

Part 1:

<https://medium.com/commonsstack/the-magic-behind-the-commons-simulator-part-1-game-flow-8fa2bed8437b>

Part 2:

<https://medium.com/commonsstack/the-magic-behind-the-commons-simulator-part-2-initialization-93da1b8b3abe>



Jeff Emmett

3:20 PM Apr 13

Hahaha, yes!

Another way I've heard cadCAD PSUBs described that really helped me is like turns in a board game round.

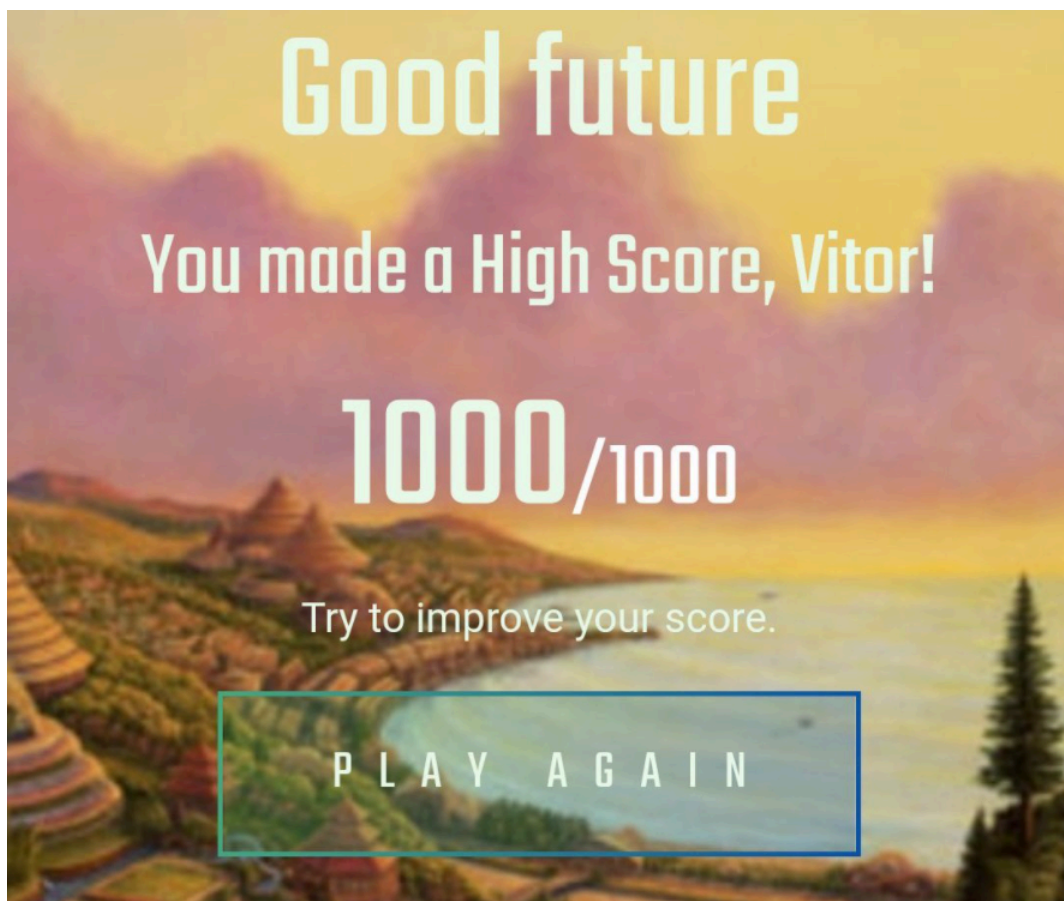
PSUB = Partial State Update Block

Reply or add others with @

The Magic Behind the Commons Simulator - Part 3: The Closest Look

This is Part 3 of 3 our “Deep Dive” into the inner workings of the [Commons Simulator](#) where we get into the details of simulation mechanisms. Much credit to [Lauren Luz](#) for co-authoring this piece with much fun, very deep thought and so amazing writing skills.

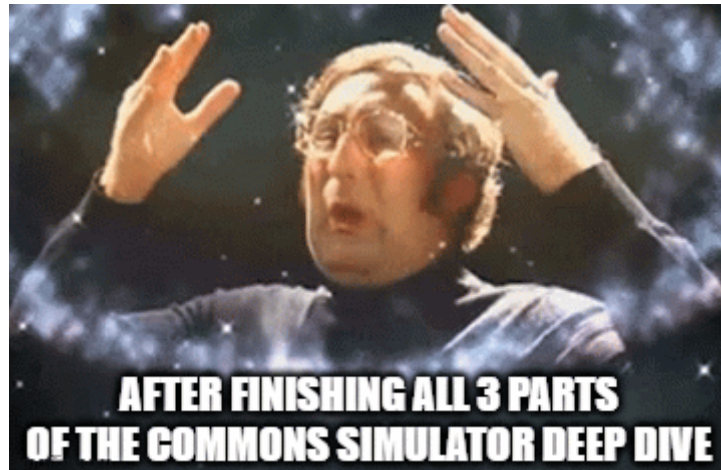
At last, the moment we’ve all been waiting for! This is the third and final part of our series of blog posts unraveling the mystery of the cadCAD backend that goes into powering the Commons Simulator’s artistic, gamified, sci-fi and *fun* frontend. The Commons Simulator is set in a dystopian future where life as we know it on Earth has been destroyed. You, as the game hero, have the opportunity to travel back-in-time with cadCAD to help the [RadicalxChange](#) community design a regenerative Commons that has the potential to change the course of society and save the world from certain doom.



Saving the Future with the Commons Simulator

In the previous posts, we discussed the game flow, the setup, the states of variables as they change throughout simulation, and simulation initialization. This is the necessary foundation required by the system before the real magic begins. If you haven't yet already, we

recommend reading [Part 1](#) and [Part 2](#) of the series in order to truly enjoy the feeling of your mind being blown as the pieces come together with the description of simulation mechanisms below 📌



....

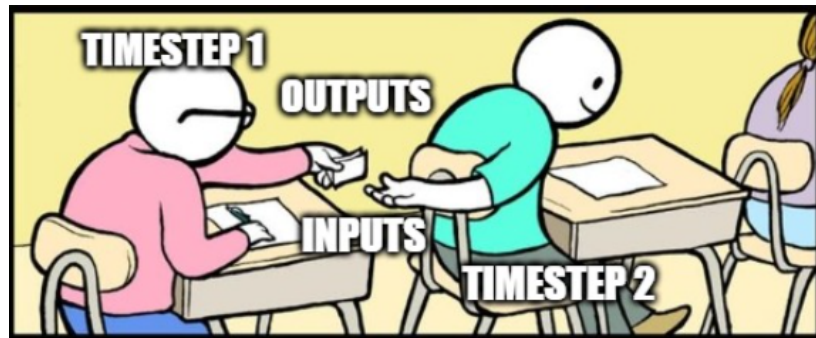
The mechanisms we get into in this piece are really where things get interesting. These are the operations that are performed iteratively and in sequence, changing and updating the states of each of the key variables. In our model, a sequence of 14 mechanisms are executed sequentially during every timestep (“day”) for a total simulated time period of 3 years (or 1095 “days”). When we finally reach the end of the simulation period, 15,330 (14 x 1095) iterations on the states and mechanisms of the model have been performed! This is an amazing demonstration of how cadCAD is used as means for complex system modelling 🤖

So let’s get to it! Finally, for “the Closest Look” at the Magic Behind the Commons Simulator, we proudly present: **The Mechanisms**.

....

Simulation Mechanisms

During each timestep, the following mechanisms are performed in sequence. The output data from each timestep is then passed to the following timestep (as the new input data) and the mechanisms are performed again. The same mechanisms are performed each day, but the data changes throughout the simulation as it gets passed from one timestep to the next.

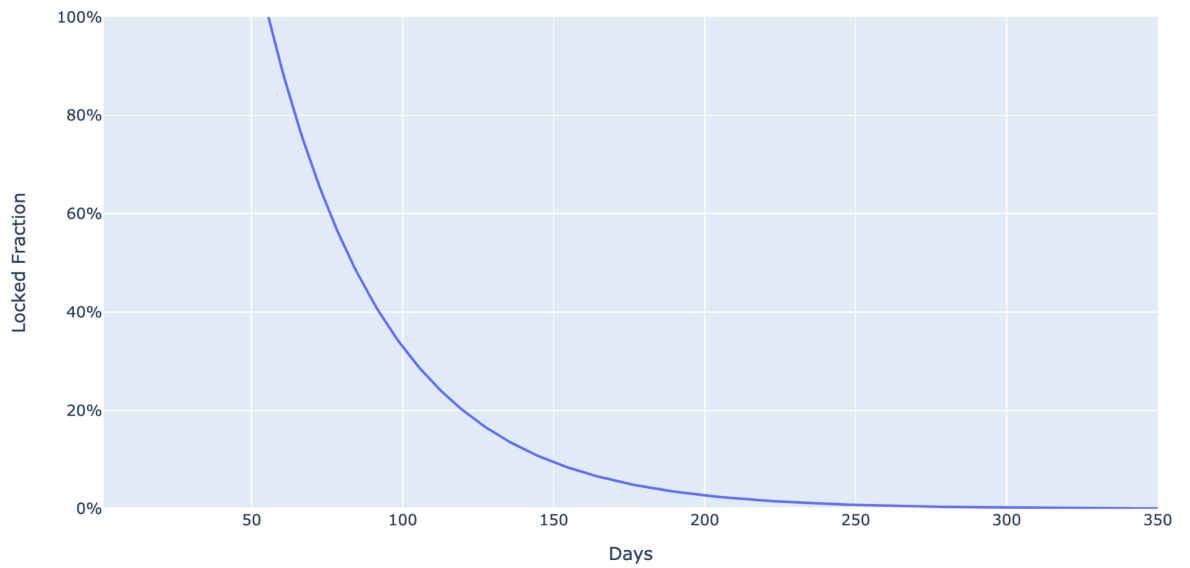


1. Generate new participants

This mechanism mimics the effect of new participants entering the Commons, whether it be in support of the vision and mission, or merely in pursuit of profit from speculated token price increase. New participants can join the Commons at the start of each new day. **The likelihood of people coming in depends on the general sentiment** (excitement or contentment of the community) and the **expectation/speculation of token price increase**.

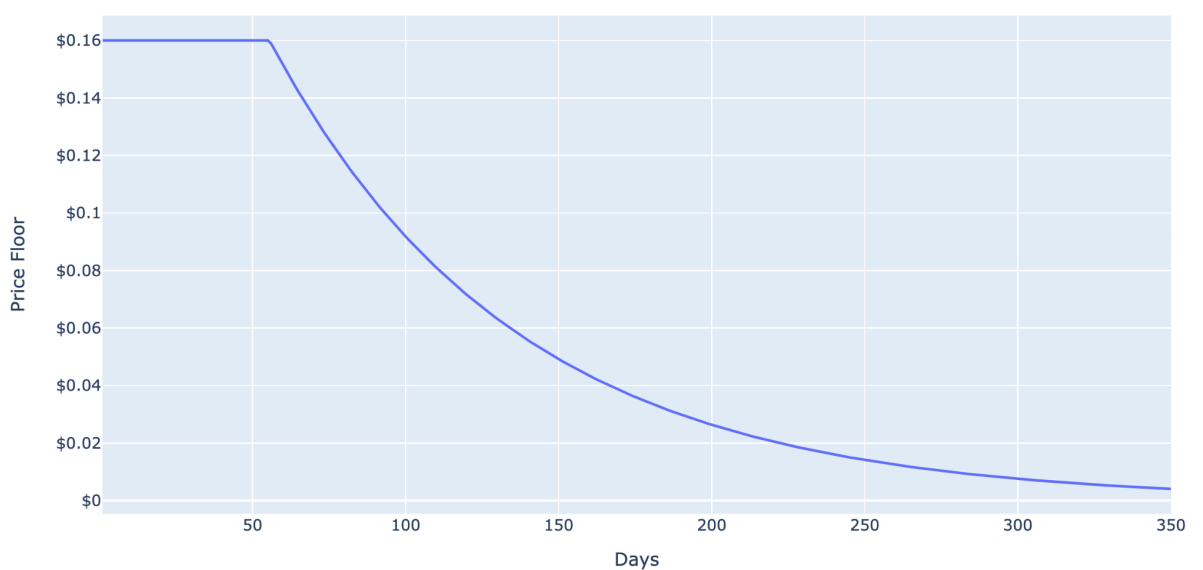
In the early stages of the Commons, a portion of the Hatchers' tokens are locked (i.e., cannot be sold) for a period of time called the "lockup period". As you will soon learn, the lockup period makes the Commons initially more attractive to speculators and therefore increases the likelihood of more participants entering into the Commons in those particular days.

The figure below is a visual representation of the ratio of tokens locked as a function of time in days. During the first 55 days of the lockup period, all of the Hatchers' tokens are locked. Between 55 and 120 days, the Hatchers' tokens are unlocked relatively quickly. At 120 days, 80% of the initial locked tokens are unlocked, and after this point the rate of unlocking tapers.



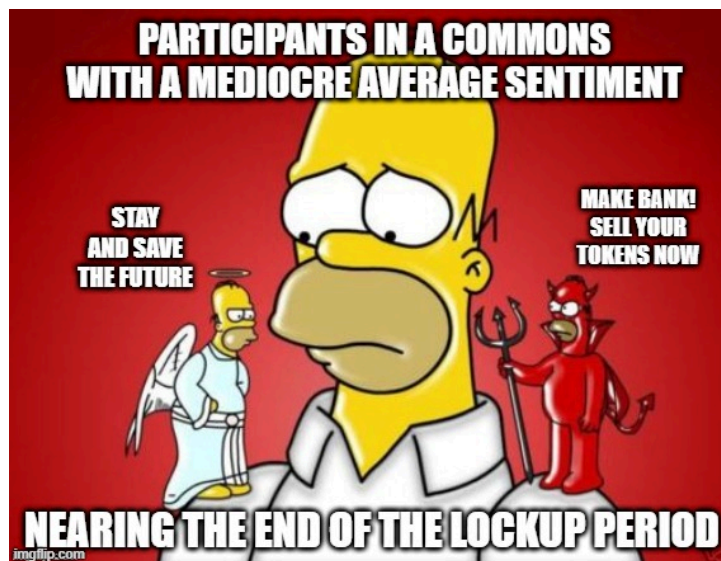
Percentage of unlocked tokens over time

During the early stages more people are likely to join the Commons because they know the price will not drop below a threshold due to the lockup period. The figure below visually represents the token price floor, i.e. the lowest price possible if all the unlocked tokens are burned. Before day 55, the price floor remains at a value of \$0.16 since the Hatchers have their tokens fully locked. From day 55 to 120, the token price floor drops from \$0.16 to \$0.07 when 80% of the Hatchers' tokens are unlocked. When all the Hatchers' tokens are unlocked, the token price floor goes to zero because it is theoretically possible for all of the tokens to be burned.



Minimum possible price over time (as tokens get unlocked)

As more of the Hatchers' tokens become unlocked, speculators are more likely to exit the Commons to take advantage of high token price and high profit potential, and to avoid losses incurred as a result of Hatchers being able to sell their tokens and affect token price. However, the likelihood of larger numbers of people leaving the Commons is in fact more dependent on the sentiment: A high sentiment after the lockup period indicates that people believe in the Commons and want to stay despite profit potential. A low sentiment makes it more likely that people will sell their tokens and exit.



After determining how many new participants enter the Commons, the simulation defines how much each participant will invest to mint new tokens. According to the bonding curve, when new tokens are minted the price of the token increases, so **the influx of new participants raises the token price**. The amount of increase depends on how many tokens are minted, the previous price, and the bonding curve parameters.

2. Generate new proposals



This mechanism represents the action of new proposals being submitted by participants. Every participant, regardless of when they enter the Commons, has the opportunity to generate new proposals. In this mechanism, new proposals are created by randomly assigned participants. The decision of a participant to create or not create a proposal depends on the following:

- How much funding is already requested by other proposals: If there are fewer proposals requesting funds, there will be a higher probability that more people will create new proposals.
- The median affinity of the Commons: Each participant has an assigned numerical affinity for each proposal. The affinity represents how much an individual participant likes a particular proposal. The median affinity is the median for all proposals for all participants in the Commons. A high medium affinity increases the likelihood that an individual participant will create a proposal.
- The Max Spending (Funding Pool): This is the maximum percentage of remaining funds in the Funding Pool that can be used for a single proposal. This is a system parameter and one of the inputs that the player must choose during the game.

After a proposal is created, it is added to the Commons and participants are assigned an affinity for the new proposal. The distribution is made so that many participants will have a low affinity for a specific proposal and only a few will have a high affinity for it. **This basically mimics the**

real-life scenario in which a few people feel very excited about a new project while most people are somewhat indifferent.

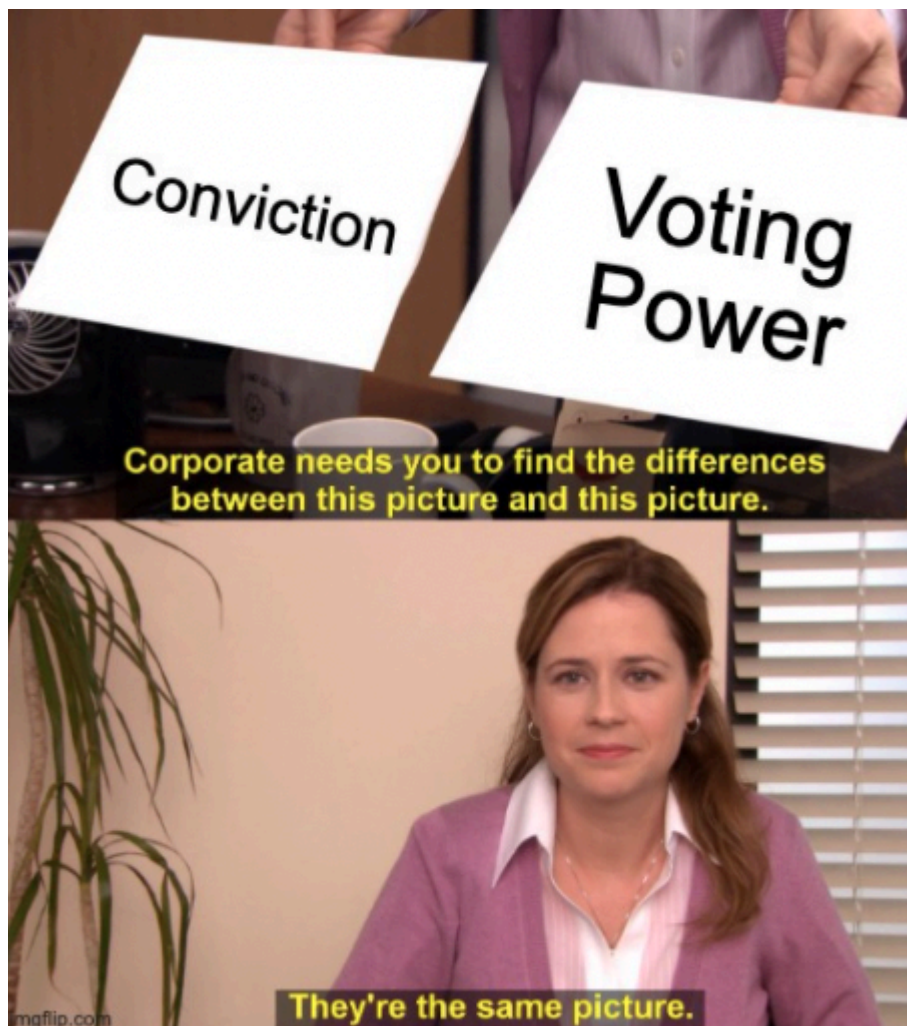
3. Generate new funding via speculation



In this mechanism, additional funds are generated for the Common's Funding Pool as a result of speculation and the system parameter, **Exit Tribute**. The Commons uses an [augmented bonding curve](#), meaning that the token price moves up and down with a deterministic relationship to the token supply. Speculators can enter and exit the Commons purely to buy and sell tokens for profit, without any intention to create or support proposals.

When participants decide to sell by burning tokens, a percentage (the Exit Tribute) of the funds released go to the Funding Pool. This is actually the only way that the Commons can generate funds for the Funding Pool after the Hatch, therefore some amount of speculation is actually advantageous. **A smaller Exit Tribute makes the Commons more attractive to speculators and increases the volume of trade.**

<https://imgflip.com/i/5avv1d>



4. Update proposals' age and conviction thresholds

This mechanism updates the age (in days) of each proposal on each timestep. According to conviction voting, the collective preference for a proposal needs to reach a set threshold in order for the proposal to become active. The conviction threshold depends on two values: the current size of the Funding Pool at the given moment, and the amount requested by the proposal. The conviction voting threshold is higher if a proposal is requesting a large *percentage* of the amount of funds in the Funding Pool at that time.

In the Commons Simulator, the player must set the “Max Spending %” parameter as one of the inputs collected. This is the maximum percentage of funds in the Funding Pool that can be spent on a single proposal at one time. If the percent of the total in the Funding Pool required by the proposal is the Max Spending amount, **then all of the voting tokens would have to be staked behind that proposal for an infinite amount of time for the proposal to reach the conviction**

needed to be funded. Obviously this is only theoretically possible and proposals requesting above the Max Spending will not pass until the quantity of funds in Funding Pool increases to the point where Max Spending is greater than the proportion required by the proposal.

5. Compare proposals' conviction and thresholds

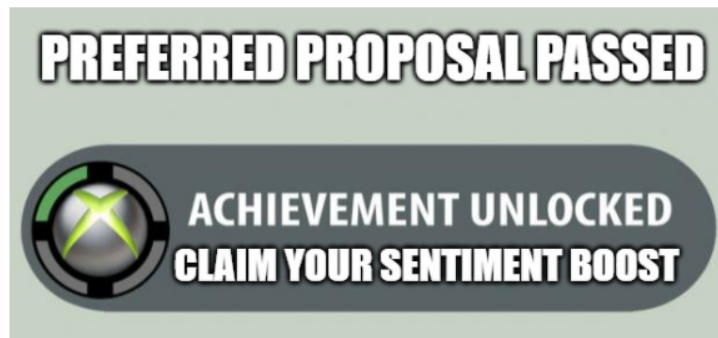


After defining a specific conviction threshold for each proposal, this mechanism calculates each proposal's conviction according to the [conviction voting design](#). Next, it compares each proposal's conviction value to the conviction threshold for that proposal. If the conviction on some proposals is smaller than its threshold, it is simply held as a candidate waiting to be approved in the future. If the proposal's conviction is higher than its threshold, it changes from candidate to active.

When a proposal passes from candidate to active, it receives the requested funds and those funds are deducted from the Funding Pool.

6. Update participants' sentiment when a proposal becomes active

After a proposal becomes active and receives funds, all the participants that have an affinity for the proposal gain a sentiment boost because the proposal they like will be able to be executed.



7. Proposals fail or are completed



At this point, all the proposals that are active have received the requested funds to achieve their proposed goals. This mechanism calculates which proposal completes its goal and which fails. It is out of the scope of this simulation to predict the inner complexities that define whether a specific proposal succeeds or not. Instead, **the simulation uses a probabilistic distribution for failure and success using the heuristic that bigger projects (those with more funds requested) are more complex and thus have a smaller rate of success than smaller proposals.**

8. Update participants' sentiment when a proposal fails or is completed



This mechanism is similar to mechanism #6 where we update the participants' sentiments based on their affinities for proposals that become active. The main difference here is that the sentiment of each participant is modified based on whether or not an active proposal succeeds in accomplishing its mission. Participants that have an affinity for failed proposals experience a decrease in sentiment. The participants with an affinity for successful proposals experience an increase in sentiment.

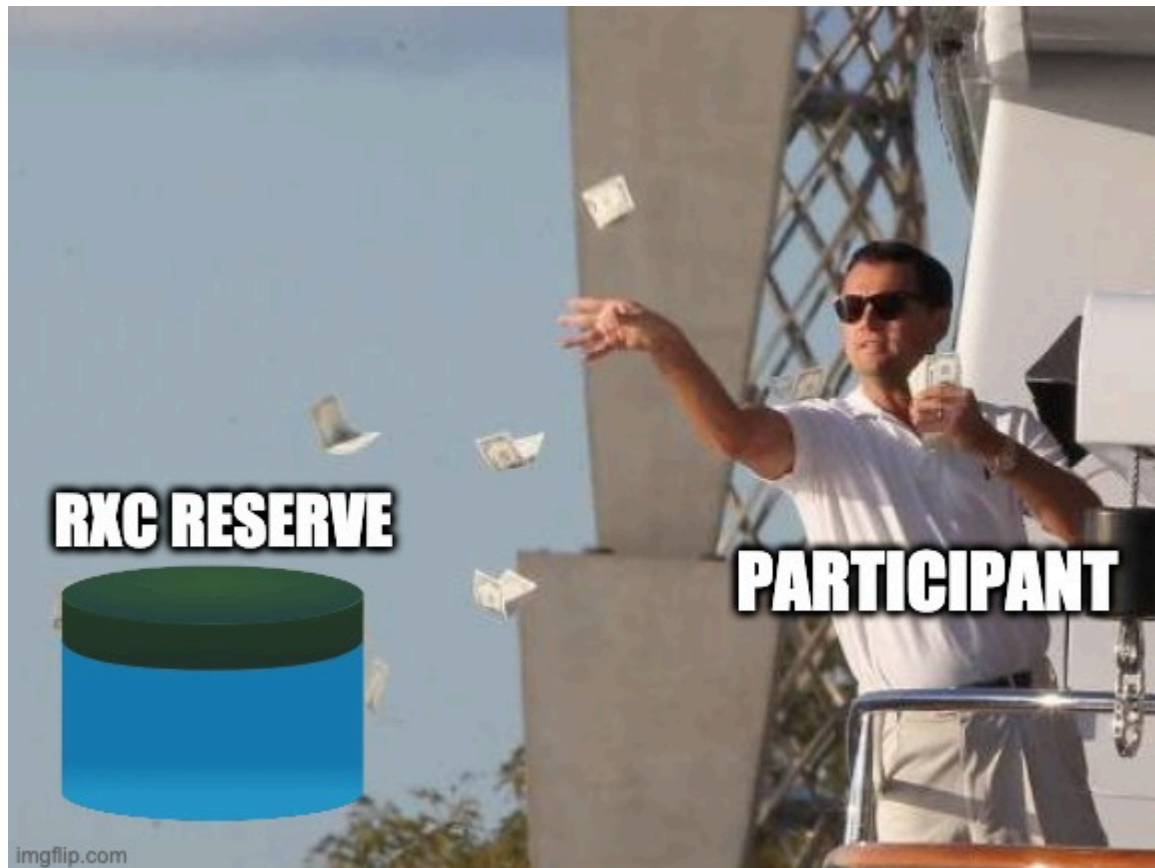
9. Participants vote on proposals according to affinity

This mechanism simulates the voting process in which participants vote on proposals by staking their tokens according to their affinity for each proposal. First, each participant chooses which proposals to vote on based on their affinity for each proposal. After this process, each participant has an associated list of proposals they wish to support, regardless of the number of tokens they will stake. Second, the amount of tokens that will be staked on each proposal is calculated based on the participant's affinity for that proposal and the amount of tokens held by the participant. Both locked and not-locked tokens can be staked on proposals.

10. Calculate proposals' conviction

This mechanism performs the magic of conviction voting, calculating each proposal's conviction every timestep based on the tokens staked by the participants in the previous mechanism.

11. Participant decides to buy tokens



This mechanism simulates the act of buying tokens, whether as a result of speculation or trust in the Commons. Every participant can "buy" tokens in the Commons at any timestep. The simulator uses an augmented bonding curve, so every time a participant buys a token it is minted and the token price increases according to the bonding curve. Participants do not pay fees for minting tokens, but do have to deal with [slippage](#).

On every timestep, the participants must decide whether or not to mint more tokens. The amount of tokens minted depends on the participant's individual sentiment towards the Commons and some constant value engagement multipliers. This means that if the general sentiment of the Commons is low (for example, if there are few proposals getting funding and proposals are also failing to meet their goals) people will not trust the Commons and will be less willing to put in more funds. In the same way, if the sentiment is decaying, few people are buying into the

Commons, and few tokens are being minted, speculators will expect the token price to be bearish and will not have much interest in minting new tokens for profit gain.

12. Participant decides to sell tokens



This mechanism is similar to #11 but in this case, the participants have the chance to "sell" their tokens. Again, due to the bonding curve, the tokens are not actually sold to anyone, but rather are burned and some part of the funds withdrawn from the RxC Reserve are given back to the participant. In this mechanism, the Exit Tribute plays a major role as it is the percentage of the funds withdrawn from the Reserve Pool, "taxed" from the participant, and added to the Funding Pool to fund future proposals.

In this mechanism, each participant must decide whether or not to sell their tokens. This decision is impacted mostly by the general sentiment towards the Commons. If the sentiment is low, it indicates that participants do not believe in the Commons and their projects to support the RadicalxChange movement. The speculators observing a Commons with a low sentiment will expect the token price to be bearish and will begin selling their tokens before the token price drops even more. If the sentiment is high, the participants have pressure to hold their tokens and contribute to the Commons by staking them on the proposals they believe in. Speculators would expect the token price to be bullish as more people enter the Commons and more tokens are minted, and would therefore expect to see profits as the token price increases.

13. Participant decides if he or she wants to exit



In addition to burning part of their tokens through sale, each participant can choose to burn all of his or her tokens and exit the Commons. In this mechanism, we simulate the extreme case in which a participant does not want to participate anymore.

In every timestep, each participant has a “chance” to exit the Commons. For that to occur, the individual sentiment of a participant must be lower than a predefined threshold, e.g., 50%. If the participant’s sentiment is lower than the sentiment threshold, he or she has a chance to leave the Commons. This chance is defined probabilistically based on the sentiment and other predefined engagement rates. Two participants with sentiments lower than the threshold, e.g. 15% and 40%,

have different chances of leaving the Commons and the participant with the lower sentiment is more likely to leave.

When a participant leaves the Commons all of his or her tokens are burned and the appropriate funds are returned to the participant (after the Exit Tribute is discounted and sent to the Funding Pool).

14. Participants' sentiment decays

This mechanism simply decreases every participant's sentiment by a small value every timestep, representing the natural decaying of sentiment as time passes and the hype of the launch of the Commons decays. This creates a need for new exciting things to happen in the Commons to keep the participants engaged.



An example of a situation with a large declining sentiment is a Commons with very few participants and very many proposals. In this scenario, no proposals can pass the conviction threshold and become active. Therefore, no projects can succeed, no progress can be made to advance RadicalxChange initiatives, and the overall sentiment of the Commons just decays.

....



Wow, you made it!

Thank you so much for reading to the end. We hope that this piece brought you a bit more insight into the inner workings of the Commons Simulator and (hopefully!) that this inspires you to go a little further down the rabbit hole of exploring [cadCAD](#) as a tool for modeling complex systems.

The Commons Simulator aims to demonstrate the potential of simulation as a design tool for empowering communities to design their own economies. We wanted it to be beautiful, to be interactive, to be educational and to stimulate conversation around the topic of using technology to bring people together around initiatives that lead to a regenerative future.

So what are you waiting for? Show us what you learned by [playing the game](#) and sharing your score! Or help us build the next level by supporting us with a 👉👉 [donation on Giveth.io](#) 👉👉 You could also [join our team](#) or [check out our repo on GitHub](#) to get involved.

Don't forget to join the [Trusted Seed](#) so you are eligible to participate in the real upcoming [Hatch of the TE Commons](#)!

Thanks to Lauren Luz, Griff Green, Danielle Gennety, Santiago Gonzalez Toral, Merlin Egalite, and Marko Prljic for content, edits, and suggestions on this piece.