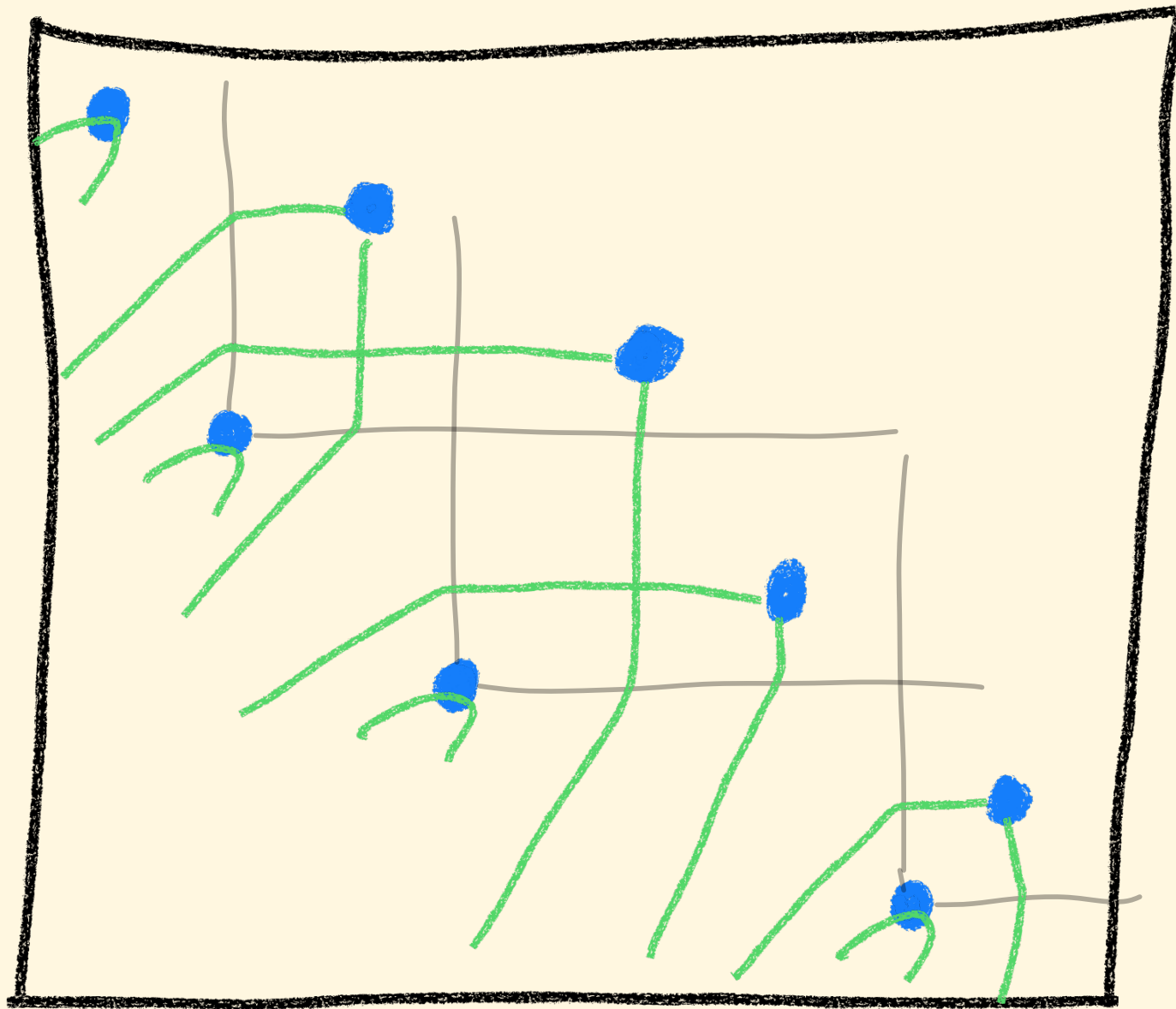
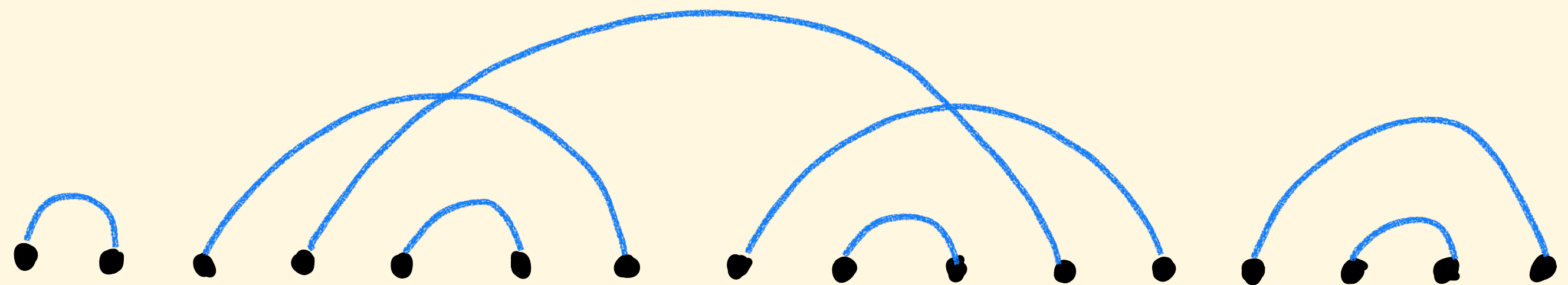
Greedy matchings are in bijection with permutations.



Dart games as matchings

Define a *short pair* to be a pair $\{i, i + 1\}$ in a matching and a *long pair* to be any other pair.

Under this bijection, the hits correspond to short pairs and the misses correspond to long pairs.



Dart games as matchings

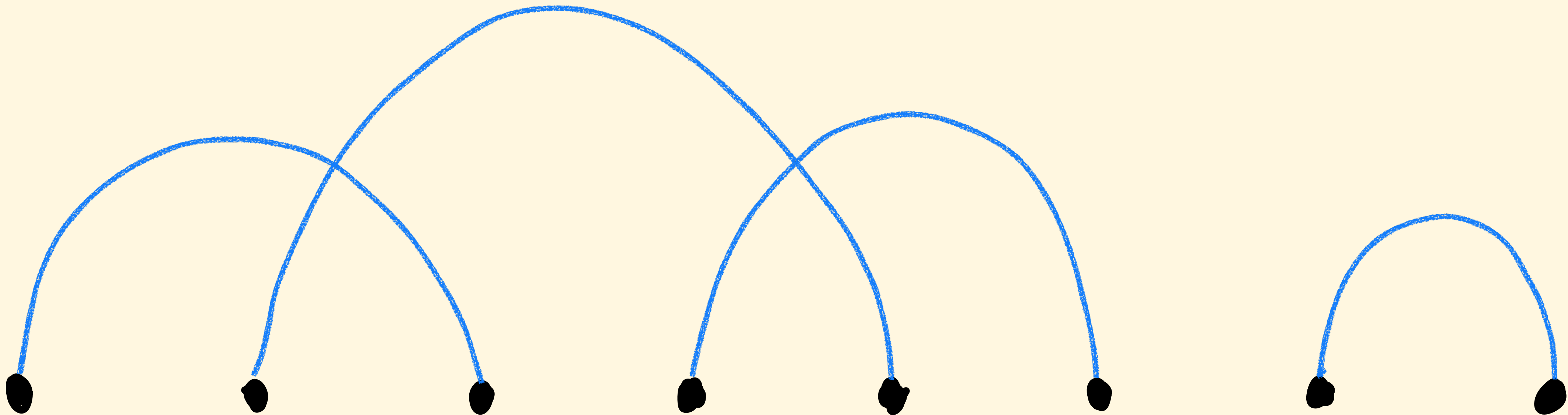
So, the dart games with n throws and m players are in bijection with the greedy matchings with n total pairs, $m - 1$ long pairs, and that don't end with short pair.

We're going to count these!

Dart games as matchings

We can build all greedy matchings in a unique way by

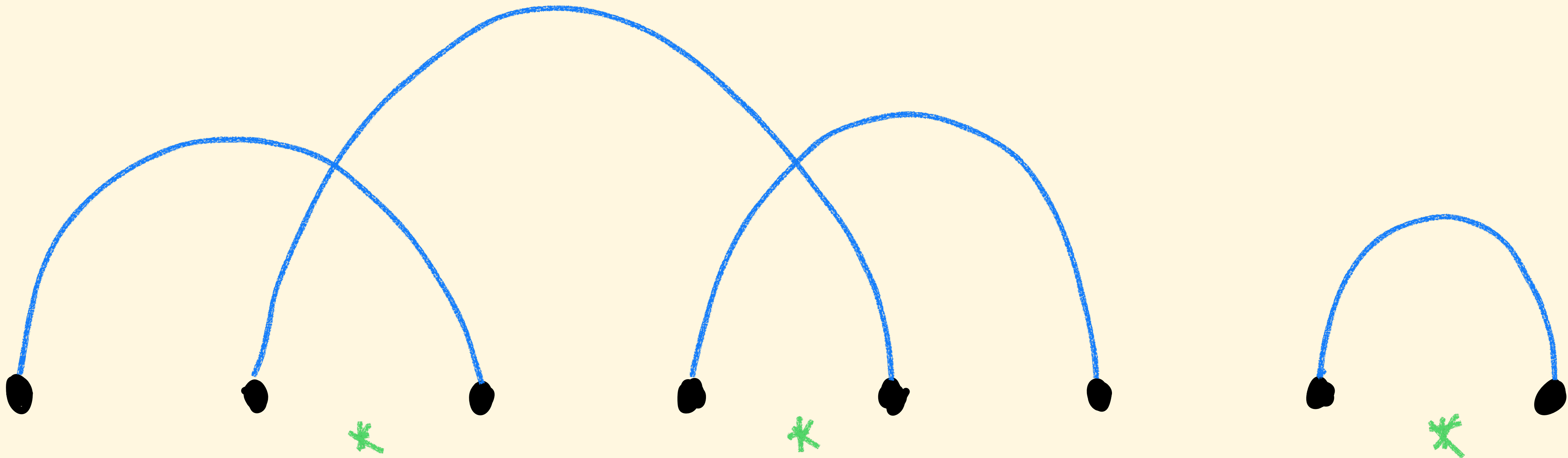
- starting with any perfect matching (these will end up as the long pairs)
- insert any number of short pairs in the gaps, making sure to add at least one in any gap between an open on the left and a close on the right
- to make a valid game, don't put any short pairs at the end



Dart games as matchings

We can build all greedy matchings in a unique way by

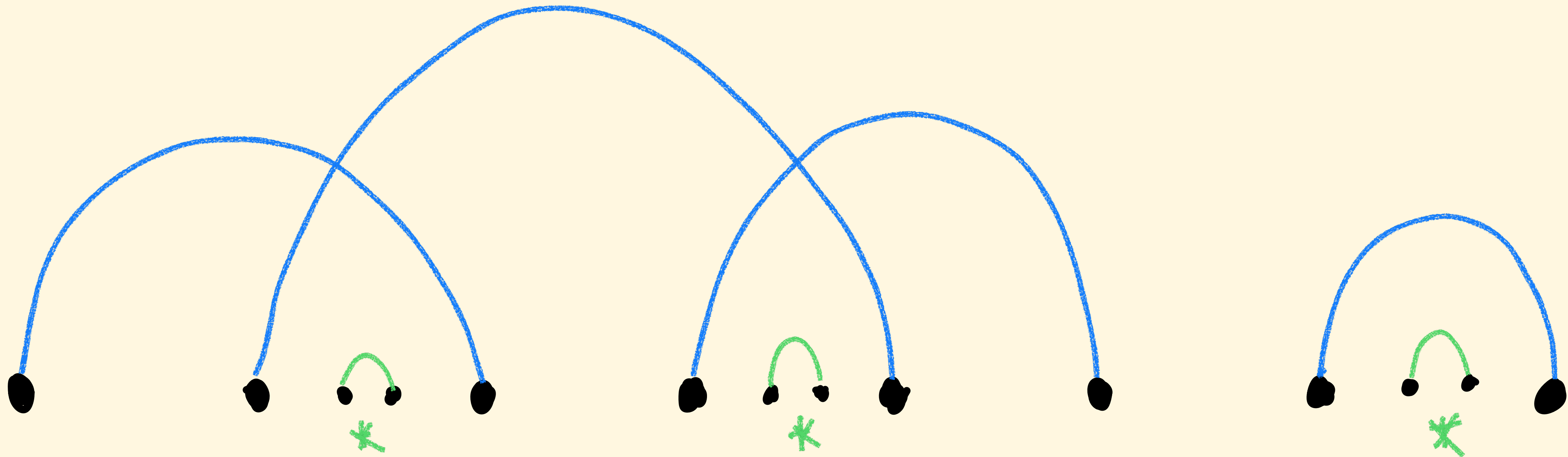
- starting with any perfect matching (these will end up as the long pairs)
- insert any number of short pairs in the gaps, making sure to add at least one in any gap between an open on the left and a close on the right
- to make a valid game, don't put any short pairs at the end



Dart games as matchings

We can build all greedy matchings in a unique way by

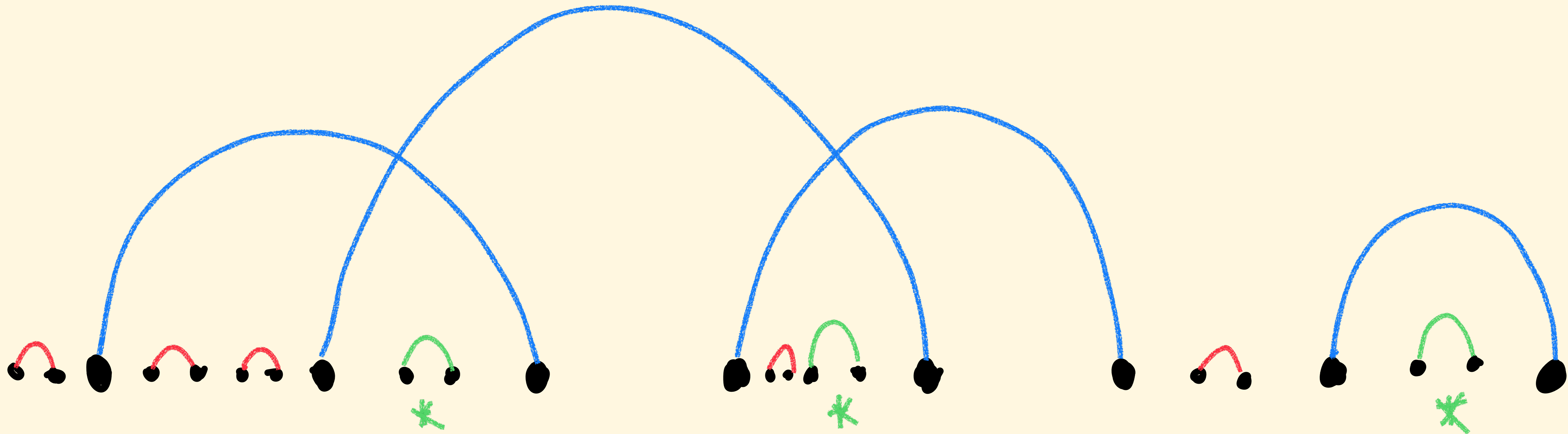
- starting with any perfect matching (these will end up as the long pairs)
- insert any number of short pairs in the gaps, making sure to add at least one in any gap between an open on the left and a close on the right
- to make a valid game, don't put any short pairs at the end



Dart games as matchings

We can build all greedy matchings in a unique way by

- starting with any perfect matching (these will end up as the long pairs)
- insert any number of short pairs in the gaps, making sure to add at least one in any gap between an open on the left and a close on the right
- to make a valid game, don't put any short pairs at the end



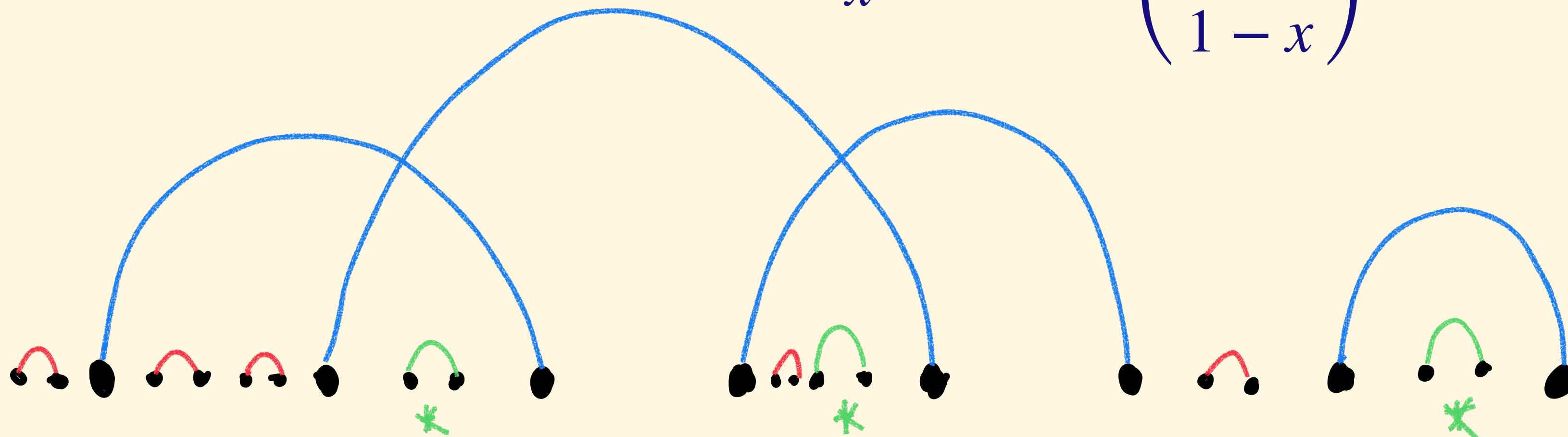
Dart games as matchings

Let $\mathcal{M}_\ell$ be the set of perfect matchings with ℓ pairs.

For any $M \in \mathcal{M}_\ell$ let $\text{star}(M)$ be the number of gaps between an opening on the left and a closing on the right.

For any $M \in \mathcal{M}_{m-1}$, the generating function for greedy matchings whose long pairs form the matching M and that do not end in a short pair is

$$x^{m-1+\text{star}(M)} \left(\frac{1}{1-x} \right)^{2(m-1)}$$



Dart games as matchings

Let $f_m(x)$ be the generating function that counts all greedy matchings with $m - 1$ long pairs and that do not end with a short pair.

$$f_m(x) = \sum_{M \in \mathcal{M}_{m-1}} x^{m-1+\text{star}(M)} \left(\frac{1}{1-x} \right)^{2(m-1)}$$

Refining by winning player

In any game, the order in which the players are eliminated forms a permutation σ that we call the *elimination order*.

Each elimination order $\sigma \in S_m$ defines a *modulus sequence*

$$\text{mod}(\sigma) = (\text{mod}_1(\sigma), \dots, \text{mod}_{m-1}(\sigma))$$

that encodes, up to a modulus, how many hits there are between each miss.

Refining by winning player

Example: $\sigma = 31452$

The number of hits before the first miss (which eliminates player 3) is equivalent to $2 \bmod 5$.

The number of hits between the first and second miss (which eliminates player 1) is equivalent to $2 \bmod 4$.

The number of hits between the second and third miss (which eliminates player 4) is equivalent to $1 \bmod 3$.

The number of hits between the third and last miss (which eliminates player 5) is equivalent to $0 \bmod 2$.

$$\text{mod}(31452) = (2, 2, 1, 0)$$

Refining by winning player

If $\text{mod}_1(\sigma) = a$ then we must insert $a + mq$ (for some $q \in \mathbb{N}$) short pairs in the gaps before the first close of M , including at least one just to the left of the first close.

Refining by winning player

If $\text{mod}_1(\sigma) = a$ then we must insert $a + mq$ (for some $q \in \mathbb{N}$) short pairs in the gaps before the first close of M , including at least one just to the left of the first close.

Suppose there are ℓ openings to the left of the first close (thus $\ell + 1$ gaps), and among them we are going to insert Q short pairs (including one just to the left of the first close). There are $\binom{Q - 1 + \ell}{\ell}$ ways to do this.

Refining by winning player

If $\text{mod}_1(\sigma) = a$ then we must insert $a + mq$ (for some $q \in \mathbb{N}$) short pairs in the gaps before the first close of M , including at least one just to the left of the first close.

Suppose there are ℓ openings to the left of the first close (thus $\ell + 1$ gaps), and among them we are going to insert Q short pairs (including one just to the left of the first close). There are $\binom{Q - 1 + \ell}{\ell}$ ways to do this.

Generating function trick: if $F(x) = x^Q$ then

$$\frac{1}{\ell!} \cdot x \cdot \frac{d^\ell}{dx^\ell}(x^{\ell-1}F(x)) = \binom{Q - 1 + \ell}{\ell} x^Q$$

Refining by winning player

So if we define

$$\text{cut}_\ell(F) := \frac{1}{\ell!} \cdot x \cdot \frac{d^\ell}{dx^\ell}(x^{\ell-1}F(x))$$

then this is the linear operator with the effect

$$x^Q \mapsto \binom{Q-1+\ell}{\ell} x^Q$$

If we are going to place $a + mq$ short pairs interleaving ℓ openings, the generating function for these arrangements is

$$\text{cut}_\ell(x^a + x^{a+m} + x^{a+2m} + \dots) = \text{cut}_\ell\left(\frac{x^a}{1 - x^m}\right)$$

(x tracks only the short pairs added)

Refining by winning player

For elimination order σ and modulus sequence $(\text{mod}_1(\sigma), \dots, \text{mod}_{m-1}(\sigma))$, we need to insert

Refining by winning player

For elimination order σ and modulus sequence $(\text{mod}_1(\sigma), \dots, \text{mod}_{m-1}(\sigma))$, we need to insert

- $\text{mod}_1(\sigma) + mq_1$ short pairs before the first closing

Refining by winning player

For elimination order σ and modulus sequence $(\text{mod}_1(\sigma), \dots, \text{mod}_{m-1}(\sigma))$, we need to insert

- $\text{mod}_1(\sigma) + mq_1$ short pairs before the first closing
- $\text{mod}_2(\sigma) + (m - 1)q_2$ short pairs between the first and second closings

Refining by winning player

For elimination order σ and modulus sequence $(\text{mod}_1(\sigma), \dots, \text{mod}_{m-1}(\sigma))$, we need to insert

- $\text{mod}_1(\sigma) + mq_1$ short pairs before the first closing
- $\text{mod}_2(\sigma) + (m - 1)q_2$ short pairs between the first and second closings
- $\text{mod}_3(\sigma) + (m - 2)q_3$ short pairs between the second and third closings

Refining by winning player

For elimination order σ and modulus sequence $(\text{mod}_1(\sigma), \dots, \text{mod}_{m-1}(\sigma))$, we need to insert

- $\text{mod}_1(\sigma) + mq_1$ short pairs before the first closing
- $\text{mod}_2(\sigma) + (m - 1)q_2$ short pairs between the first and second closings
- $\text{mod}_3(\sigma) + (m - 2)q_3$ short pairs between the second and third closings
- ...
- $\text{mod}_{m-1}(\sigma) + 2q_{m-1}$ short pairs between the last two closings

Refining by winning player

For elimination order σ and modulus sequence $(\text{mod}_1(\sigma), \dots, \text{mod}_{m-1}(\sigma))$, we need to insert

- $\text{mod}_1(\sigma) + mq_1$ short pairs before the first closing
- $\text{mod}_2(\sigma) + (m - 1)q_2$ short pairs between the first and second closings
- $\text{mod}_3(\sigma) + (m - 2)q_3$ short pairs between the second and third closings
- $\dots$
- $\text{mod}_{m-1}(\sigma) + 2q_{m-1}$ short pairs between the last two closings

For $M \in \mathcal{M}_{m-1}$, define the gap sequence $\text{gap}(M) = (\text{gap}_1(M), \dots, \text{gap}_{m-1}(M))$ where

- $\text{gap}_1(M)$ is the number of openings before the first closing
- $\text{gap}_i(M)$ is the number of openings between closing $i - 1$ and closing i

Refining by winning player

Putting this all together: the generating function for valid games with m players and elimination order σ , where x tracks the number of throws, is

$$x^{m-1} \sum_{M \in \mathcal{M}_{m-1}} \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

Refining by winning player

Putting this all together: the generating function for valid games with m players and elimination order σ , where x tracks the number of throws, is

$$x^{m-1} \sum_{M \in \mathcal{M}_{m-1}} \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

Define $S_m^k = \{\sigma \in S_m : \sigma(m) = k\}$.

Refining by winning player

Putting this all together: the generating function for valid games with m players and elimination order σ , where x tracks the number of throws, is

$$x^{m-1} \sum_{M \in \mathcal{M}_{m-1}} \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

Define $S_m^k = \{\sigma \in S_m : \sigma(m) = k\}$.

The generating function for all games with m players in which player k wins is:

$$W_{m,k}(x) \left(= \sum_{n \geq 0} w_{n,m,k} x^n \right) = x^{m-1} \sum_{\sigma \in S_m^k} \sum_{M \in \mathcal{M}_{m-1}} \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

Refining by winning player

$$W_{m,k}(x) \left(= \sum_{n \geq 0} w_{n,m,k} x^n \right) = x^{m-1} \sum_{\sigma \in S_m^k} \sum_{M \in \mathcal{M}_{m-1}} \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

This is a sum over roughly $2^{m-1}(m-1)!$ ² pairs (σ, M) , and we can manage it for up to 5 or 6 players.

Refining by winning player

$$W_{m,k}(x) \left(= \sum_{n \geq 0} w_{n,m,k} x^n \right) = x^{m-1} \sum_{\sigma \in S_m^k} \sum_{M \in \mathcal{M}_{m-1}} \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

This is a sum over roughly $2^{m-1}(m-1)!^2$ pairs (σ, M) , and we can manage it for up to 5 or 6 players.

Improvement: compress matchings down to Dyck paths

$$W_{m,k}(x) = x^{m-1} \sum_{\sigma \in S_m^k} \sum_{D \in \mathcal{D}_{m-1}} h(D) \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

where $h(D)$ is the product of the heights of the down-steps of D .

Refining by winning player

$$W_{m,k}(x) = x^{m-1} \sum_{\sigma \in S_m^k} \sum_{D \in \mathcal{D}_{m-1}} h(D) \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

This is a rational generating function!

Refining by winning player

$$W_{m,k}(x) = x^{m-1} \sum_{\sigma \in S_m^k} \sum_{D \in \mathcal{D}_{m-1}} h(D) \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

This is a rational generating function!

$$\begin{aligned} W_{3,1}(x) &= \frac{x^3 (x^7 - x^6 + 9x^5 + 4x^4 + 2x^3 + 12x^2 - 2x + 2)}{(x-1)^4 (x^2 + x + 1)^3} \\ &= 2x^3 + 12x^5 + 20x^6 + 18x^7 + 63x^8 + 80x^9 + 81x^{10} + \dots \end{aligned}$$

Refining by winning player

$$W_{m,k}(x) = x^{m-1} \sum_{\sigma \in S_m^k} \sum_{D \in \mathcal{D}_{m-1}} h(D) \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

This is a rational generating function!

$$\begin{aligned} W_{3,1}(x) &= \frac{x^3 (x^7 - x^6 + 9x^5 + 4x^4 + 2x^3 + 12x^2 - 2x + 2)}{(x-1)^4 (x^2 + x + 1)^3} \\ &= 2x^3 + 12x^5 + 20x^6 + 18x^7 + 63x^8 + 80x^9 + 81x^{10} + \dots \end{aligned}$$

$$W_{4,1}(x) = \frac{(x^{20} + x^{19} + 22x^{18} + 115x^{17} + 271x^{16} + 532x^{15} + 1143x^{14} + 1875x^{13} + 2753x^{12} + 3256x^{11} + 3557x^{10} + 3539x^9 + 3251x^8 + 2372x^7 + 1601x^6 + 849x^5 + 486x^4 + 223x^3 + 61x^2 + 6x + 6) x^4}{(x-1)^6 (x+1)^4 (x^2+x+1)^3 (x^2+1)^4}$$

Refining by winning player

$$W_{m,k}(x) = x^{m-1} \sum_{\sigma \in S_m^k} \sum_{D \in \mathcal{D}_{m-1}} h(D) \prod_{i=1}^{m-1} \text{cut}_{\text{gap}_i(M)} \left(\frac{x^{\text{mod}_i(\sigma)}}{1 - x^{m+1-i}} \right)$$

This is a rational generating function!

$$\begin{aligned} W_{3,1}(x) &= \frac{x^3 (x^7 - x^6 + 9x^5 + 4x^4 + 2x^3 + 12x^2 - 2x + 2)}{(x-1)^4 (x^2 + x + 1)^3} \\ &= 2x^3 + 12x^5 + 20x^6 + 18x^7 + 63x^8 + 80x^9 + 81x^{10} + \dots \end{aligned}$$

$$W_{4,1}(x) = \frac{(x^{20} + x^{19} + 22x^{18} + 115x^{17} + 271x^{16} + 532x^{15} + 1143x^{14} + 1875x^{13} + 2753x^{12} + 3256x^{11} + 3557x^{10} + 3539x^9 + 3251x^8 + 2372x^7 + 1601x^6 + 849x^5 + 486x^4 + 223x^3 + 61x^2 + 6x + 6) x^4}{(x-1)^6 (x+1)^4 (x^2 + x + 1)^3 (x^2 + 1)^4}$$

$$W_{8,1} = \frac{p(x)}{q(x)}, \text{ where } p(x) \text{ and } q(x) \text{ have degree 156.}$$

Converting to a probability

We have:

$$\begin{aligned} W_{3,1}(x) &= \frac{x^3 (x^7 - x^6 + 9x^5 + 4x^4 + 2x^3 + 12x^2 - 2x + 2)}{(x-1)^4 (x^2 + x + 1)^3} \\ &= 2x^3 + 12x^5 + 20x^6 + 18x^7 + 63x^8 + 80x^9 + 81x^{10} + \dots \end{aligned}$$

Converting to a probability

We have:

$$\begin{aligned} W_{3,1}(x) &= \frac{x^3 (x^7 - x^6 + 9x^5 + 4x^4 + 2x^3 + 12x^2 - 2x + 2)}{(x-1)^4 (x^2 + x + 1)^3} \\ &= 2x^3 + 12x^5 + 20x^6 + 18x^7 + 63x^8 + 80x^9 + 81x^{10} + \dots \end{aligned}$$

We want:

$$\hat{W}_{3,1}(x) = \frac{2}{3!}x^3 + \frac{12}{5!}x^5 + \frac{20}{6!}x^6 + \frac{18}{7!}x^7 + \frac{63}{8!}x^8 + \frac{80}{9!}x^9 + \frac{81}{10!}x^{10} + \dots$$

Converting to a probability

We have:

$$\begin{aligned} W_{3,1}(x) &= \frac{x^3 (x^7 - x^6 + 9x^5 + 4x^4 + 2x^3 + 12x^2 - 2x + 2)}{(x-1)^4 (x^2 + x + 1)^3} \\ &= 2x^3 + 12x^5 + 20x^6 + 18x^7 + 63x^8 + 80x^9 + 81x^{10} + \dots \end{aligned}$$

We want:

$$\hat{W}_{3,1}(x) = \frac{2}{3!}x^3 + \frac{12}{5!}x^5 + \frac{20}{6!}x^6 + \frac{18}{7!}x^7 + \frac{63}{8!}x^8 + \frac{80}{9!}x^9 + \frac{81}{10!}x^{10} + \dots$$

Trick:

$$\hat{W}_{3,1}(x) = \mathcal{L}^{-1} \left\{ \frac{1}{x} W_{3,1} \left(\frac{1}{x} \right) \right\}$$

Converting to a probability

$$\begin{aligned}\hat{W}_{3,1}(x) &= \mathcal{L}^{-1} \left\{ \frac{1}{x} W_{3,1} \left(\frac{1}{x} \right) \right\} \\ &= 1 + \frac{1}{18} e^x (x-1) (3x^2 + 8) + \frac{1}{9} \left(\cos \left(\frac{\sqrt{3}x}{2} \right) (3x+5) (x-1) + \sin \left(\frac{\sqrt{3}x}{2} \right) \sqrt{3} (x^2 - 4x - 3) \right) e^{-\frac{x}{2}}\end{aligned}$$

and the win probability is

$$\hat{W}_{3,1}(1) = 1 - \frac{2}{\sqrt{3}e} \sin \left(\frac{\sqrt{3}}{2} \right)$$

Converting to a probability

Recall from earlier:

$$P_{5,1} = 1 - \frac{5}{3} \sum_{\Phi_3(\alpha)=0} e^{-\alpha}(\alpha - 1) - \frac{25}{4} \sum_{\Phi_4(\alpha)=0} e^{-\alpha} - \frac{1}{5} \sum_{\Phi_5(\alpha)=0} e^{-\alpha}(4\alpha^3 + 5\alpha^2 - 6\alpha - 8)$$

$$\Phi_3(x) = 1 + x + x^2$$

$$\Phi_4(x) = 1 + x^2$$

$$\Phi_5(x) = 1 + x + x^2 + x^3 + x^4$$

The sums over primitive roots of unity are coming from the fact that the denominators in the rational generating function factor into powers of $(1 - x^a)^b$.

How far can we push it?

This method works up to 9 or 10 players, but to do 10 players requires very big computational resources.

$$P_{10,1} = 0.1697191929$$

$$P_{10,2} = 0.1208295203$$

$$P_{10,3} = 0.1047135410$$

$$P_{10,4} = 0.09642179456$$

$$P_{10,5} = 0.09126507812$$

$$P_{10,6} = 0.08769663200$$

$$P_{10,7} = 0.08505171410$$

$$P_{10,8} = 0.08299520678$$

$$P_{10,9} = 0.08133889051$$

$$P_{10,10} = 0.07996842969$$

How far can we push it?

This method works up to 9 or 10 players, but to do 10 players requires very big computational resources.

$$P_{10,1} = 0.1697191929$$

$$\begin{aligned}
 & - \frac{30 \left(\sum_{\alpha=\text{RootOf}[Z^6+Z^5+Z^4+Z^3+Z^2+Z+1]} e^{-\alpha} (1089068 \alpha^5 - 956149 \alpha^4 - 4593704 \alpha^3 - 7084788 \alpha^2 - 6555207 \alpha - 3402765) \right)}{7} \\
 & + \frac{5 \left(\sum_{\alpha=\text{RootOf}[Z^6+Z^3+1]} e^{-\alpha} (11532313 \alpha^5 + 21003597 \alpha^4 + 20644858 \alpha^3 + 22161679 \alpha^2 + 16647347 \alpha + 3339166) \right)}{3} \\
 & - \frac{\left(\sum_{\alpha=\text{RootOf}[Z^4-Z^3+Z^2-Z+1]} (1190153945 \alpha^3 - 153397242 \alpha^2 + 640760616 \alpha - 888945812) e^{-\alpha} \right)}{50} - \frac{\left(\sum_{\alpha=\text{RootOf}[Z^4+Z^3+Z^2+Z+1]} e^{-\alpha} (160352687 \alpha^3 - 3951676 \alpha^2 - 265827614 \alpha - 263375462) \right)}{250} \\
 & - \frac{15 \left(\sum_{\alpha=\text{RootOf}[Z^4+1]} e^{-\alpha} (49106457 \alpha^3 + 58839532 \alpha^2 + 34105854 \alpha - 10606929) \right)}{16} - \frac{7 \left(\sum_{\alpha=\text{RootOf}[Z^2-Z+1]} e^{-\alpha} (-9791310 + 95690447 \alpha) \right)}{36} - \frac{\left(\sum_{\alpha=\text{RootOf}[Z^2+Z+1]} e^{-\alpha} (342994 + 259847 \alpha) \right)}{108} \\
 & + \frac{5 \left(\sum_{\alpha=\text{RootOf}[Z^2+1]} e^{-\alpha} (-1788749 + 65410 \alpha) \right)}{64} + 1 - \frac{4949 e^{-1}}{11520}
 \end{aligned}$$

$$P_{10,8} = 0.08299520678$$

$$P_{10,9} = 0.08133889051$$

$$P_{10,10} = 0.07996842969$$

Second approach: guess-and-check

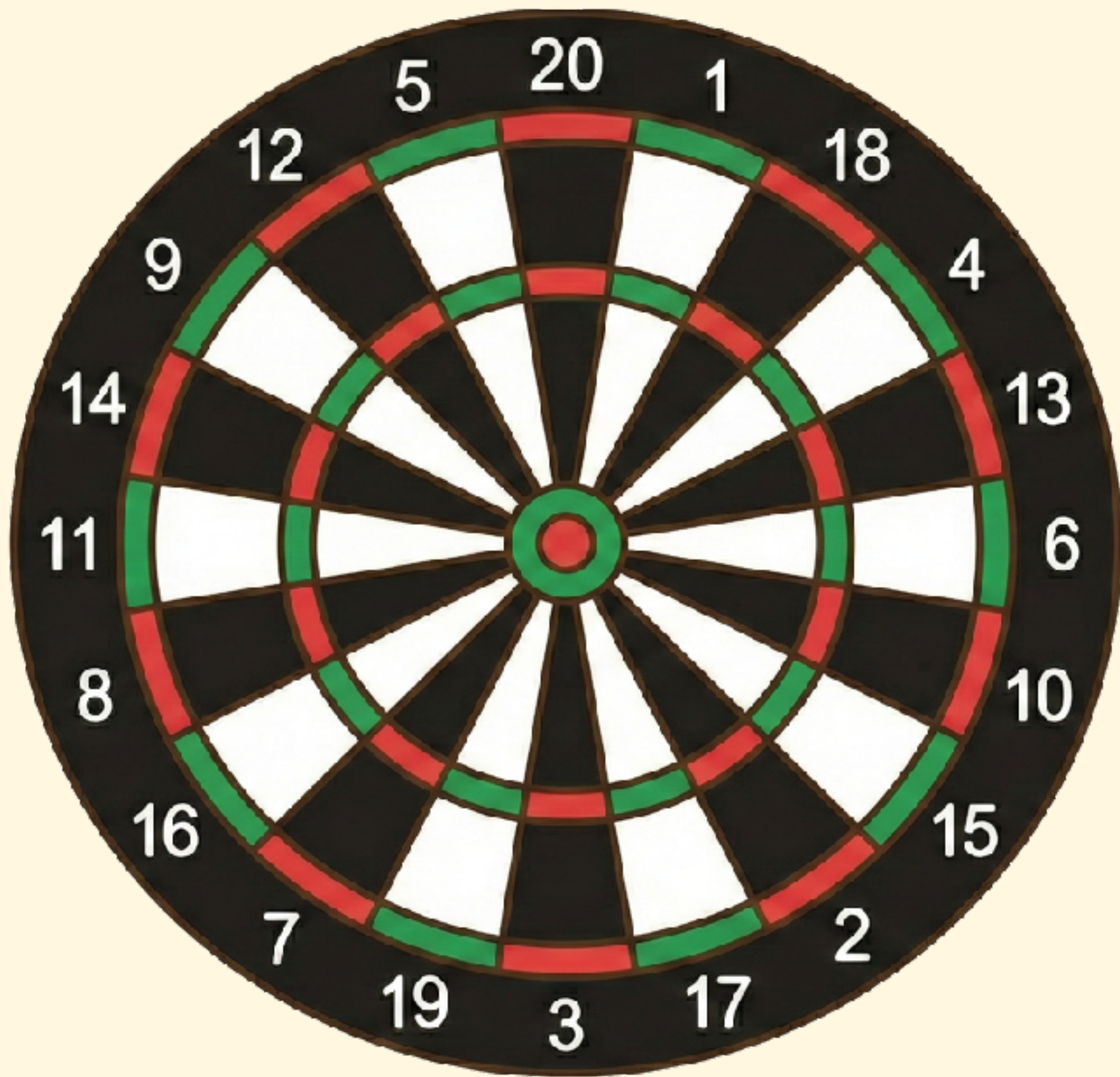
Just a very brief overview...

WLOG, we can assume the players throws are drawn from $\text{Unif}[0,1]$, with 0 being a bullseye and 1 being as far from the bullseye as possible.

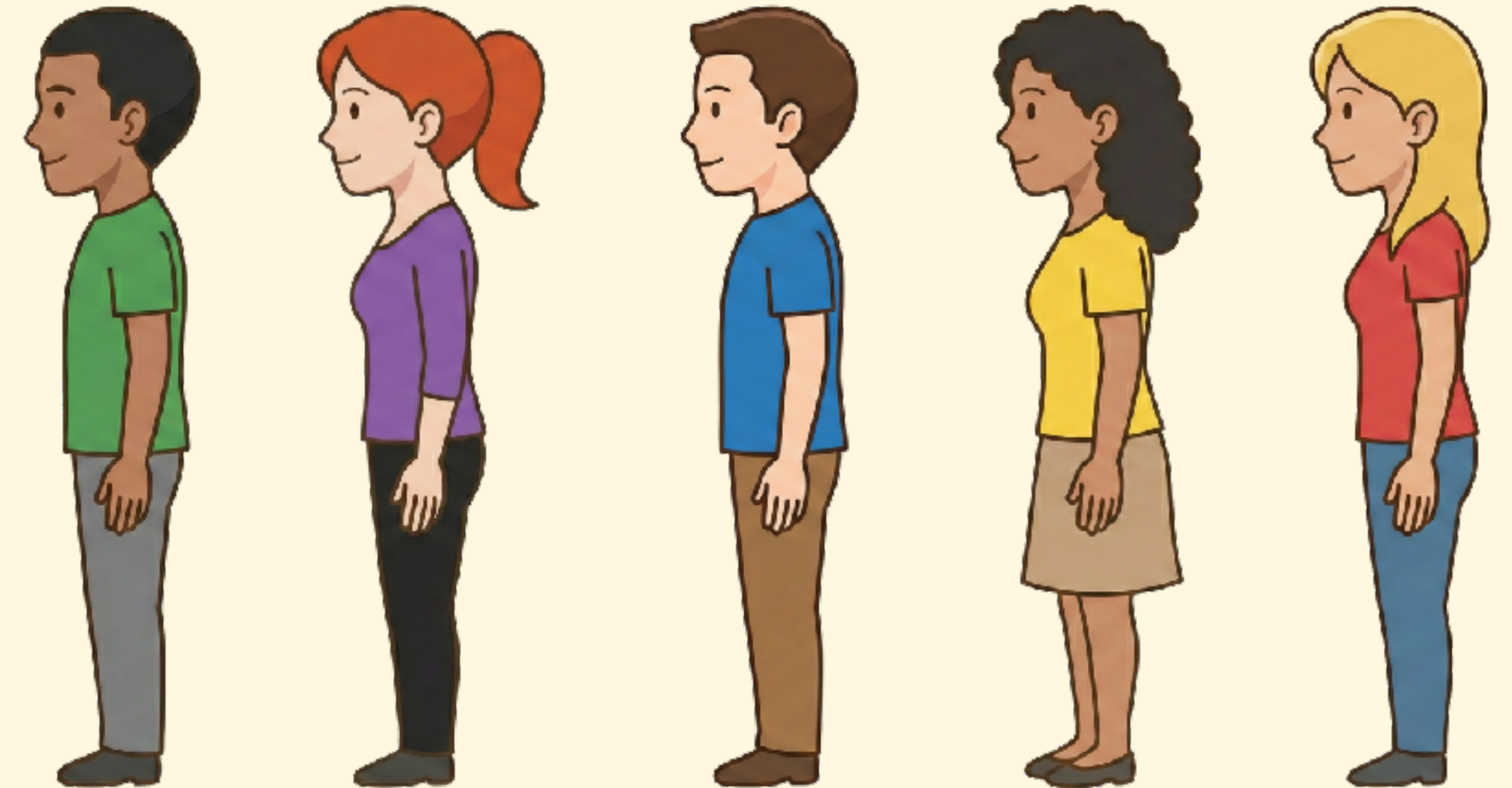
Andy Niedermaier defines functions $P_{m,k}(x)$ to be the probability that the player whose is currently k^{th} in line will win in a game in progress that currently has m players and in which the current best throw is x .

$$P_{m,k} = P_{m,k}(1)$$

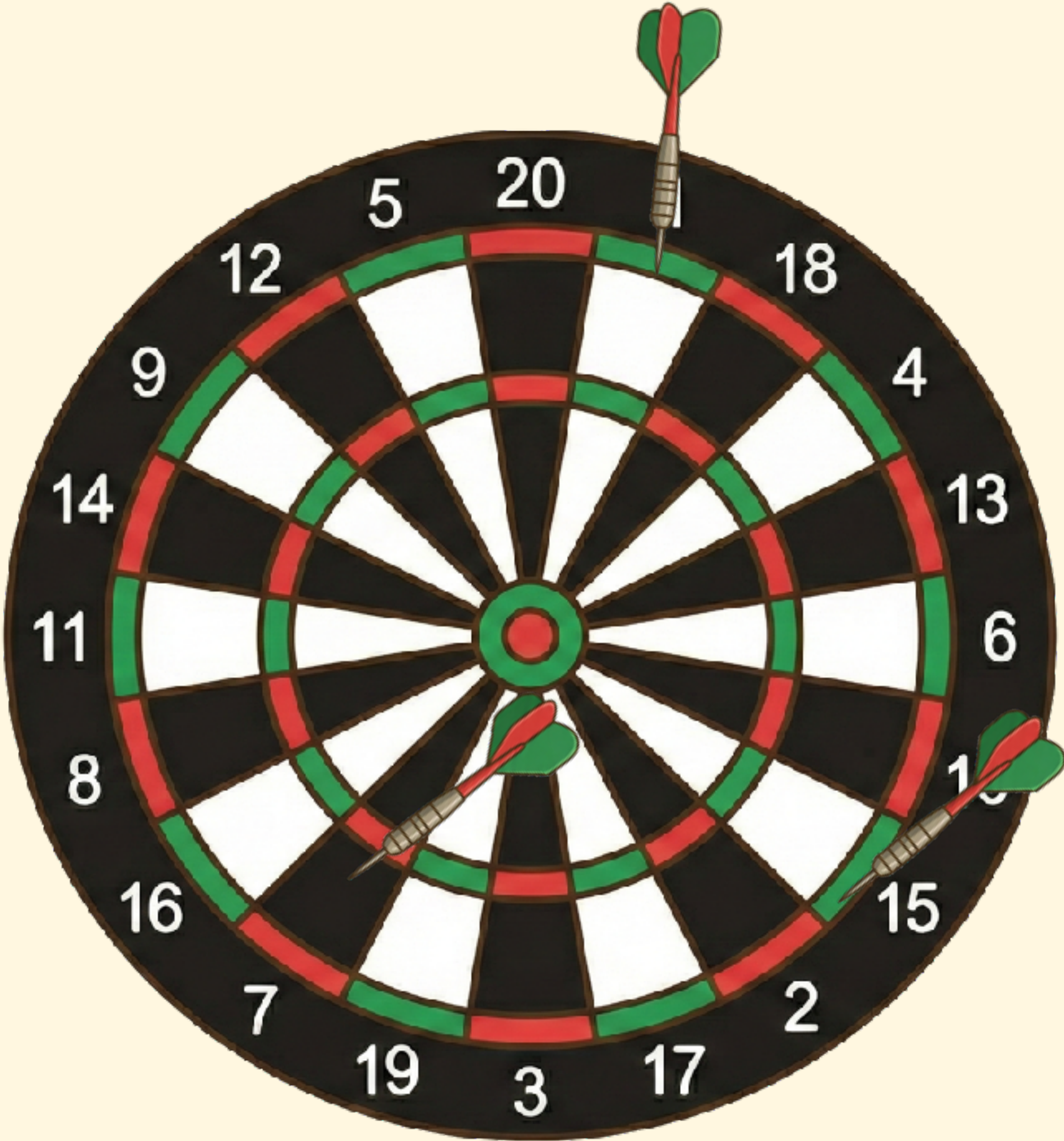
Andy Niedermaier's Dart Game



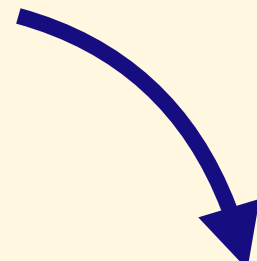
$$P_{5,4} = P_{5,4}(1)$$



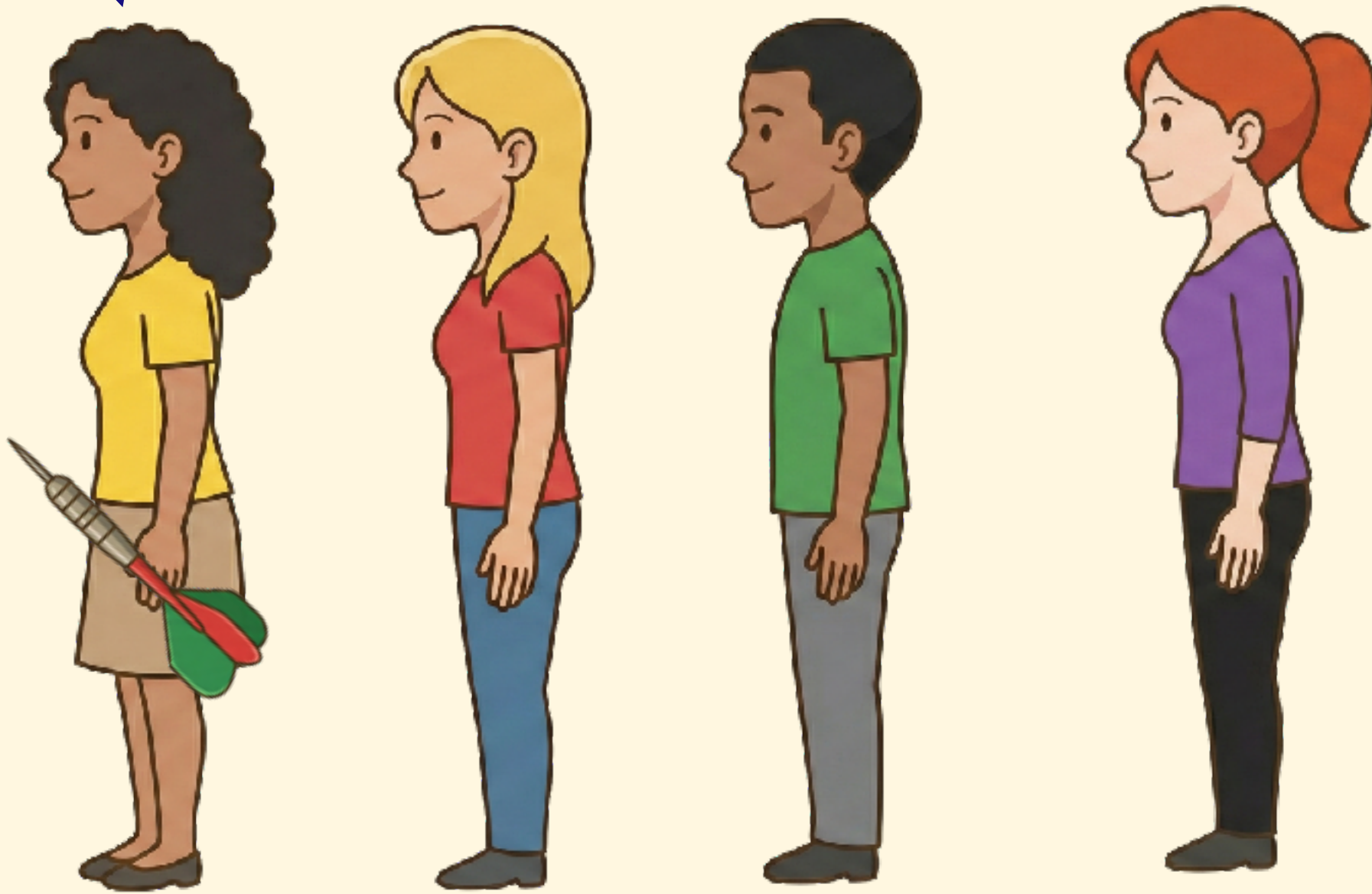
Andy Niedermaier's Dart Game



$$P_{4,1}(0.7)$$

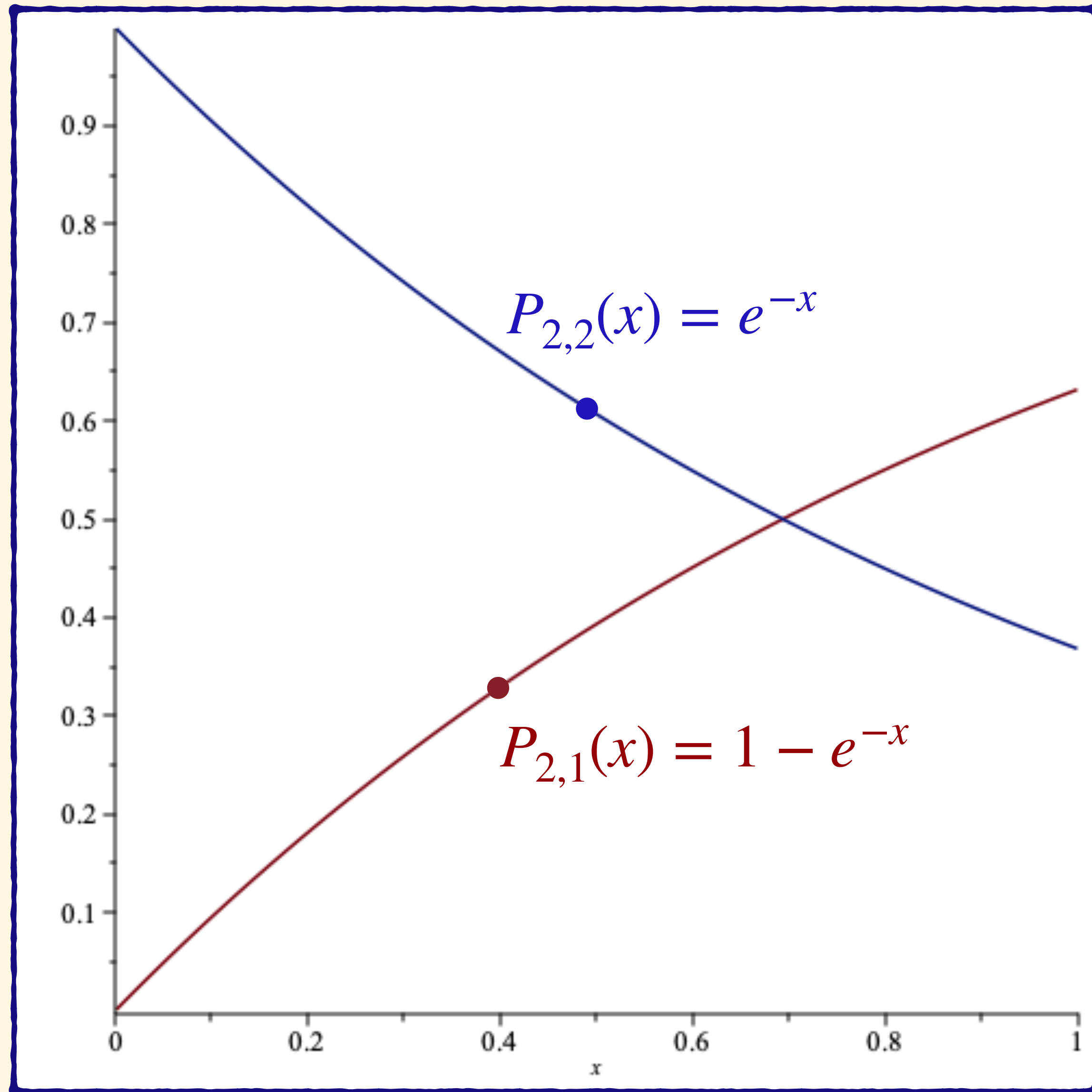


$$P_{4,3}(0.7)$$



Player 4 hit.

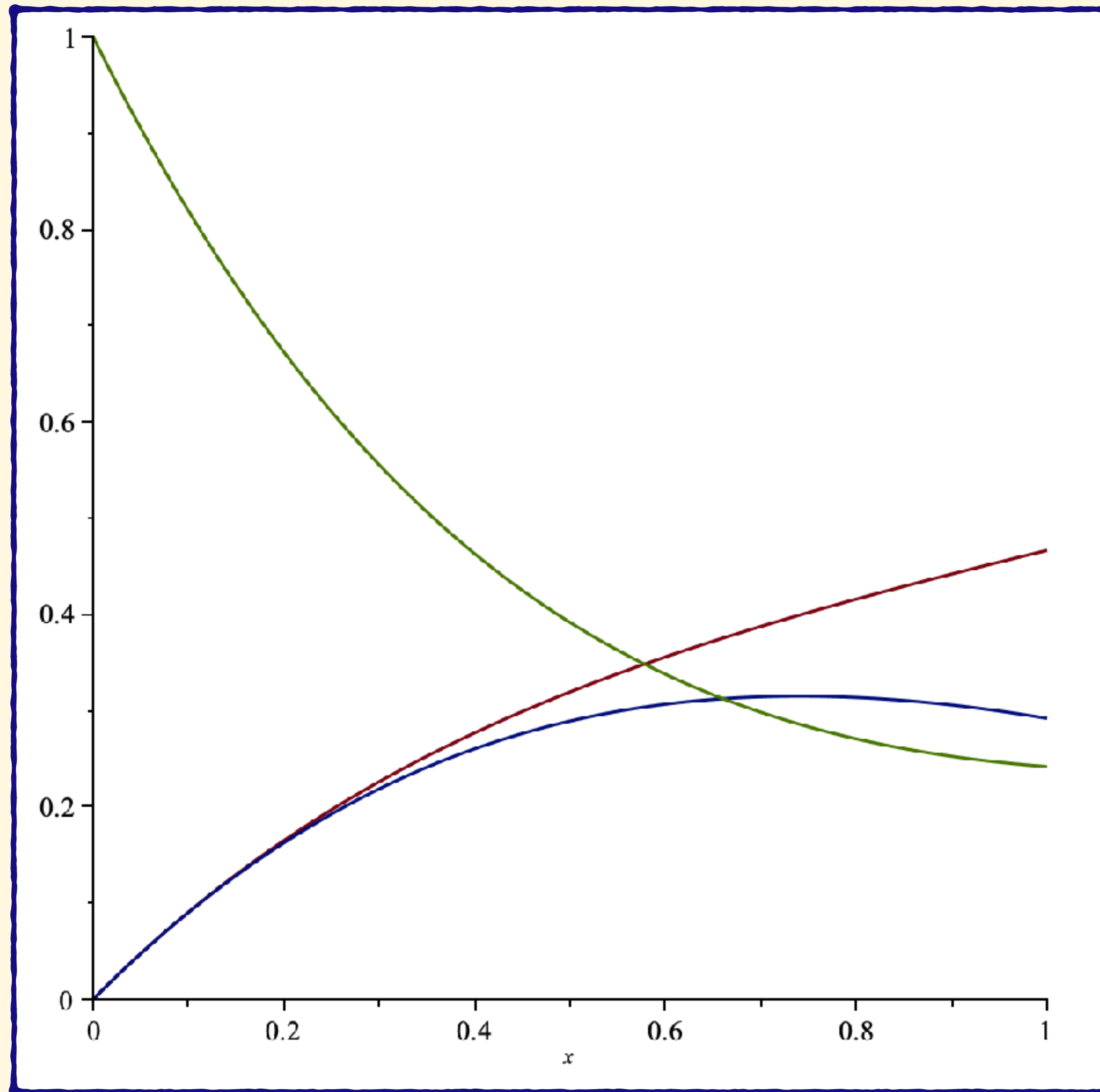
Second approach: guess-and-check



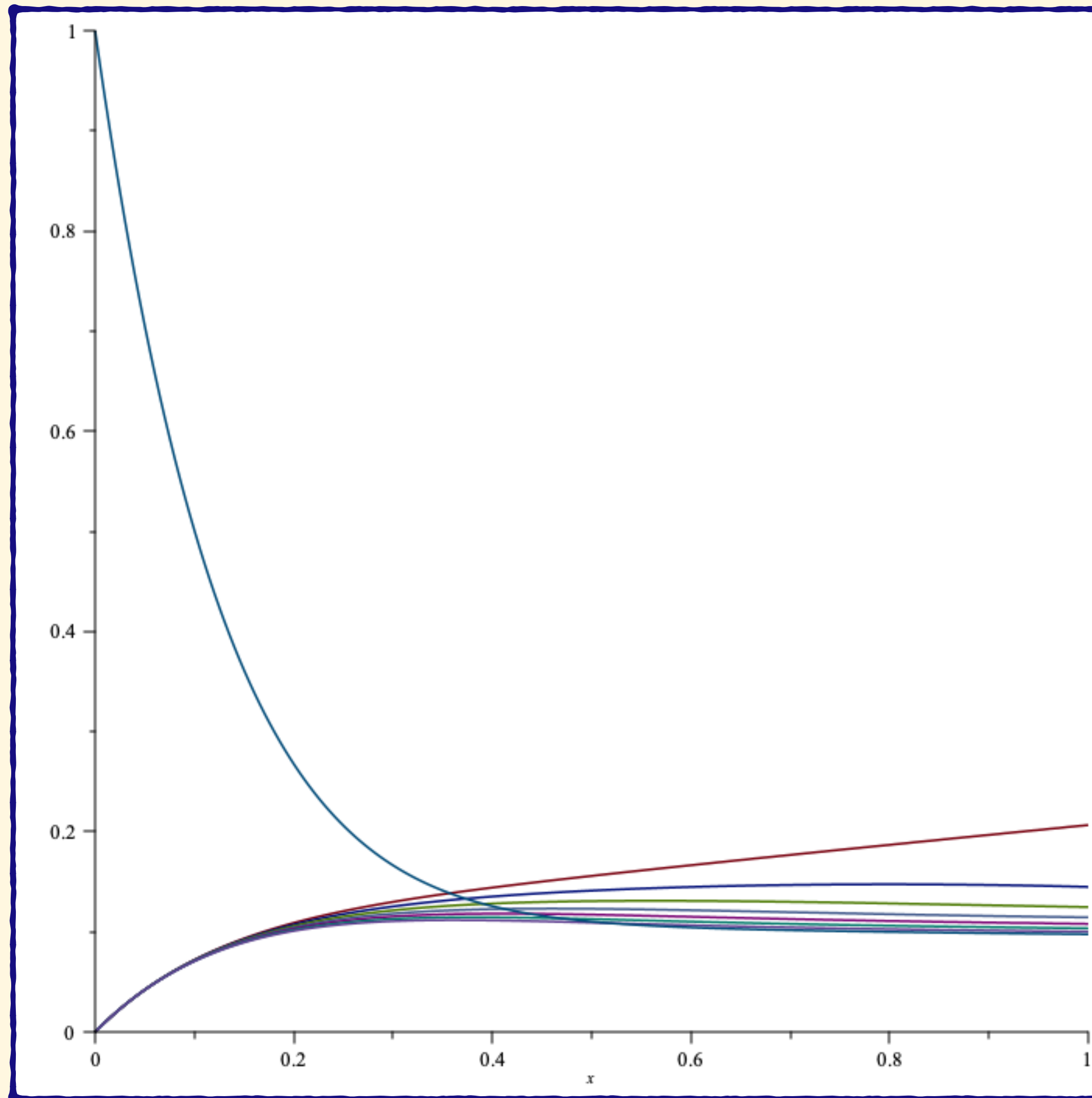
The probability that the player 2nd in line wins if two people are left and current best is 0.5 is ≈ 0.61

The probability that the player in front wins if two people are left and current best is 0.4 is ≈ 0.33

Second approach: guess-and-check



Second approach: guess-and-check



Second approach: guess-and-check

For a 2-player game, the functions $P_{2,1}(x)$ and $P_{2,2}(x)$ satisfy the system:

$$P_{2,1}(x) = \int_0^x P_{2,2}(u) du$$

$$P_{2,1}(x) + P_{2,2}(x) = 1$$

$$P_{2,1}(0) = 0$$

Second approach: guess-and-check

For a 2-player game, the functions $P_{2,1}(x)$ and $P_{2,2}(x)$ satisfy the system:

$$P_{2,1}(x) = \int_0^x P_{2,2}(u) du$$

$$P_{2,1}(x) + P_{2,2}(x) = 1$$

$$P_{2,1}(0) = 0$$

If the current best is x , the player at the front stays in if they throw a value $u \in (0, x)$.

“Sum” (integrate) over all such u the probability they will go on to win the whole game, which is $P_{2,2}(u)$ because they are now the second person in line with a current best of u .

Second approach: guess-and-check

For a 2-player game, the functions $P_{2,1}(x)$ and $P_{2,2}(x)$ satisfy the system:

$$P_{2,1}(x) = \int_0^x P_{2,2}(u) du$$

$$P_{2,1}(x) + P_{2,2}(x) = 1$$

$$P_{2,1}(0) = 0$$

The win probabilities for each player add up to 1, regardless of the state of the game.

Second approach: guess-and-check

For a 2-player game, the functions $P_{2,1}(x)$ and $P_{2,2}(x)$ satisfy the system:

$$P_{2,1}(x) = \int_0^x P_{2,2}(u) du$$

$$P_{2,1}(x) + P_{2,2}(x) = 1$$

$$P_{2,1}(0) = 0$$

If the current best is a bullseye, the player in front will lose (and the player in back will win)

Second approach: guess-and-check

$$P_{2,1}(x) = \int_0^x P_{2,2}(u) du$$

$$P_{2,1}(x) + P_{2,2}(x) = 1$$

$$P_{2,1}(0) = 0$$

Maple can solve the system, and we can plug in $x = 1$ to find $P_{2,1}(1)$ and $P_{2,2}(1)$ which are the win probabilities for player 1 and 2 in a 2 player game.

These are not the same as the functions $\hat{W}_{2,1}(x)$ before, even though plugging in $x = 1$ to both gives the win probability.

Second approach: guess-and-check

3 players:

$$P_{3,1}(x) = \int_0^x P_{3,3}(u) du$$

$$P_{3,2}(x) = (1 - x)P_{2,1}(x) + \int_0^x P_{3,1}(u) du$$

$$P_{3,1}(x) + P_{3,2}(x) + P_{3,3}(x) = 1$$

$$P_{3,1}(0) = P_{3,2}(0) = 0$$

If the front player hits with u , “sum” over their win probabilities at their new spot as third in line.

Second approach: guess-and-check

3 players:

$$P_{3,1}(x) = \int_0^x P_{3,3}(u)du$$

$$P_{3,2}(x) = (1 - x)P_{2,1}(x) + \int_0^x P_{3,1}(u)du$$

$$P_{3,1}(x) + P_{3,2}(x) + P_{3,3}(x) = 1$$

$$P_{3,1}(0) = P_{3,2}(0) = 0$$

Middle player wins in two ways:

(1) player in front misses (probability $1 - x$), and now middle player is first in line in a 2 player game still with current best x , so their win probability is now $P_{2,1}(x)$.

Second approach: guess-and-check

3 players:

$$P_{3,1}(x) = \int_0^x P_{3,3}(u) du$$

$$P_{3,2}(x) = (1 - x)P_{2,1}(x) + \int_0^x P_{3,1}(u) du$$

$$P_{3,1}(x) + P_{3,2}(x) + P_{3,3}(x) = 1$$

$$P_{3,1}(0) = P_{3,2}(0) = 0$$

Middle player wins in two ways:

(2) player 1 hits at value $u \in (0, x)$, and now middle player is first in line, still in a 3 player game, with current best u . (Integrate over u .)

Second approach: guess-and-check

3 players:

$$P_{3,1}(x) = \int_0^x P_{3,3}(u) du$$

$$P_{3,2}(x) = (1 - x)P_{2,1}(x) + \int_0^x P_{3,1}(u) du$$

$$P_{3,1}(x) + P_{3,2}(x) + P_{3,3}(x) = 1$$

$$P_{3,1}(0) = P_{3,2}(0) = 0$$

Second approach: guess-and-check

3 players:

$$P_{3,1}(x) = \int_0^x P_{3,3}(u)du$$

$$P_{3,2}(x) = (1-x)P_{2,1}(x) + \int_0^x P_{3,1}(u)du$$

$$P_{3,1}(x) + P_{3,2}(x) + P_{3,3}(x) = 1$$

$$P_{3,1}(0) = P_{3,2}(0) = 0$$

Maple can solve.

Second approach: guess-and-check

4 players:

$$P_{4,1}(x) = \int_0^x P_{4,4}(u)du$$

$$P_{4,2}(x) = (1-x)P_{3,1}(x) + \int_0^x P_{4,1}(u)du$$

$$P_{4,3}(x) = (1-x)P_{3,2}(x) + \int_0^x P_{4,2}(u)du$$

$$P_{4,1}(x) + P_{4,2}(x) + P_{4,3}(x) + P_{4,4}(x) = 1$$

$$P_{4,1}(0) = P_{4,2}(0) = P_{4,3}(0) = 0$$

Maple can solve.

Second approach: guess-and-check

5 players:

$$P_{5,1}(x) = \int_0^x P_{5,5}(u)du$$

$$P_{5,2}(x) = (1-x)P_{4,1}(x) + \int_0^x P_{5,1}(u)du$$

$$P_{5,3}(x) = (1-x)P_{4,2}(x) + \int_0^x P_{5,2}(u)du$$

$$P_{5,4}(x) = (1-x)P_{4,3}(x) + \int_0^x P_{5,3}(u)du$$

$$P_{5,1}(x) + P_{5,2}(x) + P_{5,3}(x) + P_{5,4}(x) + P_{5,5}(x) = 1$$

$$P_{5,1}(0) = P_{5,2}(0) = P_{5,3}(0) = P_{5,4}(0) = 0$$

Maple cannot solve.

Second approach: guess-and-check

Very brief outline of the steps:

- Iterate the system to get terms in the Taylor series expansion of each $P_{m,k}(x)$.
- Guess the generating function solutions based on the terms.
- Plug these guesses into the system and check that they are a valid solution.

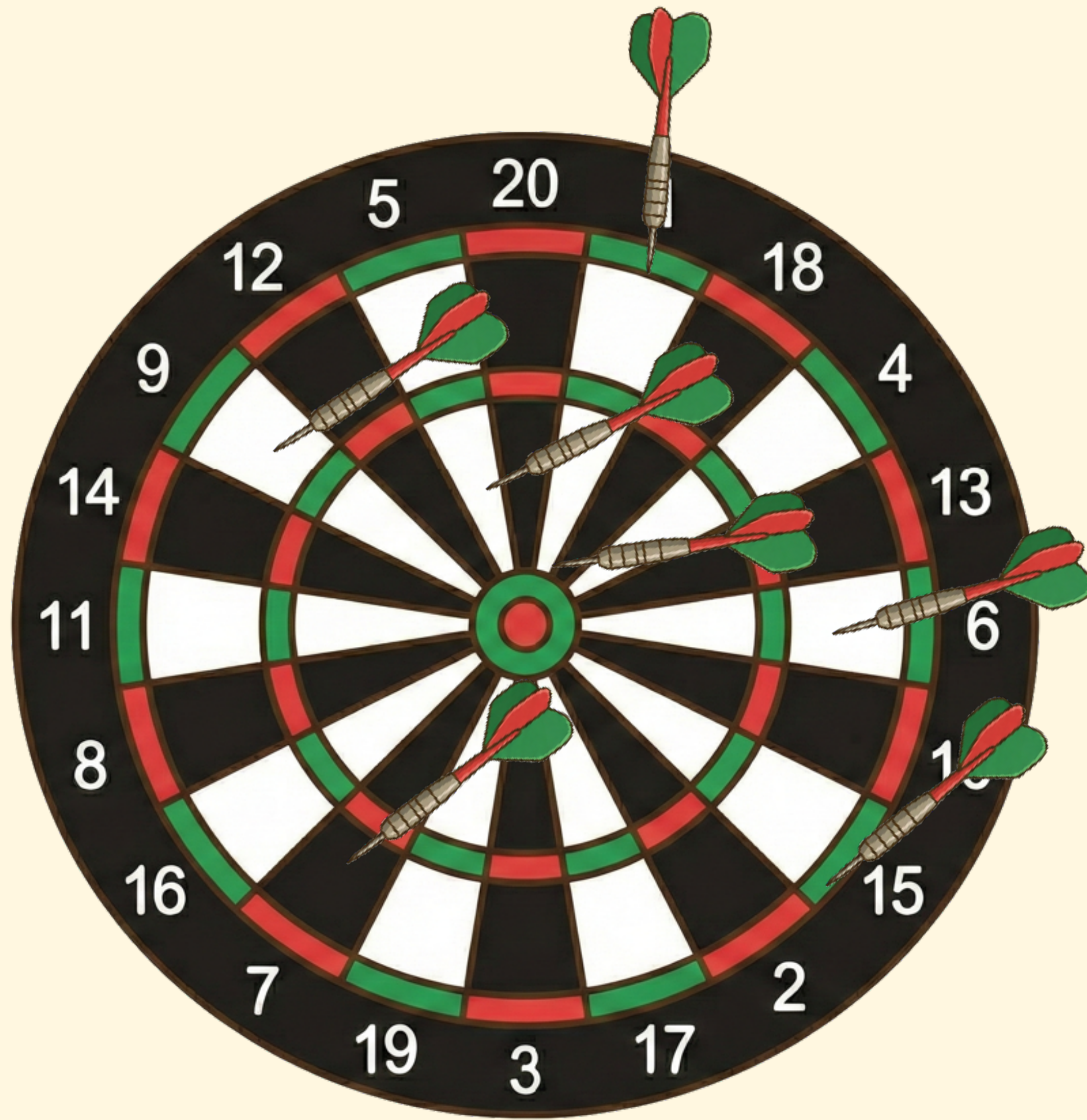
The slow part is iteration, but this method still works up to around 15 players, and I suspect there are tricks to speed up the iteration significantly.

Which is better?

Approach 1 is more combinatorial, computes win probabilities for individual elimination sequences, and uses permutations!

It also *proves* that the generating functions counting games are always rational, and the win probabilities come from evaluating their inverse Laplace transform at a point.

Approach 2 works on games with more players, is less combinatorially interesting.



Thank you!