



## Computational Governance for Shared Living

Daniel Kronovet, Joseph DeSimone, Seth Frey

[kronovet@gmail.com](mailto:kronovet@gmail.com), [thedesimone@gmail.com](mailto:thedesimone@gmail.com), [moctodliamg@gmail.com](mailto:moctodliamg@gmail.com)

Github: <https://github.com/zaratanDotWorld/choreWheel>

Talks: [Slack Bot Boarding House \(video\)](#), [Chores: a 21st century chore wheel \(video\)](#)

*NB: This project is now known publicly as “Chore Wheel”*

## I. Introduction

As housing costs continue to climb, the development of quality, affordable housing remains a continual challenge. It is increasingly apparent that shared living situations – housing configurations where residents have private bedrooms but share common facilities like kitchens, bathrooms, and living rooms – are an important part of the solution.

Perhaps the greatest benefit afforded by shared living situations is the avoidance of redundant infrastructure (e.g. one large kitchen, rather than three small ones), which drive down costs. However, the provisioning of common housing resources introduces new coordination challenges (e.g. “who does the dishes”). Historically, such challenges have been overcome through informal norms (e.g. a “dish-zero” rule), deliberative decision-making processes (e.g. house meetings), or basic coordination mechanisms (e.g. an analog chore wheel). But such solutions are unreliable, burdensome, and often too simplistic to meet the needs of a large and varied population. As Oscar Wilde famously quipped, “the trouble with socialism is that it takes up too many evenings.” We can do better.

Mirror is poetic technology: a [suite of tools](#) meant to support the healthy functioning of a shared living environment. Drawing influences variously from cognitive science, computer science, electoral theory, economics, cybernetics, and game design, it functions with four top-level design goals:

1. No managers or privileged administrative roles
2. Simple and intuitive inputs
3. Humans for sensing and judgment, machines for bookkeeping
4. Continuously available, asynchronous processes

Although many of these principles can benefit social structures outside of shared living, we have intentionally chosen this setting for the specific advantages it provides. For example, and by

contrast, a government workplace would likely also benefit from simple and intuitive inputs. But the work done there, being *linear* (i.e. novel, creative) in nature, makes leaderless organization deeply challenging. In a residential environment, much of the work is *cyclical* – continuous, repetitious, and not meaningfully different between iterations – more naturally allowing for specialized solutions which deemphasize novel ideation and emphasize resource balancing and peer accountability.

Further, unlike the highly distributed and anonymous settings of online communities, residential environments, by virtue of being shared physical spaces where participants spend a significant amount of their time, provide arguably the maximal number of opportunities for the informal, out-of-band communication essential for eliciting empathy and identification, and building relationships and friendships. As such, we should *assume* the existence of a coherent social sphere, with the technical system merely providing measurement hooks into accountability and enforcement logic. The aim, then, is the creation of an objective external representation that closely *mirrors* the subjective inner state (e.g. organic norms and culture) that we take as a given.

Note that Mirror does not claim to capture all ideation, decision making, and deliberation necessary for a shared living environment. Nor does its use preclude the need for ongoing investments in education and culture. Rather, it takes its cue from the Pareto principle: a set of simple, general processes which, given a reasonably trained population, manages the most common 80% of scenarios, leaving the remaining 20% to be handled by locally-determined, informal processes.

## Design

### Three Institutional Layers

The overall design of Mirror can be understood in terms of three layers, evoking the three layers described in Elinor Ostrom's seminal *Governing the Commons*. The first, or **constitutional layer**, involves the design of the modules themselves. In this first layer, the design of the entire system and its implementation are up for discussion. There are no constraints, as software can be changed in arbitrary ways. The constitutional layer can be understood as governing the system *from without* by changing rules themselves.<sup>1</sup>

The second layer, the **political layer**, involves participants collaboratively setting explicit parameters that govern the behavior of the system. An example would be choosing the frequency with which a certain chore is to be performed. In the political layer, residents have

---

<sup>1</sup> Note that while future iterations of Mirror will be developed in collaboration with residents and informed by the experience of practice, this first version is being developed by a small team of social scientists with deep experience in shared living.

control over the system's behavior, but only within the constraints set by the constitutional layer. We can think of this as governing the system *from within*.

Third and finally, the **operational layer** involves residents *individually* interacting with the system given the constraints created by the constitutional and political layers. In this third layer, residents complete and verify chores, vote on issues, and procure supplies.

This three-layer design is meant to balance flexibility with simplicity – keeping daily interactions clear and straightforward, and providing residents with a structured means for shaping and controlling their environment, while still allowing for unstructured, open-ended changes to be made as needed.

### Cheap Information

A guiding motivation for Mirror is the reduction of the cost of information. As observed in *Governing the Commons*, the cost of information is inextricably linked to the design of the system itself. A well-designed system, which makes high-quality information cheaply available, will lead to consistently higher-quality decisions and thus better outcomes. Mirror achieves this by placing an “event stream” at the center of every module. Every action, ultimately an attempt to claim some house resource, creates an event. This can then be interacted with by all residents, most simply in the form of an endorsement or a challenge.

### Permissionless by Default

A major design motif for Mirror is “permissionless by default.” Whenever possible, synchronous voting should be avoided. In practice, this means that most actions take the form of challenge-response. In such a system, any resident can propose an action (e.g. such as making a purchase out of a shared account). If there is no response to the proposal by other residents, the action will be allowed – and likely occur – after a set period of time. This will be recorded as having passed with a vote of 1-0, representing implicit consent. However, if other residents do not abstain, they may either oppose or support it with their own votes.<sup>2</sup> For major actions, a minimum number or percentage of votes in favor may be required, so as to encourage residents to “do their homework” and establish support prior to initiating the vote.

This approach allows uncontroversial actions to go forward unimpeded (due to a lack of opposition), while allowing for controversial actions to be decided by vote. This “lazy consensus” approach mimics the processes successfully practiced by groups such as the [Apache Software Foundation](#) and [Wikipedia](#). To both discourage initiating frivolous voting and encourage

---

<sup>2</sup> An open research question is whether this type of dynamic vote should have vote counts displayed publicly during the vote, or kept secret until the end. While often vote counts are kept secret, in this case we have them public, so residents know if they need to “drum up more support” for their side. This is somewhat akin to how the US Senate operates. We hypothesize that for small, known communities, public vote tallies are a useful source of information.

participation in out-of-band communication, residents who propose failed actions will receive a small penalty.

### Chat-based Interface

A second major design motif for Mirror is an orientation around chat-based interfaces. It is currently being developed as a set of Slack applications but is, in principle, portable to Discord, or any extensible chat platform. The vision is for residents to interact with Mirror via a series of chat bots, allowing governance interactions to occur seamlessly alongside other house communication. Each module lives in a dedicated channel and interacts with residents via an events log, which is a series of messages providing information and interactivity. To avoid spam in these channels, they will be read-only for residents. However, residents may add comments and reactions to help keep them engaged with the channels without disrupting their utility. Organizing all interactions as events in a log has positive knock-on effects for auditability and reliability, as any specific state can be reconstructed from the underlying event stream.

### Anonymity and Identity

One critical design consideration is the appropriate role and degree of anonymity. What actions must be taken publicly and which can be private? No one should have to respond to anonymous criticism, yet publicly identifying oneself can be intimidating and thus disenfranchising. Ultimately, we choose to require identity for *initial* actions (e.g. completing a chore, issuing a challenge, or making a purchase), but allowing all *votes* to be anonymous. In this way, at least one person is always linked to any action but the majority of the inputs can be private.

### Subjective Inputs

Last but not least, Mirror chooses to use only *subjective* inputs. This means that explicit surveillance is not necessary, and communities using Mirror can sidestep invasive measures practiced elsewhere such as mounting a camera behind the sink to see who leaves dirty dishes. Such explicit information-gathering approaches create an uncomfortable environment, turn the home into a public sphere, and introduce a new class of measurement error. The constrained physical environment allows for frequent eyeballs to perform the same monitoring function in a more pleasant, less invasive way, while also providing a few degrees of discretion (e.g. “wobble room”).

## II. Modules

Mirror (currently) encompasses the following modules:

- *Chores*, for keeping things clean,
- *Hearts*, for managing behavior,
- *Props*, for supporting spontaneity,
- *Rules*, for establishing norms,

- `Things`, for managing supplies,
- `Art`, for managing common space,
- `Hangs`, for managing events,
- `Subs`, for handling subletters,
- `Friends`, for supporting recruitment.

In the spirit of good game and learning design, Mirror's modules are designed to be adopted progressively. That is to say, new members need not be introduced to the fullness of the system at the outset and can get by with a minimum of interactions, picking up more advanced features as needed.

## Chores

*Should feel **better** than: passive-aggressive frustration, confronting roommates about dishes.*

*Should feel **worse** than: everyone spontaneously and generously contributing equally.*

*Everyone wants a revolution, but nobody wants to do the dishes — Dorothy Day*

The `Chores` module is the centerpiece of the Mirror suite, designed to allow groups to collaboratively, flexibly, asynchronously, and intuitively coordinate repeated tasks. It is a trope that roommates fight over dirty dishes, but one that reveals a fundamental truth: it is a significant challenge to maintain a common space in an equitable manner. We can think of existing chore systems as existing on a freedom-security continuum, with a structureless “do-ocracy” at one end, followed by a simple chore wheel, and then by a more involved chore schedule. In this frame, `Chores` can be seen as a fundamental advance, simultaneously providing high levels of freedom and security.

The political layer of `Chores` involves residents determining the set of tasks to be completed and then attributing relative weights by importance. Any resident can propose a new chore (defined as a string of text) be added to the set or for that one be removed via a challenge-response flow. If after three days the proposal passes, the change is made.

Determining the relative importance of the chores is done via pairwise preferences, with some inspiration coming from spaceship simulators such as *FTL: Faster than Light*. Think “more power to the engines!” Rather than have residents assign percentage weights to each chore, residents can choose in a pairwise way to assign *more* value to one chore and *less* to another. These pairwise preferences can then be aggregated and translated into numerical weights, using the techniques described [here](#) and [here](#). The intention and hope is that this process will be more intuitive than sitting and thinking abstractly about the value of all chores simultaneously, and reduce the burden of contemplating the overall system when changes are made to the chore set. Weights can be adjusted continually *as needed*. For example, if a resident feels that dishes are being overvalued relative to sweeping then she may specifically reallocate priority from sweeping to dishes, leaving the rest of the priorities unaffected. The aggregate weights are

a result of every resident's inputs, meaning that new residents are immediately franchised and able to exert a degree of influence from day one.<sup>3</sup>

Once these political questions are resolved, the operational layer of *Chores* asks residents to complete chores such that the total value of their chores meets a monthly token-denominated requirement (e.g. 300 tokens per month). Rather than create and enforce a chore schedule – necessitating a schedule-maker and creating the problem of “missed shifts”, a common phenomena in which a resident fails to perform their scheduled chore, creating orphaned work – *Chores* allows anyone to complete any chore at any time. To enable this, chores are not scheduled on a calendar (e.g. “Monday night dishes”), but rather assigned a value that *accumulates over time*. Residents claim this value by doing the chore. Once completed, the value resets to 0, reflecting the fact that no work needs to be done, and begins to accumulate again at a rate conferred by the weights determined at the political layer. In this fashion, residents have the flexibility of doing the work they want, when they want, while simultaneously ensuring that all residents make similar overall contributions (as determined by token values), without the need for a schedule.

When a chore is claimed, an event is posted to the *Chores* channel. Residents may optionally *verify* or *challenge* the chore's completion via anonymous voting, with Slack threads allowing for unstructured discussion. Non-response is interpreted as implicit acceptance and a minimum of two positive votes is required, analogous to having a second resident “sign off”, or verify, a chore. This subjective evaluation system is lightweight and allows for norms to enter the system – the assessment is based less on an absolute standard and more on whether people *mostly feel fine* about the work that was done. In cases where it is difficult to reach consensus, small interventions such as posting pictures of the completed chore or creating simple one-page “how-to” guides enumerating the expectations for each chore can aid observers in determining completeness. In the worst case, if it proves chronically difficult to reach consensus on a chore, that chore should be re-defined until it becomes sufficiently easy to determine completeness and performance quality.

This system has a number of positive effects. **First**, the incorporation of an auction mechanism means that unpopular chores will be done less often or assigned a higher weight. As such, residents doing unpopular chores will receive more tokens, naturally balancing the subjective cost of an unpleasant chore with a higher reward.<sup>4</sup> **Second**, the use of a monthly token requirement means that residents have significant flexibility in when they work. Some residents might want to do a little work every day whereas others may prefer to work in a few long spurts.

---

<sup>3</sup> A key consequence of this *ad-hoc* approach to chore weighting is that although the number of pairs grows *quadratically* in the number of chores, residents are not expected to pick pairs at random. Rather, residents will initiate this flow only when an under-valued chore is identified. As such, the number of meaningful options is *linear* in the number of chores, since the resident must only choose the chores from which to take weight.

<sup>4</sup> Amusingly and initially unbeknownst to the authors, a similar but more rudimentary version of this scheme was used by the visionary community of B.F. Skinner's *Walden Two*.

Both of these styles and everything in between are naturally supported. **Third**, the system is agnostic to the frequency with which chores are done. If dishes are weighted at 30 tokens per day, there is no distinction from the system's perspective between doing dishes once for 30 tokens and three times over the day for 10 tokens each. This creates ample room for the residents' subjective judgment on when and how they work. To prevent the cognitive overload caused by many very small chores – imagine someone claiming dishes twenty times in an hour – filling up the log, we can impose a minimum cooldown window (three hours or so). **Fourth**, the system creates a natural pressure to *do work*, as chores that are subjectively over-valued will be claimed quickly in a sort of labor arbitrage.

Two examples will make these dynamics clear. In the **first**, both Aoki and Boris like clean bathrooms but hate scrubbing toilets. As a result, they both assign large weights to “bathroom clean” so that it receives significant funding, incentivizing others to do it. Corinne doesn't mind nearly as much so she gladly cleans the bathroom, claims the large payout and meets her obligation in a fraction of the time. In the **second**, both Dennis and Eowyn like doing the dishes as they find it meditative. They find themselves in friendly competition, doing dishes even when the token bounty is low. As a result, the kitchen remains remarkably clean at a low cost, and tokens can be re-routed to other chores that are less intrinsically pleasant to the household.

A market-based coordinating mechanism is not without drawbacks. For all its faults, a chore schedule does provide a feeling of structure and regularity. In contrast, a market mechanism may create a feeling of transactionalism, and inter-resident competitiveness (e.g. “oh no, you cleaned the living room? I was about to do that...”). To mitigate this, we allow residents to “reserve” a chore in advance, holding a chore for some number of hours. To balance incentives, the value of a reserved chore should be frozen at the time the chore is reserved. The earlier I reserve a chore, the more value I give up. This disincentivizes “camping” on chores, or holding high-value assets merely to deny them to others. Further, since a reserved chore can now be “missed”, if a resident does not complete the chore by a given time, they receive a half-heart penalty (see *Hearts* below). This approach could be extended further to allow residents to put reservations on “repeat” at some frequency. Note that this does not guarantee the *value* of the chore each time it is reserved.

To mitigate feelings of transactionalism, we impose a relatively unintuitive exchange rate between tokens and the prevailing currency. Ultimately, every token has a dollar value, as the chore requirement is underwritten by a cash deposit, but residents should be discouraged from thinking in dollar terms to any extent possible. The hope is that the tokens can be seen as an internal community currency used for bookkeeping and accounting rather than as direct payment for work. An exchange rate of 3:2 seems like a reasonable place to start here, confounding simple doubling or halving.

The conversation concerning the relative pros and cons of explicitly valuing work has a long and nuanced history and no clear best answer. In all likelihood, a hybrid approach will be best. Note that the design of *Chores* in no way precludes or prevents the long-term development of a

socially-contained superstructure of norms and habits, by which residents develop routines and knowledge of one another's preferences. In fact, an ideal outcome would be to have the flow of work become frictionless and mutually beneficial, merely underwritten by an explicit accounting. It will be interesting to observe the patterns in which such structures develop.

## Future Directions

### Non-monetary Enforcement

While having *Chores* points be resolved into rent credits and subsidies allows for desirable emergent behaviors (e.g. allowing residents with busy, high-paying jobs to indirectly subsidize those with flexible, lower-paying jobs via differential chore contributions – the “busy lawyer supporting the aspiring writer” scenario), it is careless to assume that the system will always reach a mutually satisfactory equilibrium. It is possible, even likely, that such a scheme could lead to abuse and exploitation with less affluent residents becoming de-facto servants for their wealthier “peers”. Although the likelihood of these bad outcomes is unknown, we have taken preventative steps to curtail their potential emergence.

One way to avoid this problem is to route points into an intermediary asset or currency. In our case, we use hearts. Residents who fall short of their monthly chore points requirement would lose hearts, essentially a reputational currency representing responsible accomplishment of household duties, rather than pay additional rent. Alternatively, residents who have a points surplus would earn back hearts, representing superior commitment to household wellbeing, rather than receive a rent credit. This approach places *Chores* squarely in a symbolic space where residents cannot buy their way out of labor.

This approach is attractive in that it insulates the domestic sphere from the heterogenous economic conditions of the residents, putting everyone on an equal playing field. It is, however, limited in that it precludes the flexible, emergent behaviors that made points-as-cash so appealing in the first place. Ultimately, neither scheme is inherently better or worse. The success of each method depends on the particular cultural context of the household.

Fortunately, the choice is not binary, as we can subject markets to democracy. To do this, we introduce a “market-index” parameter, ranging from 0 to 100, which governs what percent of a chore obligation can be paid for with actual currency. This can either be set by a vote or as the median of continually adjustable individual inputs. A market index of 100 gives the pure market approach initially described, an index of 0 gives the purely symbolic system, while an intermediate value mixes the two: a value of 50 says that no more than half of one's chore obligation can be paid for.



## Forgiveness

Currently, residents who do not meet their full obligation suffer some penalty. It is worth considering whether including some non-penalized margin-of-forgiveness (e.g. 10% of the total requirement) would increase the perceived legitimacy of any penalties.

## Hearts

*Should feel **better** than: uncomfortable house meetings, feeling helpless and lacking agency.*

*Should feel **worse** than: everyone spontaneously getting along.*

The `Hearts` module is a general-purpose norm-management tool, supporting residents in maintaining standards of conduct. A key thesis for `Hearts` is that although people may be reluctant to issue monetary fines or permanent-feeling penalties, they may be more comfortable dealing with symbolic, temporary warning signs. Broadly, `Hearts` is meant to stand in for the rehabilitative (and punitive) functions of a house meeting, allowing for behavioral warning and reward to occur asynchronously, via a mechanism that is symmetrically available to all residents.

“Hearts” are an arbitrary unit drawn from the video-game vernacular (e.g. *The Legend of Zelda*). Each resident begins with some number of hearts (e.g. five), and loses or gains them as the result of various processes and actions. Losing some of your hearts causes you to lose access to certain modules (e.g. making purchases via `Things`), a type of progressive disenfranchisement giving a tangibility to the penalty. Losing all of your hearts signals serious behavioral issues meriting a more significant intervention and a large fine, which can optionally be put towards counseling.

More so than the other modules, `Hearts` relies *explicitly* on culture to function. Residents must be comfortable issuing challenges when expectations are violated – ultimately, the entire system both depends on and cultivates the capacity for individuals to hold each other accountable. Although this can be seen variously as a bug and a feature, we see it as a feature: rather than create a system in which residents have no responsibility (and thus no agency or opportunities for growth), we create one that rests on the willingness to initiate action. Mirror is not an authority. It is instead a lightly-structured reflection of the authority of the residents. As James Scott observes in [Seeing like a State](#):

*Formal order... is always and to some considerable degree parasitic on informal processes, which the formal scheme does not recognize, without which it could not exist, and which it alone cannot create or maintain.*

In this sense, houses using Mirror become [training environments](#) for building up capacity for civil society, as the ability to speak up to correct a perceived wrong is developed through continuous practice. As Elinor Ostrom observes in her seminal *Governing the Commons* (p. 94):

*Even though it is frequently presumed that participants will not spend the time and effort to monitor and sanction each other's performances, substantial evidence has presented that they do both in [common-pool resource] settings.*

That said, minimal usage of Hearts does not necessarily indicate a dysfunctional culture – it could equally indicate a highly functional culture, in which norm violations are minor enough to be handled more informally or are simply so rare as to not require frequent use of explicit sanctions. Conversely, high usage of Hearts does not necessarily indicate a healthy culture. There is as much potential to use Hearts in a passive aggressive way as there is to use it to simply but honestly and bravely confront and correct misbehavior. It is our expectation that Hearts will not be used for small, one-off infractions, but rather as a means of addressing serious, chronic, and broadly-understood misbehavior. Engaging with the system will ideally occur only when a problem becomes too difficult to broach through informal means, helpfully mitigating the abuse potential of the system.

The core of Hearts is a simple but general challenge-response flow. At any time, any resident may publically challenge another, describing some norm violation that occurred. This may be in reference to explicit house rules, but that is not a necessary condition of a challenge. The other residents then vote anonymously on the issue, protecting them from retaliation and encouraging honest assessment over interpersonal affinities. Depending on the final vote tally, either the challenger or the defendant will lose a heart. For the first three hearts, a minimum of 40% positive votes are required. For the final two hearts, a minimum of 70% positive votes are required. This increasing quorum allows small groups to more easily put up symbolic “warning signs”, while still requiring something closer to full consensus to impose meaningful penalties.

Note that this functionality is not designed as an end-to-end rehabilitative process. Instead, it is meant as a final piece of bookkeeping. The more the incident at hand has been discussed and understood in advance, the more fluid this final bookkeeping step will be. The minimum of positive votes acts as an important asymmetric filter. Since the issuing of a challenge is not free given the emotional labor created for all residents, explicit challenges should be discouraged before informal discussions have been had and positive support established.

Hearts “regenerate” at a rate of one per month (at minimum).

The goal of Hearts is to allow residents to set and enforce norms and standards of conduct, provide a “release valve” for social tension, and simultaneously protect individuals from abuse. The symbolic and regenerative nature of hearts will hopefully make it less emotionally taxing both for residents to issue challenges and for people to receive penalties – a hypothesis which must be proven out in practice.

## Future Directions

### Depersonalized Norm-setting

As the principal norms-enforcement mechanism, `Hearts` can be extended in various directions.

As mentioned, `Hearts` is agnostic to the presence of explicit rules. As a consequence, it is possible that `Hearts` will feel *too* general-purpose, bordering on inhumane, inasmuch as personal challenges can come seemingly unprovoked and out of nowhere. Although implicit structure could develop to address this (such as the emergence of informal house leaders who issue challenges on behalf of shyer housemates, in return for social esteem – in Ostrom’s words, “private benefits are allocated to those who monitor”), an explicit norm-setting mechanism may prove useful. Such a mechanism helps to *explicitly* determine community norms in advance, in a depersonalized, incident-agnostic way, making it easier for residents to collectively and comfortably determine when something has really gone wrong.

One approach would be to set house rules via a conventional house meeting process. Another would be to adopt a separate tool, such as the excellent [AllOurIdeas](#), for soft norm-setting. A third approach would be to develop a separate module, let’s call it `Rules`, which adopts explicit norms using the lazy consensus challenge-response pattern and explicitly gates `Hearts` by requiring every challenge to reference a pre-established norm or alternatively, provide a lower quorum requirement for pre-established norms. Thus, the first (or second, etc) time an incident occurs, it is addressed in a depersonalized way, and only in the face of consistent problems do personal challenges occur.

All of these mechanisms could help to differentiate the rehabilitative and punitive aspects of `Hearts`, improving overall system semantics and thus ease-of-use.

### A Note on Harassment

Not all types of bad behavior can be easily challenged in public. Sexual harassment comes to mind. Serious violations of societal and legal norms are poorly handled by `Hearts`. In these cases, prudence trumps principle. Referring the case out to a third-party mediator is appropriate. All residents should be able to contact this mediator privately, who can then investigate the case without the knowledge of the other residents and if appropriate, present the case to the property manager for appropriate action. In extreme cases, law enforcement may need to get involved.

## Props ✨

While the `Chores` module supports structured coordinated activity, the `Props` module supports unstructured, spontaneous activity through ad-hoc peer recognition. At any time, any resident may “plus-plus” another (e.g. ``krono++ for helping me build my bookcase``).

Giving props in this way has two effects. First, the recipient immediately earns back one-quarter of a heart, at most once per gifting resident per week. Second, the acknowledgment is recorded in a graph and, employing a similar technique as is used for chore weights in *Chores*, is used to determine the “most appreciated” resident over any given period. Each month, all the props given in the past month will be aggregated and this resident given special recognition (e.g. akin to an “employee of the month” award), with symbolic weight and potentially other benefits.

The hope is to [overcome the limitations](#) of the explicit accounting of contributions. That is, avoid discouraging contributions that fall outside of the explicit scope, by allowing for arbitrary and unstructured acknowledgment of contributions.

## Rules

The *Rules* module allows residents to develop a list of plaintext house rules, which are non-binding but can act as an aid to coordination.

Residents at any time may propose the addition, deletion, or modification of a house rule, expressed as a title followed by a paragraph of plaintext. Votes run for 72 hours. For a proposal to pass, it must receive *either* a supermajority approval or a simple majority approval over two weeks, with the second vote being initiated automatically. The intention here is to encourage consensus-formation while still allowing controversial rules to be enacted at a higher cost.

As mentioned in the description of *Hearts*, an optional feature is to allow any challenges that explicitly reference pre-established rules to have a lower quorum requirement, representing a smaller need to establish a group consensus.

## Things

*Should feel **better** than:* sending frequent Venmo requests, managing expense spreadsheets.  
*Should feel **worse** than:* everything being magically in-stock.

The *Things* module allows residents to make purchases out of a shared account via a challenge-response flow, with inspiration coming from the shopping interface of *The Sims*.

The set of items available for purchase is currently curated in the constitutional layer (i.e. by the property manager), who will determine the set based on need and availability (e.g. choosing a preferred brand of soap, etc). Once set, any resident can purchase any item at any time. If other residents feel that the purchase was frivolous, they can challenge the purchase, causing the resident to potentially lose a fraction of a heart (i.e. one quarter for small purchases, one half for large ones) as a mild warning against misusing shared funds. Depending on the value of the purchase, some minimum threshold of affirmative votes may be necessary – one vote per \$50 seems like a reasonable starting place.

## Art

The `Art` module is a simple variation on the basic challenge/response flow used to manage decorations in common space. The essential idea is that any resident can *anonymously* request that a decoration or other object in common space be removed. A vote runs for 72 hours, after which time the decision is made. The choice to allow anonymous challenges is to lower the emotional bar for participation. As the consequences are minor, the cost of challenging should be lower.

Unlike the other modules, `Art` is not currently integrated into `Hearts`. As such, there are no consequences for creating or losing challenges.

## Hangs

The `Hangs` module is a tool to manage events in common spaces. Any resident can schedule an event in any common space using the house calendar so long as that time or location is not currently booked. Once the event has occurred, a message is posted to the channel asking residents to evaluate the event. If the event receives a negative evaluation, the resident who hosted the event loses some amount of hearts, proportional to the length of the event. As a starting point we propose a half-heart per hour.

The goal of this mechanism is to give residents the freedom to schedule and host events that includes a feedback mechanism to ensure that the needs of the residents are respected.

## Subs

The `Subs` module is a relatively simple tool for supporting subletters. Here, any resident can request to sublet their room to a non-resident for some number of weeks, with a minimum duration determined by the house. The request must be approved by a simple majority, at which point the subletter temporarily replaces the traveling resident in `Mirror` for the duration of the sublet. Any lost hearts, chore penalties, etc accumulated by the subletter will be transferred to the resident upon their return, in that way they are “on the hook” for the subletter’s behavior.<sup>5</sup>

## Friends

The `Friends` module is a more specialized tool for selecting new roommates. It has the twin and potentially contradictory goals of giving residents agency in choosing whom they live with (even approaching lazy consensus) and simultaneously encouraging a process that is as inclusive and efficient as possible.

---

<sup>5</sup> This may be a too strong penalty.

Inspiration for `Friends` comes from the famous “[secretary problem](#)” of decision theory. Essentially, new applicants are placed in a queue, which is hidden from the residents. Once per day, the applicant at the head of the queue is posted to the appropriate channel. Residents will see some appropriate information – perhaps a short biography, a radio plot of interests, and so on. Residents can then vote on the applicant, with a strong positive vote requirement (e.g. on the order of 70%). If the applicant is rejected, then the next day another applicant is presented.

The process continues until either an applicant is accepted or some fixed number of days (e.g. 14) have elapsed, at which point the property manager chooses the applicant. This cutoff is to prevent residents from keeping the room vacant indefinitely. Knowing that they only have so much time to pick someone and that they can't know who is coming next will hopefully encourage residents to satisfice, taking the first acceptable candidate that comes along – although the low rejection threshold ensures that even minority opinions will be respected.

### III. Conclusion

Ultimately, successful shared living depends on three factors: the right **structure**, the right **environment**, and the right **culture**.

Right structure pertains to the formal systems and processes used to organize the activities of the community. These systems must manage resources, coordinate activities, and resolve conflicts. Mirror attempts to provide this first factor.

Right environment pertains to physical space. The environment should be sturdy and straightforward, with ambiguity removed whenever possible (e.g. an ice machine eliminates the need to refill ice trays, and open racks make kitchen equipment hard to misplace).

Right culture pertains to the particular blend of people and their relationships and intentions. The residents must have the skills and motivations to engage with their environment and each other, and a good organization should invest in cultivating those skills (e.g. by providing regular educational opportunities for personal development).

If these three factors can be consistently brought together, exciting things are possible.

### Acknowledgments

We would like to thank Jacob Abolafia and Cory Doctorow for their helpful feedback on earlier versions of this design, as well as the Berkeley Student Cooperative for continuing to demonstrate that accessible resident-driven governance is not only possible, but desirable.

## IV. Appendices

### Gaining and Losing Hearts

The most significant difficulty facing the `Hearts` mechanism is the establishment of the proper emotional relationship to losing hearts. Ideally, small heart losses should be a matter of course, so that losing a single heart does not feel like a significant event. By including multiple mechanisms for both gaining and losing hearts, small losses and gains should be regular occurrences. The following is an enumeration of every mechanism for gaining and losing hearts.

#### Gaining Hearts

- Monthly regeneration (1 heart)
- Receiving props ( $\frac{1}{4}$  heart, max once per week per sending resident)

#### Losing Hearts

- Losing a `Hearts` challenge (1 heart)
- Failing to meet the `Chores` requirement (variable)
- Making a frivolous purchase with `Things` (variable)
- Hosting an unsuccessful event with `Hangs` (variable)

Ideally, residents should be comfortable losing around one heart per month, equivalent to the baseline regeneration rate. Losing hearts is not necessarily a bad thing -- it reflects a certain boldness and vigor in exercising agency in the environment. Establishing this cultural norm will require some work outside of `Mirror` and should be a priority for founding residents.


### Earning Achievements

In addition to monetary rewards and penalties, symbolic awards (“achievements”) play a role in recognizing and rewarding behaviors. Broadly, achievements are not earned permanently, but are based on recent behavior. The following is an enumeration of the various achievements.


#### Chores

- A medal is earned for completing some minimum number of points in a given chore
- Medals are displayed next to the resident’s name whenever completing that chore
- Only contributions in the last twelve months are considered
- 🏅 for completing 50 points
- 🏆 for completing 250 points
- 🏆 for completing 1000 points

#### Hearts

- A shield is earned for successfully prosecuting three challenges<sup>6</sup>
- Shields are displayed next to a resident's name whenever making a challenge
- Shields last for twelve months, and can accumulate
- 

#### Props

- A trophy is earned for being the "most-propped" resident in a given month
- Trophies are displayed next to a resident's name whenever receiving a prop
- Trophies last for six months, and can accumulate
- 

---

<sup>6</sup> As is noted in Ostrom (p. 95), quoting Elster: "punishment almost invariably is costly to the punisher, while the benefits from punishment are diffusely distributed over the members." It stands to reason that privatizing at least some of the benefits would offset this personal cost, which we do via this symbolic reward.