

Magic: Made Mundane*

Marc LeBrun, Fixpoint Inc. 12/13/2010

This describes a basic card game whose design is inspired by, and indebted to, the brilliant Magic: the Gathering (also known as M:tG) created by *Richard Garfield* and ably developed and marketed commercially by *Wizards of the Coast*, and scouts the huge design and implementation space spanned by these two extremes. This work is offered as “open source”, to encourage collaborative exploration of that space.

For many years I naively dismissed M:tG as some inscrutable but harmless pastime that fantasy-prone adolescents could spend allowance money on. However after an introduction by a professional computer scientist and subsequent play and study I've come to better appreciate the considerable intellectual scope M:tG successfully achieves in its design and implementation. I now view M:tG as an innovative and exotic programming system that I believe deserves wider understanding. This write up is intentionally very concrete, in order to motivate and fuel hands-on exploration.

The plan here is to define a drastically simplified game Magic: Made Mundane (also known as M:MM) that eliminates as much of the attractive complexity, color and flavor of M:tG as possible, yet (hopefully) retains the bare character of the essential basic game play. M:MM can be played off-line with ordinary standard playing cards.

A key motive behind this reduction is to provide a specification whose base version might be readily realized in a computer implementation, then later be extended via programming modules to support a spectrum of variations for educational and recreational purposes.

We begin with the core M:MM specification, followed by a discussion of variations, extensions, comparisons with M:tG and other topics.

Game Components, Layout & Setup

Players & Turns. There are two *players*. The first player is determined by a coin flip or similar agreeable mechanism. Initially the first player is referred to as the *attacker*, while their opponent is the *defender*. Subsequently they take *turns*, swapping roles.

On each turn a *battle* occurs between the attacker and the defender. After the battle concludes either the defender has lost and the game is over, or the roles of the players reverses, with the attacker becoming the defender, and vice versa, and the next battle ensues. The players continue taking turns battling until the game is over.

Copyright © 2010 by Marc LeBrun, Some Rights Reserved. Free Use with Attribution.

Author Contact: Fixpoint Inc., 448 Ignacio Blvd. #239, Novato, CA 94949

mlb@fxpt.com



Magic: Made Mundane by Marc LeBrun is licensed under the Creative Commons Attribution 3.0 License.

Lives. Each player is allocated 17 *lives*. A battle may cause the defender to lose one or more lives. When the defender's lives are reduced to zero or less they lose, the attacker wins, and the game is over.

Card Deck. The game is played using ordinary playing card *decks* of 52 cards, labeled in the usual order, Ace through King. Jokers are not used. Each player has one full deck. The suits of cards (club, heart, diamond or spade) are ignored. Cards are divided into two types based on their value:

A thru 7 are called *critters*, while

8 thru K are called *credits*.

The value of critter cards, 1 thru 7, is relevant to game play. The value or rank of credit cards is always ignored.

Cards from different player's decks are never mixed together during game play—each player arranges their cards entirely within their individual playing spaces.

Areas. Each player's space is partitioned into six *areas* containing zero or more cards: their *hand*, three piles—their *deck*, their *dump*, and their *bank*—and two playing fields—their *front* and their *rear*.

Hand. A player's *hand* is a small collection of cards (usually a half-dozen or fewer) whose values are visible only to that player.

The *deck*, *dump* and *bank* are facedown piles whose contents are thus hidden during game play. However the number of cards in each pile is always available to all parties, and so may be tracked, or carefully counted, if need be.

Deck Pile. A player's *deck* initially contains all 52 cards, shuffled. The player draws cards sequentially from their deck into their hand. Cards are never added to the deck. When a card is drawn its value then becomes known to that player, but not to their opponent.

Dump Pile. A player's *dump* is a pile for cards that they discard from play. It is initially empty. Cards are never removed from the dump.

Bank Pile. A player's *bank* is a pile of credit cards they have placed into play. It is initially empty. Credit cards are never removed from the bank.

Critter cards are played individually faceup in the *front* and *rear* fields rather than piled, so as to be visible to, and manipulated by, the players. During play the front and rear fields may contain any number of critters, or may be empty.

Rear Field. Critters in the *rear* fields comprise a set of supporting resources that may be used by the player, but are not currently involved in battling.

Front Field. The attacker's and defender's *front* fields abut. During battle, any opposing critters are aligned so that they face each other across the common border.

At the very beginning of the game each player draws a hand of 6 cards, thus leaving 46 in their deck. Their dump, bank, rear and front areas are initially empty.

Turn Cycle

Each *turn* proceeds through four phases: *build-up*, *attack*, *defense* and *mop-up*.

Build-Up Phase. The *build-up* phase at the start of the turn is itself further sub-divided into four steps, *draw*, *deposit*, *deploy* and *discard*, as follows:

Draw Step. During the *draw* step the attacker *must* draw one card from their deck into their hand. If their deck is empty they lose the game (this is to prevent tedious or otherwise unwinnable end-games).

Deposit Step. At the *deposit* step the attacker *may* play one credit from their hand into their bank, or they may pass on doing this.

Deploy Step. During the *deploy* step the attacker *may* play one or more critters from their hand into their rear area (or pass on this). The sum of the values of the newly played critter cards must not exceed the number of credit cards in the attacker's bank pile (metaphorically, the attacker is paying for new critters, as limited by their available credit)

Discard Step. If the attacker has more than 6 cards in their hand, they *discard* enough cards of their choosing into their dump so that 6 remain.

Attack Phase. Now the attacker performs the *attack* phase by choosing and moving any number of critters (including none) from their rear area to their front.

Defense Phase. When the attacker is finished, the defender performs the *defense* phase, by choosing to move some critters from their rear into 1-on-1 opposition, joining battle against specific attackers. Not all critters available for defense need be so used, and some attackers may face no defender. (Thus the number of defenders never exceeds the number of attackers).

Mop-Up Phase. In the *mop-up* phase the consequences of each battle are determined individually, in any convenient order, as follows:

Unopposed Critters. If the attacking critter is *unopposed*, its value is deducted from the lives of the defender. If the difference is zero or less the defender loses and the game is over. If a positive residue remains the attacker is returned to the rear and the mop-up phase continues with another attacker.

Opposed Critters. If the attacking critter is *opposed* by a defender, then their values are compared. If either critter's value exceeds that of its opponent it is returned to the rear, for possible play again on future turns. Otherwise (if a critter's value is less than, or equal to, its opponent's) it is removed from play and placed in the dump (thus at least one, and maybe both, combatants get dumped).

Eventually, when all critters have been removed from the front, mop-up is done, the turn is over, the players swap roles, and the game continues until a player wins.

Example

The attached diagram shows a snapshot of an imaginary turn from the attacker's point of view, taken between the conclusion of the defense phase and the start of the mop-up phase. (This is intended for illustration only, and doesn't necessarily reflect realistic game play.)

The deck, bank and dump piles for each player are indicated in a typical layout where "card flow" proceeds roughly from the player's left to their right, although any convenient configuration of the areas may be arranged to taste.

The attacker's hand contains four critters—2♣, 3♦, 3♠ and 6♥—along with one credit card—the J♦. (The defender's hand is, of course, hidden from the attacker's viewpoint.)

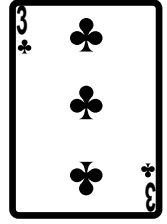
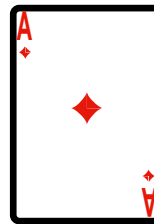
The attacker's rear area contains a single 2♥ that has been deployed but is not engaged in this turn's battle, while the defender has deployed two critters in the rear—A♦ and 3♣.

There are four attacking critters on the front line—A♠, 4♣, 5♦ and 6♣. The A♠ is unopposed, while the remaining three are opposed by the defender's 5♠, 5♣ and 5♥, respectively.

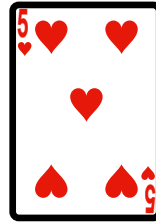
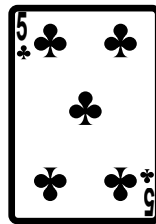
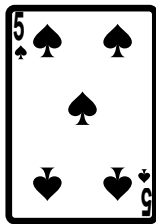
During the subsequent mop-up phase, the unopposed A♠ will inflict the loss of one life point on the defender and then be returned to the rear. In the 4♣ vs 5♠ match the 4♣ will be sent to the attacker's dump, while in the 5♦ vs 5♣ conflict *both* will go to their respective player's dumps, whereas in the 6♣ vs 5♥ meet the 5♥ will go to the defender's dump. The surviving 5♠ and 6♣ will be returned to the respective player's rear areas.

Magic: Made Mundane

Copyright © 2010 by Marc LeBrun, Some Rights Reserved



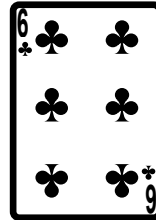
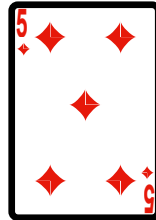
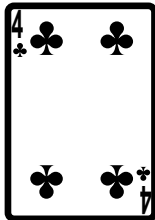
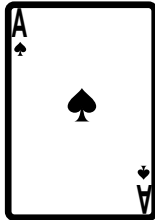
Defender



Defending

Attacker

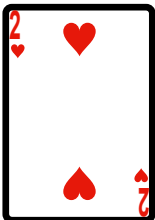
Unopposed



Attacking

Front

Rear



Hand



Deck



Bank



Dump

Tunable Design Parameters

Besides the core mechanics, a few basic parameters of M:MM are derived roughly based on those of M:tG. Since a “canonical” M:tG deck contains 60 cards, while there are only 52 cards in an ordinary standard deck, the scale factor would in principle be $52 / 60 \approx 0.87$. However various considerations suggested the values listed here (these should be checked and possibly tuned by further play testing):

Parameter	M:tG	M:MM
Cards in deck	60	52
Critter cards	36	28
Credit cards	24	24
Cards in Hand	7	6
Player’s Lives	20	17

The main difference is the relatively higher proportion of credit-to-critter cards in the M:MM deck (in M:tG these are called “lands” and “creatures”). This was chosen in an attempt to produce a faster-moving game. In M:MM the average critter “casting cost” is 3.5 credits, and since there are exactly four of each rank the “mana curve” (in M:tG terms) is a flat line at 4 cards for all ranks from 1 to 7. Unlike M:tG there are no “spells” other than critters in M:MM, and the “colorless” game design is otherwise devoid of features of interest. Therefore the choice of a credit-heavy design was made to help prevent what M:tG players refer to as “mana starvation” (in M:MM perhaps this should be called a “credit crunch”).

This balance could be adjusted in several ways if desired. For one, eights could be made critters. But while this reduces the available credit supply, it also raises the average casting cost. Another possibility might be to introduce the two Jokers usually packaged in regular decks, perhaps giving them a rank of 1 or even 0.

Extensions

A conceptual highly-multidimensional game-design space extends from M:MM as a minimal origin, out towards M:tG, which presents an ever-evolving canonical model of the polar extreme in complexity and richness under the continually innovative auspices of *Wizards of the Coast* and its player communities.

As envisioned above, M:MM could be implemented as a suite of software “base classes”, with the behavioral extensions like those sketched below derived from this foundation. Interestingly, many potential incremental extensions turn out to be orthogonal or only loosely coupled, so there are numerous possible variations that can be built on the basic core game—each offering different combinations of extensions, ranging from just a few to many.

The extensions sketched below are intended to motivate exploration of this space. Of course many are directly inspired by M:tG, and represent only a small sample of

its vast inventiveness. To gain a fuller appreciation of its scope, the web offers many excellent research materials on M:tG's design and implementation, and of course there is no substitute for actual play, which is highly recommended.

As compared with M:tG, the most striking simplifications of M:MM include the elimination of player *deck design* and *card abilities*. Many other interesting extensions can be classified as *additional card properties* or *gameplay mechanics*.

Deck Design

In M:MM the players start with identical (randomly shuffled) standardized playing card decks. In contrast, M:tG players construct their decks by drawing from the thousands of available card designs—indeed this variety supports a large market where physical and digital cards are actively bought, sold and traded. Perhaps as much enjoyment is delivered through designing and tweaking decks as by actual game play, which is viewed as a test arena or proving ground for designs as well as a match contest between players' capabilities.

One way to introduce more creative control into M:MM would be to allow players to "stack" their decks into any order of their choice, rather than shuffling them. This would appear to trivialize the game, but it is not entirely clear if there is a single universally optimal order!

For example it would seem to make sense to arrange a deck so that many critters can be deployed early, say by alternating credit and critter cards. However another viable arrangement might be to instead front-load the deck with credit cards, followed by a blitz of more powerful higher-rank critters.

With multiple options—since a player won't know how their opponent's deck will be stacked—there may be complex trade-offs, choosing between dueling strategies and counter-strategies in the so-called "meta game". In M:tG there are many forms of intransitive "rock-paper-scissors" tangles of this sort, and their appearance in this slightly-extended yet still wholly deterministic variant of M:MM is interesting.

An alternative to ordering their entire decks, that would retain the chance element, would be to allow the players to select only their opening hands, and then shuffle the remaining cards.

This small twist still captures some of the most fascinating aspects of M:tG. Pre-designed card sets in essence create an exotic kind of "programming model" wherein a non-deterministic runtime environment makes "instructions" available for execution in random order. While a player exercises immediate tactical control during an individual game, the deck designer's strategic ideas play out against other design choices across the multiple matches in the meta-game. This seems analogous to, for instance, "classical" versus "hypermodern" styles in chess, but it is embodied concretely in a particular deck, rather than a vague "book" of moves and heuristics.

The ultimate extension for M:MM along the lines of M:tG is to allow players to create decks containing non-standard collections of cards (to be obtained from other packs of standard cards in an off-line implementation). For example a “swarm” deck might be created by adding four aces, or perhaps two of the “heavy” sevens might be replaced with threes, and so on.

To maintain playability M:tG’s rules do impose some constraints—such as requiring decks to have at least 60 cards, with no more than 4 of any given card. There are also some sensible design rules-of-thumb—such as that decks should consist of about 40% “credit cards”, so a player doesn’t become frozen for long stretches due to resource “starvation”.

Card Abilities

Another striking characteristic of most M:tG cards is that they carry additional textual instructions which can affect the play, modify default behaviors and even supersede basic rules.

These are generically referred to as *abilities* of the card. The activation of these abilities may be triggered on various conditions—for example putting the card into play, dumping it, a player’s explicit choice, or simply on various aspects of the current state of the game.

An impressively vast and diverse collection of these abilities have been invented for use in M:tG. Only a tiny sample of possibilities these suggest for M:MM extensions can be described here.

In the following, the token <card> is used in these templates to refer to the card bearing the text (kind of like “self” in OOP). M:tG cards have colorful names; for standard playing cards names like “two-of-clubs” or “jack-of-diamonds” would be substituted.

- *If <card> is dumped, return it to your hand instead.*
(Good news: the card won’t go away! Bad news: the card won’t go away!)
- *<card> cannot attack alone. At least one other critter must also attack.*
- *While <card> is in play all aces have rank 2.*
- *Only critter cards with odd-numbered values may oppose <card>.*
- *When <card> is deployed, attacker and defender dump all cards in their banks.*
- *When <card> attacks it has rank 9. After mop-up your life total becomes zero.*
(An attack using this card had better defeat the opponent before the end of the turn!)

Since abilities could potentially do anything, designing collections of them that interact in interesting and viable ways involves a yet higher order of creativity and

analysis than deck design (which selects from these pre-designed cards). For M:tG this is the domain of *Wizards of the Coast*. The challenge is considerable; even with an experienced staff and extensive testing these professionals aren't perfect, and have even, on rare occasions, had to ban or "recall" cards whose abilities enabled ruinous exploits or had other unexpected consequences. It is hoped that M:MM might open up this fascinating "meta-meta-game" level to a wider community.

Implementation of abilities on the core M:MM base entails three interesting sub-problems: *associating* the abilities with cards, providing an *API* for the "play environment" and defining coherent semantics for *interacting abilities*.

Association. In an online embodiment linking abilities to cards simply involves adding an association between the card and the sub-module implementing the ability. For offline, to avoid defacing the physical cards, an auxiliary "ability table" mapping card names to the text descriptions might be used (note that composing these requires an almost legalistic or mathematical quality standard for precise expression, since they are interpreted by human players rather than a computer!).

The API for the play environment needs to allow the ability modules to access all aspects of the current state, including the status and sequence of every card in each area, and any associations such as battle oppositions, as well as other "ephemeral" bookkeeping information, such as the players' life totals, which player is attacker, which defender, what phase and step the play is currently in and whose turn it is to act. This information might be provided via explicit queries, standard events or the like. The API also needs to provide support for the ability module to affect the environment, such as to cause cards to be drawn or dumped, update status, change state or trigger events. If the core implementation provides a graphical UI this will also need to be open to extension for ability-specific animations or other effects.

The semantics of ability interactions is a very crucial cross-cutting aspect for any implementation. Even before it was implemented on-line, M:tG had the concept of a "stack" on which "spells" were pushed when cast, in order to provide an orderly model for "resolution" of effects. For example if a counter-spell to a spell is played the semantics must respect their ordering for proper resolution. Such dependencies can be complex. For example the effect of a card's ability may depend on another card. If that card is dumped due to independent factors, does the effect persist or disappear? Because abilities are general "fully-programmable" features, designing coherent semantics to resolve such questions for any given implementation is a rich area for further exploration.

Additional Card Properties

Cards in M:tG are often endowed with further additional properties. These might be "mixed in" to M:MM, in various forms, alone or in combinations as extensions.

Color is an interesting orthogonal property that increases potential complexity and also is used as a subtle "aesthetic organizing influence" for game design. In M:MM

the obvious colors are red and black; however the four suits—♣, ♦, ♥ and ♠—provide a finer differentiation that might be made to serve in a way more analogous to M:tG's five colors—○, ●, ●, ● and ●—(in fact six, as there are ● “colorless” or “any color” options too).

At the most basic level, in M:tG the colors behave like different “currencies”. The M:MM analog would be that ♦ credit cards could only be used to deploy ♦ critter cards, and so on. Implementing this extension is fairly trivial but as such merely complicates the game without really making it more interesting.

However the M:tG designers also exploit their colors in a more intuitive way (as a “meta-meta-meta-game” level feature?) by associating each color with a distinct sort of “personality”, which is reflected in the “flavor” of spells or abilities of that color. Details of the elaborate M:tG color system may be pursued elsewhere, but some simple examples might be that a protective, supportive or healing effect is most typically ○ or perhaps ●, an effect that “hurts me as much as it hurts you” is likely ● or maybe ●, while a “tricky” ability such as a universal counter-spell is ●-ish, “generic” effects are ● and so on.

Color “personality” introduces an interesting five-way “rock-paper-scissors” twist on the “meta-game”, since different styles of gameplay will result depending on whether “natural” allies or antagonists are in conflict. A ●-vs-○ conflict has a very different feel from a ●-vs-○ match. What “suit personalities” might be created for M:MM that might similarly differentiate ♠-vs-♥ from ♦-vs-♥ games?

A further possible complexity is cards or abilities that involve more than one color. However because of the likely limitations of a probabilistically more-restricted credit support base, multicolor M:tG decks most frequently just utilize one dominant color, with a single additional accent color. For example $\frac{3}{4}$ of the cards may be the main color, and $\frac{1}{4}$ the other. Typically M:tG cards require only one or two explicitly colored credits to deploy, with the remaining credit allowed to have any color.

Power, Strength and Cost are unified as a single value, the rank, in basic M:MM. However these often have different individual values in M:tG, expressed in different contexts. *Power* is the effective rank of the card when attacking, *strength* is the rank for defense, while *cost* is the required credit for deployment. With standard offline cards it is simplest to make these all equal to the printed rank. However these properties could also be extended by using a table lookup, as for abilities. When these parameters are independent, as in M:tG, interesting new possibilities arise, such as critters that rank zero for attack but non-zero on defense (called “walls”) or, conversely, “aggressive” but “weak” critters with high attack rank whose defensive value is low, and so on.

Instants, Sorceries, Enchantments, Equipment, etc. These are non-critter, non-credit card types in M:tG. They all have a deployment cost, but different default

behaviors thereafter. For example an Instant's effect is scoped to a precise point in the stack resolution sequence, Sorceries apply only to that turn, Enchantments are persistent (like critters, but don't attack or defend) while Equipment is a kind of enchantment whose effect scoped to another particular card. Any or all of these might be incorporated to extend the core M:MM implementation.

Art & Backstory are additional dimensions of contextual and esthetic value added by M:tG. However since they don't directly affect gameplay they won't be addressed here, although the interested reader or implementer is highly encouraged to pursue these topics as additional ways to create extensions of M:MM that are more broadly appealing and attractive.

Gameplay Mechanics

There are countless elaborations of M:tG gameplay mechanics that might be "ported" to M:MM. Many of these are fairly specialized (eg "flying" critters, that may only be opposed by other flyers) and their implementation would be fairly modular. A few whose impacts that might be more pervasive are listed here.

Tapping augments cards in play with an additional bit of state that is visually indicated by rotating the card 90°. Numerous card behaviors are affected by whether they are tapped or not, only presenting their key effects when in the untapped state, when tapped by a player, and so on. For example credit cards are tapped when "spent" to deploy other cards, and tapping often is required to trigger abilities. Tapping and untapping help control the scope and timing of effects and other events, either when deliberately initiated by the player, or automatically by abilities or other rules. The tapping mechanic is in fact so pervasive in M:tG that there is a step explicitly allocated that untaps the player's tapped cards in the initial phase of their turn.

Counters are an additional class of game pieces that sometimes modify the action of a card. For example counters might be placed on a critter to increase its value. There are different flavors of counters, with different effects and behaviors. They are similar to equipment in that they are typically are scoped to a single card, but they aren't themselves cards, and usually disappear when their target is dumped. Counters arise most naturally as bookkeeping devices in off-line play. As on-line extensions counters require a fair amount of additional support. They complicate the user interface visually and mechanically, so other approaches (eg simply increasing the displayed value of a card) are sometimes easier alternatives.

Multi-way battles involve critters opposing each other in groups. This seems a simple natural concept, but specifying the details of the play mechanics and resolution algorithms to determine the outcome for multi-way matches can entail surprisingly non-trivial design and implementation effort. In fact in the early versions of M:MM this feature was naively imitated ("how hard could it be?") but was removed after the required descriptions turned out to be almost as lengthy as all the rest of the rules!

Many game formats beyond the basic scheme are also possible. Extending from two players to multiplayer naturally provides new design options (eg teams vs. free-for-all). An orthogonal option is to constrain the cards and decks. M:tG releases cards in coordinated sets regularly, and tournaments are often limited to these sets. There might also be options for very small “sudden death” decks, large decks (eg 104 cards) that can have statistically distinct behavior in play and so on.

Sideboards are very small sets of additional cards that a player may substitute for other cards in their decks between rounds in a multi-game set. This allows the players to tweak their “meta-game strategy” somewhat based on what strategies their opponents present. For example a defensive deck might be made more aggressive, or specific abilities might be swapped. Of course this also significantly extends the design and implementation complexity by introducing new objects that persist across games—a tournament is no longer just a concatenation of independent games!

Taking mulligans is a representative of the many creative “miscellaneous” ideas that can be “mixed into” gameplay almost independently. This clever feature allows a player to reject their initial hand and draw a new one, but with one fewer card. This may be repeated to the point where the player starts with an empty initial hand! Supporting features like mulligans requires an implementation framework flexible enough to accommodate unusual new plug-in behaviors, while still reliably presenting enough useful common structure to enable robustly defining those modules in the first place.

Acknowledgements

Thanks to Fred Garrett for introducing me to Magic: the Gathering, as well as review of these ideas and write-up, and to Don Knuth and David LeBrun for their input.