Multi-agent learning

A similar BSc assignment

Gerard Vreeswijk, Intelligent Software Systems, Computer Science Department, Faculty of Sciences, Utrecht University, The Netherlands.

Saturday 6th June, 2020

Plan for today

Plan for today

- Comparing our programming assignment with a similar programming assignment in the bachelor course "Introduction to Adaptive Systems".

Plan for today

- Comparing our programming assignment with a similar programming assignment in the bachelor course "Introduction to Adaptive Systems".
- Discussing a solution app of this BSc assignment.

Plan for today

- Comparing our programming assignment with a similar programming assignment in the bachelor course "Introduction to Adaptive Systems".
- Discussing a solution app of this BSc assignment.
- Discussing the concept of a grand table for the first time.

Plan for today

- Comparing our programming assignment with a similar programming assignment in the bachelor course "Introduction to Adaptive Systems".
- Discussing a solution app of this BSc assignment.
- Discussing the concept of a grand table for the first time.
- Getting to know the replicator dynamic.

Plan for today

- Comparing our programming assignment with a similar programming assignment in the bachelor course “Introduction to Adaptive Systems”.
- Discussing a solution app of this BSc assignment.
- Discussing the concept of a grand table for the first time.
- Getting to know the replicator dynamic.
- Getting to see phase diagrams of the replicator.

Plan for today

- Comparing our programming assignment with a similar programming assignment in the bachelor course “Introduction to Adaptive Systems”.
- Discussing a solution app of this BSc assignment.
- Discussing the concept of a grand table for the first time.
- Getting to know the replicator dynamic.
- Getting to see phase diagrams of the replicator.
- Studying behaviour of the replicator dynamic on the grand table.

There is a similar programming assignment in the bachelor

The BSc assignment "strategieën vergelijken"

Inleiding Adaptieve Systemen | cs.uu.nl/docs/vakken/ias/main

www.cs.uu.nl/docs/vakken/ias/main.php?page=opdracht_ne

powerdot verbatim

cursus Inleiding Adaptieve Systemen | Opleiding Kunstmatige Intelligentie 2019-2020

Opdracht 2: strategieën vergelijken

Inleiding

Als het prisoner's dilemma meerdere keren wordt gespeeld, zijn veel strategieën denkbaar. Denk aan: altijd samenwerken, altijd verzaken, tit-for-tat, enz. We noemen een strategie *succesvol* als deze in vergelijking met andere strategieën een hoge opbrengst heeft. Bv. All-C is in de aanwezigheid van veel andere strategieën niet zo succesvol omdat All-C makkelijk van zich laat profiteren. Tit-for-tat is in de loop der jaren succesvol gebleken. Het presteert beter dan veel andere strategieën. Maar daarna zijn er toch ook serieuze concurrenten van TFT verschenen, denk aan Pavlov (a.k.a. "win stay, lose shift"). De vraag is nu: gegeven een aantal strategieën, bepaal welke de meest succesvolle is.

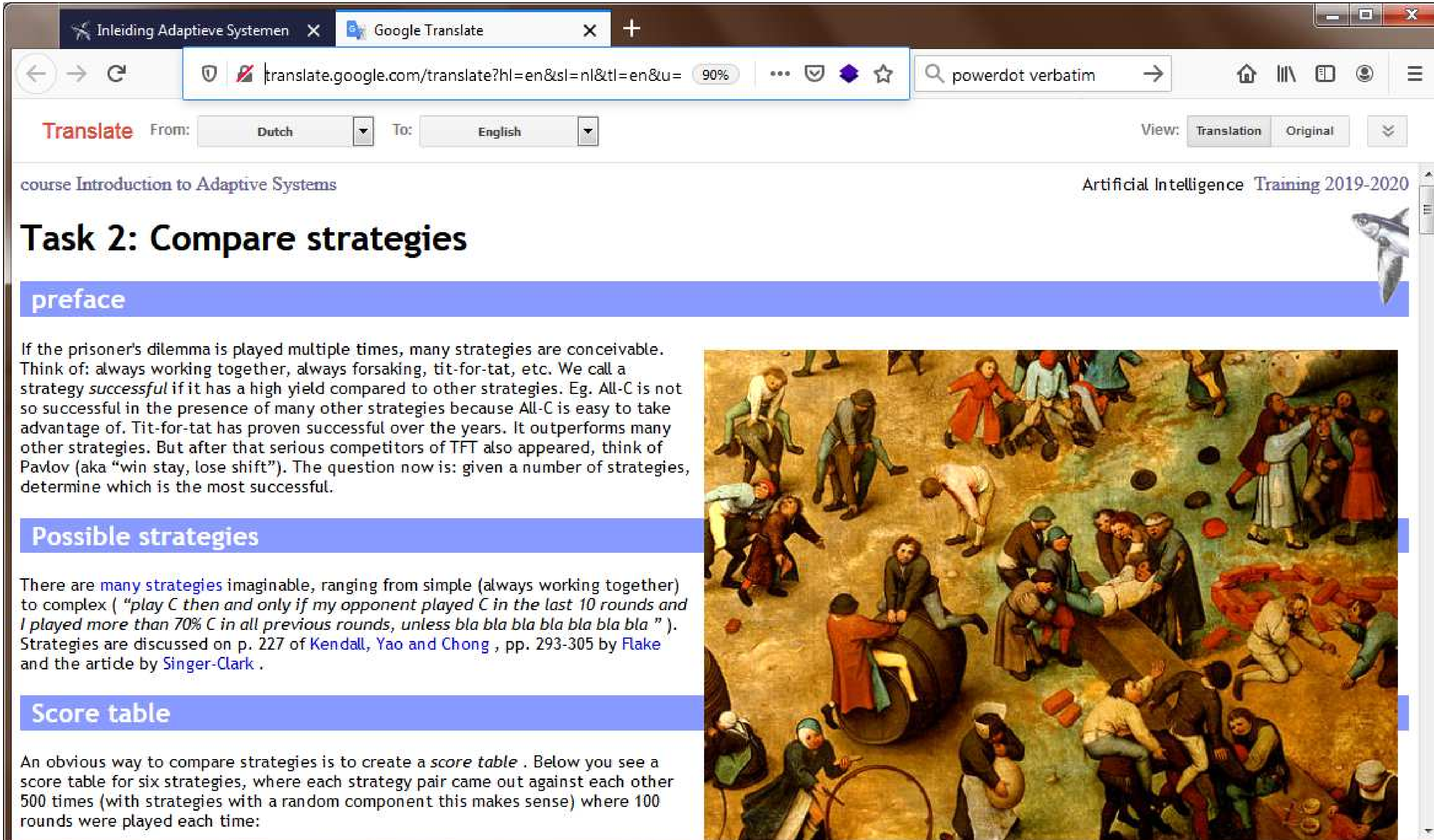
Mogelijke strategieën

Er zijn zeer veel strategieën denkbaar, varierend van eenvoudig (altijd samenwerken) tot complex (*"speel C dan en slechts dan als mijn opponent in de afgelopen 10 ronden C speelde en ik in alle voorgaande ronden meer dan 70% C speelde, tenzij bla bla bla mits bla bla bla"*). Strategieën worden besproken op bijvoorbeeld p. 227 van Kendall, Yao en Chong, pp. 293-305 van Flake en het artikel van Singer-Clark.

Score-tabel

http://www.cs.uu.nl/docs/vakken/ias/main.php?page=opdracht_netlogo_2_strategieen

The BSc assignment “Compare Strategies”



http://translate.google.com/translate?hl=en&sl=nl&tl=en&u=http://www.cs.uu.nl/do

Comparison

BSc assignment

MSc assignment

Comparison

BSc assignment

MSc assignment

1. Reactive strategies

   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, $\dots\}$.

Comparison

BSc assignment

1. Reactive strategies

   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, ...$\}$.

MSc assignment

1. Adaptive strategies / reply rules

   $S = \{$ `No-regret`, `Fictitious Play`, ϵ-`Greedy`, `Bully`, ...$\}$

Comparison

BSc assignment

1. Reactive strategies
   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, `...` $\}$.
2. Game suite
   $G = \{$ `prisoner's dilemma` $\}$.

MSc assignment

1. Adaptive strategies / reply rules
   $S = \{$ `No-regret`, `Fictitious Play`, ϵ`-Greedy`, `Bully`, `...` $\}$

Comparison

BSc assignment

1. Reactive strategies

   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, $\ldots\}$.

2. Game suite
   $G = \{$ `prisoner's dilemma` $\}$.

MSc assignment

1. Adaptive strategies / reply rules

   $S = \{$ `No-regret`, `Fictitious Play`, ϵ-`Greedy`, `Bully`, $\ldots\}$

2. Game suite varies. E.g., $G =$ a set of certain constant sum games $\cup$ a set of certain tragedy of the commons games.

Comparison

BSc assignment

1. Reactive strategies

   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, $\ldots\}$.

2. Game suite

   $G = \{$ `prisoner's dilemma` $\}$.

3. S is tailored to optimal performance on the prisoner's dilemma.

MSc assignment

1. Adaptive strategies / reply rules

   $S = \{$ `No-regret`, `Fictitious Play`, ϵ`-Greedy`, `Bully`, $\ldots\}$

2. Game suite varies. E.g., $G =$ a set of certain constant sum games $\cup$ a set of certain tragedy of the commons games.

Comparison

BSc assignment

1. Reactive strategies

   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, $\ldots\}$.

2. Game suite
   $G = \{$ `prisoner's dilemma` $\}$.

3. S is tailored to optimal performance on the prisoner's dilemma.

MSc assignment

1. Adaptive strategies / reply rules

   $S = \{$ `No-regret`, `Fictitious Play`, ϵ`-Greedy`, `Bully`, $\ldots\}$

2. Game suite varies. E.g., $G =$ a set of certain constant sum games $\cup$ a set of certain tragedy of the commons games.

3. Reply rules are designed to perform well in all games.

Comparison

BSc assignment

1. Reactive strategies

   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, $\dots\}$.

2. Game suite
   $G = \{$ `prisoner's dilemma` $\}$.

3. S is tailored to optimal performance on the prisoner's dilemma.

4. S is empirically tested.

MSc assignment

1. Adaptive strategies / reply rules

   $S = \{$ `No-regret`, `Fictitious Play`, ϵ-`Greedy`, `Bully`, $\dots\}$

2. Game suite varies. E.g., $G =$ a set of certain constant sum games $\cup$ a set of certain tragedy of the commons games.

3. Reply rules are designed to perform well in all games.

Comparison

BSc assignment

1. Reactive strategies
   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, $\dots\}$.
2. Game suite
   $G = \{$ `prisoner's dilemma` $\}$.
3. S is tailored to optimal performance on the prisoner's dilemma.
4. S is empirically tested.

MSc assignment

1. Adaptive strategies / reply rules
   $S = \{$ `No-regret`, `Fictitious Play`, ϵ`-Greedy`, `Bully`, $\dots\}$
2. Game suite varies. E.g., $G =$ a set of certain constant sum games $\cup$ a set of certain tragedy of the commons games.
3. Reply rules are designed to perform well in all games.
4. Elements of S are often theoretically analysed.

Comparison

BSc assignment

1. Reactive strategies
   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, $\dots\}$.
2. Game suite
   $G = \{$ `prisoner's dilemma` $\}$.
3. S is tailored to optimal performance on the prisoner's dilemma.
4. S is empirically tested.
5. Elements of S do not learn. However, if learning is defined as acting on history, then reactive strategies *do* learn.

MSc assignment

1. Adaptive strategies / reply rules
   $S = \{$ `No-regret`, `Fictitious Play`, ϵ-`Greedy`, `Bully`, $\dots\}$
2. Game suite varies. E.g., $G =$ a set of certain constant sum games $\cup$ a set of certain tragedy of the commons games.
3. Reply rules are designed to perform well in all games.
4. Elements of S are often theoretically analysed.

Comparison

BSc assignment

1. Reactive strategies

   $S = \{$ `All-C`, `All-D`, `Tit-for-tat`, `Pavlov`, `Eatherly`, `...` $\}$.

2. Game suite

   $G = \{$ `prisoner's dilemma` $\}$.

3. S is tailored to optimal performance on the prisoner's dilemma.

4. S is empirically tested.

5. Elements of S do not learn. However, if learning is defined as acting on history, then reactive strategies *do* learn.

MSc assignment

1. Adaptive strategies / reply rules

   $S = \{$ `No-regret`, `Fictitious Play`, ϵ-`Greedy`, `Bully`, `...` $\}$

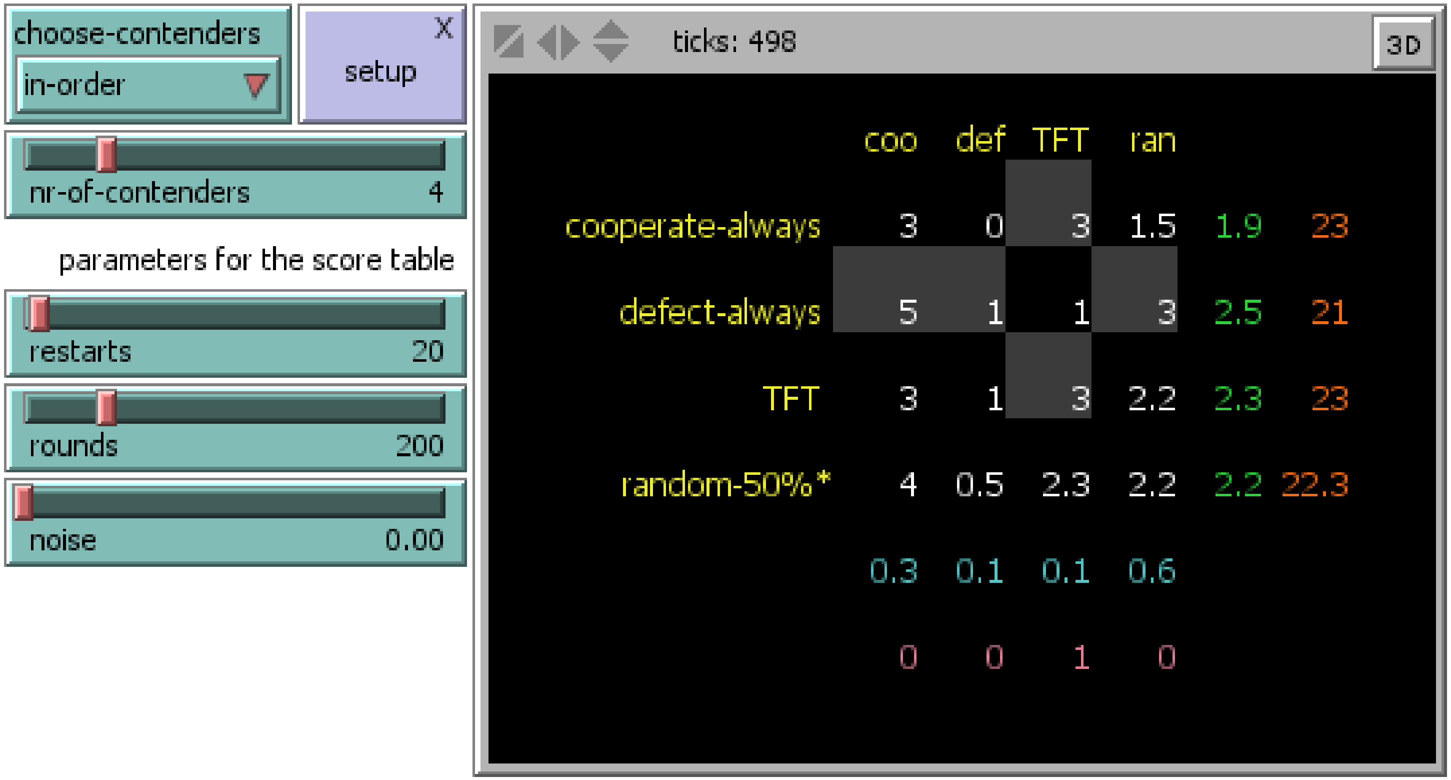
2. Game suite varies. E.g., $G =$ a set of certain constant sum games $\cup$ a set of certain tragedy of the commons games.

3. Reply rules are designed to perform well in all games.

4. Elements of S are often theoretically analysed.

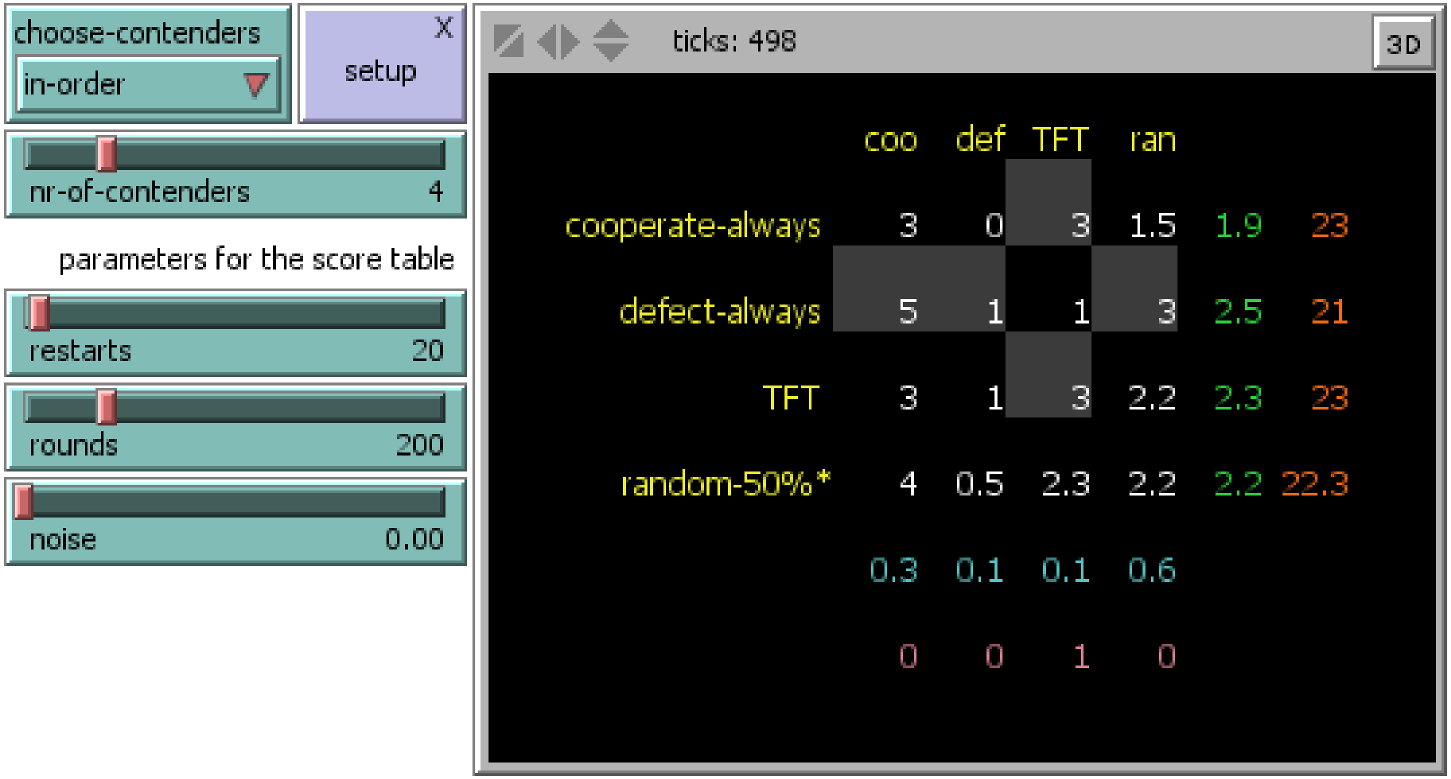
5. Elements of S do learn. `Bully`, however, does not.

Partial screenshot of BSc app



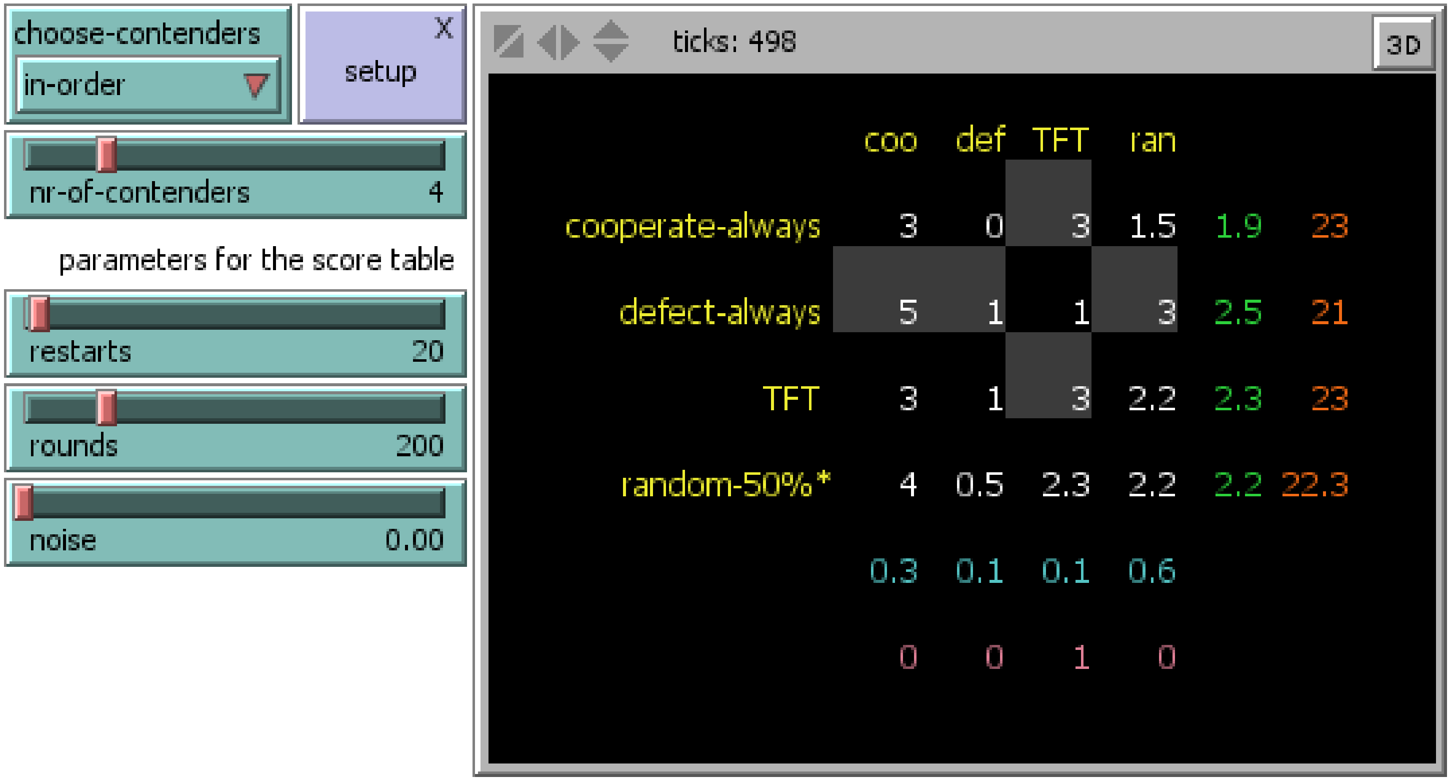
You are looking at a grand table

Partial screenshot of BSc app



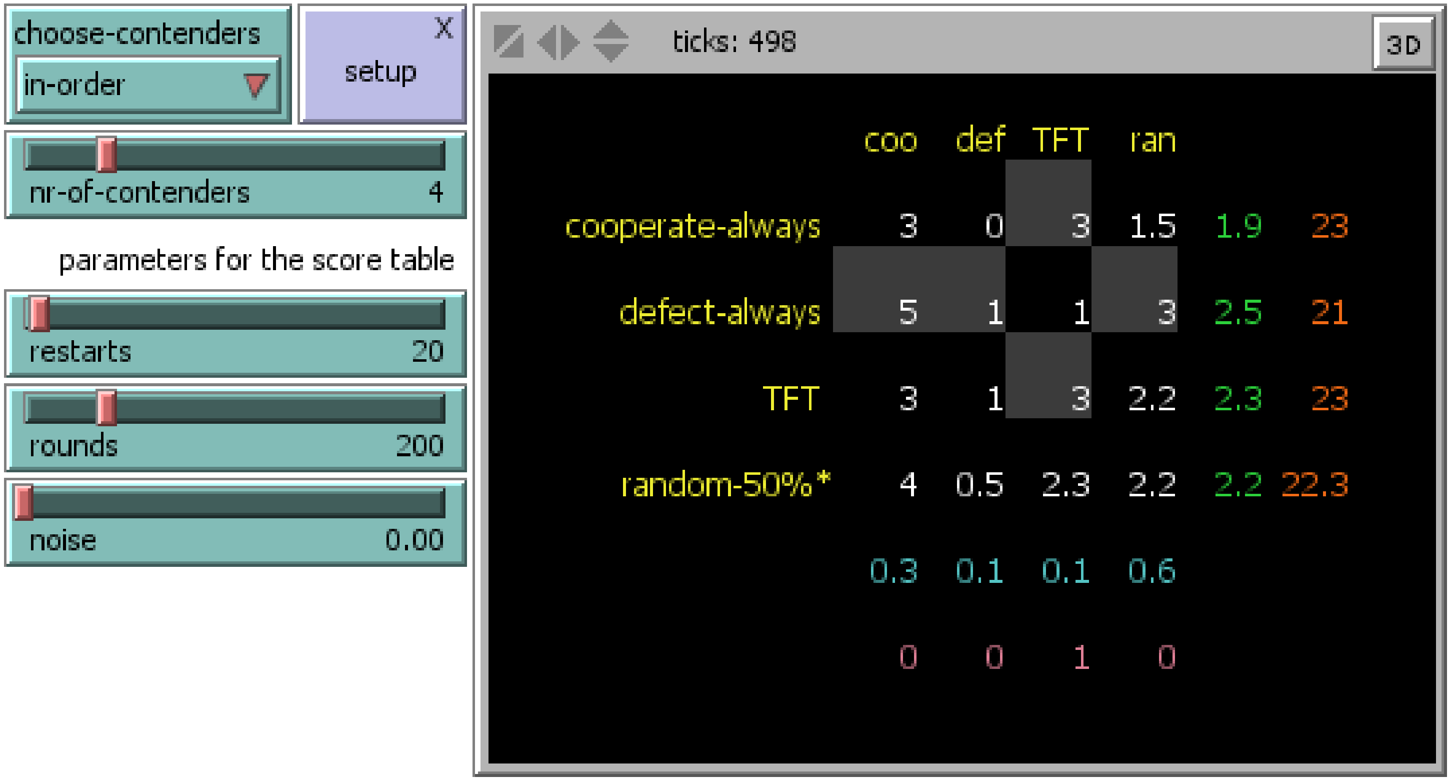
You are looking at a grand table: an overview of the average scores of repeated pairwise encounters.

Partial screenshot of BSc app



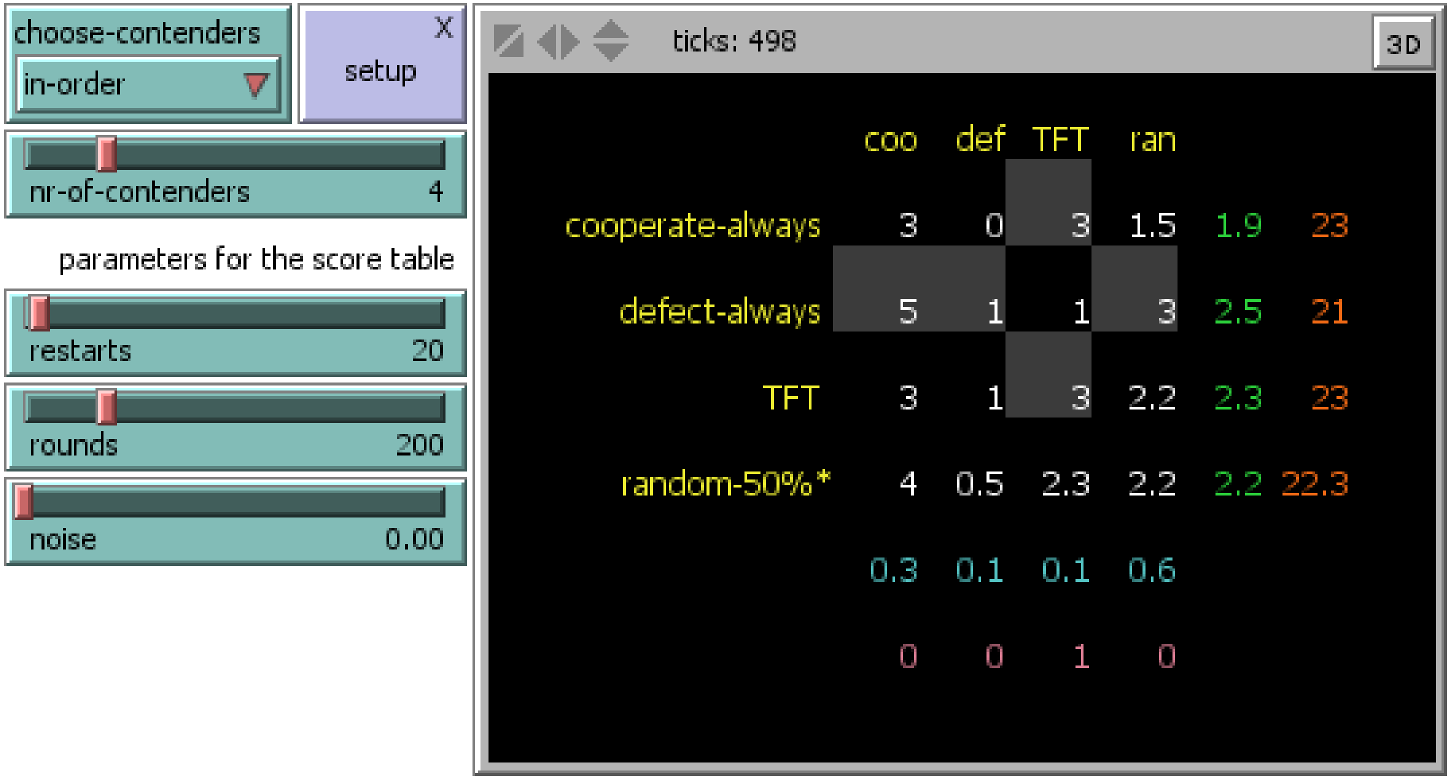
You are looking at a grand table: an overview of the average scores of repeated pairwise encounters. First row: scores of `cooperate-always` against each other individual strategy in 50 encounters of 200 rounds.

Partial screenshot of BSc app



You are looking at a grand table: an overview of the average scores of repeated pairwise encounters. First row: scores of `cooperate-always` against each other individual strategy in 50 encounters of 200 rounds. Restarts are necessary because random-50% is stochastic.

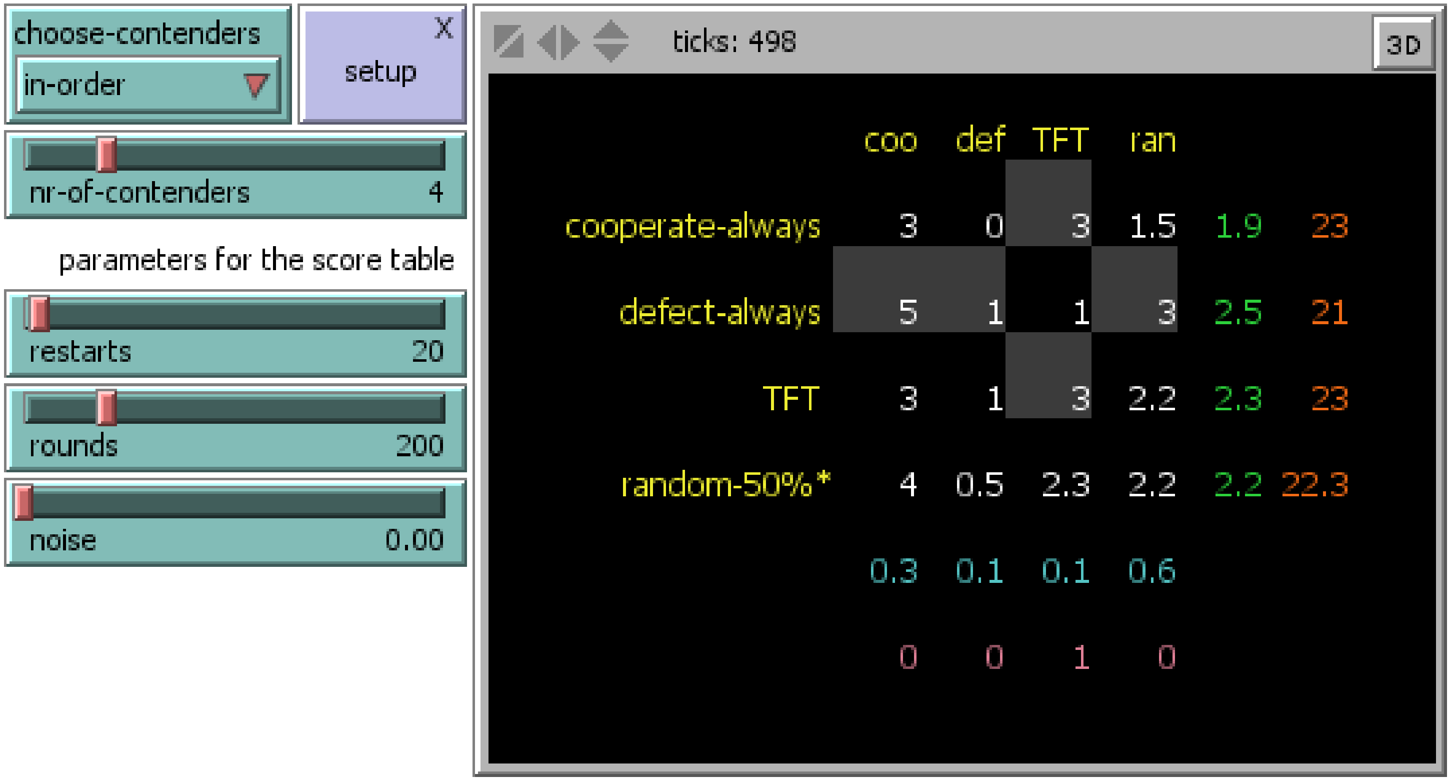
Partial screenshot of BSc app



You are looking at a grand table: an overview of the average scores of repeated pairwise encounters. First row: scores of `cooperate-always` against each other individual strategy in 50 encounters of 200 rounds. Restarts are necessary because random-50% is stochastic.

Green is average score. Orange, blue and pink are related to the replicator dynamic.

Partial screenshot of BSc app

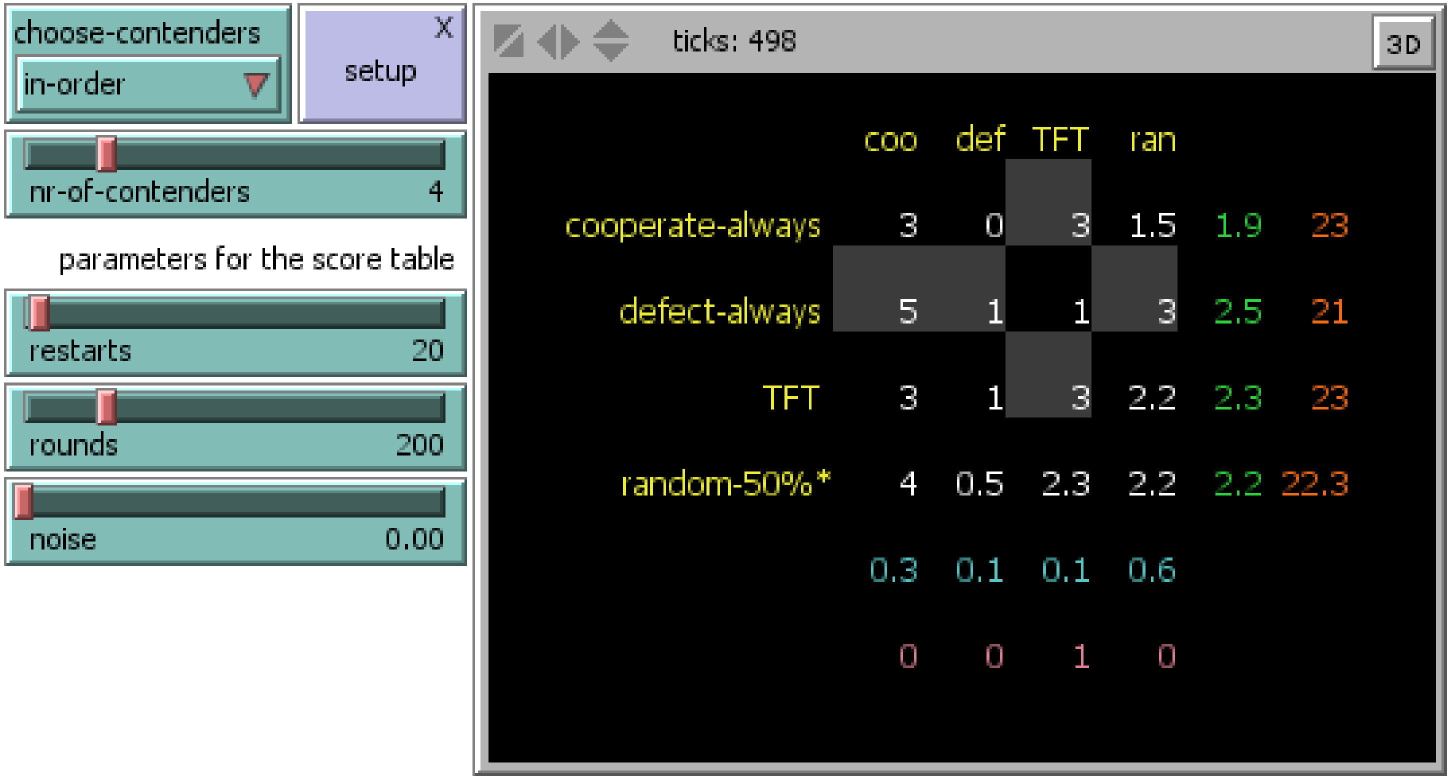


You are looking at a grand table: an overview of the average scores of repeated pairwise encounters. First row: scores of `cooperate-always` against each other individual strategy in 50 encounters of 200 rounds. Restarts are necessary because random-50% is stochastic.

Green is average score. Orange, blue and pink are related to the replicator dynamic.

Starred strategies are stochastic.

Partial screenshot of BSc app

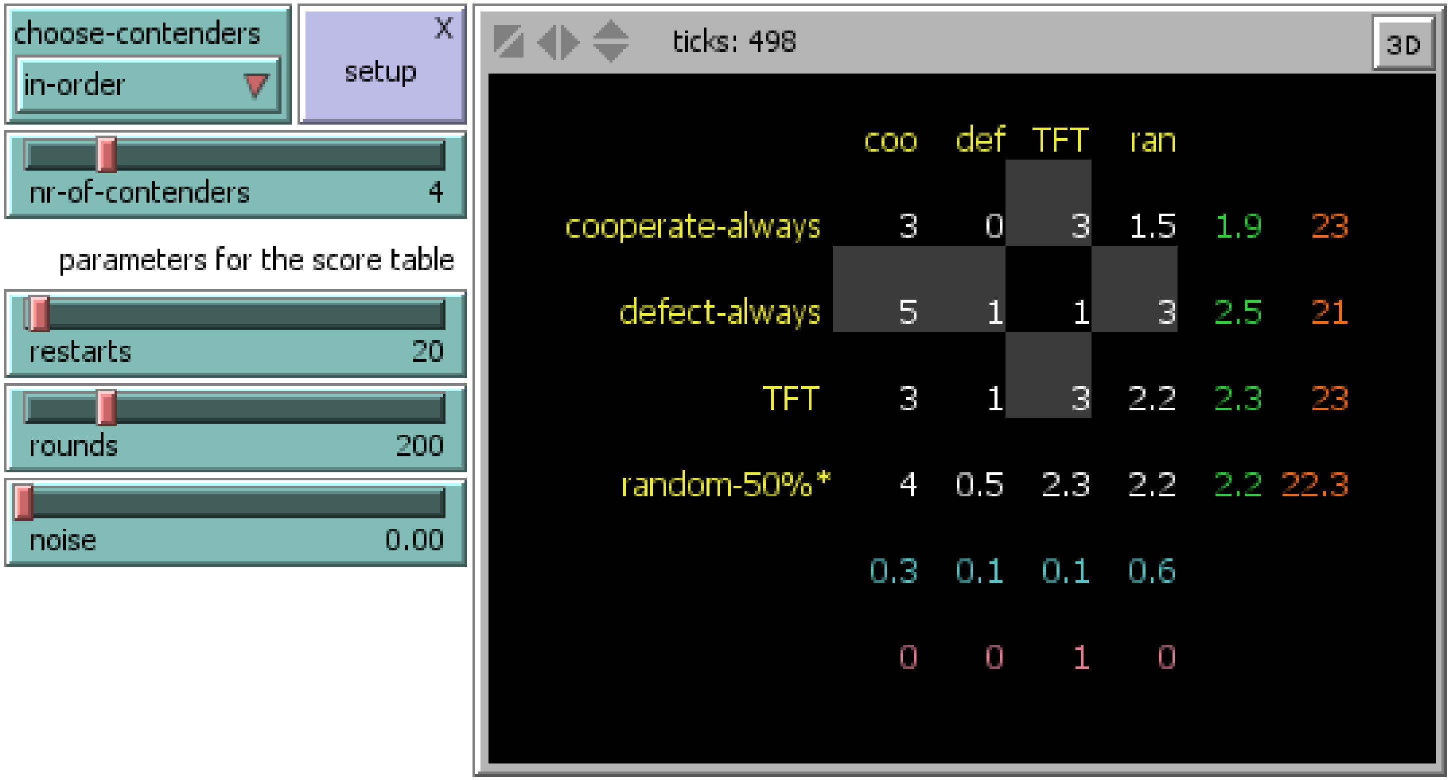


You are looking at a grand table: an overview of the average scores of repeated pairwise encounters. First row: scores of `cooperate-always` against each other individual strategy in 50 encounters of 200 rounds. Restarts are necessary because random-50% is stochastic.

Green is average score. Orange, blue and pink are related to the replicator dynamic.

Starred strategies are stochastic. Gray squares denote best (row) counter-strateg{y | ies} to column strategy.

Partial screenshot of BSc app

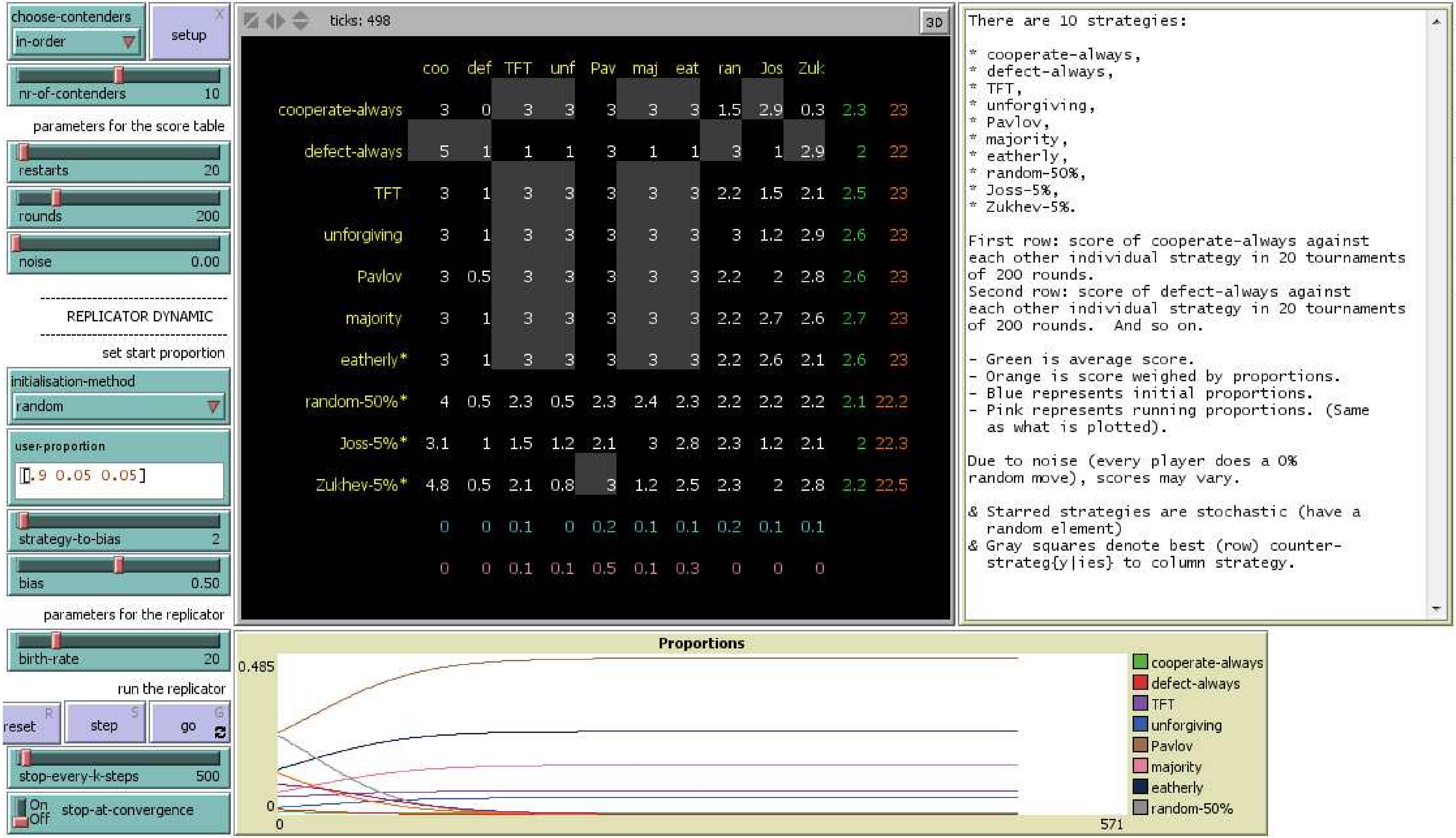


You are looking at a grand table: an overview of the average scores of repeated pairwise encounters. First row: scores of `cooperate-always` against each other individual strategy in 50 encounters of 200 rounds. Restarts are necessary because random-50% is stochastic.

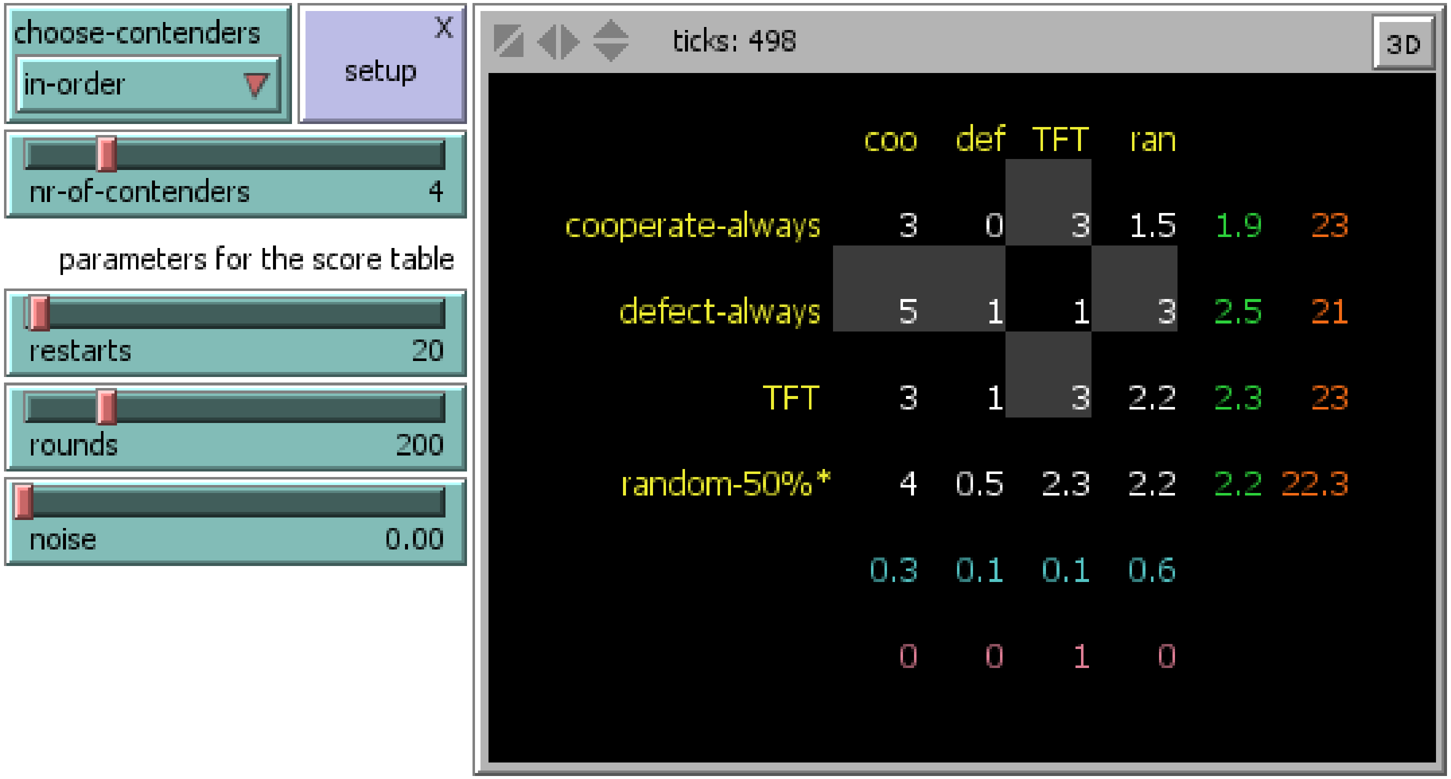
Green is average score. Orange, blue and pink are related to the replicator dynamic.

Starred strategies are stochastic. Gray squares denote best (row) counter-strateg{y | ies} to column strategy. For example, `defect-always` performs best against `cooperate-always`.

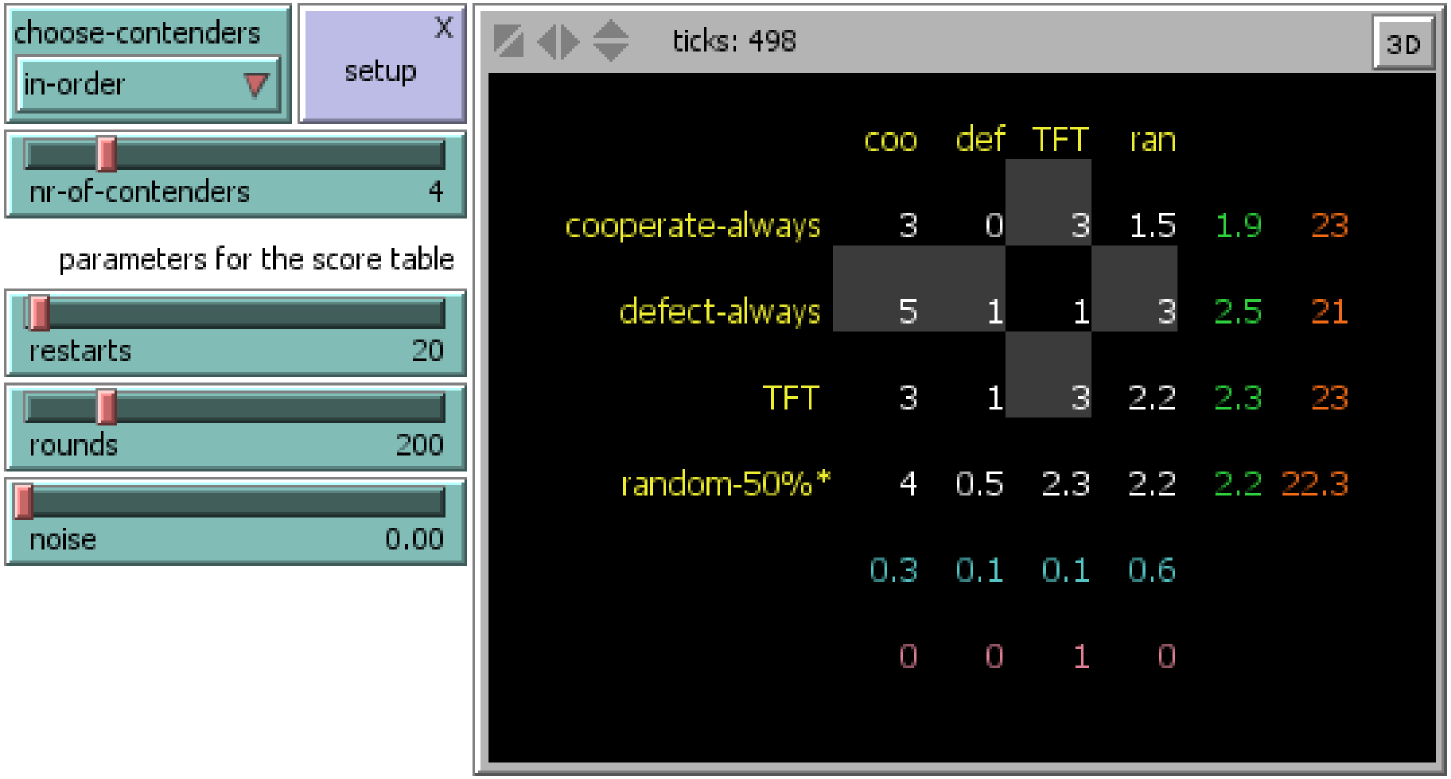
Full screenshot of BSc app



Back to the partial screenshot of BSc app

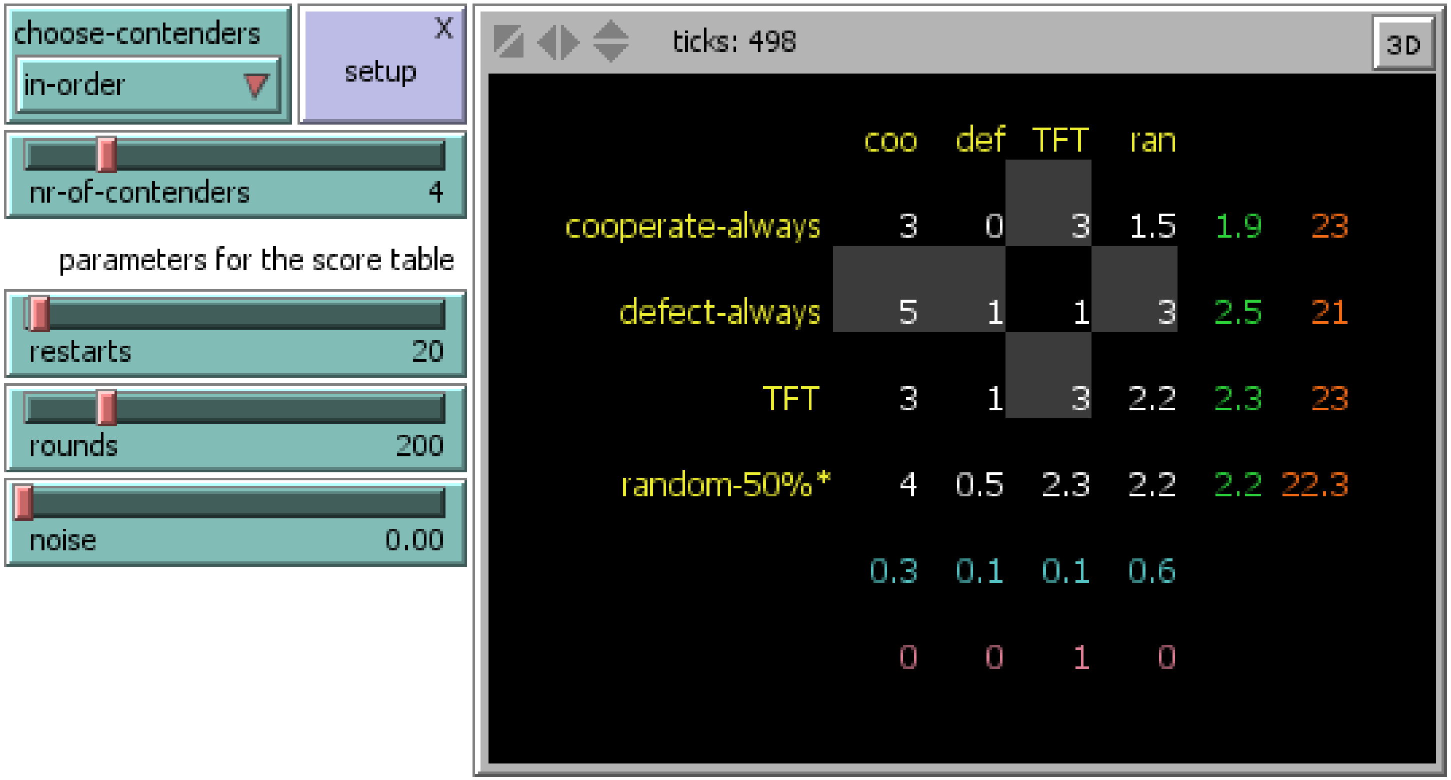


Back to the partial screenshot of BSc app



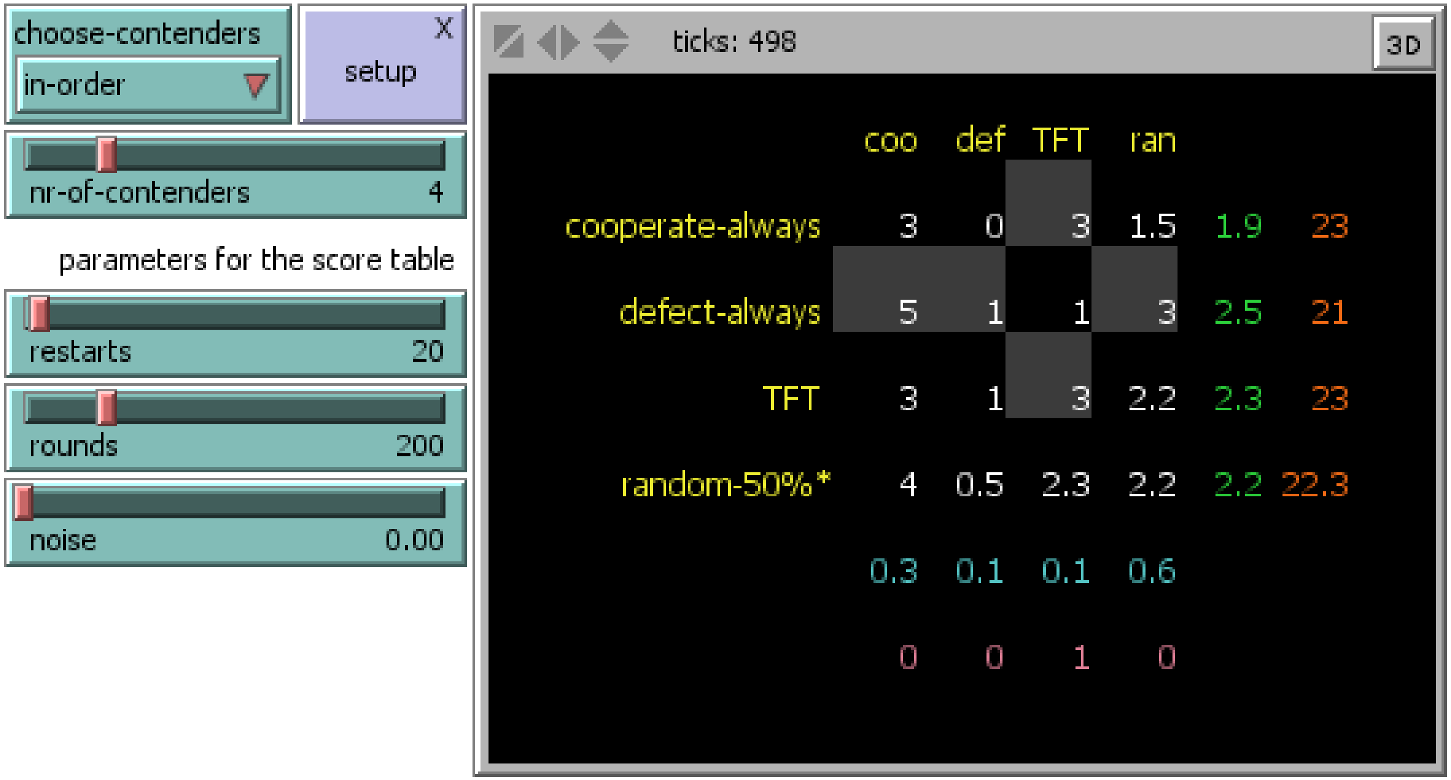
- The strategy `defect-always` seems to perform best (see row averages, in green).

Back to the partial screenshot of BSc app



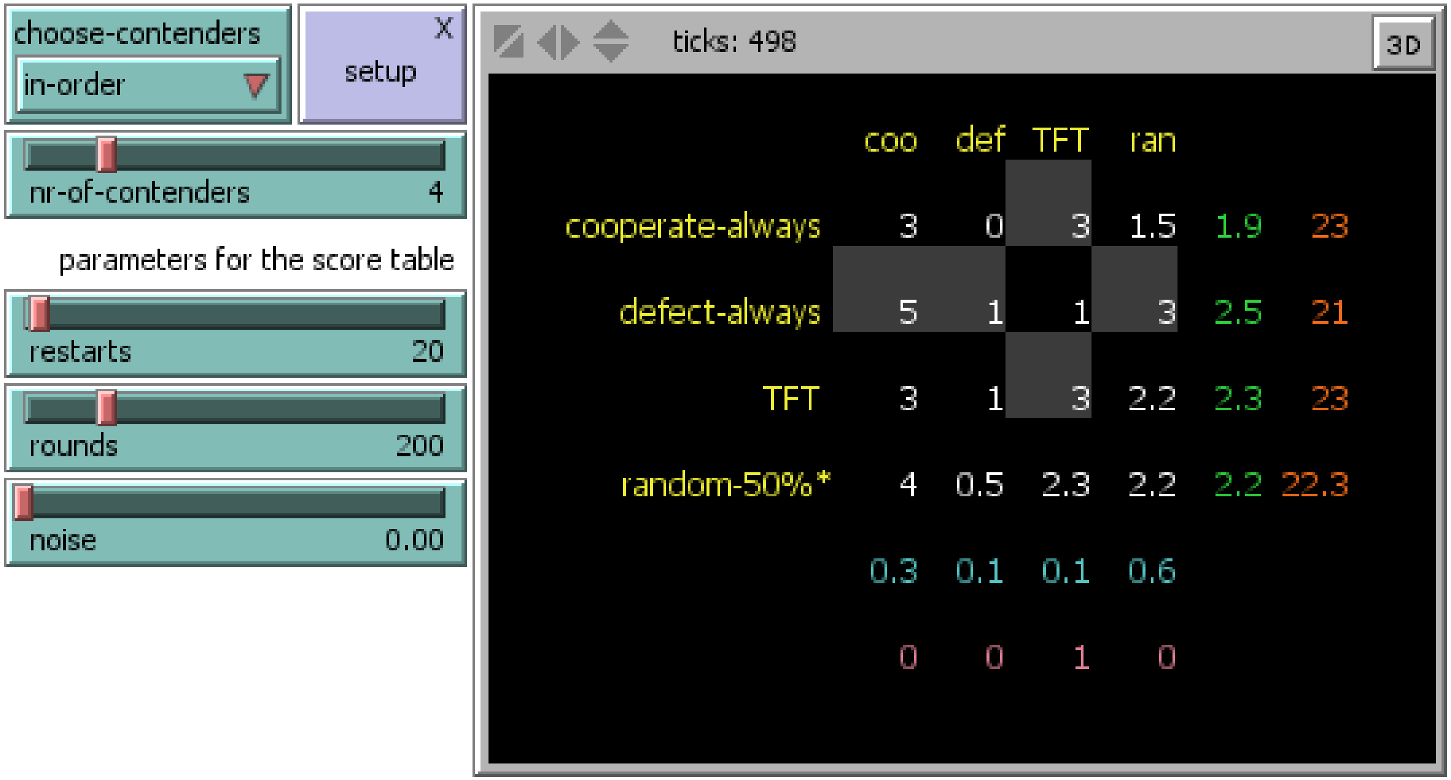
- The strategy `defect-always` seems to perform best (see row averages, in green).
- This is a premature conclusion!

Back to the partial screenshot of BSc app



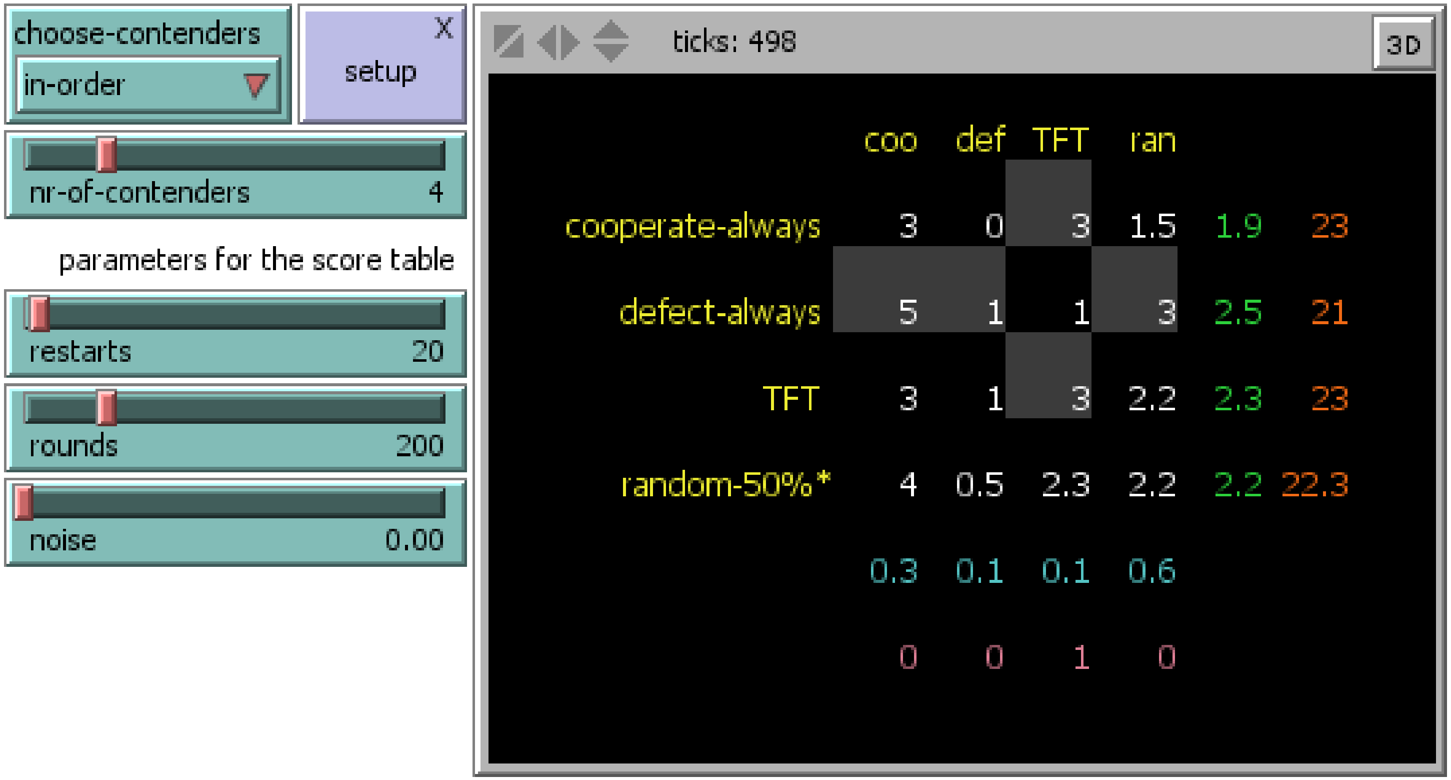
- The strategy `defect-always` seems to perform best (see row averages, in green).
- This is a premature conclusion! `defect-always` performs well because it exploits `cooperate-always`.

Back to the partial screenshot of BSc app



- The strategy `defect-always` seems to perform best (see row averages, in green).
- This is a premature conclusion! `defect-always` performs well because it exploits `cooperate-always`. Because `cooperate-always` performs poorly against almost all competitors, it should actually be taken out of the competition.

Back to the partial screenshot of BSc app



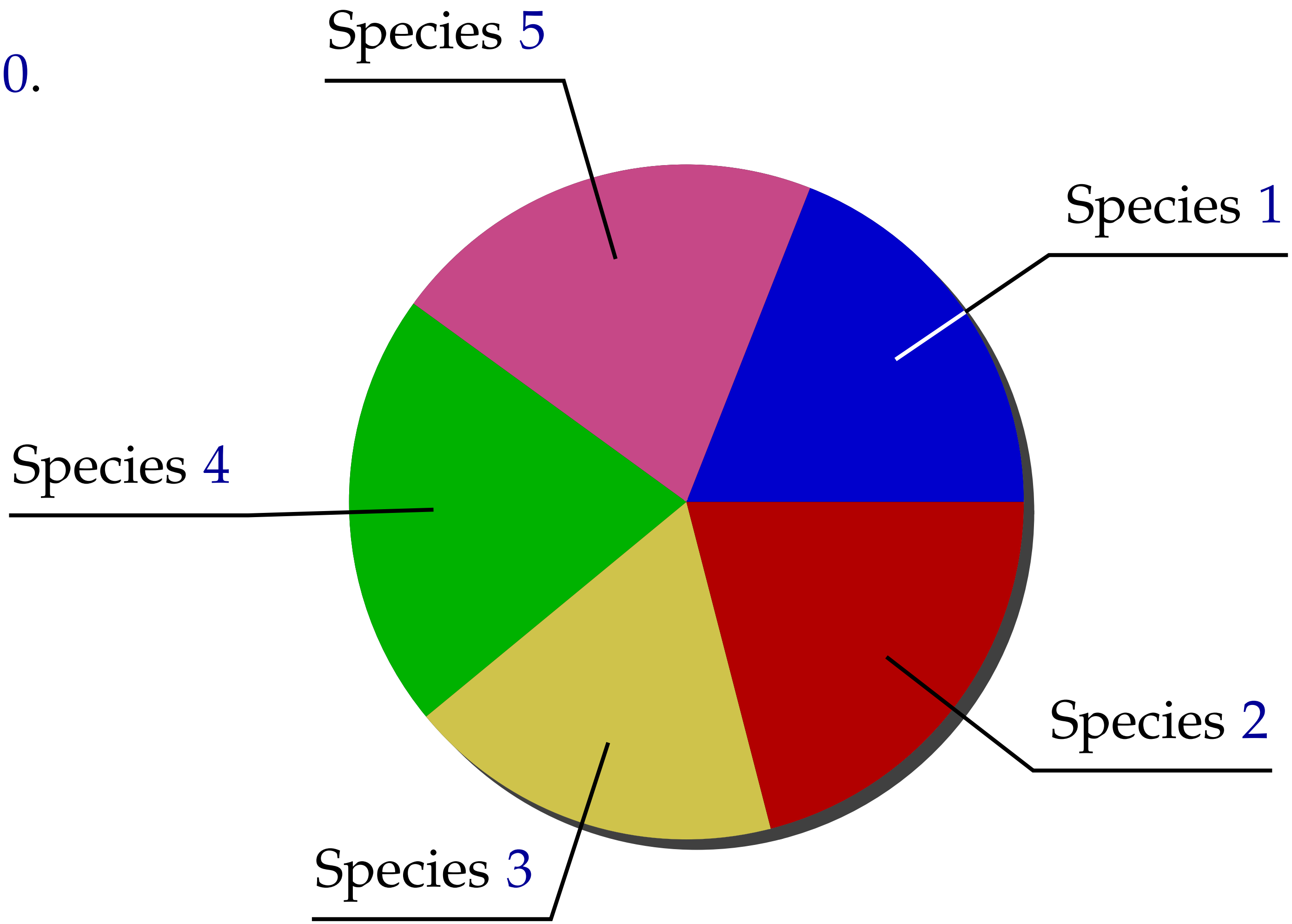
- The strategy `defect-always` seems to perform best (see row averages, in green).
- This is a premature conclusion! `defect-always` performs well because it exploits `cooperate-always`. Because `cooperate-always` performs poorly against almost all competitors, it should actually be taken out of the competition.

  But then `tit-for-tat` suddenly performs nearly as well as `defect-always`!

The replicator dynamic

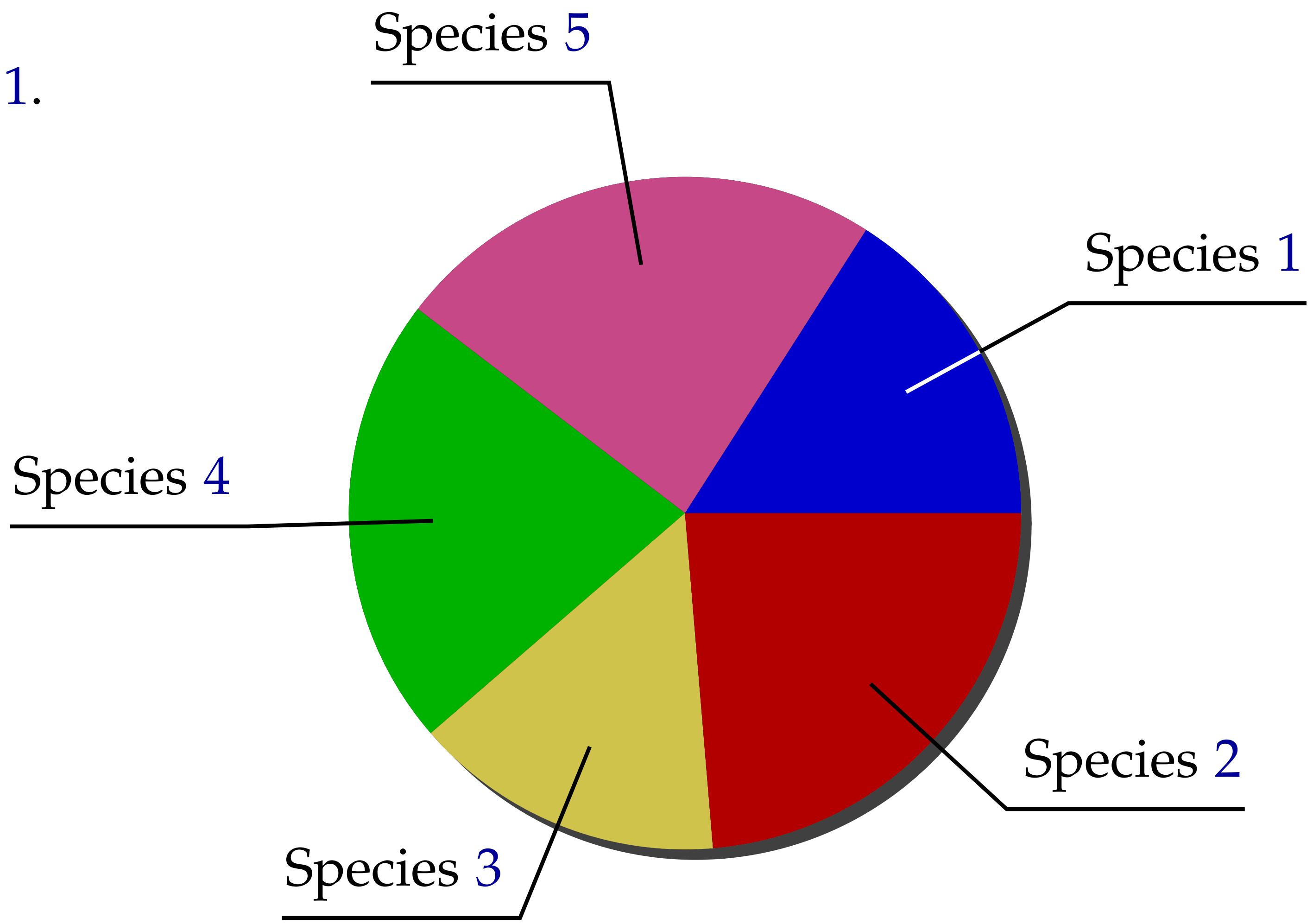
Intuitive idea of the replicator dynamic

Proportions at $t = 0$.



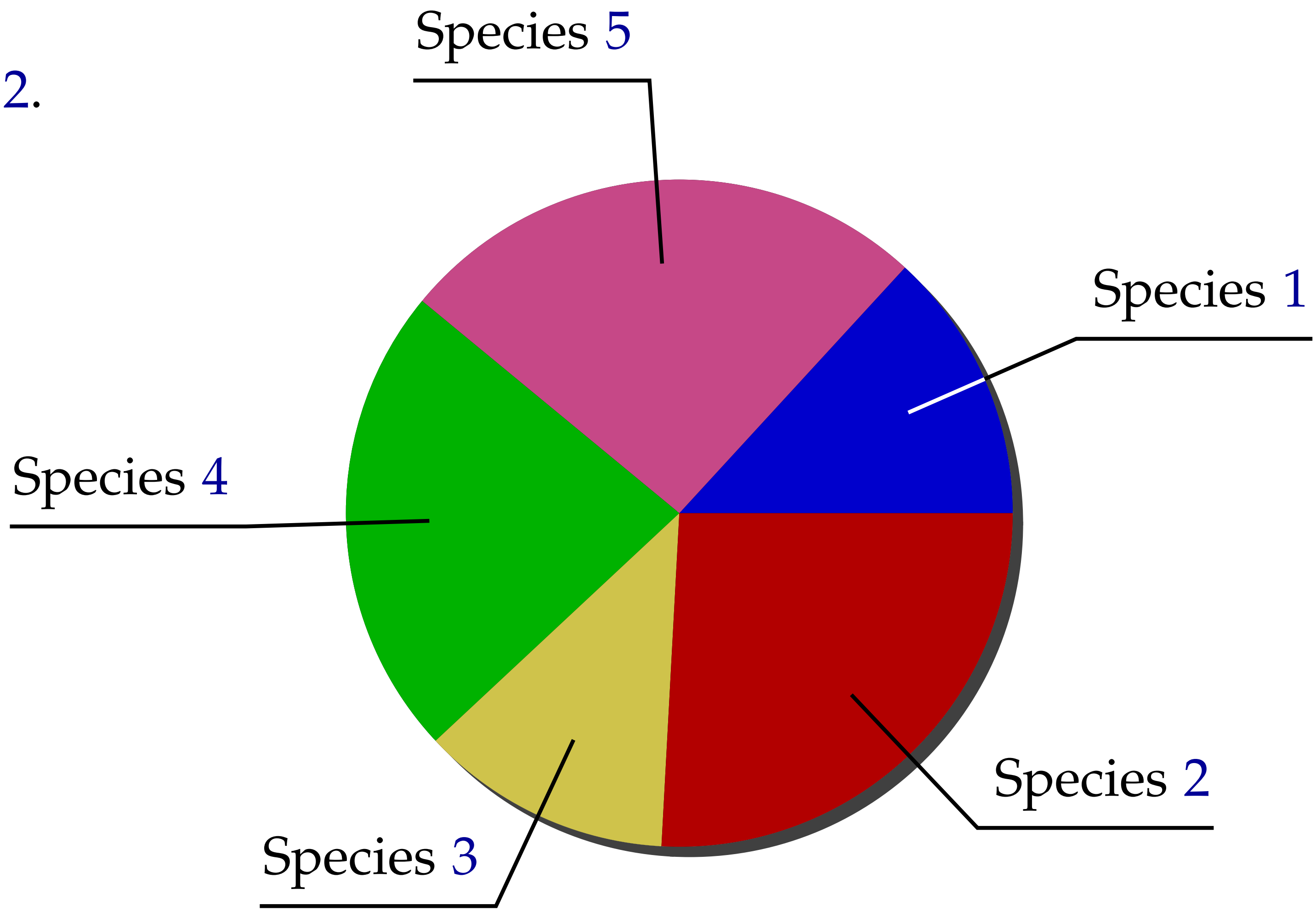
Intuitive idea of the replicator dynamic

Proportions at $t = 1$.



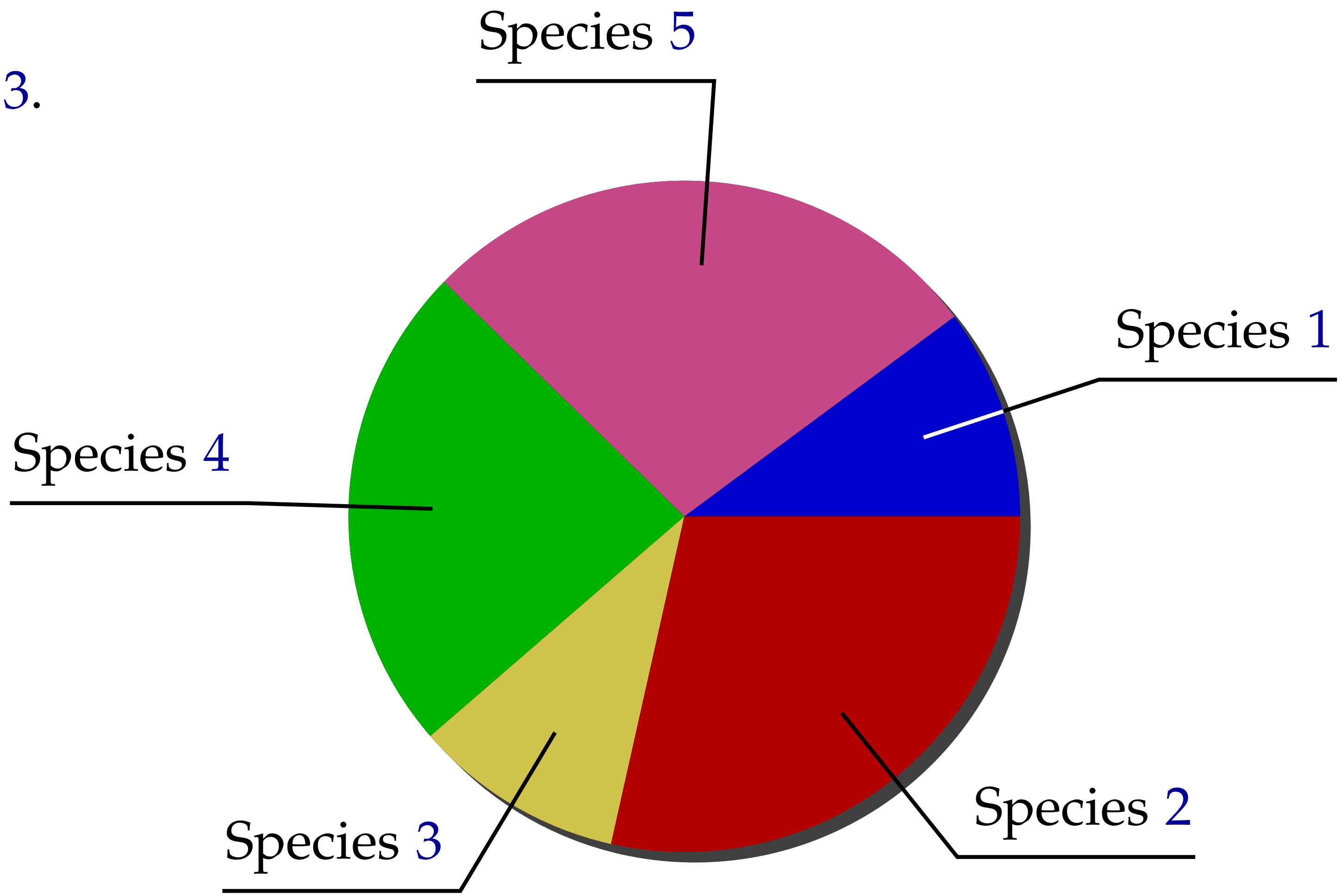
Intuitive idea of the replicator dynamic

Proportions at $t = 2$.



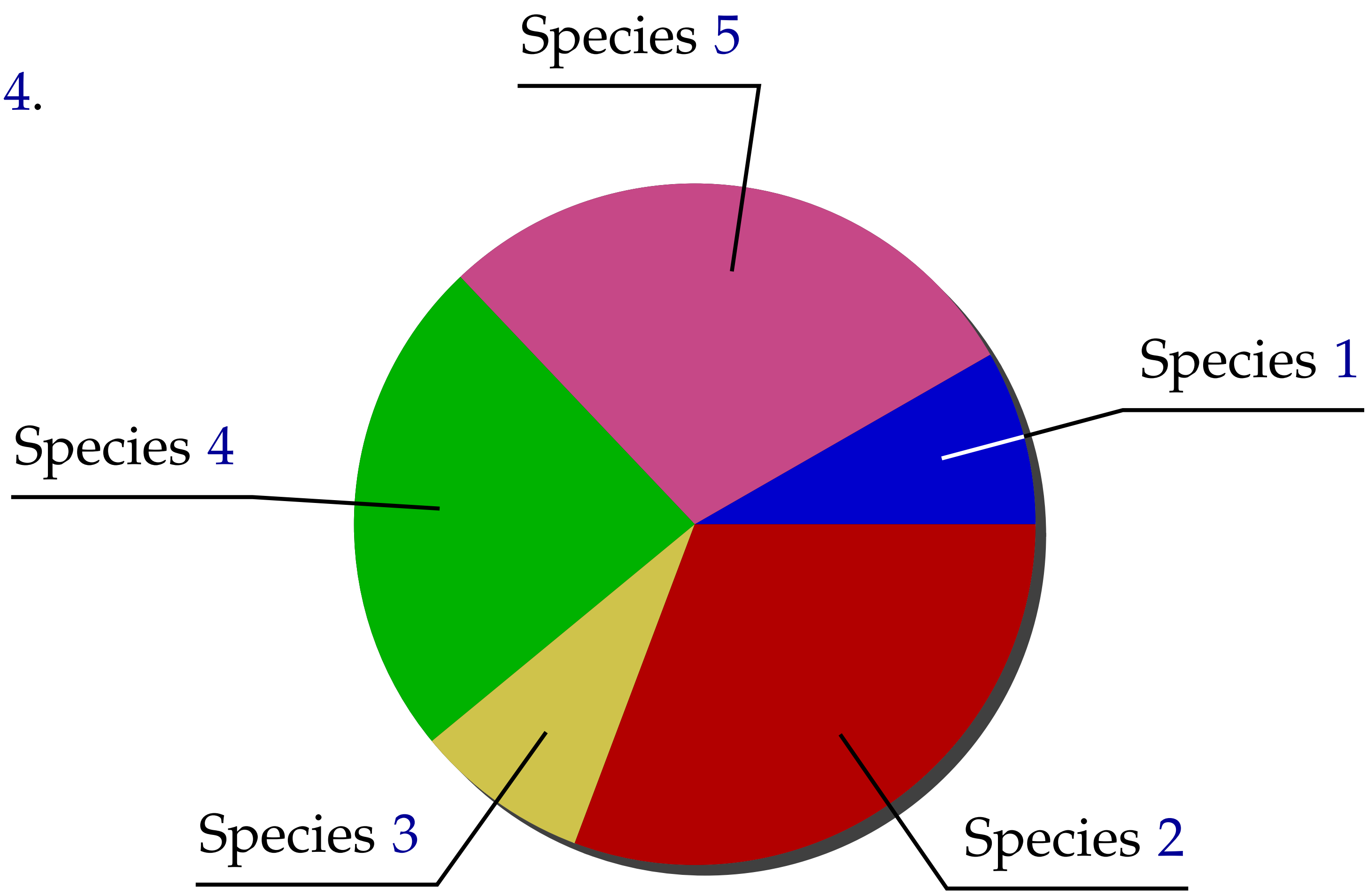
Intuitive idea of the replicator dynamic

Proportions at $t = 3$.



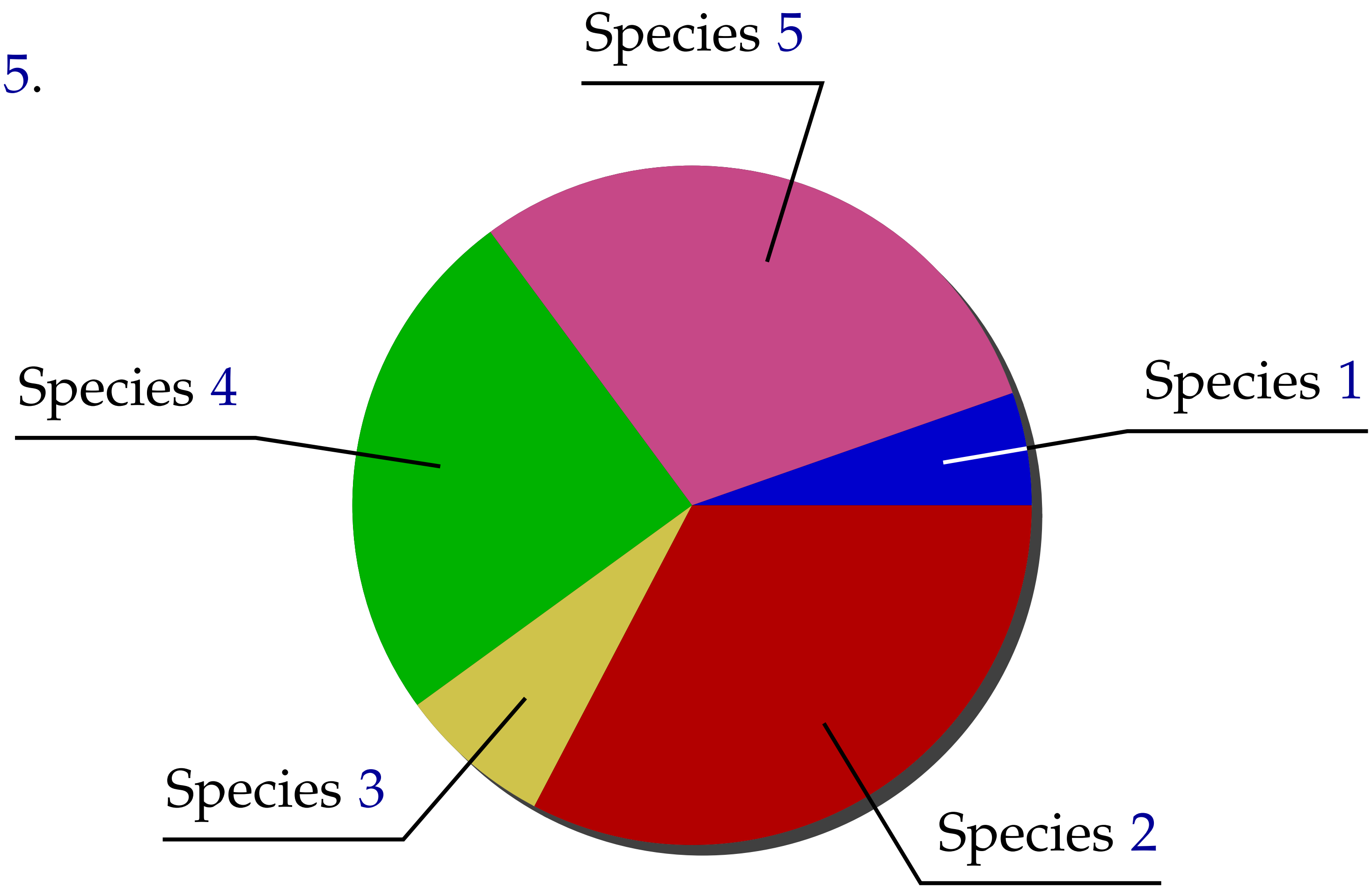
Intuitive idea of the replicator dynamic

Proportions at $t = 4$.



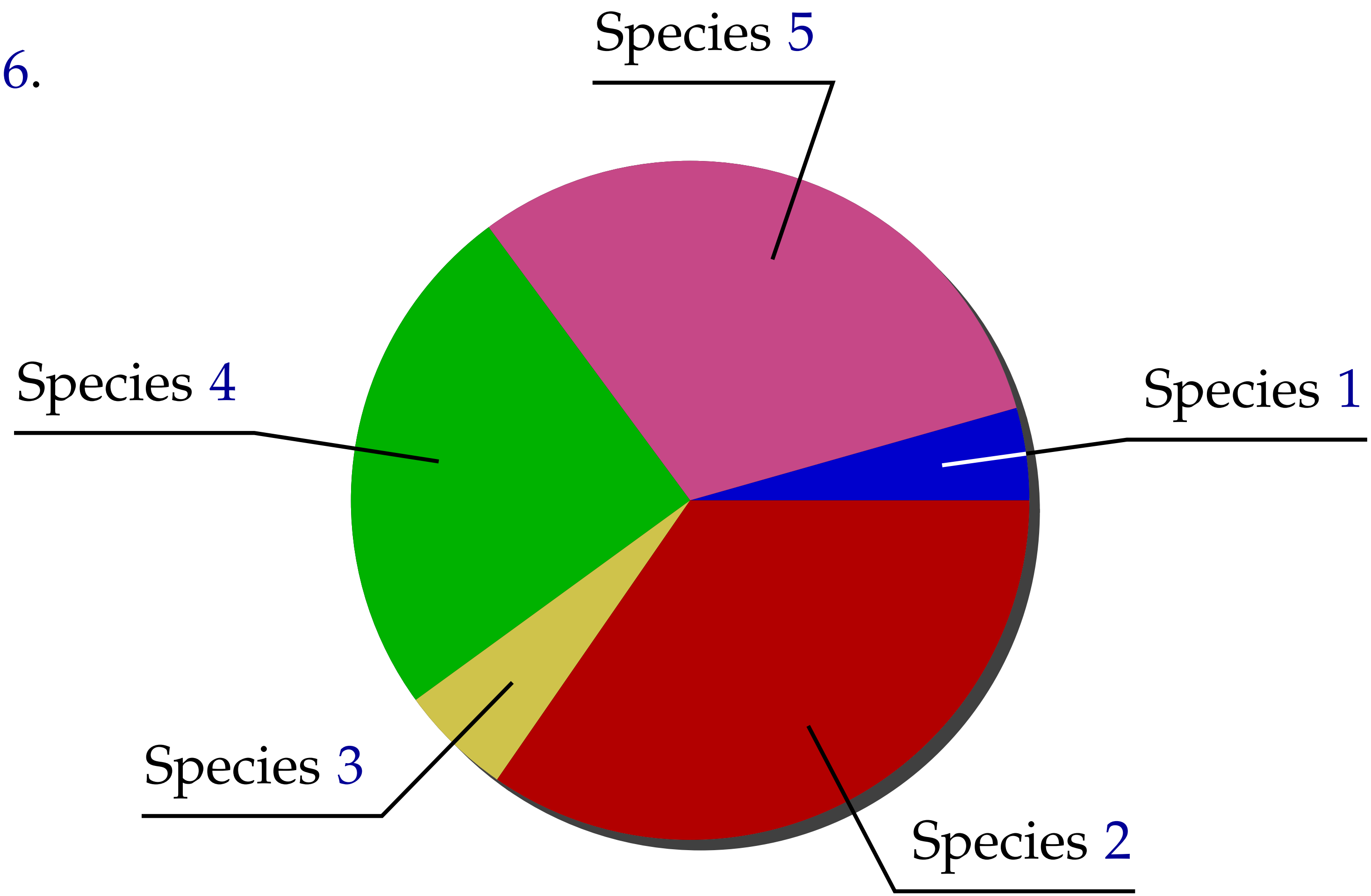
Intuitive idea of the replicator dynamic

Proportions at $t = 5$.



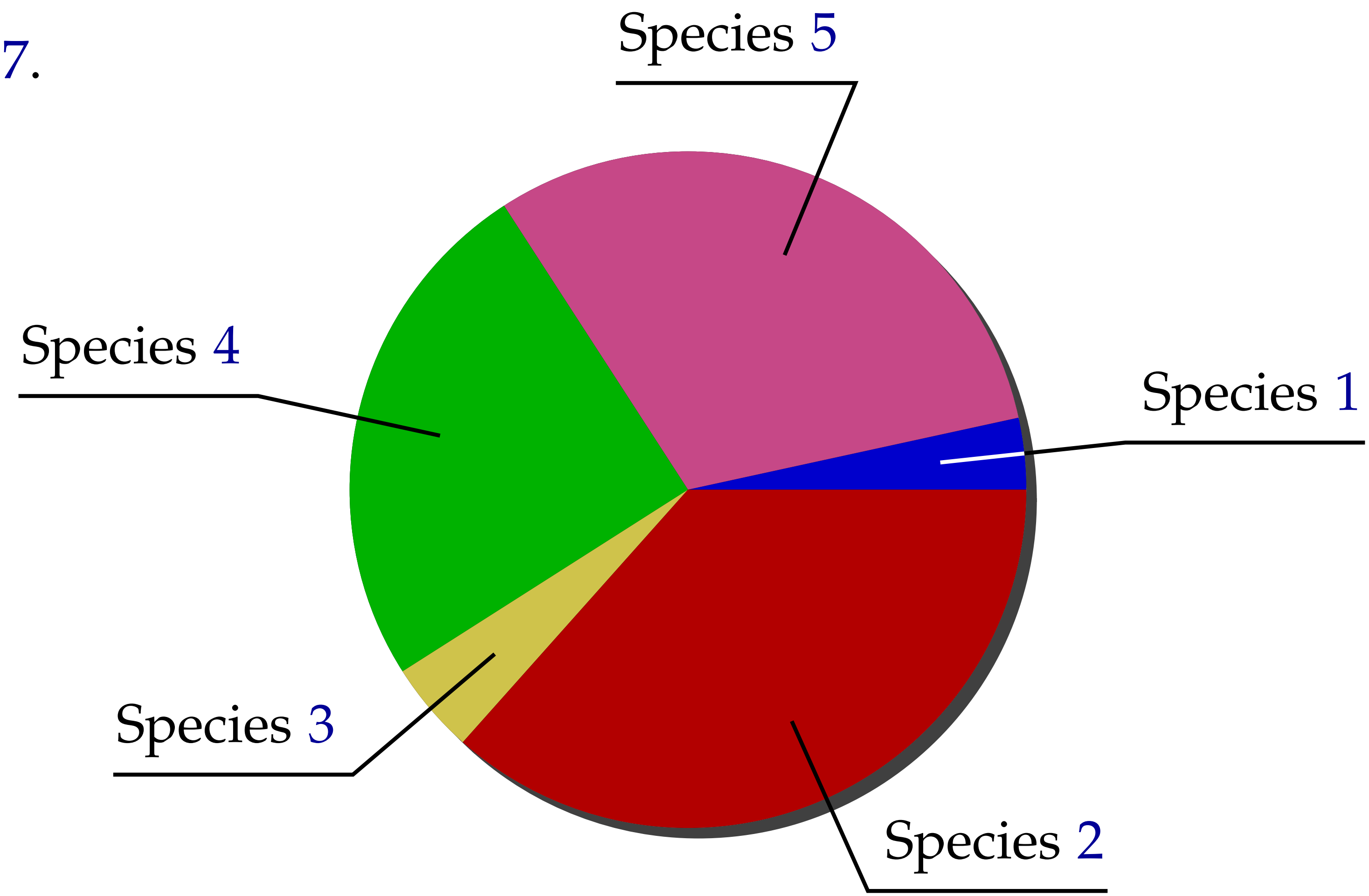
Intuitive idea of the replicator dynamic

Proportions at $t = 6$.



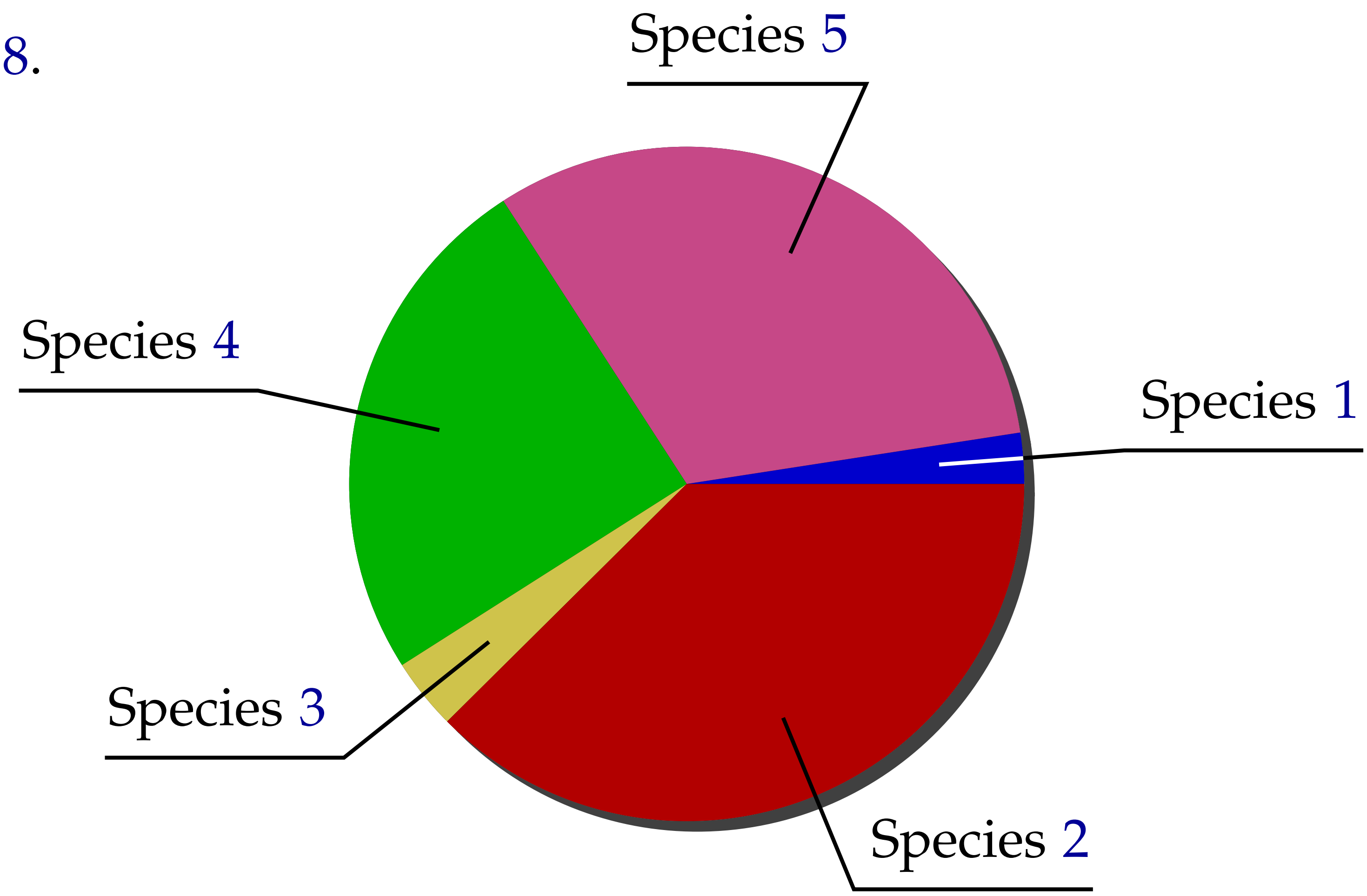
Intuitive idea of the replicator dynamic

Proportions at $t = 7$.



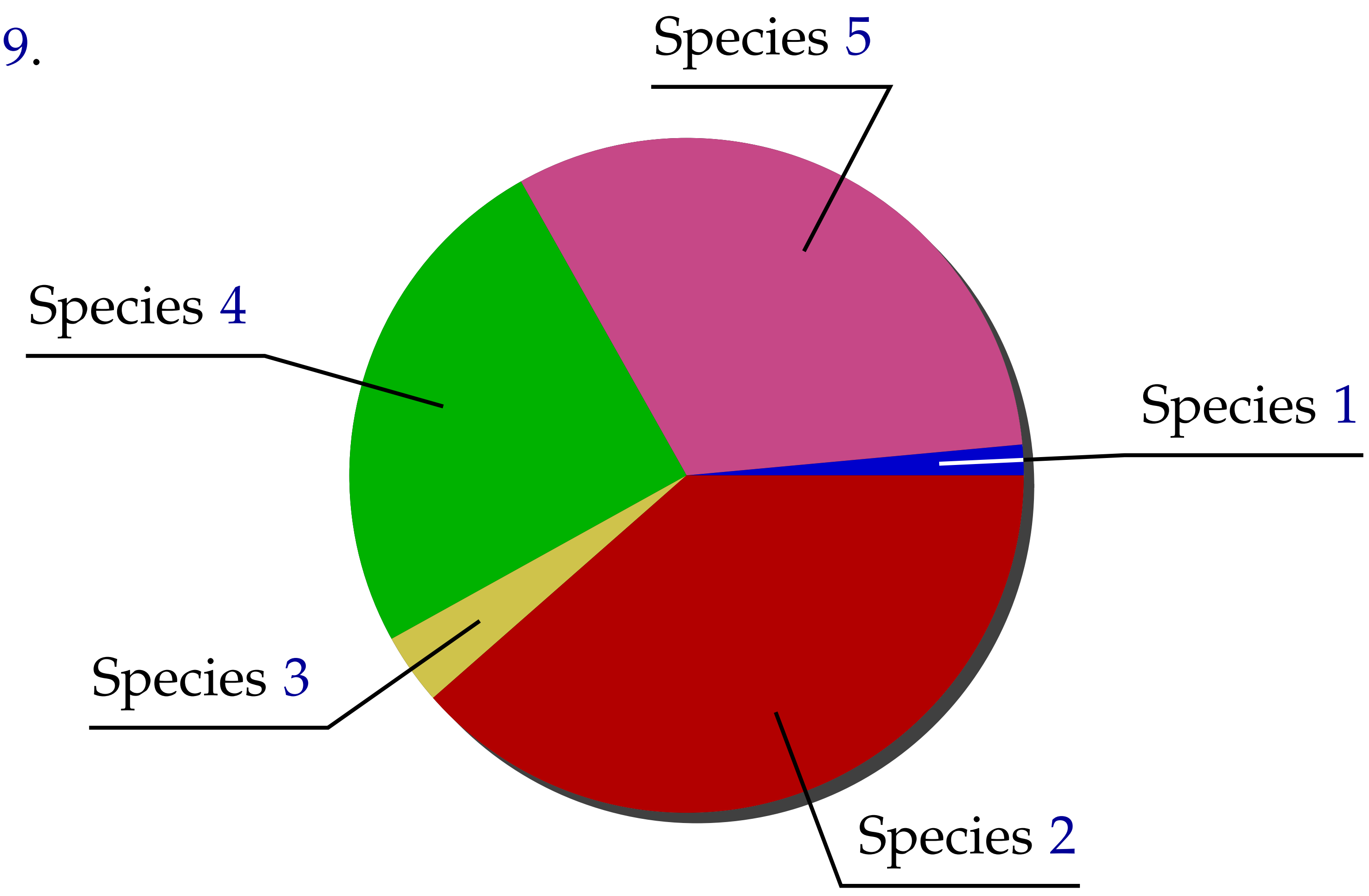
Intuitive idea of the replicator dynamic

Proportions at $t = 8$.



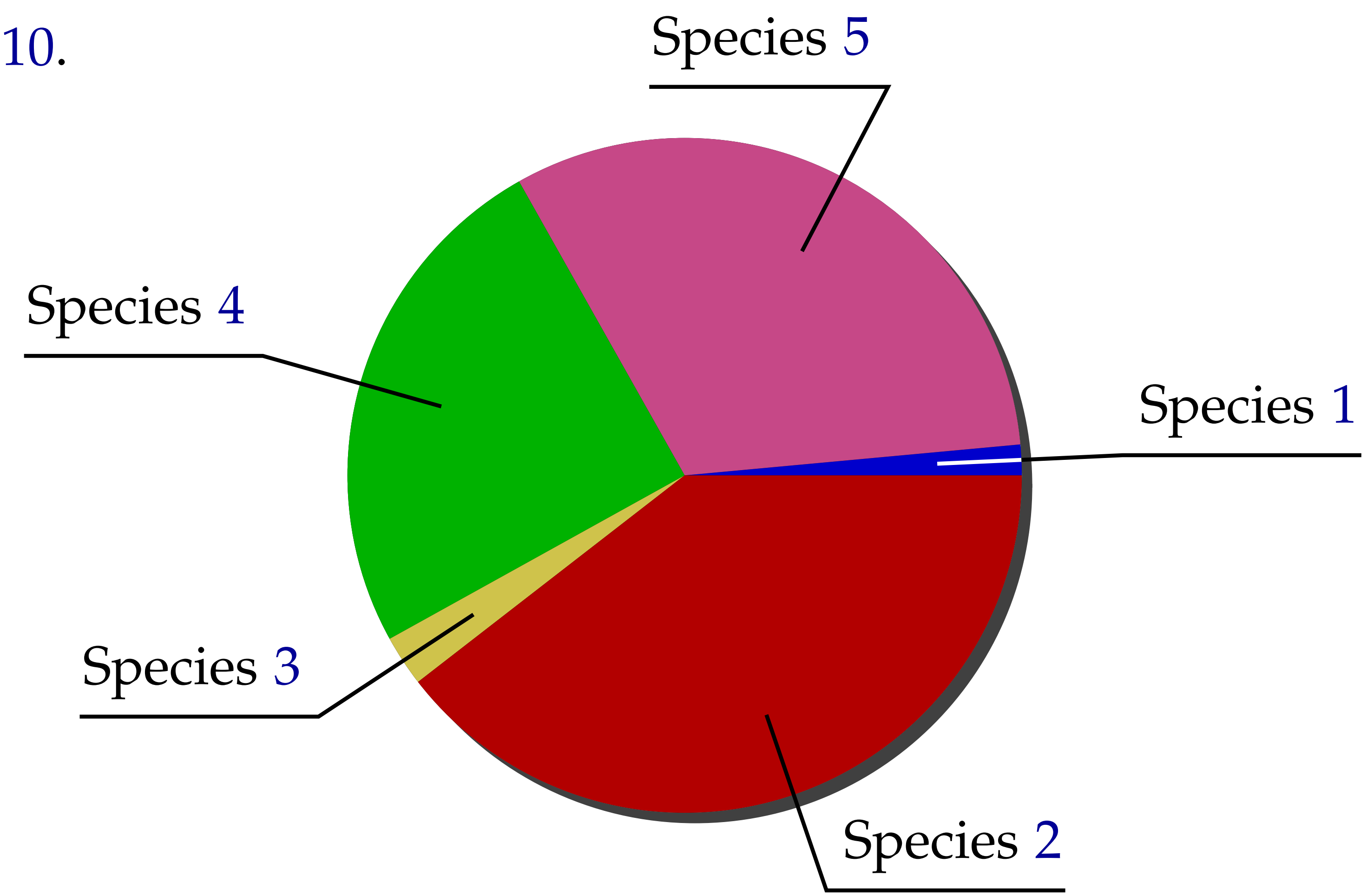
Intuitive idea of the replicator dynamic

Proportions at $t = 9$.



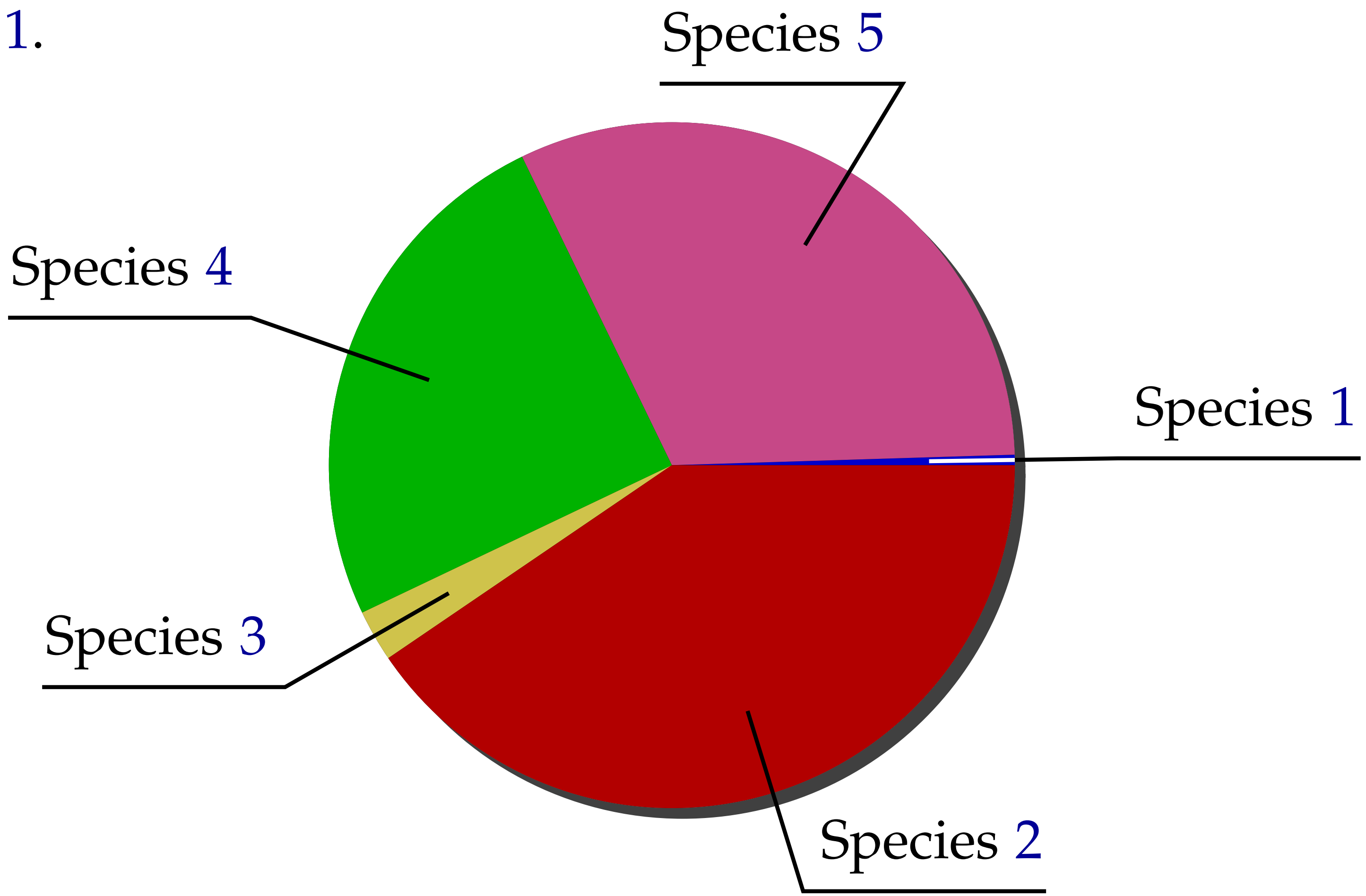
Intuitive idea of the replicator dynamic

Proportions at $t = 10$.



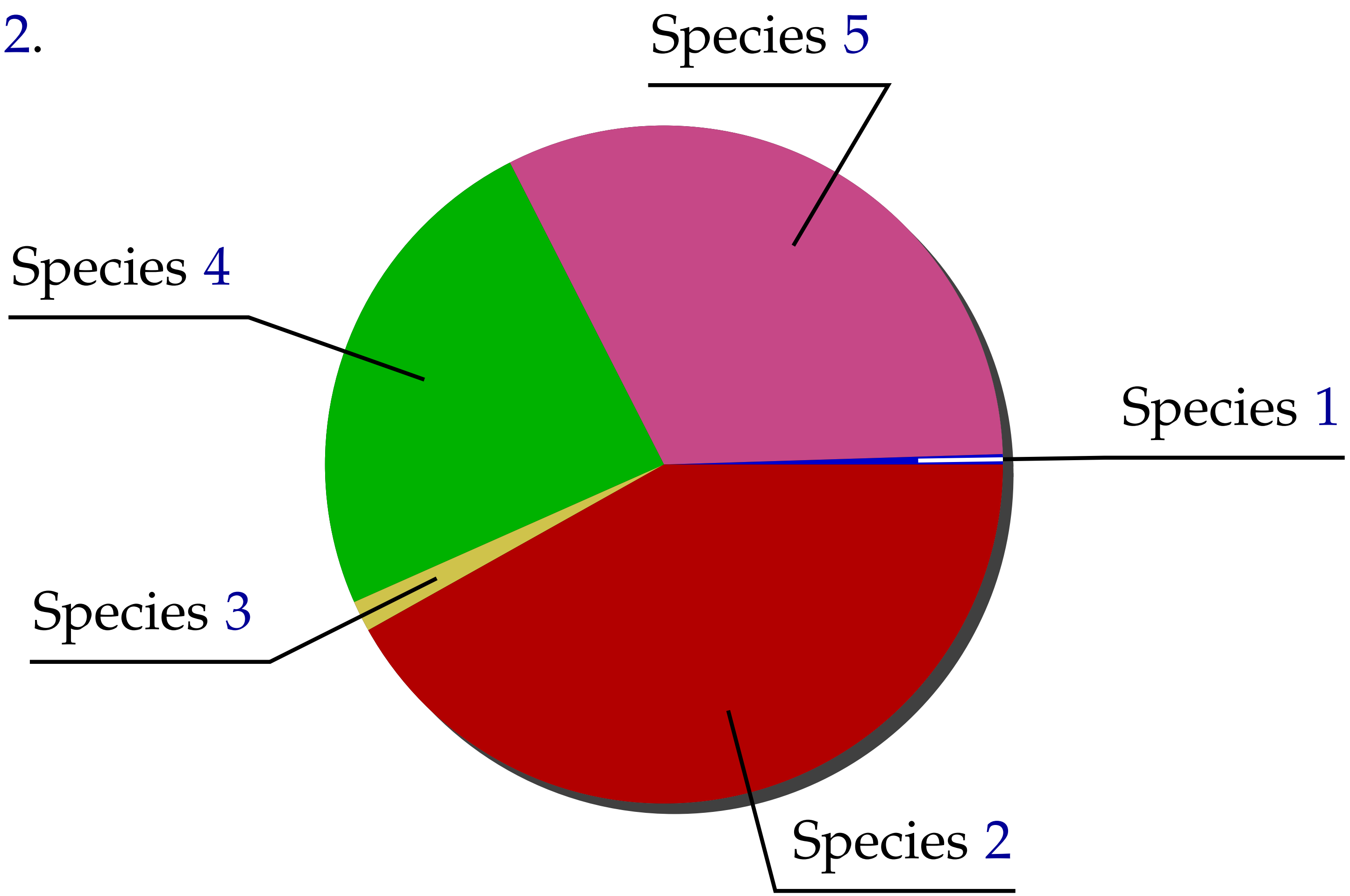
Intuitive idea of the replicator dynamic

Proportions at $t = 11$.



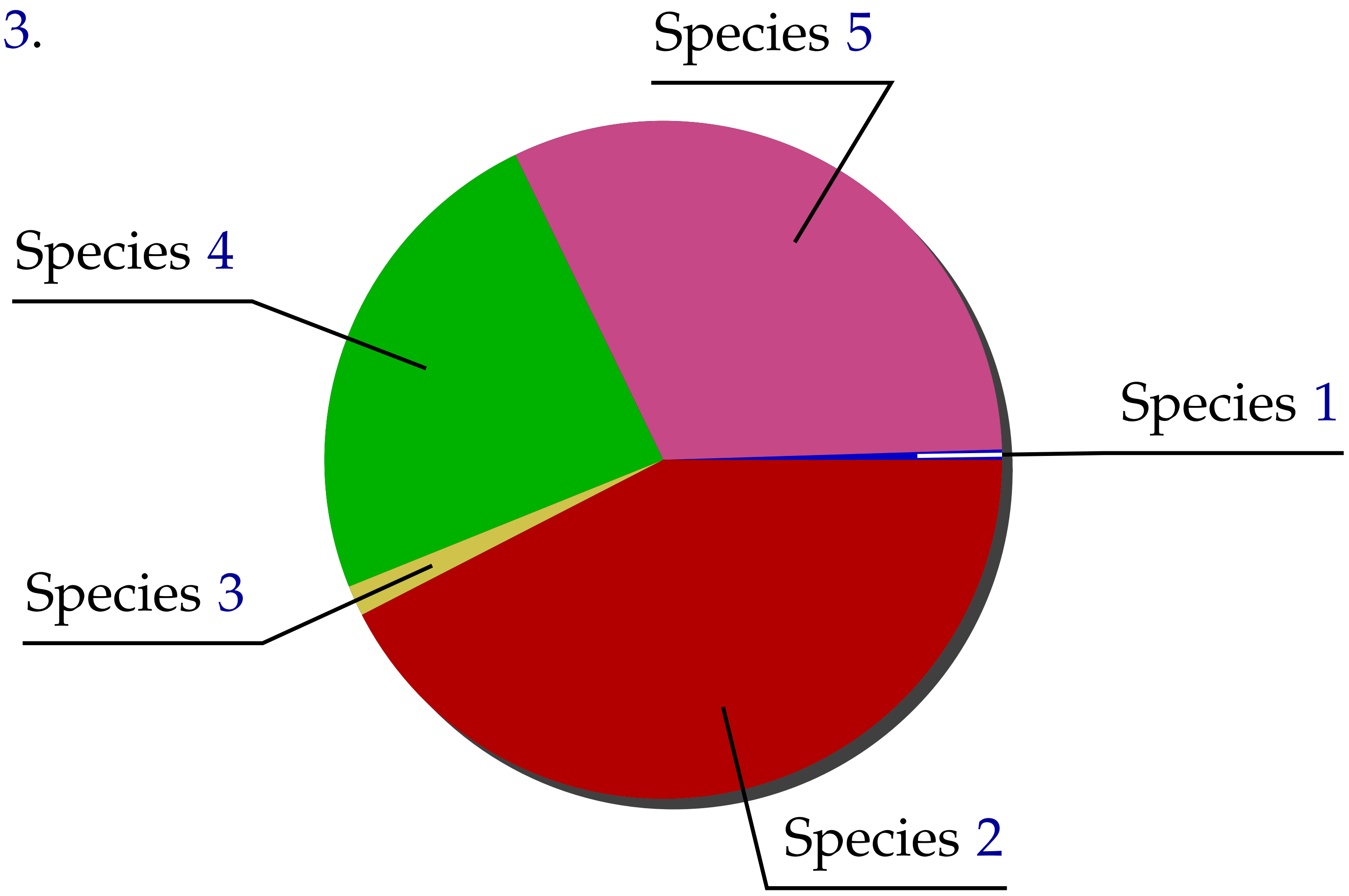
Intuitive idea of the replicator dynamic

Proportions at $t = 12$.



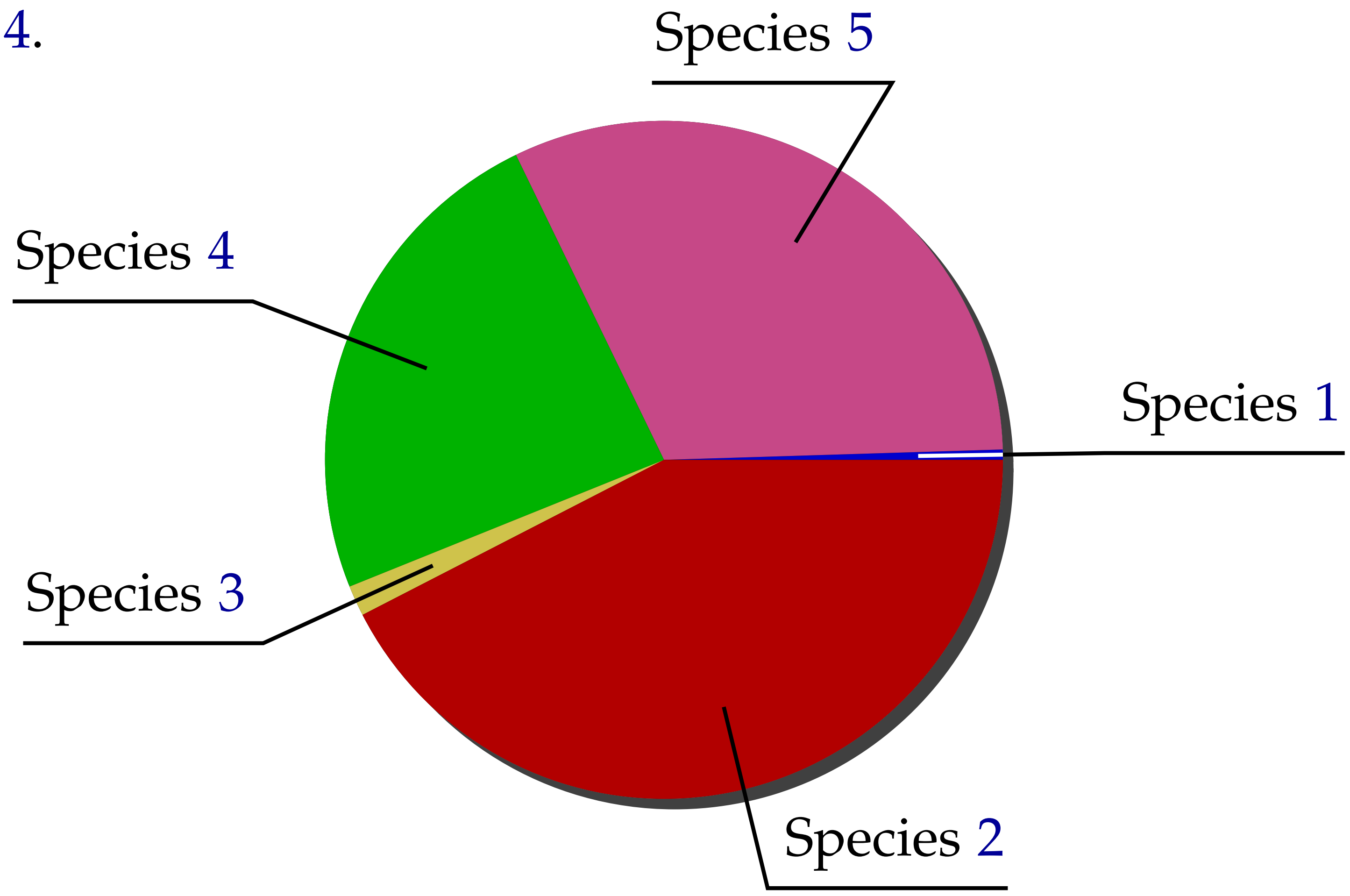
Intuitive idea of the replicator dynamic

Proportions at $t = 13$.



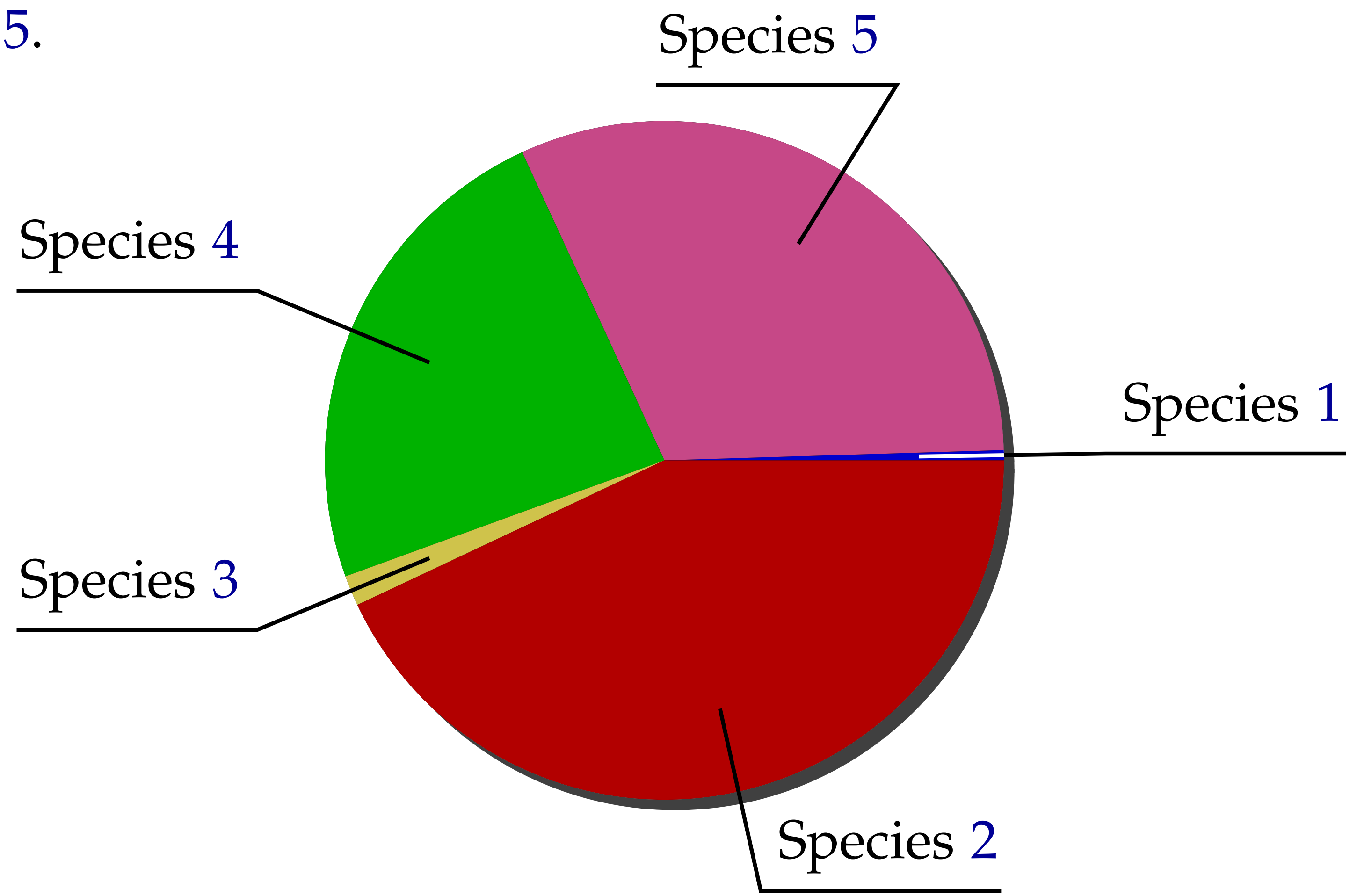
Intuitive idea of the replicator dynamic

Proportions at $t = 14$.



Intuitive idea of the replicator dynamic

Proportions at $t = 15$.



Strong species survive, even in competition with other strong species.

Replicator dynamic: circle of life

Replicator dynamic: circle of life

- Consider the following interaction matrix:

	lion	zebra	grass
lion	0	$+2$	0
zebra	-10	0	$+2$
grass	$+1$	-9	$+1$

.

-

Replicator dynamic: circle of life

- Consider the following interaction matrix:

	lion	zebra	grass
lion	0	$+2$	0
zebra	-10	0	$+2$
grass	$+1$	-9	$+1$

.

- • When species of the same kind interact, there is no gain or loss.

Replicator dynamic: circle of life

- Consider the following interaction matrix:

	lion	zebra	grass
lion	0	+2	0
zebra	−10	0	+2
grass	+1	−9	+1

.

-
  - When species of the same kind interact, there is no gain or loss.
  - Consumption yields 2. Death costs 10.

Replicator dynamic: circle of life

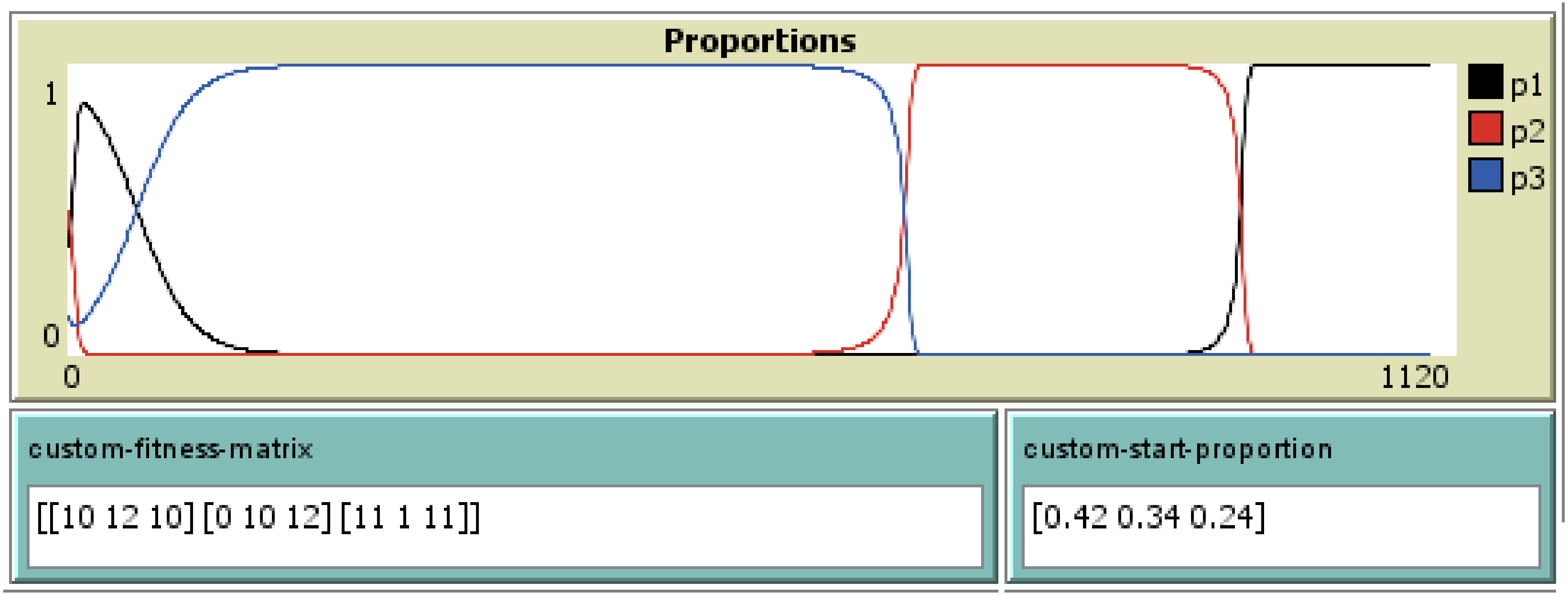
- Consider the following interaction matrix:

	lion	zebra	grass
lion	0	$+2$	0
zebra	-10	0	$+2$
grass	$+1$	-9	$+1$

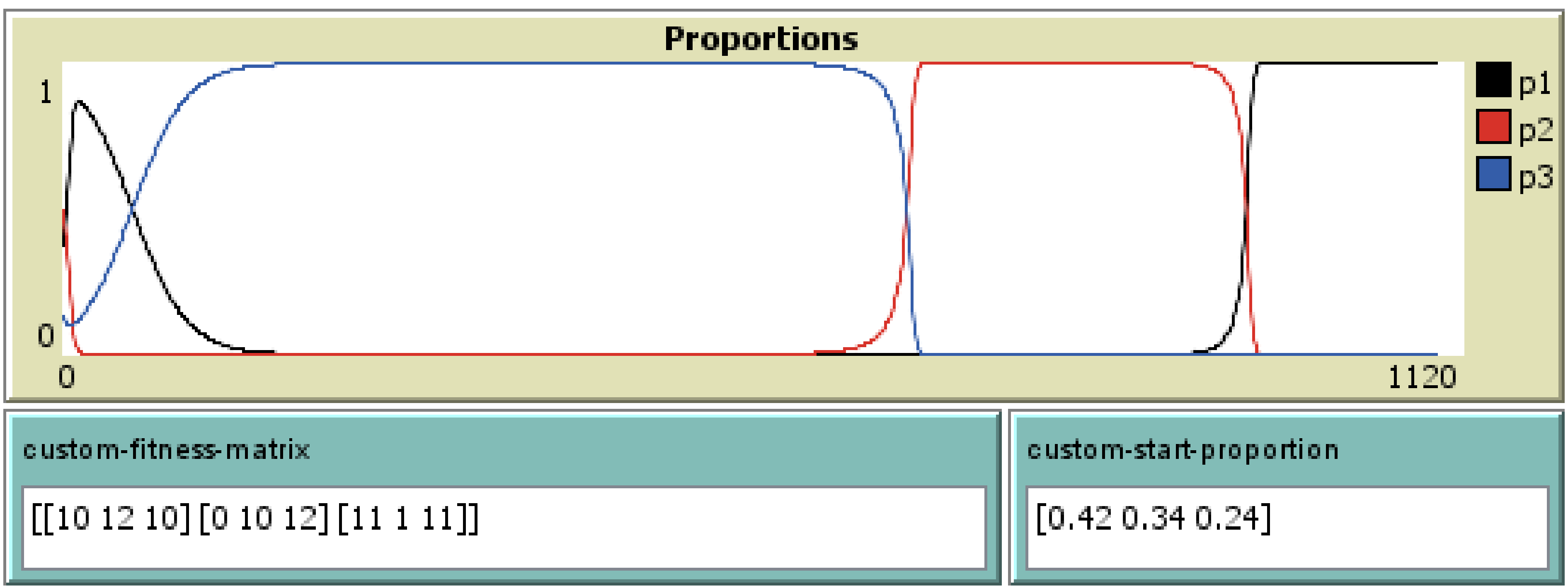
.

- - When species of the same kind interact, there is no gain or loss.
  - Consumption yields 2. Death costs 10.
  - If grass is left untouched it scores $+1$ for growth.

Replicator dynamic: circle of life

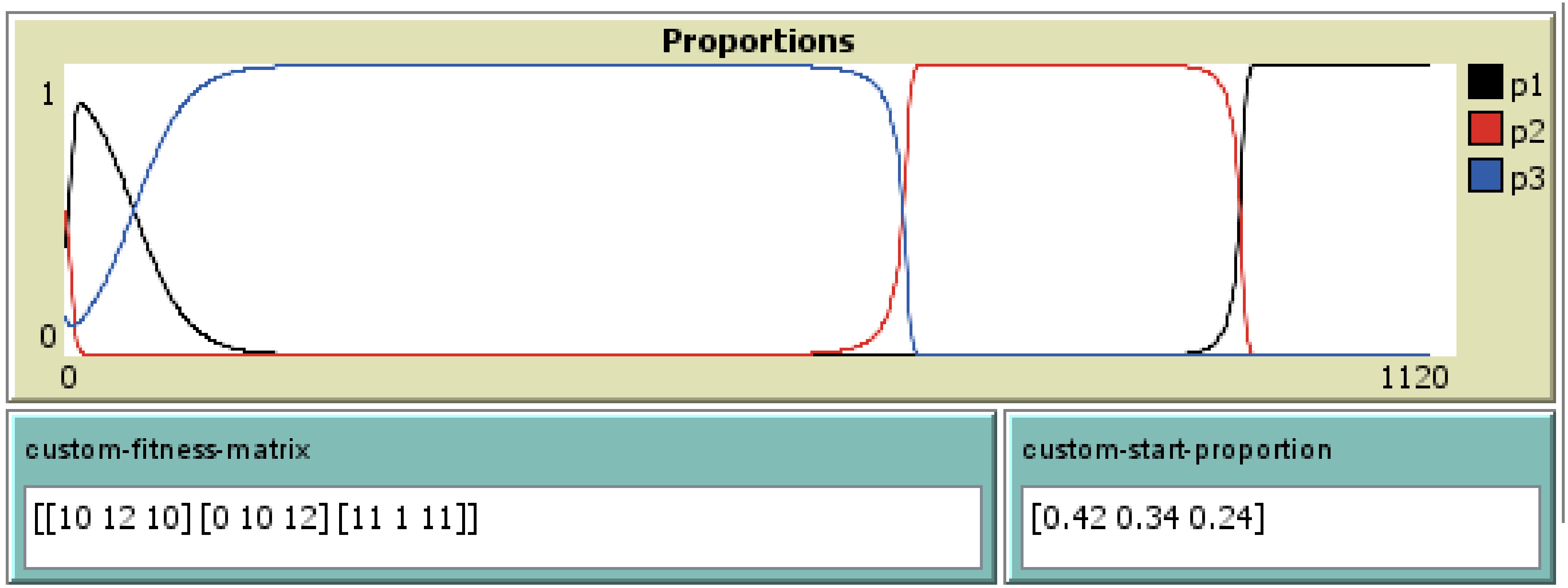


Replicator dynamic: circle of life



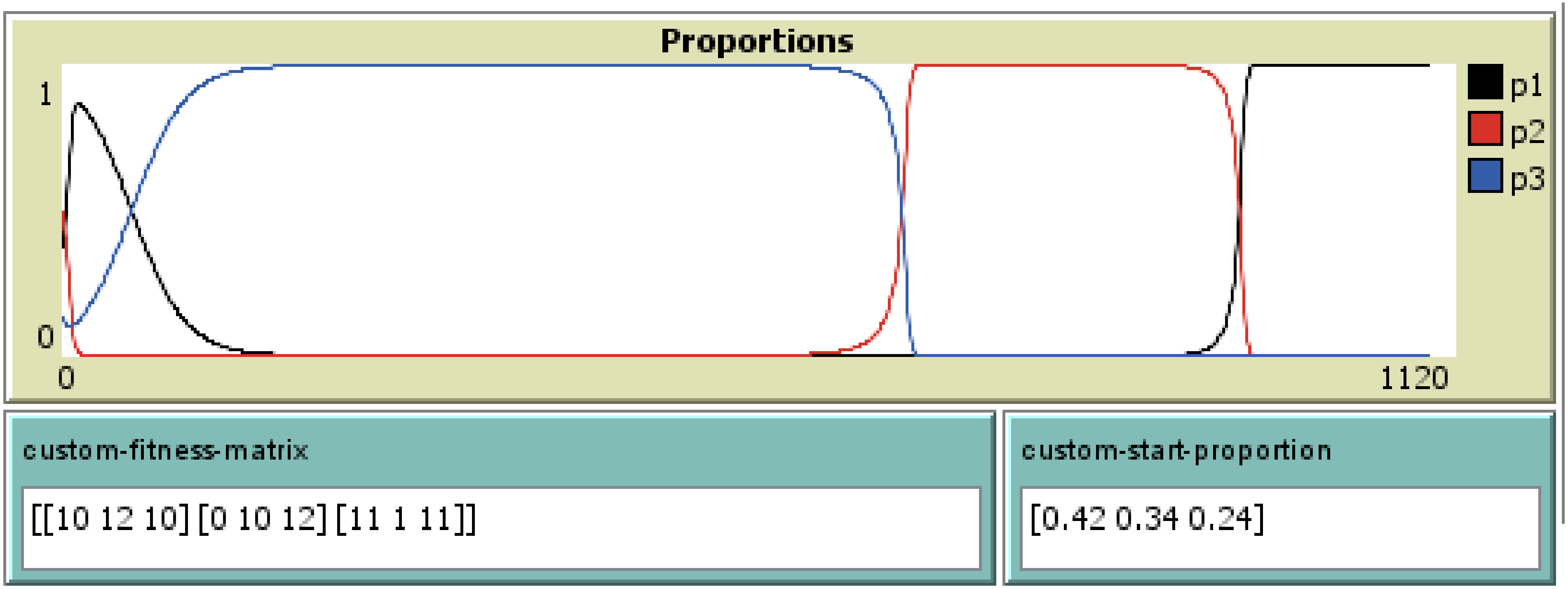
Lion is black, zebra is red, grass is blue.[1]

Replicator dynamic: circle of life



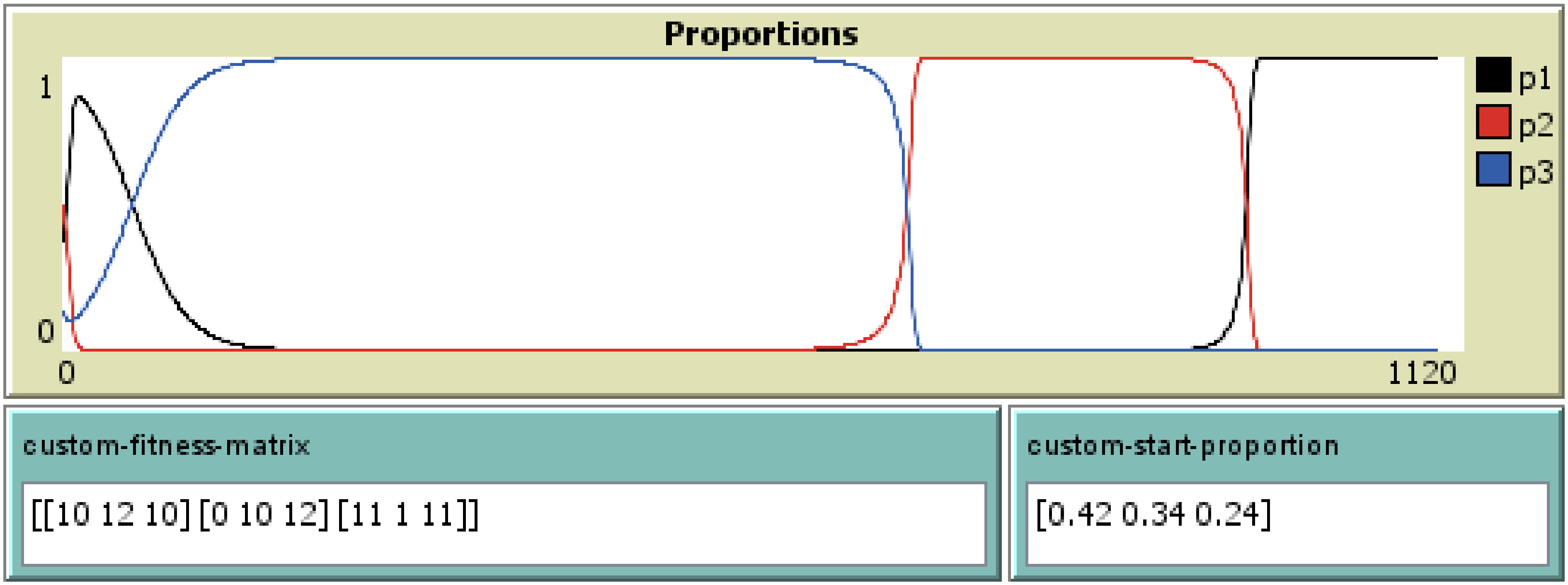
Lion is black, zebra is red, grass is blue.[1] Notice the circulation.

Replicator dynamic: circle of life



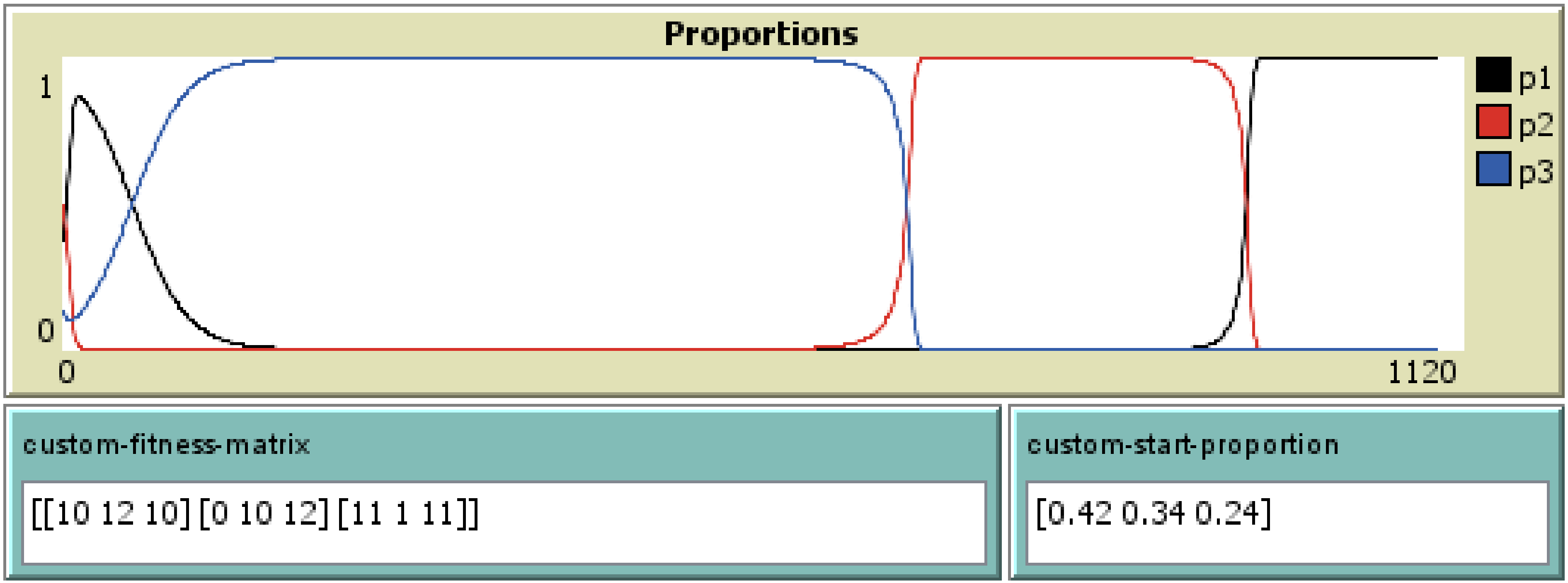
Lion is black, zebra is red, grass is blue.[1] Notice the circulation. Because the starting proportions are in Δ^2_o (the interior of Δ^2), no species will die out.

Replicator dynamic: circle of life



Lion is black, zebra is red, grass is blue.[1] Notice the circulation. Because the starting proportions are in Δ_o^2 (the interior of Δ^2), no species will die out. This follows from the replicator equation.

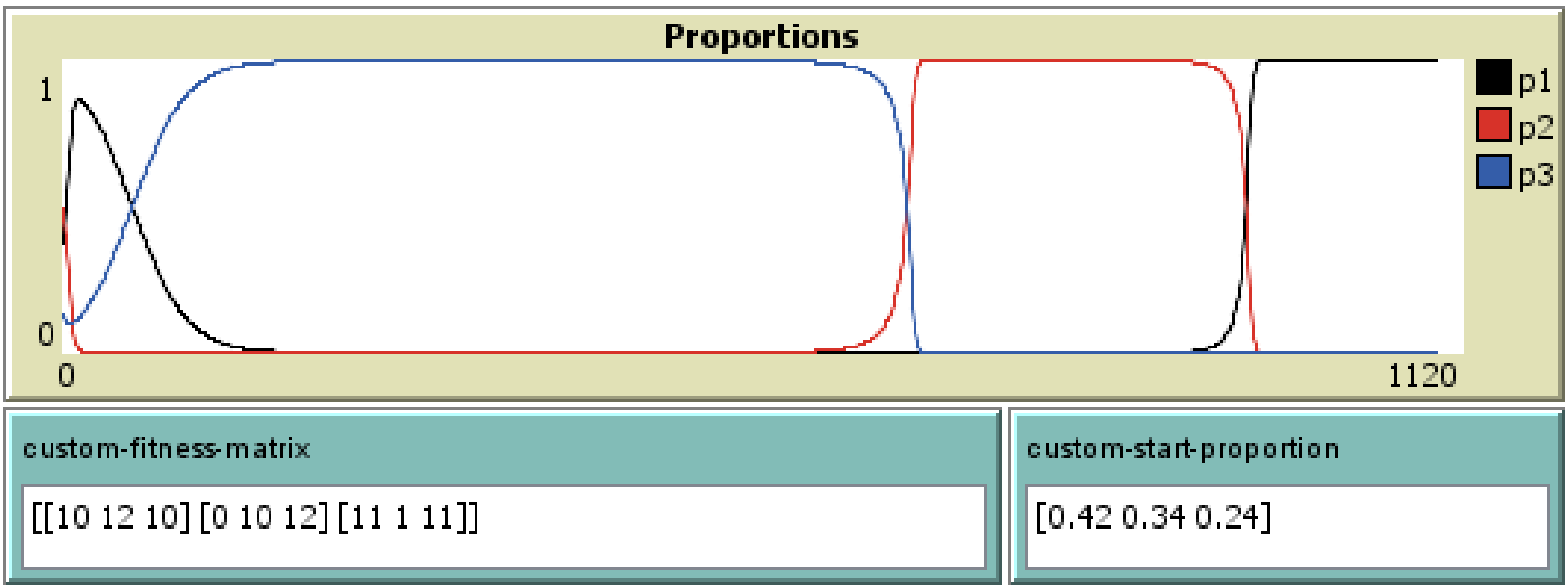
Replicator dynamic: circle of life



Lion is black, zebra is red, grass is blue.[1] Notice the circulation. Because the starting proportions are in Δ^2_o (the interior of Δ^2), no species will die out. This follows from the replicator equation.

The fitness matrix uses is the one on the previous page $+10$

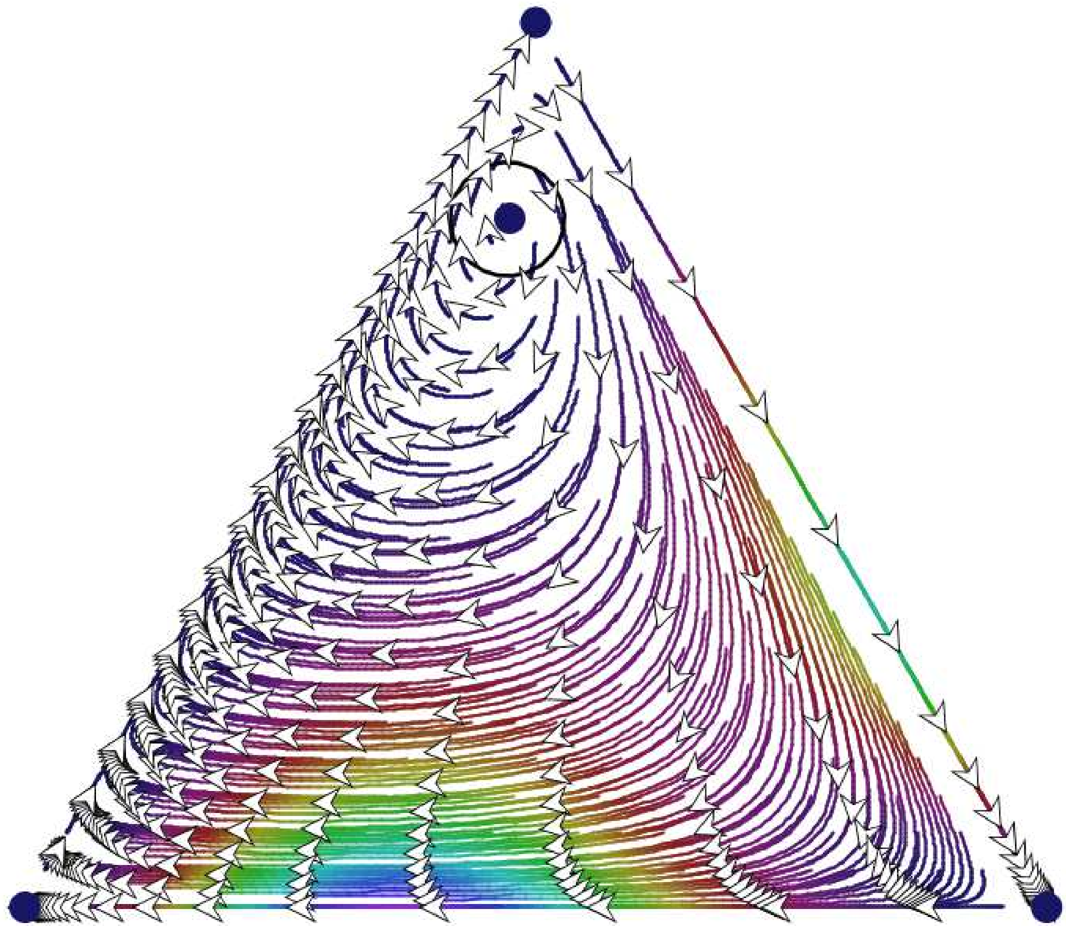
Replicator dynamic: circle of life



Lion is black, zebra is red, grass is blue.[1] Notice the circulation. Because the starting proportions are in Δ^2_o (the interior of Δ^2), no species will die out. This follows from the replicator equation.
The fitness matrix uses is the one on the previous page $+10$: the replicator equation can't handle negative numbers, and one can prove that the replicator dynamic doesn't change under linear transformations.

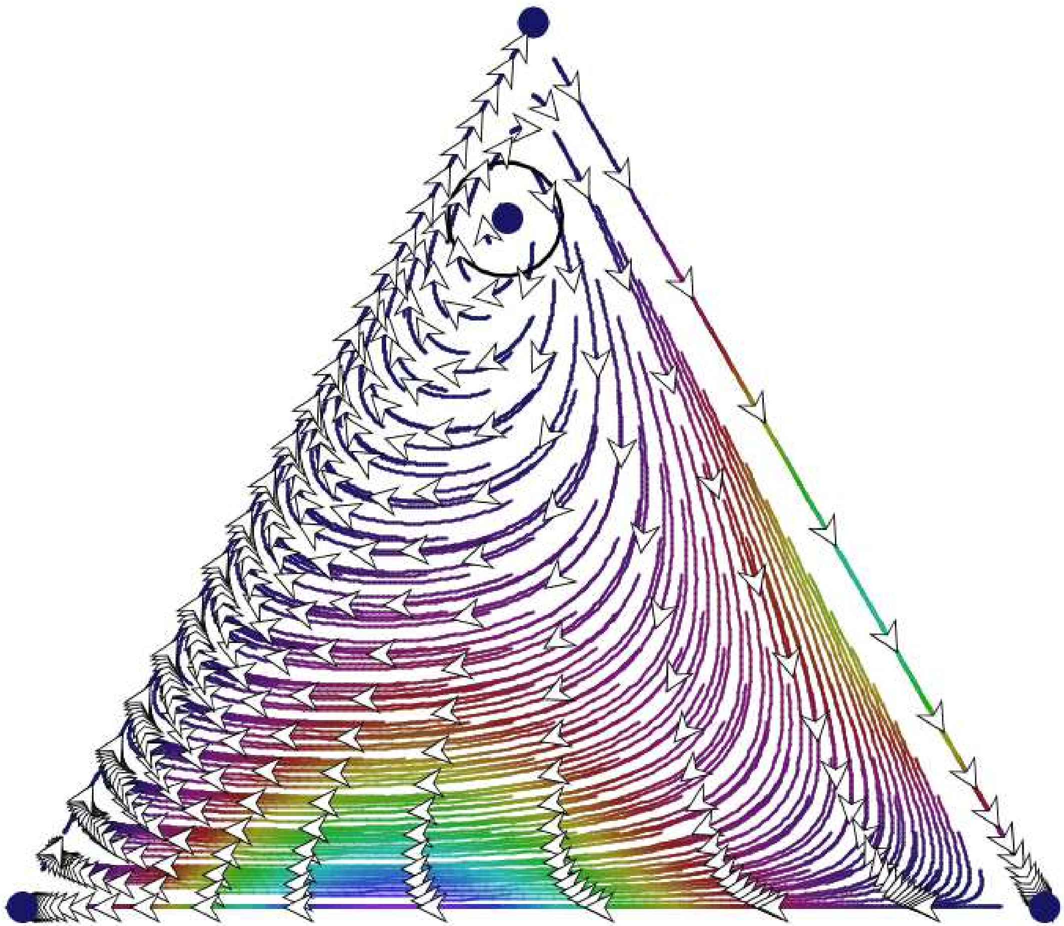
[1] https://en.wikipedia.org/wiki/Bluegrass_music

Replicator dynamic: circle of life



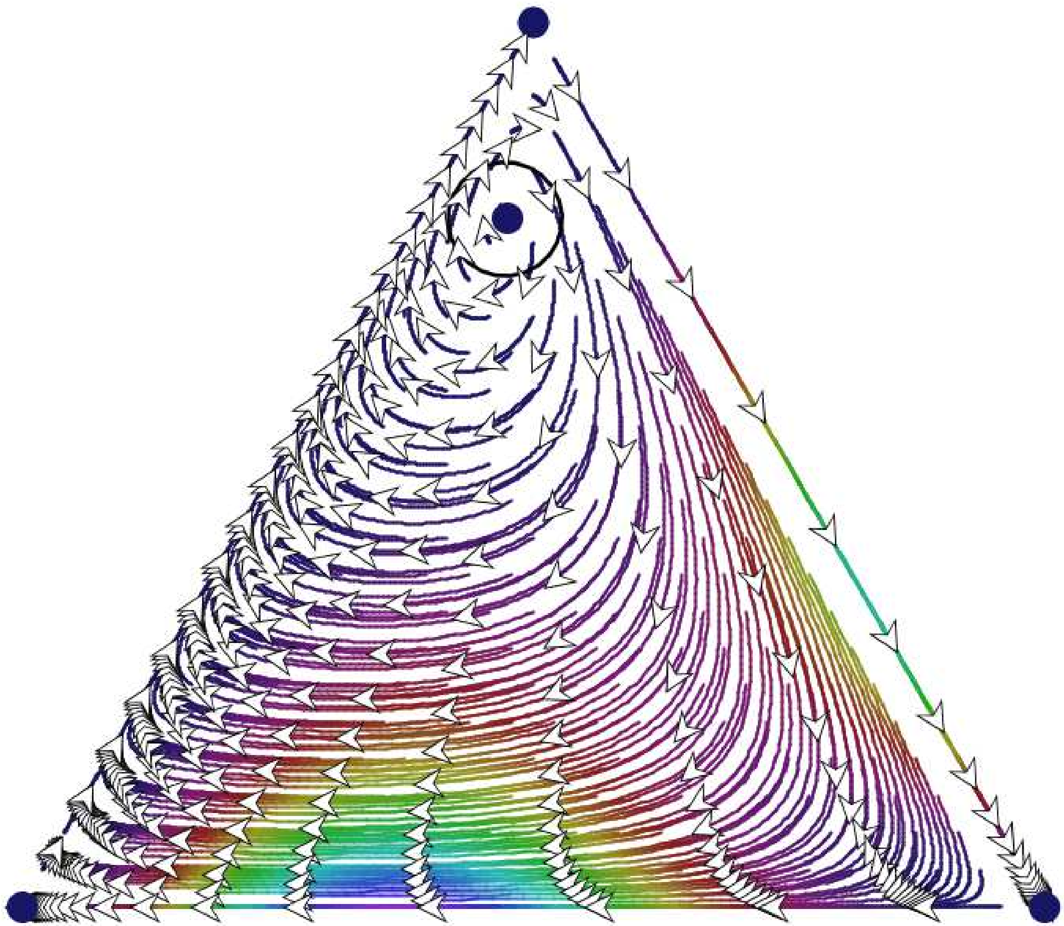
Left corner is 100% lion. Right corner is 100% zebra. Mid-base is 50% lion, 50% zebra. Top-corner is 100% grass.

Replicator dynamic: circle of life



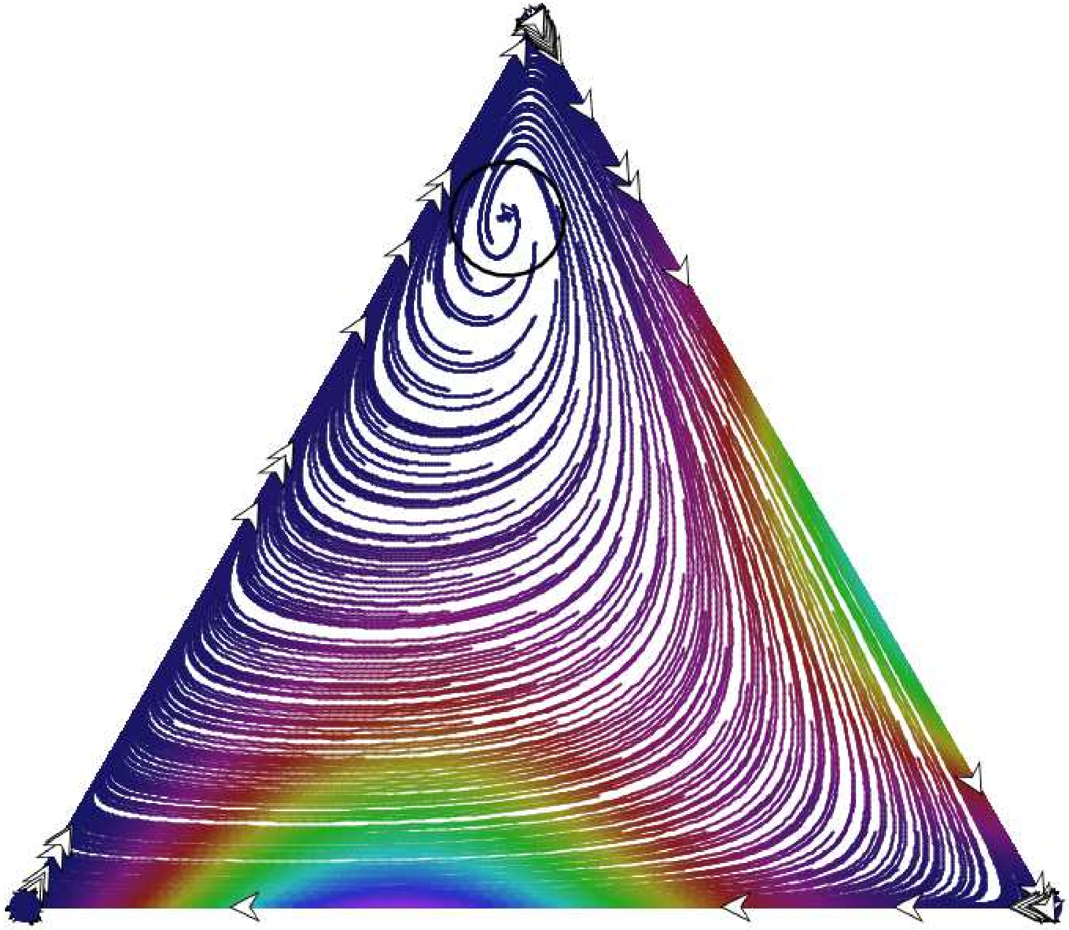
Left corner is 100% lion. Right corner is 100% zebra. Mid-base is 50% lion, 50% zebra. Top-corner is 100% grass. Blue dots indicate rest points.

Replicator dynamic: circle of life



Left corner is 100% lion. Right corner is 100% zebra. Mid-base is 50% lion, 50% zebra. Top-corner is 100% grass. Blue dots indicate rest points. Black circles indicate Nash equilibria.

Replicator dynamic: circle of life



After many iterations. Left corner is 100% lion. Right corner is 100% zebra. Mid-base is 50% lion, 50% zebra. Top-corner is 100% grass. Color indicates speed.

Simplex tutor

This app is meant to promote your intuitions about the (3-1)-simplex, and N-simplices in general. The 2-simplex enables you to represent 3D in 2D. It consists of all 0 <= x, y, z <= 1 such that x + y + z = 1.

* The blue square represents, in the unit square, the (x, y) part of (x, y, z). The value of z then automatically is 1 - x - z.
* The red triangle represents (x, y, z) in the 2-simplex (the big triangle).

setup | steer-xyz-in-simplex

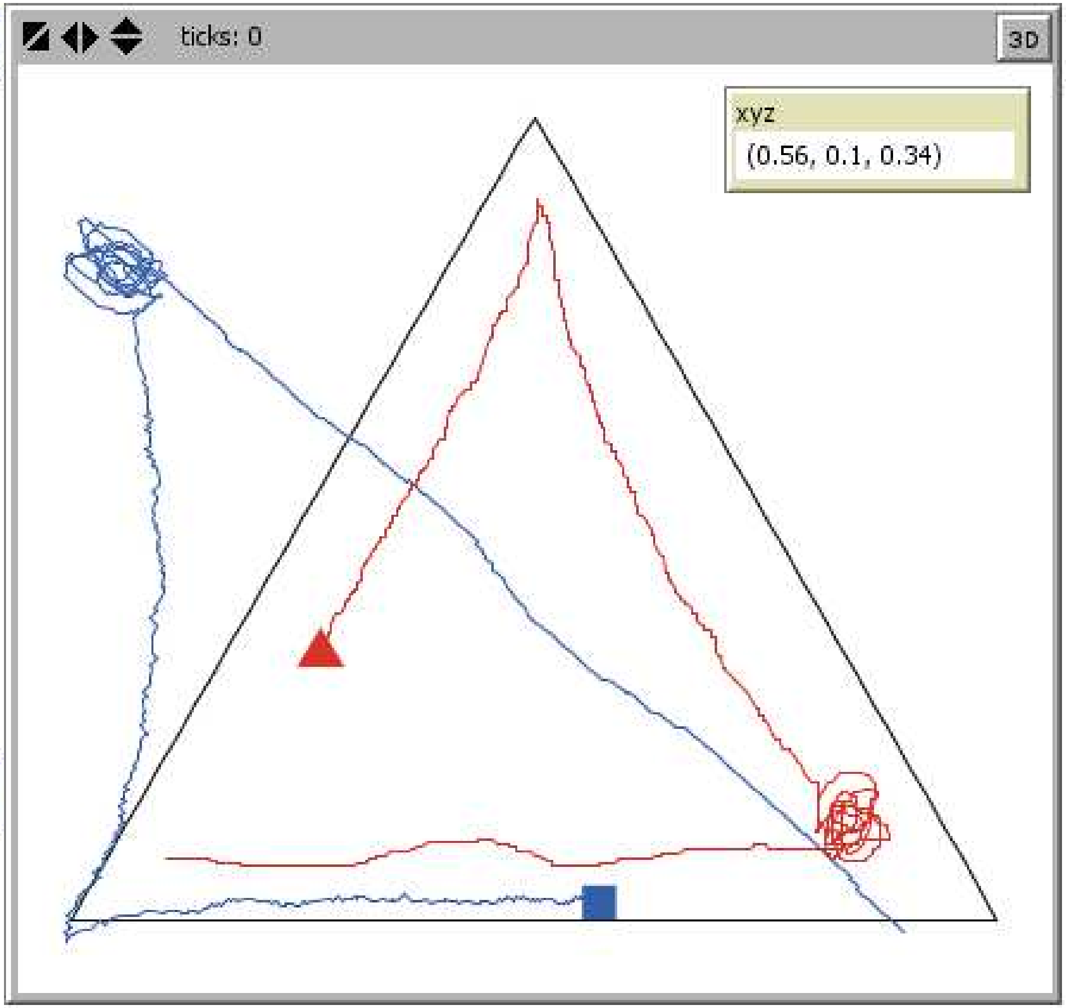
First "setup".

While "steer-xyz-in-simplex" is down, you are in charge of x, y, and z. (Just move the mouse.)

Try to do the following:
- Move red to (1, 0, 0). This is the left corner of the triangle. Can you explain the position of blue?
- Move red to (0, 1, 0). This is the right corner. Explain blue.
- Move red to (1, 0, 0). Explain blue.
- Move red to (0.5, 0.5, 0). This is mid-base. Explain blue.
- Move red to (0.5, 0, 0.5). This is mid-left. Explain blue.
- Move red in the middle of the 2-simplex. Explain blue.

pens-down | pens-up | erase-all

Click on the white outside every widget. The P on "pens-down" should light up. Move the mouse at a proper location. ONLY THEN then press P on you keyboard to start drawing.



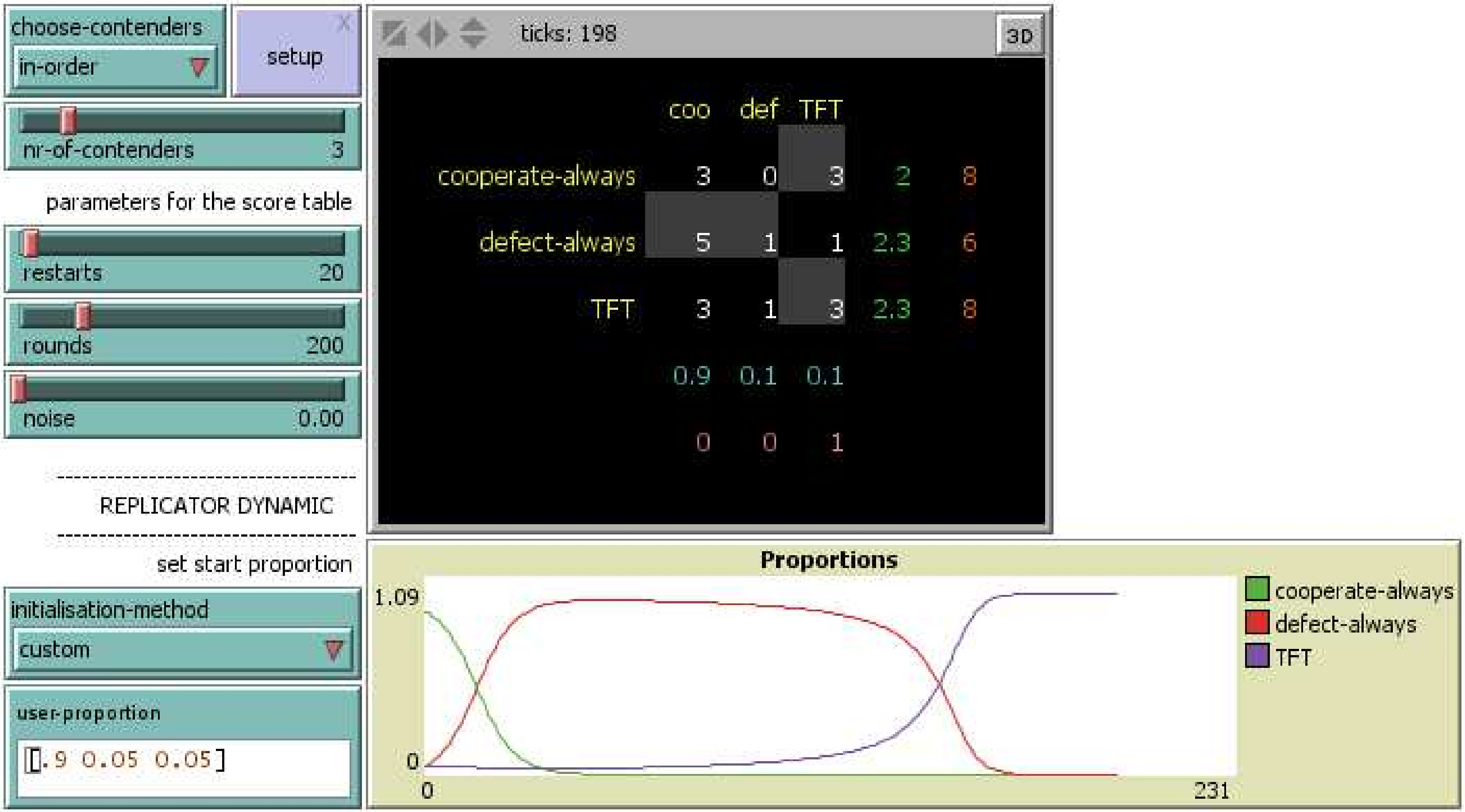
steer-xy-in-unit-square

First "setup".

While "steer-xy-in-unit-square" is down, you are in charge of x and y. (Just move the mouse.) Then z = 1 - x - y and the triple (x, y, z) is mapped on the 2-simplex.

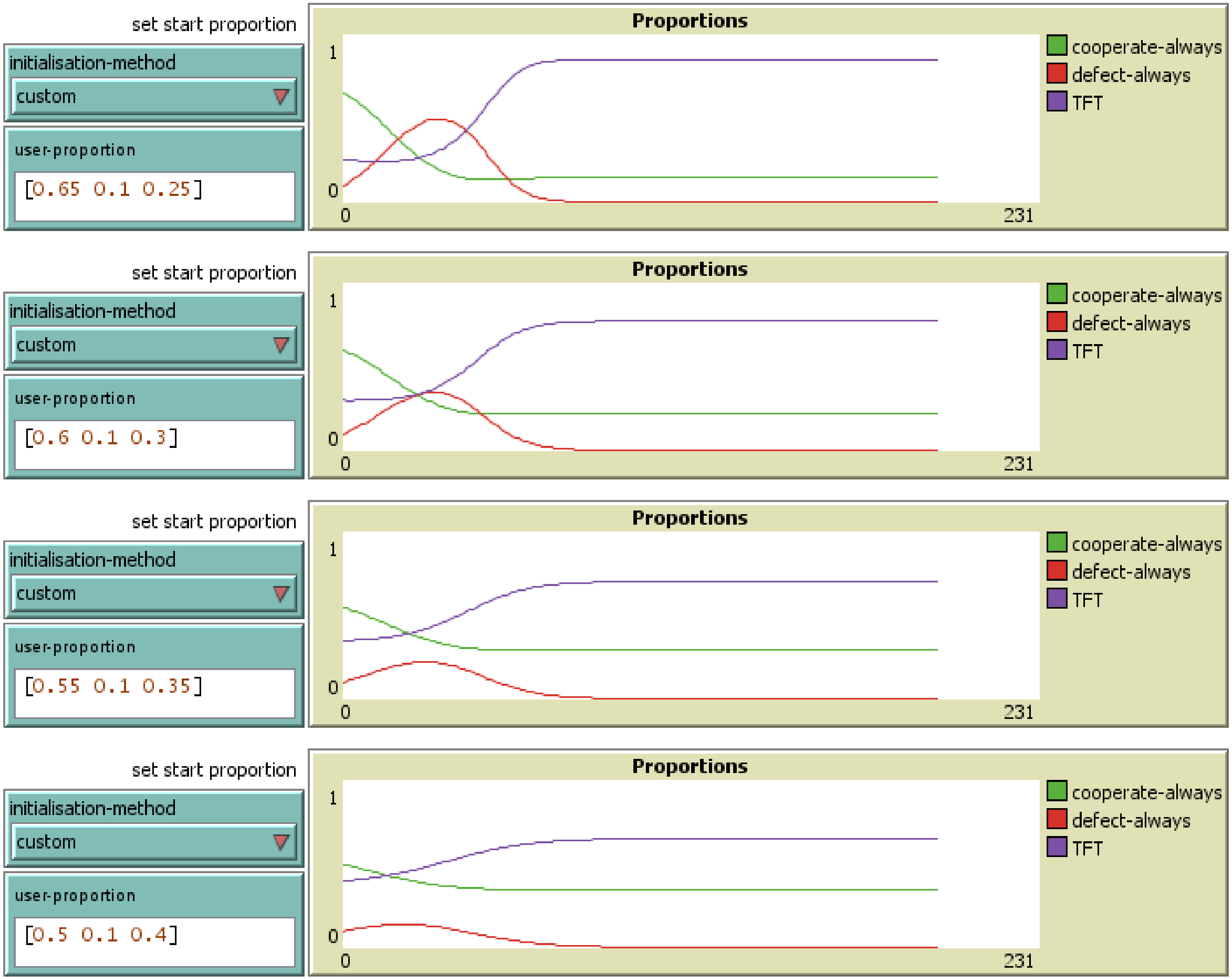
Try to do the following:
- Move blue to (0, 0). Can you explain the position of red?
- Move blue to (1, 0). Explain red.
- Move blue to (0, 1). Explain red.
- Move blue to (0.5, 0.5). Explain red.
- Move blue along the line x = 0. This is the left edge of the unit square. Explain red.
- Same with the line y = 0. (Bottom edge of unit square.)
- Move blue along the line y = 1 - x (diagonal from Texel to Maastricht). Explain red.
- Try to discover which positions of blue are legal for red. Explain.
- Which positions of blue are illegal for red? Explain.

The replicator dynamic among All-C, All-D, and Tit-for-t

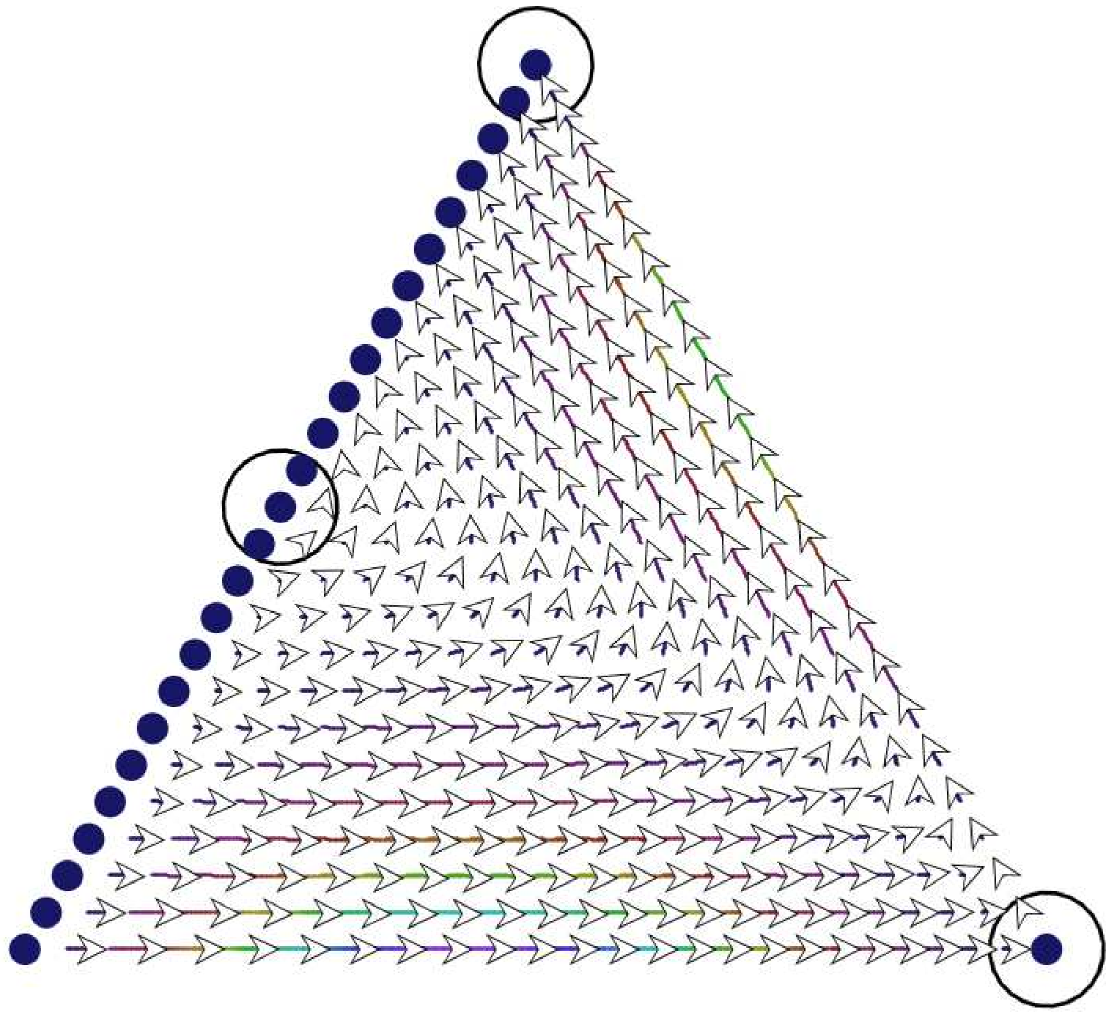


If there are many cooperators, the defectors will profit at the expense of the cooperators. TFT-ers, on their turn, prosper in the presence of themselves and defectors.

Different starting proportions lead to different rest point

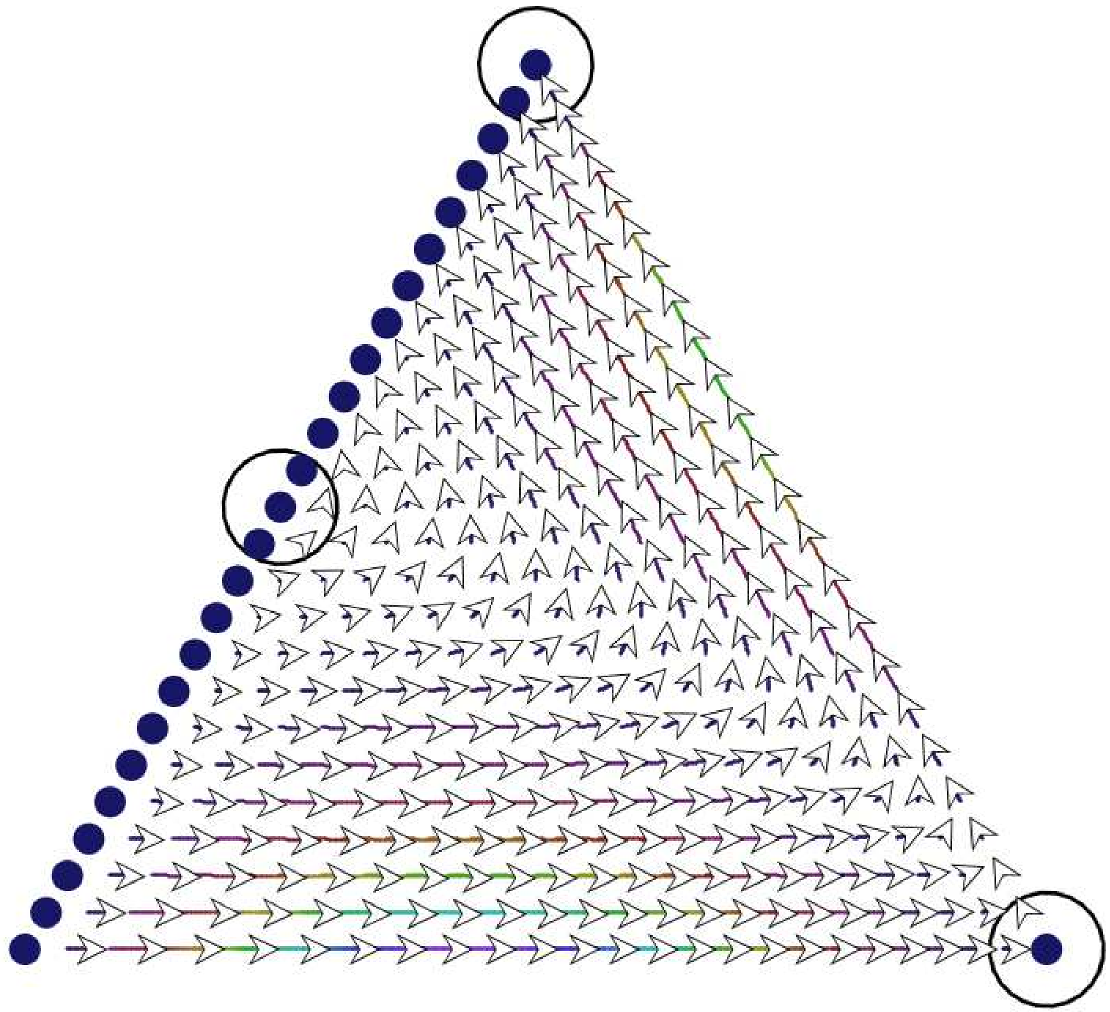


Phase space of the replicator for All-C vs. All-D vs. TFT



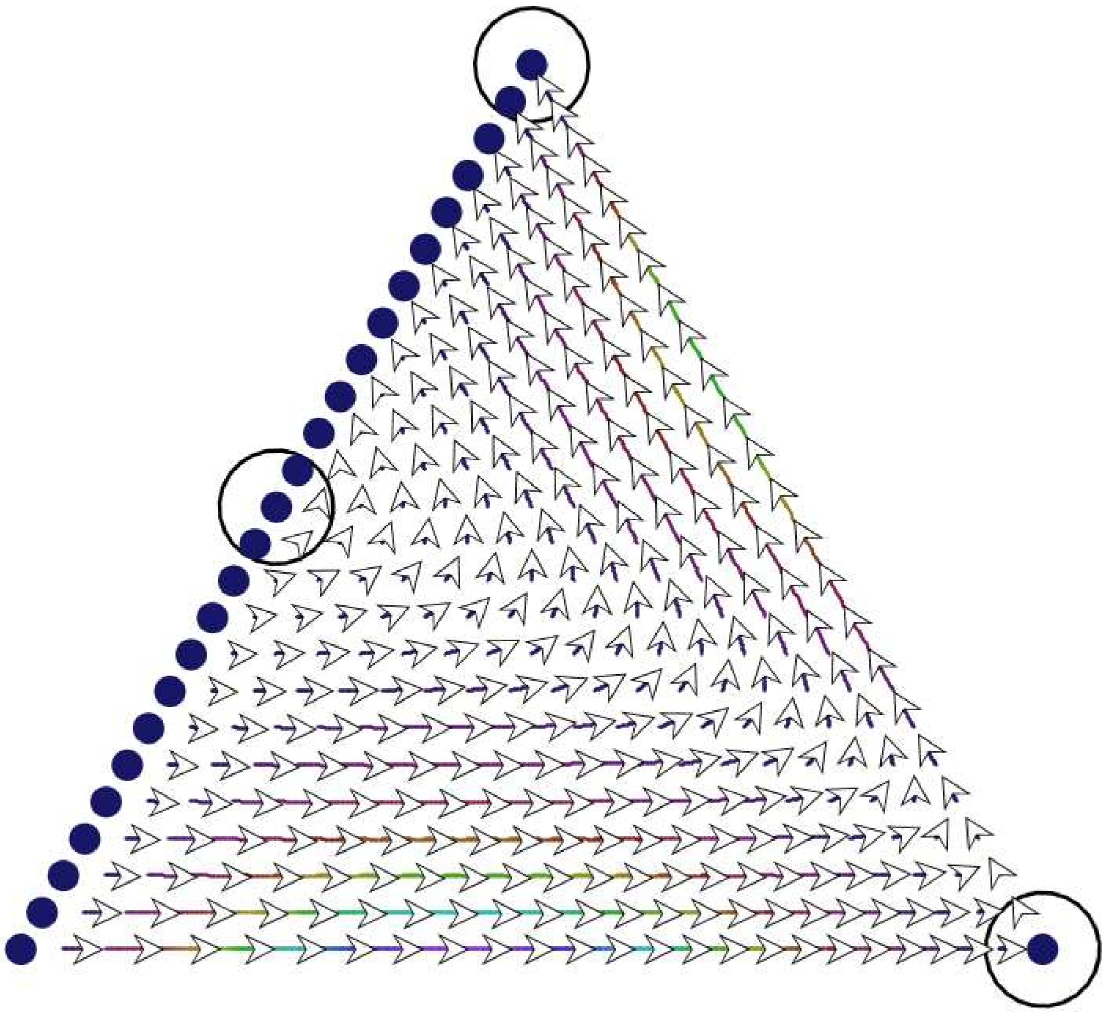
Left corner is 100% All-C. Right corner is 100% All-D. Top-corner is 100% TFT.

Phase space of the replicator for All-C vs. All-D vs. TFT



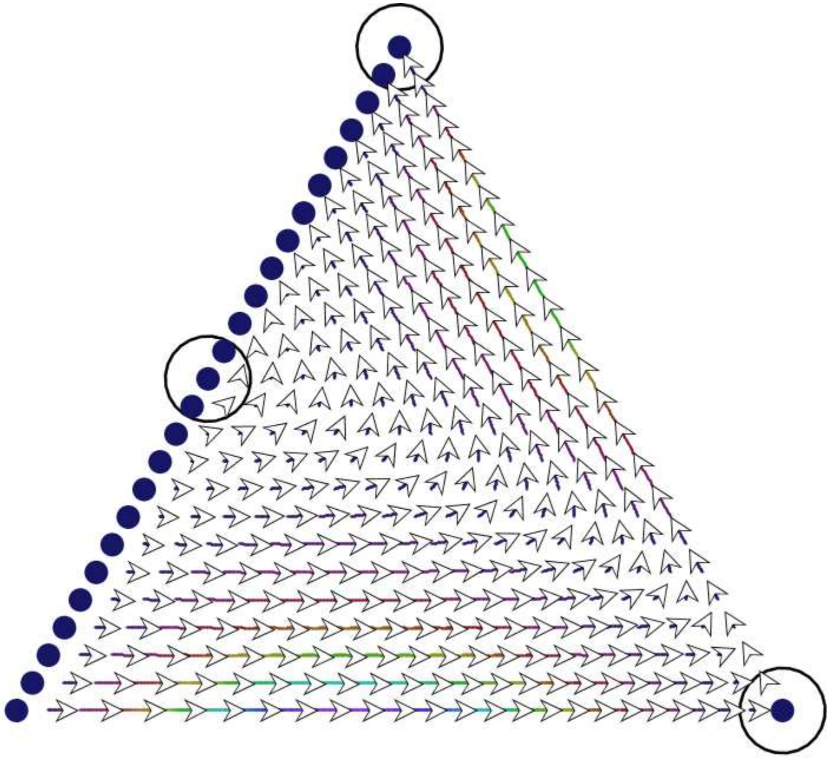
Left corner is 100% All-C. Right corner is 100% All-D. Top-corner is 100% TFT. Mid-base is 50% All-C and 50% All-D.

Phase space of the replicator for All-C vs. All-D vs. TFT



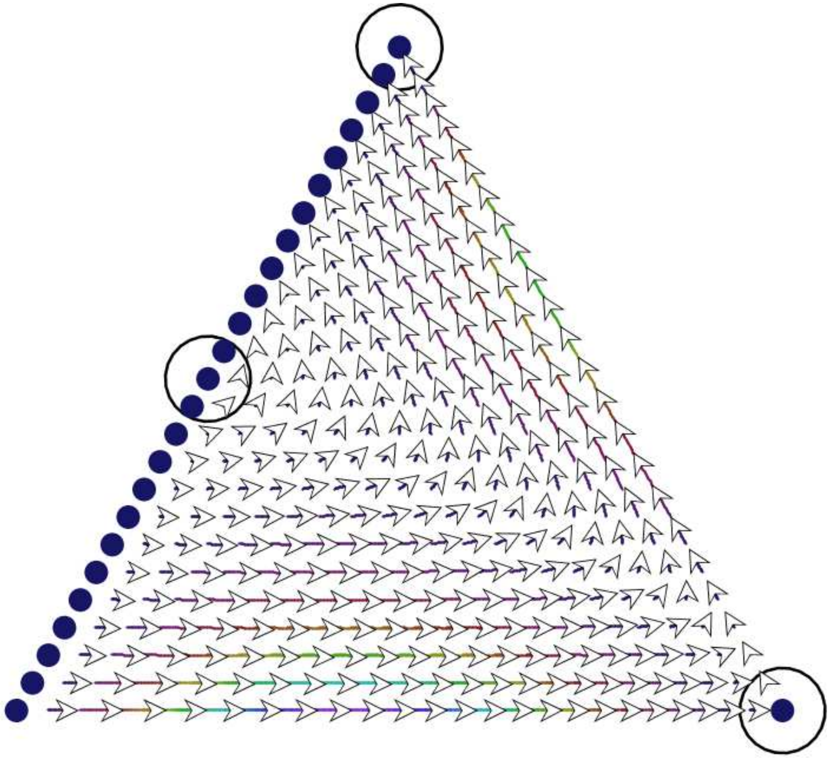
Left corner is 100% All-C. Right corner is 100% All-D. Top-corner is 100% TFT. Mid-base is 50% All-C and 50% All-D. Blue dots indicate rest points. Black circles indicate Nash equilibria.

All-C vs. All-D vs. TFT: observations



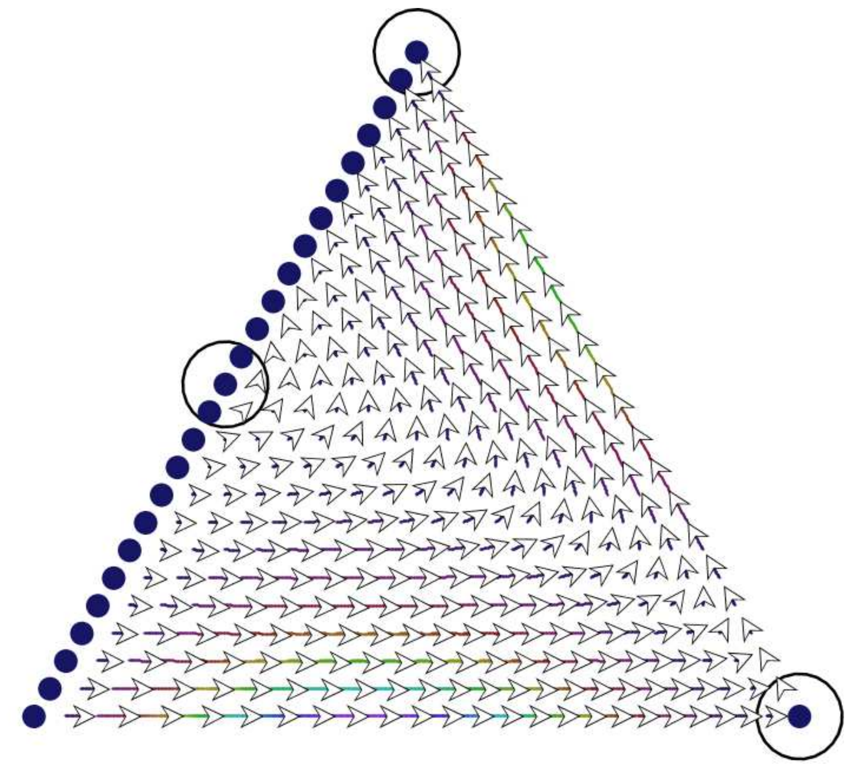
All-C vs. All-D vs. TFT: observations

- The left edge, the one without defectors, is pointwise invariant under the replicator dynamic, i.e., no change here.



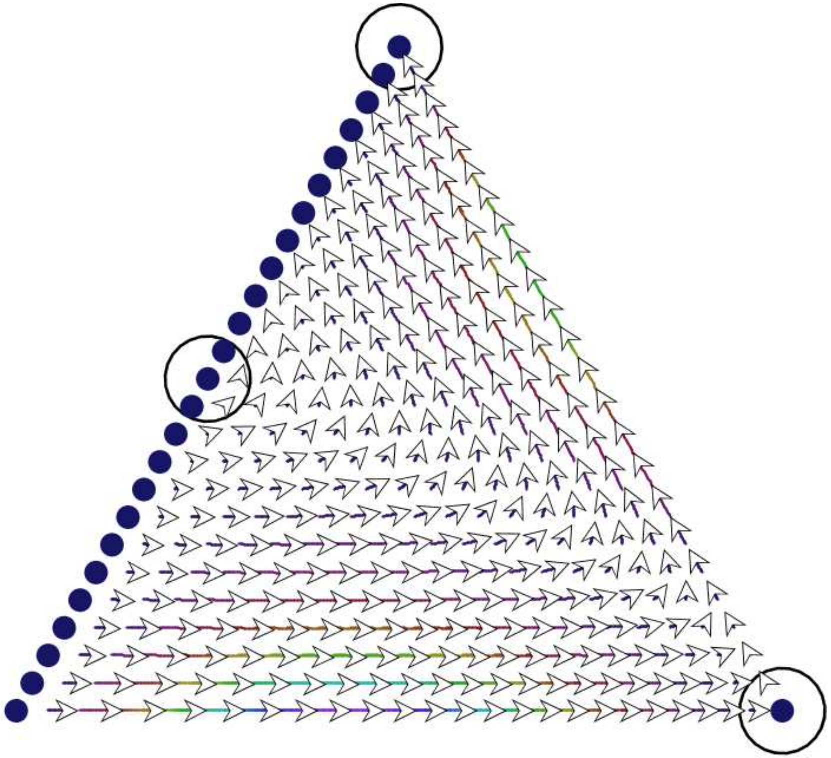
All-C vs. All-D vs. TFT: observations

- The left edge, the one without defectors, is pointwise invariant under the replicator dynamic, i.e., no change here.
- All pure populations are invariant as well. E.g., if there are only cooperators and no defectors or TFT-ers, there will be only co-operators forever.



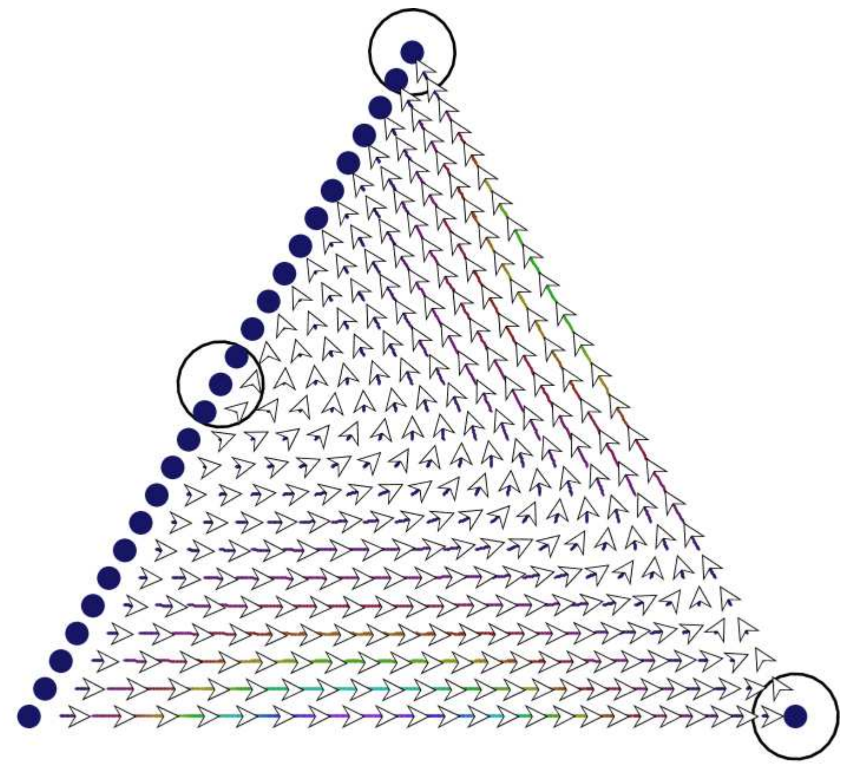
All-C vs. All-D vs. TFT: observations

- The left edge, the one without defectors, is pointwise invariant under the replicator dynamic, i.e., no change here.
- All pure populations are invariant as well. E.g., if there are only cooperators and no defectors or TFT-ers, there will be only co-operators forever.
- All Nash equilibria (open circles) are rest points (blue dots) of the replicator. (Check!)



All-C vs. All-D vs. TFT: observations

- The left edge, the one without defectors, is pointwise invariant under the replicator dynamic, i.e., no change here.
- All pure populations are invariant as well. E.g., if there are only cooperators and no defectors or TFT-ers, there will be only co-operators forever.
- All Nash equilibria (open circles) are rest points (blue dots) of the replicator. (Check!)
- A rest point is called Lyapunov stable or just stable if, once the replicator is close enough, it remains there. All Lyapunov stable rest points are Nash equilibria. (Check!)



Convergence of the replicator in a grand table

Does replicator dynamic always converge?

Does replicator dynamic always converge?

- No.

Does replicator dynamic always converge?

- No. (See circle of life.)

Does replicator dynamic always converge?

- No. (See circle of life.) On grand tables?

Does replicator dynamic always converge?

- No. (See circle of life.) On grand tables? No, not even that :-(

Does replicator dynamic always converge?

- No. (See circle of life.) On grand tables? No, not even that `:-(`
- Try to find strategies `alpha`, `beta`, `gamma` such that `beta` performs best in the presence of `alpha`, `gamma` performs best in the presence of `beta`, and `alpha` performs best in the presence of `gamma`.

Does replicator dynamic always converge?

- No. (See circle of life.) On grand tables? No, not even that `:-(`
- Try to find strategies alpha, beta, gamma such that beta performs best in the presence of alpha, gamma performs best in the presence of beta, and alpha performs best in the presence of gamma. Think rock-paper-scissors.

Does replicator dynamic always converge?

- No. (See circle of life.) On grand tables? No, not even that :-(
- Try to find strategies alpha, beta, gamma such that beta performs best in the presence of alpha, gamma performs best in the presence of beta, and alpha performs best in the presence of gamma. Think rock-paper-scissors.
- This is not enough! Payoffs must be in proportion to prevent convergence to $(0.33, 0.33, 0.33)$.

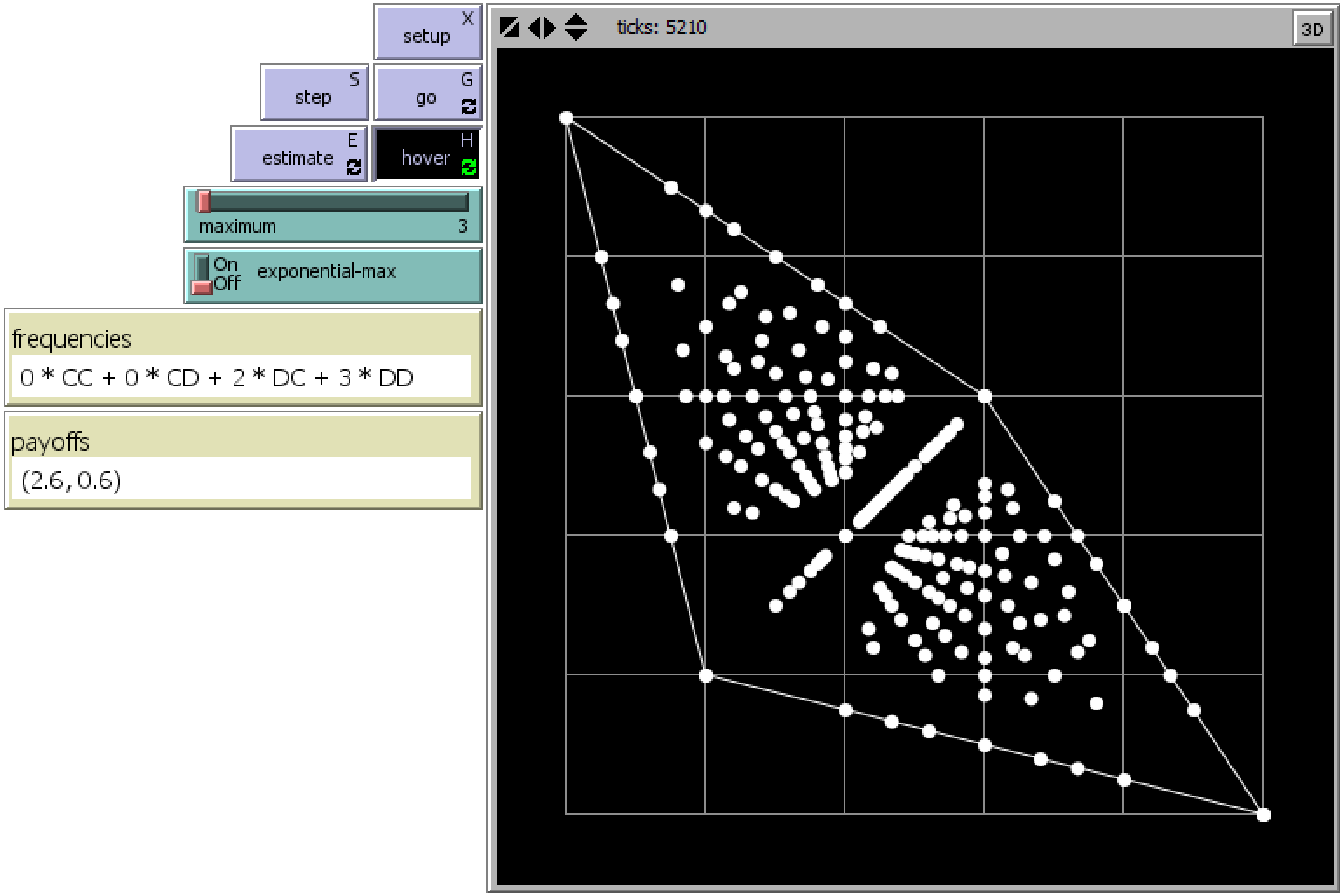
Does replicator dynamic always converge?

- No. (See circle of life.) On grand tables? No, not even that :-(
- Try to find strategies alpha, beta, gamma such that beta performs best in the presence of alpha, gamma performs best in the presence of beta, and alpha performs best in the presence of gamma. Think rock-paper-scissors.
- This is not enough! Payoffs must be in proportion to prevent convergence to $(0.33, 0.33, 0.33)$.
- Because $(0, 5)$, $(5, 0)$, and $(2.5, 2.5)$ are feasible payoff profiles [use $(C, D)^*$, $(D, C)^*$ and $((C, C)^3, (D, D))^*$, respectively], try

	alpha	beta	gamma
alpha	2.5	5	0
beta	0	2.5	5
gamma	5	0	2.5

.

Feasible payoff combinations in the PD



E.g., $2DC + 3DD$ yields a payoff profile of $(2.6, 0.6)$ in the average. Can you find the payoff profile in the screenshot? With higher max factor than 3 we can get near every other payoff profile in the convex hull of $\{(3,3),(0,5),(5,0),(1,1)\}$.

Three strategies in rock-paper-scissors relation

Three strategies in rock-paper-scissors relation

- We'd like to have the following as a grand table.

	alpha	beta	gamma
alpha	2.5	5	0
beta	0	2.5	5
gamma	5	0	2.5

.

Three strategies in rock-paper-scissors relation

- We'd like to have the following as a grand table.

	alpha	beta	gamma
alpha	2.5	5	0
beta	0	2.5	5
gamma	5	0	2.5

.

- Strategies must be able to recognize one another to emit the proper behaviour.

Three strategies in rock-paper-scissors relation

- We'd like to have the following as a grand table.

	alpha	beta	gamma
alpha	2.5	5	0
beta	0	2.5	5
gamma	5	0	2.5

.

- Strategies must be able to recognize one another to emit the proper behaviour.
-

Three strategies in rock-paper-scissors relation

- We'd like to have the following as a grand table.

	alpha	beta	gamma
alpha	2.5	5	0
beta	0	2.5	5
gamma	5	0	2.5

.

- Strategies must be able to recognize one another to emit the proper behaviour.

- • At the outset, let alpha emit "0" and let alpha read the digit emitted by its opponent. After that, alpha responds to its opponent so as to reach the payoff as specified in the grand table.

Three strategies in rock-paper-scissors relation

- We'd like to have the following as a grand table.

	alpha	beta	gamma
alpha	2.5	5	0
beta	0	2.5	5
gamma	5	0	2.5

.

- Strategies must be able to recognize one another to emit the proper behaviour.

- • At the outset, let alpha emit "0" and let alpha read the digit emitted by its opponent. After that, alpha responds to its opponent so as to reach the payoff as specified in the grand table.

  • Same for beta, but emit "1" at the outset.

Three strategies in rock-paper-scissors relation

- We'd like to have the following as a grand table.

	alpha	beta	gamma
alpha	2.5	5	0
beta	0	2.5	5
gamma	5	0	2.5

.

- Strategies must be able to recognize one another to emit the proper behaviour.

- - At the outset, let alpha emit "0" and let alpha read the digit emitted by its opponent. After that, alpha responds to its opponent so as to reach the payoff as specified in the grand table.
  - Same for beta, but emit "1" at the outset.
  - Same for gamma, emit "2".

Example play paths

Example play paths

- If a strategy plays against itself, ensure that it emits it identity–and recognizes its opponent's identity–in the first two rounds. Then behave so as to achieve the desired payoff profile.

alpha	C	C	C	C	C	D	C	C	C	D	...	$\downarrow 2.5$
	"0"		cycle through fitting action profiles									
alpha	C	C	C	C	C	D	C	C	C	D	...	$\downarrow 2.5$
	"0"		cycle through fitting action profiles									

Example play paths

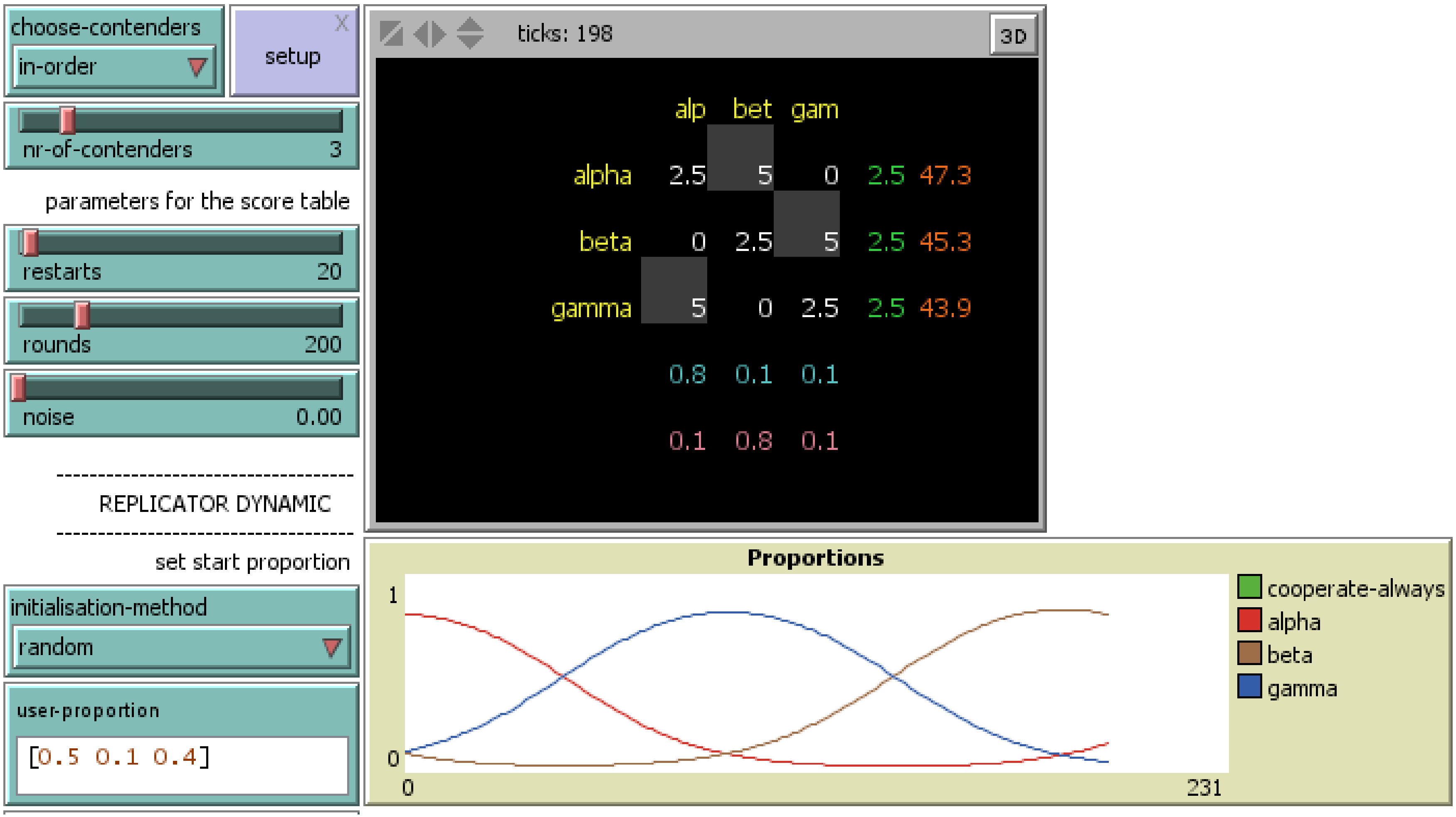
- If a strategy plays against itself, ensure that it emits it identity–and recognizes its opponent's identity–in the first two rounds. Then behave so as to achieve the desired payoff profile.

alpha	C	C	C	C	C	D	C	C	C	D	...	$\downarrow 2.5$
	"0"		cycle through fitting action profiles									
alpha	C	C	C	C	C	D	C	C	C	D	...	$\downarrow 2.5$
	"0"		cycle through fitting action profiles									

- With opponent beta:

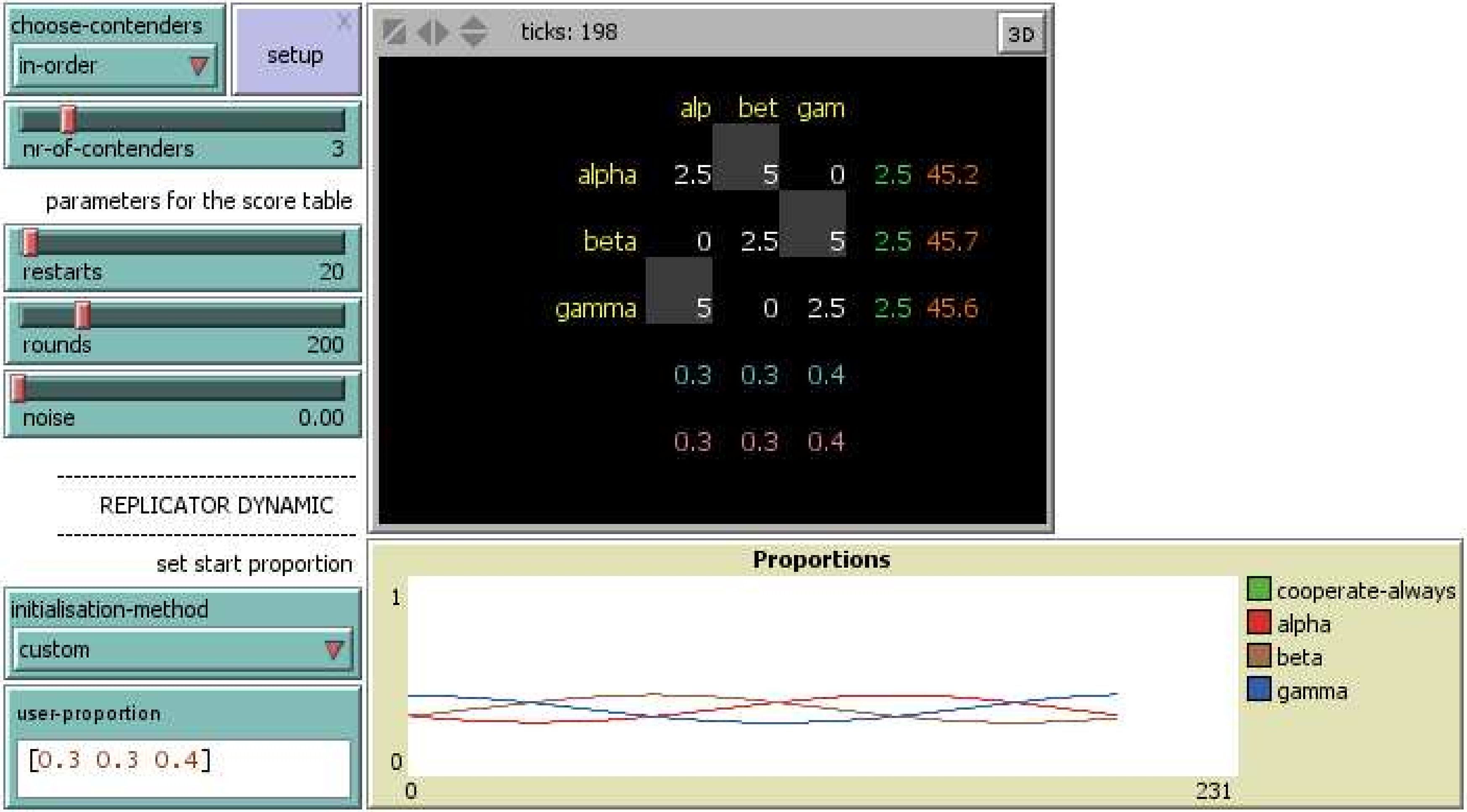
alpha	C	C	D	D	D	D	D	D	D	D	...	$\uparrow 5$
	"0"		cycle through fitting action profiles									
beta	C	D	C	C	C	C	C	C	C	C	...	$\downarrow 0$
	"1"		cycle through fitting action profiles									

No convergence in the $\{\alpha, \beta, \gamma\}$-tournament



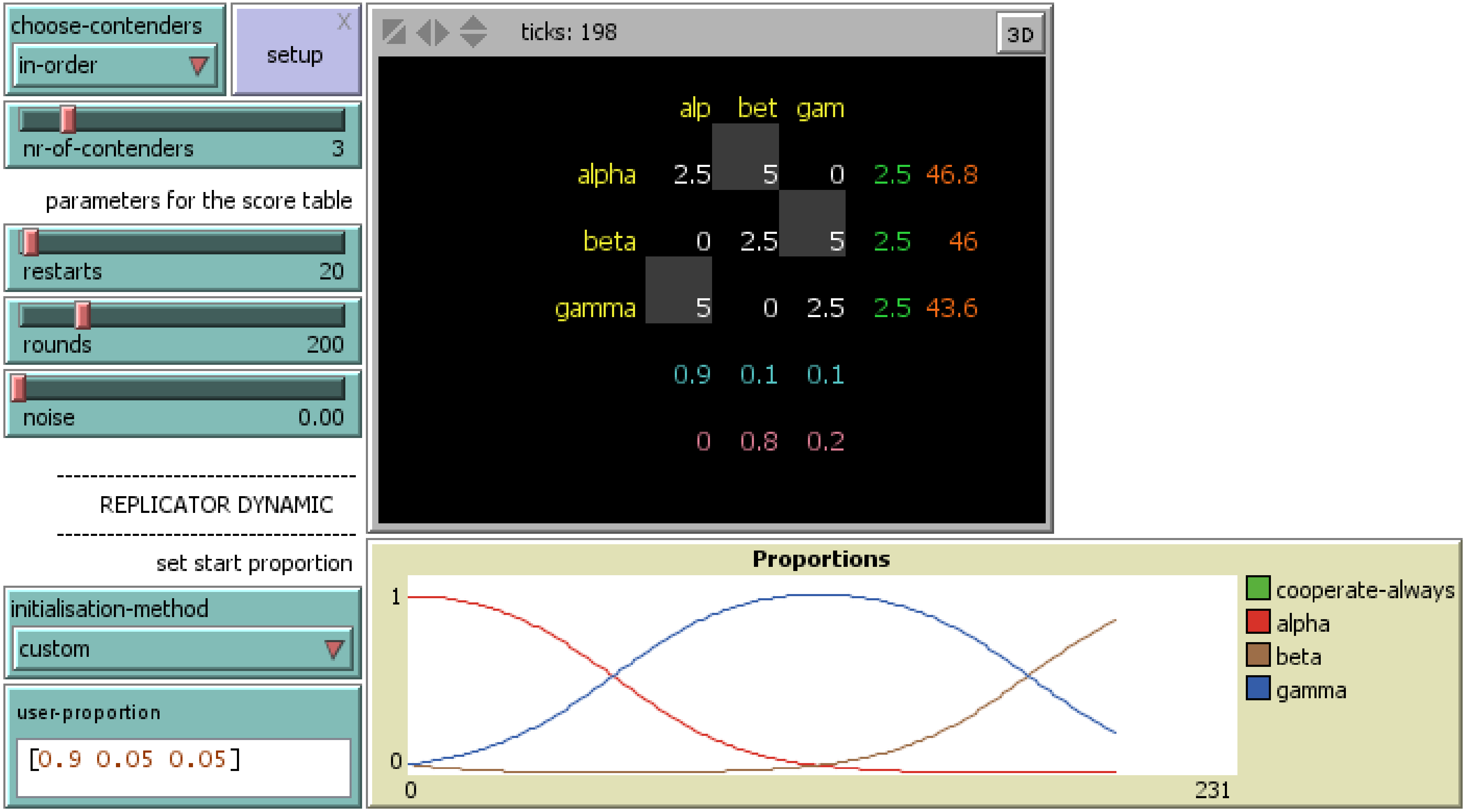
Start proportion is $(0.5, 0.1, 0.4)$.

No convergence in the $\{\alpha, \beta, \gamma\}$-tournament



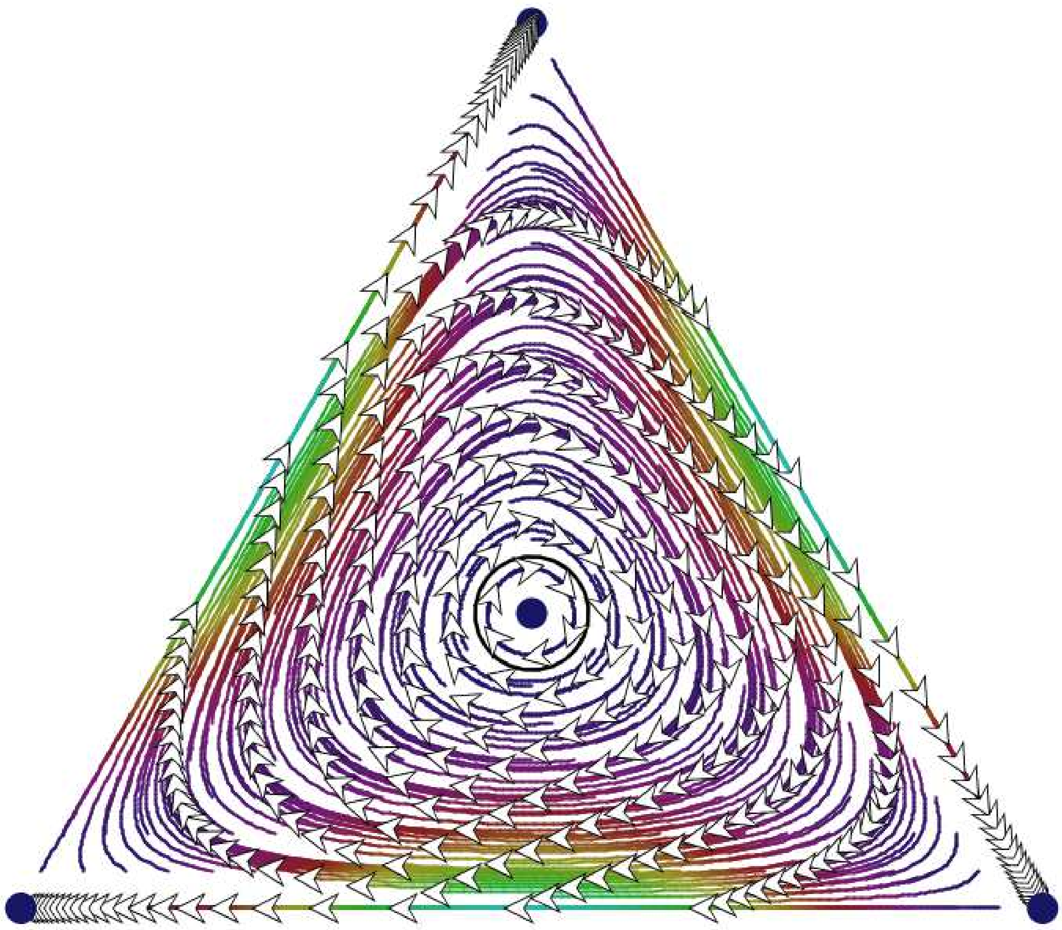
Start proportion is $(0.3, 0.3, 0.4)$, which is close to $(1/3, 1/3, 1/3)$.

No convergence in the $\{\alpha, \beta, \gamma\}$-tournament



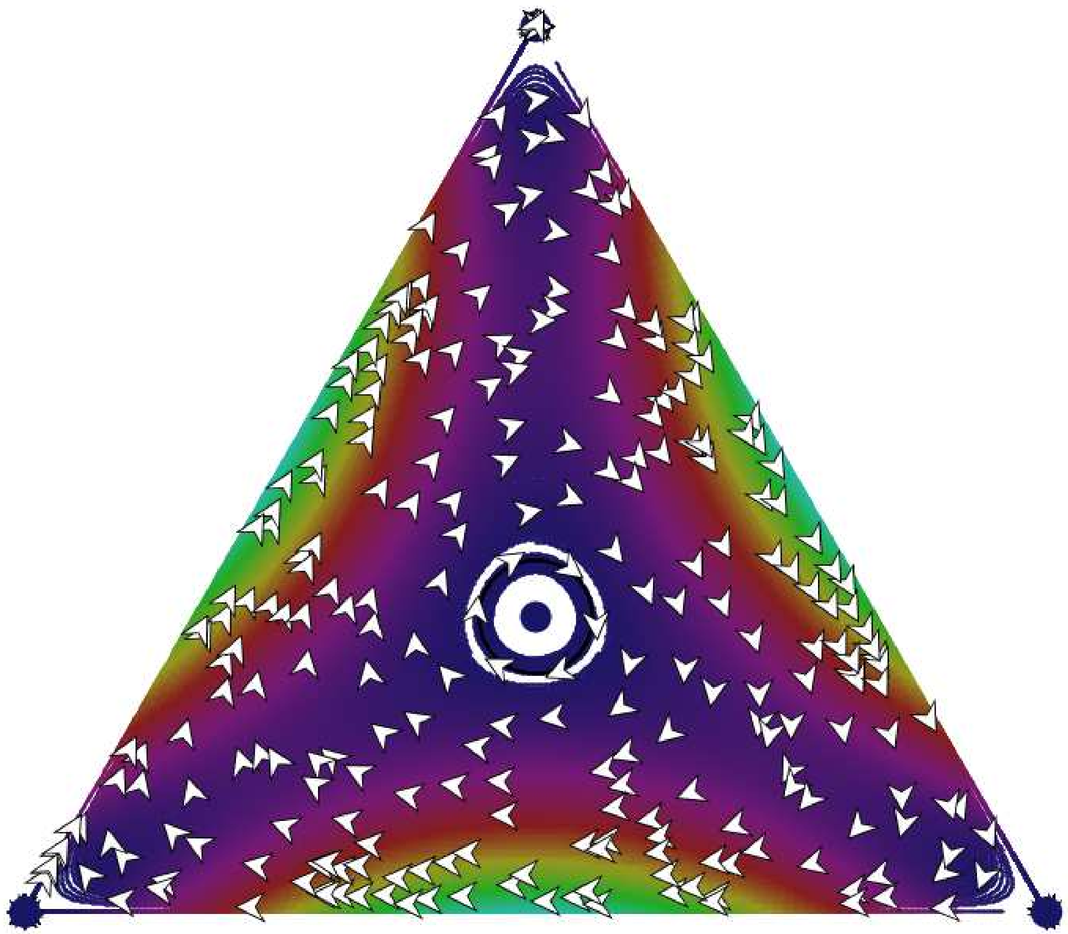
Start proportion is $(0.9, 0.05, 0.05)$, which is far from $(1/3, 1/3, 1/3)$.

No convergence in the $\{\alpha, \beta, \gamma\}$-tournament



Number of iterations $t = 40$. For the record: we have α at $(1, 0, 0)$ and β at $(0, 1, 0)$ but this does not matter because the diagram is symmetric.

No convergence in the $\{\alpha, \beta, \gamma\}$-tournament



Same, but with $t = 700$ iterations of the replicator. Color indicates speed of change.

Conclusion

Conclusion

Conclusion

- You now know:

Conclusion

- You now know:
  - The definition of a grand table.

Conclusion

- You now know:
  - The definition of a grand table.
  - Why a strategy's average row score in the grand table is not a good indicator of its quality *per sé*.

Conclusion

- You now know:
  - The definition of a grand table.
  - Why a strategy's average row score in the grand table is not a good indicator of its quality *per sé*.
  - Why the limit proportion of the replicator dynamic of an interior point in the $(n-1)$ simplex is a much better indicator of the qualities of all strategies.

Conclusion

- You now know:
  - The definition of a grand table.
  - Why a strategy's average row score in the grand table is not a good indicator of its quality *per sé*.
  - Why the limit proportion of the replicator dynamic of an interior point in the $(n-1)$ simplex is a much better indicator of the qualities of all strategies.
  - That, depending on where you start in the $(n-1)$ simplex, the replicator may converge to different proportions.

Conclusion

- You now know:
  - The definition of a grand table.
  - Why a strategy's average row score in the grand table is not a good indicator of its quality *per sé*.
  - Why the limit proportion of the replicator dynamic of an interior point in the $(n-1)$ simplex is a much better indicator of the qualities of all strategies.
  - That, depending on where you start in the $(n-1)$ simplex, the replicator may converge to different proportions.
  - That, for some grand tables, the replicator won't ever converge at all.

Conclusion

- You now know:
  - The definition of a grand table.
  - Why a strategy's average row score in the grand table is not a good indicator of its quality *per sé*.
  - Why the limit proportion of the replicator dynamic of an interior point in the $(n-1)$ simplex is a much better indicator of the qualities of all strategies.
  - That, depending on where you start in the $(n-1)$ simplex, the replicator may converge to different proportions.
  - That, for some grand tables, the replicator won't ever converge at all.
- You still don't know (probably):

Conclusion

- You now know:
  - The definition of a grand table.
  - Why a strategy's average row score in the grand table is not a good indicator of its quality *per sé*.
  - Why the limit proportion of the replicator dynamic of an interior point in the $(n-1)$ simplex is a much better indicator of the qualities of all strategies.
  - That, depending on where you start in the $(n-1)$ simplex, the replicator may converge to different proportions.
  - That, for some grand tables, the replicator won't ever converge at all.
- You still don't know (probably): ...

Conclusion

- You now know:
  - The definition of a grand table.
  - Why a strategy's average row score in the grand table is not a good indicator of its quality *per sé*.
  - Why the limit proportion of the replicator dynamic of an interior point in the $(n-1)$ simplex is a much better indicator of the qualities of all strategies.
  - That, depending on where you start in the $(n-1)$ simplex, the replicator may converge to different proportions.
  - That, for some grand tables, the replicator won't ever converge at all.
- You still don't know (probably): ......

Conclusion

- You now know:
  - The definition of a grand table.
  - Why a strategy's average row score in the grand table is not a good indicator of its quality *per sé*.
  - Why the limit proportion of the replicator dynamic of an interior point in the $(n-1)$ simplex is a much better indicator of the qualities of all strategies.
  - That, depending on where you start in the $(n-1)$ simplex, the replicator may converge to different proportions.
  - That, for some grand tables, the replicator won't ever converge at all.
- You still don't know (probably): . . . . . . see next slide.

Conclusion

You now know: . . .

You still don't know (probably):

Conclusion

You now know: ...

You still don't know (probably):

- How the replicator exactly works.

Conclusion

You now know: …

You still don't know (probably):

- How the replicator exactly works.
- What it does.

Conclusion

You now know: ...

You still don't know (probably):

- How the replicator exactly works.
- What it does.
- How Nash equilibria are related to rest points[2] of the replicator.

Conclusion

You now know: . . .

You still don't know (probably):

- How the replicator exactly works.
- What it does.
- How Nash equilibria are related to rest points[2] of the replicator.
- How compare MAL algorithms more systematically.

Conclusion

You now know: . . .

You still don't know (probably):

- How the replicator exactly works.
- What it does.
- How Nash equilibria are related to rest points[2] of the replicator.
- How compare MAL algorithms more systematically.
- How others, in particular Airiau *et al.*(2007), Bouzy and Métivier (2010), and Zawadzki *et al.*(2014) compared MAL algorithms.

Conclusion

You now know: . . .

You still don't know (probably):

- How the replicator exactly works.
- What it does.
- How Nash equilibria are related to rest points[2] of the replicator.
- How compare MAL algorithms more systematically.
- How others, in particular Airiau *et al.*(2007), Bouzy and Métivier (2010), and Zawadzki *et al.*(2014) compared MAL algorithms.
- The workings of two learning algorithms that are to be implemented in the programming assignment, viz. satisficing play and Bully.

[2]Other names: fixed point, stationary point, critical point, stagnation point, singular point, singularity, steady state. Equilibrium is also used, but this term is of course rather confusing in the presence of Nash equilibria.

What's next

What's next

What's next

- A more serious and in-depth discussion of the replicator dynamic.

What’s next

- A more serious and in-depth discussion of the replicator dynamic.
  - How the replicator works.

What's next

- A more serious and in-depth discussion of the replicator dynamic.
  - How the replicator works.
  - Properties of the replicator.

What's next

- A more serious and in-depth discussion of the replicator dynamic.
  - How the replicator works.
  - Properties of the replicator.
  - That all Nash equilibria are rest points of the replicator.

What's next

- A more serious and in-depth discussion of the replicator dynamic.
  - How the replicator works.
  - Properties of the replicator.
  - That all Nash equilibria are rest points of the replicator.

    That this doesn't hold the other way around.

What's next

- A more serious and in-depth discussion of the replicator dynamic.
  - How the replicator works.
  - Properties of the replicator.
  - That all Nash equilibria are rest points of the replicator.

    That this doesn't hold the other way around.
  - That certain rest points of the replicator, viz. those that are so-called Lyapunov stable, *are* Nash equilibria.

What's next

- A more serious and in-depth discussion of the replicator dynamic.
  - How the replicator works.
  - Properties of the replicator.
  - That all Nash equilibria are rest points of the replicator.

    That this doesn't hold the other way around.
  - That certain rest points of the replicator, viz. those that are so-called Lyapunov stable, *are* Nash equilibria.

That this doesn't hold the other way around.

What's next

- A more serious and in-depth discussion of the replicator dynamic.
  - How the replicator works.
  - Properties of the replicator.
  - That all Nash equilibria are rest points of the replicator.

    That this doesn't hold the other way around.
  - That certain rest points of the replicator, viz. those that are so-called Lyapunov stable, *are* Nash equilibria.

That this doesn't hold the other way around.

- A more extensive discussion of comparing MAL algorithms empirically, including discussion of work of Airiau *et al.*(2007), Bouzy and Métivier (2010), and Zawadzki *et al.*(2014).

What’s next

- A more serious and in-depth discussion of the replicator dynamic.
  - How the replicator works.
  - Properties of the replicator.
  - That all Nash equilibria are rest points of the replicator.

    That this doesn’t hold the other way around.
  - That certain rest points of the replicator, viz. those that are so-called Lyapunov stable, *are* Nash equilibria.

That this doesn’t hold the other way around.

- A more extensive discussion of comparing MAL algorithms empirically, including discussion of work of Airiau *et al.*(2007), Bouzy and Métivier (2010), and Zawadzki *et al.*(2014).
- An explanation of two more learning algorithms that are to be implemented in the assignment, namely, satisficing play and Bully.