Fools and Corpses

Foreword

Hello, and welcome to Fools and Corpses! Started in 2015, F&C is our ongoing passion project, driven by our love for collaborative storytelling and tabletop roleplaying. What started as a silly little homebrew has grown through three major incarnations into a fully-fledged roleplay system designed to support, build, and quantify narratives flexibly and intuitively. Today, F&C is our go-to tabletop system, allowing us to run sessions in completely custom settings with little to no preparation. We are so excited to show you what we've created. It is our genuine hope that it will encourage, inspire, and above all equip you to build something good that you can call your own.

The mission of Fools and Corpses is simple : let any party of players tell any story at any time. Stories come in all shapes and sizes, as do players, settings, and situations. To assume that we could predict all the ways someone might want to express themselves through a story is simply absurd. This means that above all, Fools and Corpses must be flexible, intuitive, and easily extensible. F&C isn't here to put boundaries on the story you're telling, it's here to enable and enhance your stories by allowing them to be profoundly collaborative. The starting point for such a system is language. Language is open-ended, adaptable, and intuitive. It's one of the first mental tasks we learn to do, and it's the foundation for our reasoning and communication for the rest of our lives.

All that is missing from language is a consistent quantification of what is said. So that's Fools and Corpses' primary task : quantify what you're already saying, so it can have systemic meaning that matches its semantic meaning. F&C achieves this by taking what you've said about your story and adding numeric values to its exceptional factors. Then you do the very simple work of deciding which factors apply in the scenario you're describing.

By letting the story do the "leading", you let the narrative be the system. What can you do in F&C? Whatever the story says you can. This means the moment you can think of a setting, character, or story, it is ready to play. F&C doesn't tell you what is possible and what isn't. That's your job. The story's rules are the rules. F&C just tells you how to quantify them.

Naturally, this leaves a ton of gray areas in play. That is by design. When we were writing Fools and Corpses, we viewed this uncertainty as a feature, not a failure or mistake. It allowed us to make Fools and Corpses profoundly expressive in a way that videogames and heavily systematized tabletop games just aren't. Removing this gray area would mean destroying the flexibility, expression, and ease-of-use that F&C is built upon.

This requires that the people at the table work together to tell the story, and more importantly, that they trust each other. On its most fundamental level, Fools and Corpses is a trust-based game. The players must trust the Narrator to represent the world well and the Narrator must trust the players to act in good faith when relating to the world they've made. Ultimately, everyone at the table is on the same side- not the side of the Narrator or the side of the players, but the side of the narrative. We're all here to make something good, and trust is central and necessary for that work.

If this type of gray area is a turn-off for you, or if this kind of trust makes you uncomfortable, that's okay. There are plenty of more structured games out there that might be a better fit for you. But, if we could be so bold, we'd like to challenge you a little bit. The trust this system requires is small compared to the payoff you get in return. Trusting the people you're building with makes so much more possible than building solely along codified, predictable lines. Give it a chance!

Playing F&C is a creative endeavor, something you do with your friends and an adventurous spirit to see what comes out the other end. More often than not you'll find the world you've created will shift and evolve as you and your friends play in it. This is expected and intended. When playing FnC, you're creating something collaboratively. That means the game will change over time to fit what all the players jointly want it to be. Here at F&C, we think that's awesome. We want the stories we tell, the worlds we create, and the product we make to be deeply and profoundly affected by those who participate in it.

So let's take a look at these rules and see what we can make with them- something new, something meaningful, and above all something good.

See you out there,

The Fools and Corpses Team

Table of Contents

Introduction

Fools and Corpses is a tabletop role-playing system designed to be simple, approachable, intuitive, and infinitely extensible. It achieves this by quantifying the story you are already telling, allowing any story in any setting to be told at a moment's notice. The words you're already using to describe the events of your story become the numbers you add to your dice rolls to determine how the story unfolds. The rules below exist as a guide to help you translate those words into numbers and create a fair, balanced game without sacrificing the narrative power of the story.

Players and Narrators

In any given game of Fools and Corpses, there are two kinds of participants : players and Narrators. Each game has exactly one Narrator and many players. There is no hard limit to the number of players that can be included in a game, but larger groups tend to be harder for the Narrator to manage. We recommend a group of players (often referred to as a "party") of about four, but have seen success with groups as small as one or as large as seven.

Before play begins, the Narrator and the players should spend some time discussing what they want out of this experience. This discussion should include things like the setting they want to tell a story in, how long that story will likely be, and what activities will be common in the story. The party or Narrator may suggest using a setting that has been written by someone else, such as a piece of beloved fiction, or even a purpose-made setting designed for F&C or another TTRPG. Since Fools and Corpses is designed for building custom worlds, it's not uncommon for parties and Narrators to invent settings themselves, sometimes before play begins or even while play is ongoing. If you wish to create your own setting, we recommend having at least some idea of the content and boundaries of that setting before play begins.

Each player in Fools and Corpses has a character they control during play. That character's choices, appearance, and actions are up to the player, but should attempt to remain consistent with the tone of the world the story is taking place in. Once each player has prepared their character, the Narrator describes the world and circumstances in which the characters find themselves. The players describe how their characters interact with these situations, and the Narrator responds with how the world, non-player characters, and circumstances react to the player character's actions, and play continues.

The Narrator should attempt to maintain a fun, consistent, and balanced expression of the world during play, allowing for creativity on the part of the players, even if it's not what the Narrator expected. This includes allowing the players to make mistakes and letting them get themselves into jams. Getting yourself out of a situation of your own making is often much more compelling than a problem contrived by the Narrator, and it imbues the player's decisions with a sense of meaning and weight. It's this feeling of relevance, sometimes called *agency*, which makes TTPRGs so compelling. What the players do really matters, and it matters in a way that encourages further decision making, action, and creativity.

Types and Attributes

Types

Every entity in Fools and Corpses is defined by one or more words or phrases which describe it. The most important of these is the **type**. It is the baseline that all other aspects are measured against. Every item, character, skill, or ability has a type. Usually a thing's type is just its name, or maybe more accurately its *noun* or *label*. Knife, uppercut, dog, hovership, deception, archery- these are all *types*. Types are rarely strictly defined, and live almost entirely in the realm of assumption- they are just what you assume a thing to be without any sort of modification. They represent the baseline from which any *specific* instance of that type is measured. Types don't usually contribute numerically to gameplay, but rather define what's possible for that thing and provide the GM with context for their decisions. To define an instance of a type further and start numerically influencing gameplay, you must add an *attribute*.

Attributes

Attributes are *encapsulations of any single significant aspect of a particular object or activity that differentiates that object or activity from others of the same type*. That's a wordy way of saying that an attribute defines the difference between a *generic* object, a knife for example, and a *specific* object, like a dull, short, rusty knife. A knife can be assumed to be a few things: short, able to be used one-handed, sharp enough to cut, with a metal blade and a handle. All deviations from these assumptions are accounted for by attributes. Attributes contain two important pieces of information:

1. **Description** : A single word or phrase that describes what the attribute contains. This can be as simple as "strong", in the case of a trait, or "sharp", perhaps applied to an item. It can also be as complex as "Fought and beat the head of the Broken Bottles Gang" when applied to achievements.
2. **Rank** : A numeric value representing how much mod the attribute is likely to generate when applied to a check. This value is not always fully applied during mod calculation. Depending on the context, a situation might call for partial

application or more-than-full application of this rank. That said, an attribute’s rank should be considered its “most likely” value.

When writing an attribute, the format Description (Rank) is used, so the description and rank are easily discernible.

In some cases, (usually in the case of items and skills) an instance can have ranks without an explicit description, essentially behaving as its own attribute. This just expresses how *effective* that instance is at exemplifying its Type's innate attributes. A Shovel (4) should be considered a much better shovel than a Shovel (2), though *why* it is better is left somewhat open-ended. Instances can also have both "described" attributes and direct ranks:

```
Shield (4)
 + Heavy (2)
```

This Heavy Shield will function one way when its weight plays a role in whatever it's being used for, but another when only its explicitly shield-like attributes are at play.

Modifiers

During play, you will often need to quantify how effective a character is at a given task. This value is called a **modifier**, and it is calculated by collecting all the character's attributes (including those from items) that apply to the task and adding their ranks together into a final value. Attributes can be applied positively or negatively, depending on if they would aid or hinder the character. If an attribute applies "somewhat" or "partially", a number of ranks reflecting that marginal influence can be added to the final value (usually around half of the total ranks in that attribute). Any number of attributes from any category can be applied, so long as they could conceivably have an effect on the task at hand.

Passive modifiers represent the character's effectiveness at completing a task when they are not consciously attempting to do so. This includes things like noticing something missing in a room, detecting deception without explicitly attempting to, or mitigating damage from an attack. This modifier should only include attributes that can be effective without conscious choice, and therefore often won’t include learned skills. Attributes like Perceptive, Insightful, or Tough Skin would be effective as passive modifiers. Passive modifiers are usually not paired with a dice roll, as their application does not represent a choice or active effort on the character's part.

Active modifiers, on the other hand, represent the character's effectiveness at completing a task they are consciously trying to perform. Tasks like scaling a wall, hacking a computer, or attacking a bear are active tasks. Active modifiers are typically much higher than passive ones, as they include all attributes that can be invoked consciously, on top of those that are invoked passively. Active modifiers usually include a dice roll, as they represent active effort on the part of the character.

Making a Check

When a situation arises where the result is uncertain, the Narrator may require a player to make a **check** to determine the outcome. The first step when making a check is to collect the active modifier for the task. Depending on the context of the check, the Narrator will indicate if it is active or passive, and the player collects the modifier based on the description the Narrator provides. In any case where the application of an attribute is unclear or unsure, the player should ask the Narrator to rule on how the attribute should be applied. The Narrator is responsible for maintaining a fair, clear and consistent pattern of application.

Let's look at an example character. For simplicity's sake we're only going to look at traits for now, but the same application rules described below would apply for skills, actions, status effects and items as well:

Daisy Dancepants
- Clumsy (2)
- Smart (4)
- Quick (5)
- Dancer (4)

If this character were to attempt to dodge a falling rock, Smart (4) would be ignored (being smart doesn't help you dodge), Quick (5) would be applied positively (the faster you are, the easier it is to get out of the way in time), and Clumsy (2) would be applied negatively (being clumsy means you are likely to stumble as you try to move quickly). Dancer (4) is likely to be applied partially, probably at a value of 2 (being a trained dancer makes you better at controlling your body, but does not necessarily pertain to dodging rocks specifically). This would result in a total mod of 5.

While the player is calculating their mod, the Narrator is deciding on the difficulty class, or DC for the check. The DC represents how difficult the Narrator believes this specific action to be, including any extenuating circumstances but *excluding* any special skills or abilities the character might have (since this is represented by the mod). The DC is rated based on simple difficulty descriptions:

Difficulty	Description	DC
Easy	Things children can do	-5
Moderate	Things adults do frequently	0
Difficult	Things untrained adults find difficult	5
Hard	Things adults typically need training to do	10
Impressive	Only trained or gifted people can do these things	15
Remarkable	Even gifted people require some training to complete this	20
Amazing	Serious training is required to have a chance at doing this	30
Astounding	People at the top of their field can do this	40
Heroic	Only superhuman heroes and prodigies are equal to this task	50

For Daisy Dancepants dodging the falling rock, the check is likely within the "difficult" range- an untrained adult would need to get lucky to do it, but it certainly doesn’t *require* training, therefore the DC is 5. Now all that remains is the addition of a little chaos.

When making a check, the player rolls one positive 12 sided die and one negative 12 sided die and subtracts the negative die from the positive die. The result of this roll can be anywhere from -11 to 11, but will favor the values closer to 0. This means that the character’s mod and the DC together represent what is most likely to occur: if the mod is the same as or higher than the DC, it is most likely that the character would succeed. If the mod is lower than the DC, it is most likely the character would fail. For Daisy, this means she will *most likely* succeed, but just barely, since her mod is equal to the DC for the check. The dice are rolled and the result comes up:

```
Roll Positive 1d12, result +7
Roll Negative 1d12, result -5
Subtract 5 (negative die) from 7 (positive die), result 2
Add 5 (modifier) to 2, result 7
Total is 7
```

With a total roll of 7, the character would succeed in their dodge- though not by much. Meeting or exceeding the DC results in a success, though exceeding it further could result in an even better outcome. The same is true for rolls that come in far below the check’s DC. A serious failure should have more serious consequences than a near failure.

Contextual Knowledge

Occasionally during play, a player might lack understanding necessary to determine how difficult a check might be, or not know something their character might know. In these cases it may be appropriate for the player to ask the GM for Contextual Knowledge. This could involve the GM describing the factors that contribute to the check’s difficulty, providing information the character would know, or simply stating the numerical DC for the check.

Contextual Knowledge should also be used to account for a character’s unique perception of a situation based on their personality. The player might ask “Does my overconfident, cocky character think they can do it?” and the Narrator might say yes, even if the task is impossible.

In some cases, the GM might ask the player to roll a check to see how much their character knows, via their experience and study. The player would add relevant attributes like Historian, Chemist, Wikipedia Editor, or Pendant of Forgotten Secrets, and the GM would determine what the character knows based on the result of the check, with a DC set based on the relative obscurity of the desired knowledge.

Using Items

Using items in a check is essentially the same as using attributes. The main difference is the item must be used for the check, and cannot be applied if the character making the check is unable to use the item at the time of the check (i.e. they don't have it on their person, their hands are full, etc). Items can have attributes of their own, and those attributes are applied following the same rules as attributes found on the character. If a character were to use some specially-designed boots to cross a marsh:

```
Boots of Striding (3)
  + Waterproof (2)
```

They would likely receive a +5 mod from this item on top of their other attributes. This is because the boots indicate that they are designed to travel great distances (Boots *of Striding*), *and* that they are Waterproof, which is helpful in the wet terrain of a marsh. If the character were to use the same boots to attempt some fancy footwork during a fight, however, they would likely receive a lower mod, perhaps 1 or 2, since the waterproofing isn't relevant and the boots are designed with travel rather than combat in mind.

Guessing

In the case that a character cannot be said to know *how* to complete a task they attempt, the Narrator has a choice. If it is reasonable that the character could *guess* how to complete the task, the DC is calculated in light of the guessing, not just the difficulty of the task itself. If the task *cannot* be completed without knowledge or pre-practiced skill, the Narrator may allow the character to attempt anyway, using their resulting roll (including mods) to determine how the attempt fares, even though success is impossible. In this case, it is usually useful for the Narrator to have a number in mind that represents the check going "as well as it can, considering the circumstances", which acts as a de facto DC. If this value is beaten, the result will still not be what the character intended, but be as close as one possibly could get without the required knowledge or skill.

Checks with Opposition

Sometimes two or more parties will want to perform actions whose outcomes are mutually exclusive (e.g. one person attacking and the other dodging the attack). In these situations, the check is said to be "opposed". Any number of parties can oppose any check, but the opposing parties must declare their intent to oppose before the inciting check or any opposing checks are resolved. In this case, the two parties roll checks against each other. In addition to the two characters' attributes, the Narrator might add Contextual Modifiers to account for situational advantages one party has over the other. As an example, let's take our same dancer from earlier:

Daisy Dancepants
- Clumsy (2)
- Smart (4)
- Quick (5)
- Dancer (4)

But now, instead of a falling boulder, Daisy's dodging a laser blast from a vicious laser robot:

Deathmotron 47
- Vicious (6)
- v3 Aiming Algorithm (3)
- Deathalizer™ Hypertronic Megagun (4)

First, we already know that the dancer's mod is going to be roughly 5. While a character's mod for dodging a boulder and dodging a laser aren't *necessarily* the same, in Daisy's case there is no significant difference, as she doesn't have any additional attributes that would apply to this combat that didn't already apply to the boulder.

So on to the robot. Nearly everything about the robot is geared towards killing, so their mod is going to be fairly high: Vicious, though not specifically geared towards accuracy actions, should add at least half its rank since it would prevent any sort of hesitation from clouding the robot's intent. The v3 Aiming Algorithm and Deathalizer™ Hypertronic Megagun both add their entire rank, since they apply directly to this activity. In total, the robot's mod is 10, five more than its target. Since both parties are acting, both parties roll dice and compare totals:

Daisy Dancepants
Roll 4 (positive)
Roll -2 (negative)
(4 - 2) + 5 = 7

Deathmotron 47
Roll 5 (positive)
Roll -7 (negative)
(5 - 7) + 10 = 8

The dancer gets hit, but it's a close thing. See Damage Calculation for guidance in determining how much pain Daisy would be in in this situation. Both attacking and dodging are actions which would cost one energy for this round of combat. More on that in the Combat section.

Contextual Modifiers

In many situations there can be contextual aspects of a check that lead to its difficulty changing significantly. Usually this can be easily accounted for by a modification of DC by the Narrator. That said, sometimes it is not so clear, especially when it comes to comparing two characters. Consider a mouse attempting to attack a dragon:

Sir Squeakers the Brave
- Brave (3)
- Strong (4)
- Mouse-sized Sword (2)
- Knight (8)
- Chosen One (4)

Thrum, the Coming End
- Powerful (6)
- Armored (3)
- Ender of Ages (4)

If the mouse were to attack the dragon with his mouse-sized sword, he would do so with all of his attributes: a mod of 21 against the dragon's 13. That doesn't seem right, unless the dragon is mouse-sized itself (which hasn't been designated, and shouldn't be assumed). This is because attributes are a measurement of what is *special* about that creature, object, or action, not what can be *assumed*. The difference between *a* mouse and *a* dragon is apparent, but needs to be accounted for in the system somehow.

To resolve this issue we use an additional mod called a "contextual modifier". In this case, it's a modifier representing the ability of *a* mouse relative to the ability of *a* dragon. This mod is not recorded anywhere because it is contextual- it only exists, indeed only *can* exist, in the context of the given contested check. This mod is decided by the Narrator and should take two things into account:

1. The relative power of the *Types* of the parties involved in the check, for the purpose of that check.
2. External factors that tilt the situation in favor of one party or another.

In our example scenario, the dragon has a clear, monumental advantage over the mouse in the context of dealing and resisting damage, simply due to the fact that it is *a* dragon, and the mouse is *a* mouse. The Narrator should select a contextual modifier that adequately accounts for the difference:

Magnitude	Modifier
Mild	2

Minor	5
Moderate	10
Severe	15
Profound	20
Extreme	30
Insane	40

Here, the contextual modifier should be in the ballpark of Extreme or Insane, for a Contextual Modifier of 35. Returning to the modifiers above, Sir Squeaks keeps his 21 modifier, but is now going against the dragon's modifier of 48. That makes a bit more sense. The mouse, even with all his skill and ability, could not effectively attack the dragon even if he rolled a perfect 11 and the dragon rolled a -11. A full-frontal assault simply will not work. This isn't to say that the mouse could *never* hurt the dragon- since contextual modifiers take both the attributes of the characters *and* the current situation into play, changing the situation will change the mod. With some clever choices about *how* the mouse attacks (targeting weak spots, waiting until the dragon is asleep, etc), the contextual modifier could be reduced, or maybe even swung in the mouse's favor.

Note : As a general rule, we have found it helpful to have a range of "usual" contextual modifiers in mind when a party goes up against a particular NPC. This range isn't set in stone, and creative solutions on the part of the party will almost always result in contextual mods outside these values. It's nice to have some idea of what kind of disparity your party is up against before actually getting into the situation. This is also helpful for when players enter into combative situations with NPCs that you have not written yet or had no intention of them conflicting with.

Characters

Descriptors

Just like all things in Fools and Corpses, the very first thing that must be defined about a character is its type. Depending on the setting, this could be the character's race, species, or some aspect of their background. The documentation for the campaign setting should contain this information, as well as a few example types that could be drawn upon for inspiration. Once you have your character's type, it's time to start adding attributes. For characters there are three categories of attributes: Traits, Skills, and Abilities.

Traits

Traits are the broadest type of attribute, and are fundamental to the character's basic functions- things like Strong, Fast, Smart, Tough as Nails, Stupid as a Box of Hammers, and Utterly Repulsive. As a general rule of thumb, these are things the character *is*, not things the character *can do*. As such, they are likely to be used very frequently, perhaps in nearly every check the character makes. Traits frequently apply to passive modifiers *and* active ones.

Skills

Skills are more specific, and result from active pursuit through training and education, or simple raw talent. Skilled Mechanic, Champion Bullfighter, Professional Dancer, and Knight Regent all fall cleanly into this category. Skills can be applied to tasks that relate to a character's experience or aptitudes. When a character is doing the stuff they're "good at", skills come into play. Skills rarely apply to passive modifiers, but frequently apply to active ones.

Abilities

Abilities are the most narrow attributes a character has. These are specific tasks the character has become particularly adept in, above and beyond what is expected of someone with their skills. Pick Locks, Dodge Melee Attacks, Command Animal, and Lie Convincingly are all Abilities. They are usually expressed as verbs, and almost never apply to passive modifiers.

Note: A character does not need to have a specific ability listed under "abilities" in order to attempt it, but if the GM thinks the character would have no idea how to complete the task, the character might be guessing.

Status Effects

During the course of play, temporary effects may be applied to characters. These effects are referred to as **Status Effects**. While they function like any other attribute a character might have, Status Effects don't count towards the character's level and typically only persist for a specific length of in-game time or until some other condition is met. Things like Drunk, Sleepy, Afraid, Stunned, Magically Strengthened, or Inspired are all Status Effects. The intensity of the effect which caused the status determines the number of ranks the character receives, and the nature of the status itself determines what kind of checks it affects, and whether that effect is positive or negative.

Health

In Fools and Corpses, all characters, great and small, have 36 Maximum Health. Since contextual modifiers allow "translation" from one creature's conceptual space to another, giving different characters different health totals is unnecessary. In essence, health is a percentage, but we start with 36 instead of 100 because 36 fits better with the mods generated by the attribute system. When a character drops to zero health, they are dead. A character may become unconscious at any point above zero health, depending on the narrative context, but zero is dead.

Calculating Damage

While there are some guidelines around damage calculation when it comes to attacks (see Making an Attack), there are plenty of other instances of damage that may occur during play. However, it's important to first answer a basic question: do we actually want to deal damage? It's easy to jump to damage as a mechanism in any case where a character incurs some sort of personal harm, but it's important to remember that health represents a very specific attribute of a character. Not all forms of harm apply directly to health, and it can be very interesting to apply other detrimental effects while leaving the actual health pool untouched. Status Effects are a great way to handle this kind of behavior, as they produce gameplay effects while also requiring a narrative reaction to alleviate them.

That said, sometimes damage is just the right thing and we need to know how to rate it properly. When deciding on a damage value, it is important to know what certain damage amounts translate to in the narrative realm. Here are some narrative descriptions of damage ranges and their effects when applied all in one event :

Damage Range	Description
1-2	Superficial - cuts and bruises, general wear and tear, does not cause pause.
3-5	Painful - deep cuts, heavy bruises, might cause momentary pause, but no Wounds or Status Effects.
6-11	Significant - fractures, broken bones and deep lacerations, mild concussions. Temporary Wounds with 2-3 ranks, depending on total damage.

12-23	Severe - shattered bones, loss of limbs, extreme blood loss, loss of consciousness. Lasting Wounds with 4-7 ranks.
25-36+	Extreme - catastrophic and lasting damage that may never fully heal. Grievous Wounds with 8+ ranks. 36+ damage causes death.

Whenever a character takes damage, either during combat or otherwise, the total damage is reduced by the total ranks of any passive attributes the character has that would absorb or reduce the damage, such as Hard Scales or Diamond Skin. Equipment such as armor or energy shields often applies passively. If the character reacts in a way that reduces the damage, they may do so with active attributes such as Combat Training or Block (in a Realtime Scenario, doing so would require energy).

Wounds

If a character incurs six or more points of damage from a single event after applying their resistances, they receive a Wound. A **Wound** is a significant, semi-permanent attribute describing the damage that was incurred. This could be something like Fractured Femur, Deep Gash, Torn Pectoral, Blood Loss, or Mental Breakdown. The Wound's rank is equal to the damage that caused the wound divided by three, rounded down. Wounds persist until they are treated in a narratively-appropriate way. This may reduce the Wound's rank or remove it entirely, depending on the quality of the treatment.

While wounded, the character's Maximum Health is reduced by the total number of Wound ranks they have multiplied by 3. This means that if a single character has Broken Arm (2) and Concussion (1), their Maximum Health is reduced by 9 ([1 + 2] * 3) leaving them with 27 Maximum Health. Characters cannot regain health above their Maximum Health through resting or other mundane means, though your setting might contain special items, spells, or abilities that allow them to do so.

Any actions taken that would utilize wounded body parts incur a negative modifier from the wound attribute in proportion with the wound's relevance to the action. If you're trying to swing a baseball bat with a Broken Arm (2), you're going to get a -2 to that check, whereas running a mile with that same broken arm might only incur a -1 penalty: half the wound's ranks.

Resting

One way to restore lost health is to rest. When resting, the first thing to determine is how long it takes for the resting characters to recover in a significant way. This period is called a Rest Cycle. The duration of the cycle varies between campaign settings and characters, but for most humans it is roughly four hours. The length of a rest cycle is typically *not* affected by the context of the rest, instead remaining consistent for all characters throughout play. Upon completing the Rest Cycle, all resting characters make a Rest Check to determine how much health they have recovered.

Check Result	Description	Health Recovered
Below -18	Horrible	0
-18 to -6	Poor	3
-5 to 5	Average	6
6 to 18	Good	9
Above 18	Fantastic	12

Resting for a shorter time than a full Rest Cycle should provide reduced benefits, including health recovery and any other benefits of the rest. If a rest is interrupted before the end of a cycle, The Narrator might grant a partial recovery based on the proportion of a rest cycle which was completed. For example, if the characters rested for half of a rest cycle, the Narrator could award them half the normal health recovery for their rest check.

A Rest Check can be affected by character Attributes and Contextual Modifiers just like any other check. Attributes having to do with physical regeneration, first aid, nourishment and comfort all affect this check. Equipment, such as beds, tents, magical accommodations, food, air conditioning, etc., all add to the Rest Check modifier in proportion with how well they help the characters rest. Conversely, situations such as sleeping in armor, nearby combat, hunger, loud music, extreme heat, and sleeping with your foot out from under the covers so a demon can take you during the night could all impact the rest roll modifier negatively.

Light, non-strenuous activity, such as keeping watch, mending clothes or eating a meal can be performed for up to one-fourth of the total Rest Cycle without breaking the rest. If appropriate first aid, healing magic or medical intervention is available, the health recovered via the resting check may be applied instead to the healing of Wounds at a rate of 1 Wound rank per 3 health recovered. Keep in mind that in a non-magical setting with current technology, significant wounds won't go away overnight, potentially requiring weeks or months to fully heal. Arduous medical intervention would not allow the healer to rest, but would provide resting benefits to the one being healed.

Inventory

Inventory restrictions vary widely in different settings, and should be handled on a case-by-case basis. A basic rule of thumb is, don't worry about inventory until circumstances make it important. If an object requires special accommodations to hold or carry, work that out, otherwise, assume it goes into a pack, pocket or pocket dimension without issue. If an object would create a significant encumbrance, it is likely accounted for in the item's attributes. If not, decide how much of an encumbrance the item would create (in ranks) and record it next to the object, for easy reference later.

Check in with your players periodically to see if they are collectively or individually carrying *way too many things.* If you think they are, you might impose a temporary Status Effect such as Overencumbered (X) until they put some things down, or simply ask them to explain how they're managing it. A character whose pack is a disorganized, overstuffed mess might need to make checks to pull out needed items in good time.

All that said, if your game requires or would benefit from a systematically regulated inventory system, feel free to get crunchier with it. This could be particularly important if your setting allows for material-based magic or relies heavily on other forms of ammunition. See Suggested Systems for an example inventory system that accounts for many of the important parts of inventory management without falling into some of the pitfalls that arise from strictly weight-based inventory systems.

Achievements

Through the course of play a character can grow, increasing in power and versatility as a result of the experience they've gained through their adventures. In Fools and Corpses, this process is quantified through achievements. **Achievements** are moments of significant growth, demonstrations of increased power, or the completion of a goal. Ultimately, it's anything that gets the Narrator and the players up out of their seats and cheering, rallying around the excitement of the story and the character's role in it. The Narrator awards achievements in the moment, between sessions, or at significant milestones in the narrative. Achievements can be awarded to particular characters or to the party as a whole.

Achievements have two parts- a description and a point value. An Achievement's description is usually a phrase or sentence that indicates what the character did to gain the Achievement. The Achievement's point value, on the other hand, indicates the Achievement's value for ranking up attributes when the character levels up. Achievement points usually have to do with the amount of effort and time it took to complete:

Point Value	Description	Example
1	Single notable actions or choices	"Cut Joram the Black's head clean off"
2	Multi-action efforts, usually involving other player or non-player characters	"Infiltrated the castle and kidnapped the princess by impersonating a guardsman’s long-lost brother"
3	Large-scale events or small quests the character took part in	"Located, entered and plundered the Hoard of Kle'nk, the Great Wyrm"
4	Awarded for completing multi-quest campaigns or other special cases	"Led the Green Miner Rebellion, resulting in the fall of the Xiang Collective"

Leveling

When a player gains Achievement Points that bring their total unspent Achievement Points to 6 or more, they Level Up. At this time the player spends exactly 6 of their Achievement Points to increase the rank of existing Attributes or purchase new ones. Once leveling is complete, the character should have fewer than 6 Achievement Points still available. If a level is gained (i.e. 6 Achievement Points are spent) and there are still 6 or more unspent Achievement Points available to the character, the character should repeat the leveling process, spending achievement points 6 at a time until they have fewer than 6 Achievement Points remaining.

Player characters begin at Level 0, with only their starting 12 achievement points. Each time they Level Up, their character level increases. Character level in Fools and Corpses serves mostly as a balance tool (see Designing Encounters), having no direct effect on gameplay.

Each type of Attribute has an Achievement Point cost associated with increasing itsRank. This cost is proportional to how broad Attributes in that category are. Gaining a new Attribute costs the same number of Achievement Points as increasing one you already have. All new Attributes start at Rank 1.

Category	Cost	Breadth

Trait	3	High
Skill	2	Medium
Ability	1	Low

The specific Achievements used to increase a given Attribute's rank or gain a new Attribute must apply intuitively to the Attribute in question. For example, Prepared Two Thousand Phenomenal Omelets (2) cannot be used to increase the rank of Pick Locks, but it may be used to add ranks to Master Chef. Any number of Achievements can be used to increase a single Attribute, i.e. points from Achievement “A” and Achievement “B” can both be used to purchase a rank in Trait “C”, so long as the Achievements both relate to that Trait intuitively. Additionally, points from a given Achievement can be split freely between multiple Attributes, or some portion of the points from an achievement could be spent while the rest are saved for later.

If an Attribute’s rank exceeds 6, an additional cost is incurred when increasing its rank. For every rank above 6 an Attribute grows, the player must also choose a “negative” rank to take in another Attribute of the same breadth. This may be a decrease in an existing attribute, or a new attribute that shows the character is “below average”. This “negative” Attribute should adequately represent the opportunity cost of focusing all of one’s effort on a particular skill. For example, a character who spends all their time in the library improving their Historian Skill to rank 7 might neglect their physical health, and gain the Physically Inept (1) skill, which would be applied negatively to checks involving athleticism.

When increasing the rank of an Attribute, the Attribute's description can also be changed in addition to the rank increase if the player desires. This change should be relatively minor, narratively appropriate, and is typically representative of an evolution in the character’s abilities. There is no explicit or implicit change to the Attribute’s power associated with changing the Attribute’s description, it simply changes the situations where that Attribute can be applied. A good example would be changing the Attribute Brawler (2) to Boxer (3)- the character has grown in their ability to fight, and has honed that ability in a specific way. The shift in the “domain” of the Attribute means it can likely be applied in some new situations (perhaps a Knockout Punch or Duck-and-Weave maneuver), but can no longer be applied in others (such as Grappling). The player is never required to change an Attribute’s description, regardless of how many times they increase the Attribute’s level.

Creating a Character

When a player wants to create a character, there are a few pieces of information they must provide :

Name - What is your character called?
Age - How old is your character (if necessary for the campaign setting)?
Appearance - What does one see when looking at your character? This description does not directly generate ranks or apply to checks (it's just for flavor).
Type - What is the setting-designated type of your character (usually race)?
Background - Where does your character come from geographically and socially?
Traits - What traits does your character have?
Skills - What skills does your character have?
Abilities - What abilities has your character practiced?
Inventory - What is your character carrying? (This may be parameterized by the campaign setting or the Narrator)

Your character's name, age, and appearance are all mostly for flavor and narrative grist- they don't *necessarily* affect the numeric aspects of gameplay, unless they do so by changing the narrative in a significant way (which is true of all things in Fools and Corpses). The character's **Type** is their broad noun or label. This is often a race like Elf or Kobold, but could be something like Martian or Corporate Shill depending on the campaign setting.

Background is a special Attribute that describes the character's personal history- were they a street urchin, or a merchant's apprentice? Did they grow up as part of the royal family or as a survivalist in the wilderness of Kentucky? Once selected, the character's background automatically has two ranks added to it.

When creating a brand new character, they start with twelve achievement points to use to purchase **Attributes** (i.e. Traits, Skills or Abilities). Their prices are each listed above (see Rank Leveling). Each rank added to an Attribute allows it to move one Semantic Step from Average. For example, putting four ranks into your character’s beauty could take them from Average (0) to Mildly Attractive (1) to Pretty (2) to Beautiful (3) to Gorgeous (4).

Finally, the significant (i.e. not *assumed*) items a player character is carrying should be listed, along with their ranks. What kinds of items are allowed for a brand new character depends on the campaign setting and your Narrator, but here’s an example of a starting setup:

- One Kit, rank 3 - A set of tools or a pack with standard utility items for a person with your skills and background. The kit can be used to make a check to see if it contains a specific needed item at the time when it's needed, with the DC dependent on how likely the item is to be contained in the kit.
- Two mundane items, rank 2 - Things that are easy to come by in the setting, but not everyone carries.
- One special item, rank 4 - Something that is somewhat hard to come by, which your character acquired under special circumstances.

Realtime Scenarios

In Fools and Corpses, a **Realtime Scenario** is any situation where the exact order of actions is extremely important or the passage of time needs to be tightly controlled. The most typical Realtime Scenario is combat, but certain kinds of social situations, escape sequences, or non-combat competitions may also be considered Realtime Scenarios. There are also cases where combat-like situations are handled fluidly using the standard check system without invoking the realtime system, which is completely fine so long as everyone feels they got to have their say and the situation was resolved fairly. Indeed, the actual *actions* in Realtime Scenarios are made using the same check system as all other checks in Fools and Corpses. The realtime system just determines *when* you can make a check, and what checks you can make.

Energy and Rounds

During Realtime Scenarios, time is broken into rounds of about six to ten seconds of in-game time. During each round, every character has a pool of three energy they can expend. A character must expend at least one energy to act in any way, be that by attacking, moving, reacting to another action, activating a console, or calling out a lie. If a character takes meaningful and willful action to affect the state of play (even as a reaction to another character's action) they must spend energy to do it. Non-willful or passive interactions with the world, on the other hand, cost no energy. The round is over when all characters have expended all their energy for the round, or decided not to act further in the round. Energy is then replenished and a new round starts.

Reactions

Whenever a character takes an action, any number of other characters may choose to react. This **reaction** is a quicker action incited by the initial action that starts just after the initial action starts and finishes just before the initial action finishes. Reactions can be used to directly contest the initial action, or just to accomplish something unrelated *before* the initial action is completed. Since a reaction is an action, it costs at least one energy. Any number of characters can react to a single action, but each character can only react once before the initial action finishes. A character can only have one action in progress at a time, so the initial actor could not react *to anything* until their initial action is completed. All reactions to a given initial action must be declared before any of the reactions are resolved.

Resolving reactions functions just like resolving actions (see Making a Check), though the Narrator should apply a Contextual Modifier called a Reaction Penalty to reaction checks, making the checks harder, since they must be completed more quickly than normal actions. This modifier could have any value depending on how burdensome it is to complete the specific action quickly, but unless an action is uncommonly quick or uncommonly slow, the **Reaction Penalty should be 5 per reaction**.

A character can also react *to a reaction,* and another character could then react *to that reaction to the first reaction*. This could continue indefinitely with enough characters involved, but each layer deeper into the reaction stack we go, the actions must become faster and faster. Thus, the Reaction Penalties should become multiplicatively higher for reactions to reactions to reactions, etc.

For example, Gwendolyn attacks Stuart for the Initial Action. Stuart reacts by bashing Gwendolyn's blade out of the way with his shield, and Stuart's friend Catherine reacts to Gwendoly's attack by trying to put a knife in Gewndolyn's kidney. Here, Gwendolyn suffers no reaction penalty, while both Stuart and Catherine are reacting to the initial action and incur a penalty of 5 on their checks. Now, if Gwendolyn's friend Brian wants to stop Catherine's kidney stab by casting a paralysis spell, Brian is reacting to a reaction, which would incur double the normal Reaction Penalty: 10 subtracted from his check. Depending on how magic works in your universe, the Narrator might rule that Brian's spell is more difficult to perform quickly than a melee attack, and increase this penalty beyond 10.

Combat

Combat is the most common form of Realtime Scenario. Below are several mechanisms for quantifying combat, but they should not be viewed as the only ways to act in combat. There is simply a lot going on during combat sequences, and we wanted to provide our players with a framework for thinking about all these factors meaningfully.

Order of Play

In Fools and Corpses combat, there is no set turn order. Play proceeds on a first-come, first-served basis, with “initiative” being granted to whoever pipes up with their action first. Some characters might fling themselves into combat immediately, while others might prefer to sit back and save their energy until the opportune moment. That said, if two or more characters wish to act *simultaneously*, and the results of each character’s actions might affect the others, it may be necessary to determine the order before proceeding.

To do this, each character trying to act first should roll a check, adding initiative-relevant attributes like quick, alert, focused, Sword of Vigilance, etc. This check is not an action, but a free check (costing no energy) to see who acts most swiftly. The highest result on this check acts first, after which play proceeds as normal. The other characters can still react as normal to the winner of the initiative contest, or wait until the winner’s action has been resolved to attempt their own actions.

Making an Attack

Narrative is everything in Fools and Corpses, and combat is no exception. Players and Narrators alike are encouraged to "call their shots", i.e. not just state they are attacking a target but describe *how* they are attacking. What part of the target are they striking? Are they doing so cautiously or recklessly? How many times are they striking? This is necessary because it helps determine which attributes from both the offense and the defense apply to the check. Describing precisely how you want to attack or defend is extremely helpful in this process, and adds narrative flavor to the situation. Players can attempt to attack in any way they want, but the difficulty of the attack should be accounted for by the Narrator by applying penalties or increasing the DC. Like all other actions in combat, making an attack costs at least one energy, but depending on how the attack has been described, this cost may increase.

Granted, not everyone thinks like the characters they play (really, this is part of the fun of role playing in the first place). There may arise scenarios where the player wants their character to "do their thinking for them". This means that the *character* has training or knowledge about combat or the scenario they're in, and the player wants them to act in light of that. At times like these it's appropriate for the Narrator to step in and help describe how the character would likely act in this situation. The Narrator might request a check to determine if the character knows what to do in the present scenario. This kind of check should not cost any energy, as it doesn't represent an action but rather a determination of the current state of the character's mind.

Landing Check

The **Landing Check** determines if the attack hits its target, and is executed just like any other check (see Making a Check). The **Base DC** for any attack is determined by the weapon that is being used. This DC may increase or decrease depending on context, especially when attempting to increase damage (see Special Shots). The Landing Check is completed using the ranks from any attributes that would help the attacker *land* their attack, as well as ranks from any weapons being used in the attack. When an attack lands (i.e. the character beats the attack's DC), it means that the weapon has made contact with the target. It does *not* mean that the target has *necessarily* suffered damage from the attack, since the damage mitigation from the target has yet to be applied.

Damage Calculation

Once an attack lands, the total damage to apply to the target needs to be calculated. The starting value for total damage is Base Damage. Like Base DC, Base Damage depends most critically on the weapon used in the attack. **Base Damage** describes how much damage is dealt when using this weapon in the most basic way. This damage may be increased or decreased depending on the context, especially if the characters in the conflict have drastically different statures, or if mitigating circumstances leave one of the characters especially vulnerable (e.g. if the target is asleep). If a player describes attacking a certain body part or notable vulnerability of their target, the attack's Base DC and Base Damage should both be increased (see Special Shots).

The table below should be used as a reference, both to assign damage values and base DCs to the weapons listed in the table, and to figure out the properties of other weapons in your setting that are *not* in the table. For any weapons not listed, choose a tier with examples similar to the weapon in question. An exotic weapon could sit between tiers, or have a higher base DC but lower damage than any particular tier.

Tier	Example Weapons/Attacks	Base DC	Base Damage
1	Slap, Harmless non-weapon object	-5	1
2	Punch, Dart, Elbow, Kick, Knee, Small Knife, Shield Bash, Mildly dangerous non-weapon object	-5	2
3	Light Weapons: Light Pistol, Sling, Brass Knuckles, Dagger, Light Club, Quarterstaff, Shortsword, Spear, Light Mace, Hunting Rifle, Hunting Bow, Hunting Crossbow, Javelin, Spiked Shield	0	6

4	Heavy Weapons: Hunting Shotgun, Longsword, Warhammer, War Axe, War Bow, War Crossbow, Submachine Gun, Heavy Pistol	0	8
5	Combat Shotgun, Greatsword, Greataxe, Military Rifle, Maul, Poleaxe	5	12
6	Grenade Launcher, Machine Gun, Oversized Fantasy Weapons	10	18
7	Extremely Powerful Effects	10+	18+

Special Shots

In order to deal more damage than a weapon's Base Damage, a character must call their shot in a way that makes their attack more damaging. Essentially, they must choose to sacrifice accuracy and quickness for power, or choose a target (a headshot, for example) which is harder to hit but more vulnerable. If a character wishes to call a more damaging shot, they should reference the table below to choose their preferred strike power, balancing the increase in DC with the potential for extra damage. This can also be used to improve the utility of an attack, by striking multiple opponents or attacking at a long range. In these cases, the attack wouldn't deal any more *damage,* but its effectiveness would be improved narratively. In order to call a shot in the table below, a character must be using a suitable weapon for the effect they desire. You can't cut someone's hand off with a quarterstaff, for example, or at least, the DC mod would need to be vastly higher than +15.

DC Mod	Examples	Bonus Damage
+5	Deep cuts, Cracked ribs, Significant fleshwounds	+3
+5	Moving a short distance (~10-20ft) to strike, Mid-range shots (~20-30ft), Ranged attacks when within reach of an active melee opponent	None
+10	Broken bones, Eye gouges, Multiattacks (one target), Non-lethal headshots, Tendon cuts	+6
+10	Striking two opponents, Long-range shots (~30-100ft), Moving a long distance then striking (~20-60ft), Parrying a blow and striking as a single action	None
+15	Removing hands or feet, Deep puncture wounds, Flurries of many blows (one target), Severe Concussions, Arterial nicks	+9

+20	Evisceration, Limb removal, Throat cut, Severed arteries, Cracked Skull	+12
+20	Striking 3+ opponents, Very long-range shots (~100ft+)	None
+40 or more	Decapitation, Right between the eyes, Heart-shot, Cleave-in-twain	+18 or more

Special Shots are not the only way an attack might deal more or less damage, or have its DC increased or decreased. The Narrator should also take into account Contextual Differences between the attacker and defender, which most often manifests in their relative size. A larger creature should deal extra damage if it's striking a smaller creature, and the smaller creature should deal less damage when striking the larger one.

The examples in these tables do not cover all weapons and scenarios, so Narrators should be agile in applying penalties and bonuses based on specific narrative context.

Once the attack's damage has been determined, subtract any of the defender's relevant attributes or item ranks which would reasonably mitigate the damage. This Mitigation might include ranks from armor, Tough Skin, Hard Scales, or any similar attribute. Such Mitigation cannot reduce the damage dealt below zero. Be sure to account for the called shot when deciding whether to apply ranks from armor, scales or skin, as they may not be applicable if the attack struck an unprotected portion of the defender.

Reacting to an Attack

Any character can react to any action, and attacks are no exception. Often, the target of an attack will use a reaction to defend against the attack in some way. The defender must decide to react *before* the Landing Check is rolled, and must have enough energy to perform the reaction. Reactions can be used to attempt to evade an attack entirely, reduce the damage it deals, or accomplish an unrelated goal. If the reaction would avoid the attack entirely (e.g. a dodge, parry, or disabling action) the reaction is treated as contesting the Landing Check (see Checks with Opposition). If the reaction would mitigate the *damage* of the attack but makes no attempt to *prevent* the attack from connecting (i.e. blocking with a shield), the reaction check is not treated as contesting the Landing Check, but changes the state of play for damage calculation afterwards, typically by reducing the damage.

If a defender is actively defending to oppose the landing check, the Base DC of the attacker's weapon and any DC increases incurred by the called shot should be added as a Contextual Modifier to the defender's Defense Check.

Example Exchange

Let’s take a look at an example to make this a bit more clear. This time, let’s set up the Dancer from our earlier example against a Scoundrel. They’ve been quarreling over a necklace they stole, and the Scoundrel has snatched it up from where it lay on a table between them. The Dancer, incensed by his audacity (or perhaps just his superior speed), grabs her Trusty Blackjack and lunges over the table at the ne'er-do-well, seeking to break the fool’s jaw:

Daisy Dancepants
- Clumsy (2)
- Smart (4)
- Quick (5)
- Dancer (4)
- Trusty Blackjack (2)

Daisy’s Landing Check is affected by many of her attributes. Quick (+5), half her ranks in Dancer (+2), and her Trusty Blackjack (+2) all contribute positively, while Clumsy applies negatively (-2). In total, her modifier for the check is +7. She spends one energy on the attack.

NOTE : If any attribute would make the character hesitate, move slower, or otherwise encumber the attack, that attribute's ranks should be added negatively to the modifier. This includes weapons and items- in some cases, the ranks of an unwieldy weapon should act against the player, making it harder for them to land the hit.

Tommy Ten-Toes
- Nimble-fingered (4)
- Dumb as a box of hammers (2)
- Tough (1)
- Scoundrel (4)
- Beat-up bowler hat (1)

Tommy doesn’t use an action to resist, as Daisy’s sudden attack caught him off-guard. The blackjack is small, lightweight, and oftentimes non-lethal, meaning it lines up cleanest to Tier 2 weapons. This sets the Base DC for Daisy’s attack at -5. However, Daisy isn’t just trying to hit Tommy’s center-of-mass. She’s swinging up under his chin, making this a “non-lethal headshot”. This kind of Special Shot will tack an additional 10 points on to the DC, bringing it up to 5. Furthermore, the table’s obstruction makes the check more difficult, and the Narrator decides that this should add 2 more to the DC, resulting in a final DC of 7.

Daisy rolls and receives a 6 on the positive dice and a 4 on the negative dice, totalling 2. Adding Daisy's modifier of +7 brings the grand total up to 9! The dancer's hit lands!

Damage Calculation

Since the Trusty Blackjack is a light weapon, it deals 2 base damage. Not much to write home about. However, since Daisy went to the trouble of calling her shot, aiming for Tommy's chin instead of somewhere more robust, she has dealt an *additional* 6 damage, bringing her total to 8. Tommy's Tough (1) trait will reduce this damage by 1. Tommy's Beat-Up Bowler Hat (1) *would* reduce the damage further had Daisy aimed for the top of his head, but since the shot landed on his chin, it did nothing to protect him.

This leaves the final damage total at 7, enough to leave Tommy with a rank 2 wound. Something like a mild concussion or broken jaw would be appropriate, but the Narrator decides to get a little creative and awards Tommy Two Broken Teeth (2) instead.

Reacting to the Attack

But what if Tommy wasn't a lazy ingrate who wouldn't lift a finger to save his own skin? What if he spent energy to dodge? Daisy's mod for her Landing Check would remain the same, but Tommy would get a chance to contest the roll using any dodge-related ranks.

Being a Scoundrel (4), he's seen his fair share of tussles, but there's not much else about Tommy that will help him out of this pinch. So Tommy's Attributes yield a +4 for his check to negate Daisy's attack via dodging. The table gives Tommy a +2 Contextual Modifier to his check because the obstruction is working in his favor. If either action was significantly easier than the other, that party would receive a positive contextual modifier to reflect that, but since both parties are executing actions of comparable difficulty (DC of 5) only the table needs to be taken into account. There is one last modifier to consider for this check: the Reaction Penalty. Tommy is reacting to Daisy's attack, so he must move swiftly to avoid it. With a reaction penalty of -5, it is going to be quite difficult for Tommy to pull this off.

We'll keep Daisy's same roll from earlier, yielding a final result of 9. Since Tommy is actively dodging, he rolls his check. He must exceed Daisy's total roll of 9 in order to be successful. Let's say Tommy rolls a 11 on the positive die, and a 4 on the negative die. This yields a +7. Adding his mod of +6, the table modifier of +2 and subtracting the Reaction Penalty of -5 results in a total of 10, one point higher than Daisy's attack. The attack misses, no damage, and the results were close enough that neither party is adversely impacted by the exchange.

This can be a lot to juggle at first, so let's recap the steps for attacking and defending.

1. The attacker declares their attack and describes the action. This may or may not include Special Shots.
2. The defender decides whether to actively resist the attack or not.
3. The Narrator decides any Contextual Modifiers either party would receive.
4. The defender rolls the Contesting Check (if applicable).
5. The attacker rolls the Landing Check.
6. If the attacker exceeds the *greater* value between the DC and the Contesting Check (if applicable), the attack lands.
7. Damage is calculated based on Base Damage, Special Shots (if applicable), and active defense (if applicable).

Passive Resistance

Oftentimes, there are factors that can make an attack harder to complete even if the defending party is not actively reacting to the attack itself. For example, if a Swordmaster wanted to attack a plate-mailed enemy by artfully weaving their rapier between the armor's plates, the ranks of the armor should be included in the DC calculation for the attack. If the attack fails, it means the blade never found its way between the plates. However, if the attack succeeds, the blade found its way between the plates, meaning the armor could no longer help mitigate damage.

On the other hand, if a huge brute of a monster wanted to crush that same well-armored character under his huge club, the armor wouldn't do much of anything to prevent the hit from *landing*. In fact, it may even encumber an attempted dodge. But a successful hit doesn't mean successful *damage*- the armor may be able to soak a significant portion of the huge club's enormous weight, leaving the wearer (nearly) unscathed.

In both these cases, the armor plays a role even if the armor's wearer does not actively resist (i.e. use an energy to react to) the attack. Armor isn't the only factor that could influence the Landing Check in this way. Special types of training, magical effects, technology, and contextual aspects could all make a Landing Check more difficult without the use of energy *so long as they function passively.* A good rule of thumb is : "If you have to choose to do it, it costs energy." If it *doesn't* cost energy, it's allowed to come into play during the Landing Check passively.

Designing Encounters

The rules of combat can be a bit intimidating at first, but with a bit of guidance, designing encounters in Fools and Corpses is pretty easy. Since attributes and mod work more or less the same regardless of the context, the following guidance can be used to design non-combat encounters just as well as combat encounters, making even on-the-fly planning relatively light-weight.

Multi-enemy Encounters

The most reliable way to create a balanced combat encounter is to create a group of enemies with similar numbers and skill levels to the player characters. These enemies could be created as fully-fledged player characters, or produced in a more general sense based on the following example. Here, we split the character's modifiers into three categories: Primary Actions are the most common actions a character takes as they go about their daily lives. They have extensive training in these actions and are very, very good at them. Secondary Actions are actions they have some training in, but don't do all the time, and Tertiary Actions are actions they take rarely, but know well how to do. The bonus for all other actions should be zero. Ranks for all action types should be rounded down.

Generic Enemy
Primary Action Modifier = (Player Character Level + 3) * 2
Secondary Action Modifier = (Player Character Level + 3) * 1.5
Tertiary Action Modifier = (Player Character Level + 3)

For example, a Level 3 party might come across a band of Hired Guns, who would pose a significant challenge to the party if their numbers are similar:

Hired Gun
Primary Action Modifier: 12
Secondary Action Modifier: 9
Tertiary Action Modifier: 6

The Narrator should use their best judgment when deciding what actions fall under each category. In addition to these modifiers, give your generic enemies a damage reduction value from armor or other factors, and decide what weapons they're carrying to determine their base DC and damage for their attacks.

Boss Encounters

Another way to present a combat challenge to a party is the classic boss battle: one very powerful enemy whose prowess vastly exceeds each individual party member. Together, though, and with a little ingenuity, the party can prevail! To create a boss enemy, several factors need to be considered.

Boss Attributes

Bosses should have significantly more attribute points than the player characters. Start by deciding what makes the boss threatening. Are they a towering giant? An unparalleled blademaster? A cunning mage? Give them attributes that make them incredible at whatever they're good at. Generally, a boss should have roughly double the number of achievement points of the party they're up against. For example, a Level 5 party (42 achievement points) would be well-matched against a boss with 84 achievement points. You can also take a more general approach to your boss, and build it using primary, secondary and tertiary actions. If you choose this path, simply double the bonuses of a normal enemy for the party's level, as detailed earlier in this section.

Take care to consider the amount of damage the boss will deal. You want your boss to be threatening, calling devastating shots and taking chunks out of the players nearly every time they attack, but you don't want them to be outright killing the players in a single strike. Additionally, damage should be reduced and combat slowed down in settings without easy access to healing.

Called shots are a huge part of Fools and Corpses combat, so be prepared for your party to disable some or all of your boss' capabilities. Every boss should have multiple ways to threaten the party, in case one or more of them are damaged during the fighting through wounds or other narrative effects.

Damage Reduction

Since all characters in Fools and Corpses have 36 health, your bosses will rely on Damage Reduction to keep them alive in combat with your player characters. You can give your boss damage reduction through the use of Traits, or simply give them a static Boss Trait that defines this parameter. It is usually wise to give a boss at least two forms of damage reduction: one which can be circumvented with a little ingenuity from the party, and one which cannot.

As a general guideline, bosses should have **unavoidable damage reduction at least equal to the player characters' level.** You should tune this amount based on your specific group of player characters. If your players are all highly combat-optimized strikers, the boss will need significantly higher damage reduction if it's going to make it through even a single round of combat.

Getting around the boss' avoidable damage reduction should be integral to the party's combat strategy. A boss might have armored plating that could be damaged, hard scales that cover everything but its eyes and mouth, or a deflector shield that could be overwhelmed with a barrage of quick strikes. The values of this type of damage reduction can be extremely high, essentially making the boss invincible if the party doesn't choose their attacks wisely to circumvent it.

Be creative with avoidable damage reduction! It's your best tool to make your party think on their feet rather than simply smacking the boss until it dies.

Extra Energy

Finally, even the strongest boss will be torn to pieces by a party of many adventurers if the boss only has 3 energy to spend each round. No matter their mods, simple action economy will almost always result in an embarrassing defeat for the boss. To combat this, bosses and other very threatening enemies should be given additional energy in each combat round.

A good guideline for this extra energy is to take the number of player characters and subtract one. For example, a party of four player characters should come up against a boss with 6 energy per round.

Conclusion

These rules are offered as a simple, structured way to think about collaborative storytelling. As such, it is very important to always remember that you are telling a *story*, and the *story* should take precedence over the rules. The rules, especially the numeric ones, are here to help you think about something you're creating together, but what you want to create should be the first and most important standard for success. If you have told a story you find compelling and interesting, then Fools and Corpses has done its job, even if the numbers have been modified, tweaked, or ignored.

This is your game, your story- it belongs to you and the people you create it with. The rules are just here to make your story tangible, to give your game legs. They are here to make sure you don't have to do all the thinking we already did while putting them together. That way you can focus on what you actually want to create- something meaningful, robust, and above all, *good*. Something we can't wait to see.

Share it, won't you?

Appendix A: Optional Rules

In an effort to keep the official rules of Fools and Corpses bare-bones, we've relegated several rules and play styles to this set of optional rules. These rules can be implemented, and more importantly, tweaked as needed, and by no means represent all possible extensions to the Fools and Corpses system. We've found that in most cases it's best to establish which optional rules are in play up-front, then keep those rules consistent during play unless significant issues are encountered. If you take nothing else away from this section, let it be that Fools and Corpses is more than just a ruleset- it's a platform that is easily retrofitted to nearly any use case, so long as its central design tenants are adhered to consistently.

Multi-Energy Actions

Sometimes during combat or other Realtime Scenarios, it makes sense to dedicate additional time and effort to a single task instead of taking three separate actions. The multi-energy action rule is aimed at making this strategic choice worthwhile. When taking any action, a character can choose to expend additional energy beyond that which is required to execute the action. For each energy spent beyond the first, the character is awarded a +5 mod.

Spending additional energy is often used to fit multiple actions into the timescale of one action. This is best used to make utmost use of an opportune moment which arises in combat. For example, if Warrior Joe wishes to interrupt the spellcasting of a distant mage, he could react by spending two energy to sprint forward and grab the caster's hands before the spell is cast. Joe should roll a single check which incorporates all of his attributes related to sprinting *and* grappling, as well as the energy bonus, contesting the mage's own check for their spellcasting. Joe would suffer normal reaction penalties since he is reacting to the mage's action, and would suffer Contextual Modifiers to account for the complexity of the multi-part action.

Finally, players are encouraged to use extra energy in inventive ways to extend the capabilities of their character. Ultimately, though, it's up to the GM how this mechanic can be reasonably used in a given setting or situation.

Surprise Rounds

When a situation arises where one character or group "has the drop" on another, that character or group should gain some sort of obvious advantage over the unprepared party. This is typically referred to as a **surprise round**. In Fools and Corpses, a surprise round is a special round of a Realtime Scenario where one character or group gets to act while the other is still getting their bearings. During this round, each character of the surprising party gets to act once, while no characters in the surprised party get to react at all. This single action can use more than one energyif the Multi-Energy Actions optional rule is also in use, up to a maximum of three energy. Once each character from the surprising party has acted, the surprise round is complete and the next turn of the Realtime Scenario starts as normal.

Event Actions

In some cases, resolving a large-scale event needs to be done quickly- this may be due to time constraints on play, or simply because the exact actions and choices made during that particular event won't enhance the enjoyment of the game. At times like these, **Event Actions** are a good way to ensure that the event's resolution takes the character's strengths and weaknesses into account, while still allowing the event to resolve quickly. Event Actions can occur in one or multiple steps, depending on how expeditious the GM wants to be in concluding the event.

Each step of an Event Action starts with the Narrator describing the state of play. Based on this description, each character involved in the Event Action should decide what they would do to affect the situation and contribute to a favorable outcome. Each character then rolls to determine their success in their piece of the endeavor, adding modifiers that apply to their action. A character's contributions to each step can and should encompass multiple actions, and all modifiers that apply to *any* of those actions should be added to their roll. The outcome of these collective checks determines the result of the step. Once all checks are resolved, the Narrator describes the results and moves on to the next step.

In a case where all characters involved in the Event Action are working together toward the same goal, the Narrator should set a DC for the overall check. If half or more of the involved characters meet or exceed that DC, the overall result is success. If not, the overall result is failure. That said, the Narrator should account for *all* of the characters' checks in their narration, whether the group as a whole succeeded or failed.

If, instead, two sides are at odds to determine the outcome of the event action, the Narrator should compare the total rolls on each side to determine who emerges victorious.It may also be appropriate for the Narrator to award Contextual Modifiers in this kind of Event Action, to account for clear advantages on one side or the other.

Group Actions

One of the more fun aspects of role playing is working as a group. Oftentimes this is expressed as each individual doing their best at the role that they play in the party, but sometimes the team has to come together and pull as a group, all helping out to complete a single task. This is what's called a Group Action. A **Group Action** is when several characters all attempt to complete the same task at the same time. However, there's often more than one way to skin a cat. Each character may help out with the task in whatever way they see fit, with the understanding that indirect assistance may only apply part of a character's mod to the group's task.

For example, let's say the party is trying to tip a boulder over the edge of a cliff onto a camp of goblins. All the big, strong characters will likely just put their shoulders to the rock and push, applying all their strength-related attributes to the effort. But what of the smaller, faster characters or more intelligent ones? Their skills might be better suited to digging out the dirt on the downhill side of the boulder or constructing a lever to improve the strong characters' leverage. This collective effort results in a collective modifier, which takes into account each character's unique contribution to the effort. The group then rolls one single check and adds their collective modifier, rolling against a DC determined by the Narrator.

Blanket Attributes

Blanket Attributes are large-scale attributes that apply to an entire category or situation. These attributes are great for "tilting" a situation, region, or entire campaign setting towards desired behaviors and outcomes. They can be used for everything from expressing narrative flavor in a campaign, to magical effects in an area, to increasing tension in an argument. When a Blanket Attribute is added, the Narrator records the attribute's information (description and rank) either privately or publicly. The attribute is then taken into account during all mod calculation. Private blanket attributes are useful for creating a "feeling" or "atmosphere" in the session that the players can't quite put their finger on, while public blanket attributes are good for incentivizing certain behavior. Blanket attributes should be added before characters start interacting with the "space" where the attributes would take effect, as well as how the attributes could be affected by play (i.e. raised, lowered, or removed), if they can be at all.

Example Blanket Attributes

- "This is an action movie (10)", Public : A blanket attribute to encourage players to jump out of windows, drive cars off of bridges, and get into gunfights on top of trains.

- “Pressure cooker (3)”, Private : Measures the tension in the room where two warring factions are trying to come to peaceful terms. Player and NPC actions raise or lower this value, making it easier or harder to get people to put down their weapons and come to the table.
- “Long Crisis (4)”, Private : Static attribute to account for the strain the adventuring party has undergone while dealing with a natural disaster.

Achievements for Failure

The name “achievement” typically evokes thoughts of summiting mountains, vanquishing foes, and freeing captives. But achieving isn’t the only way you can grow- indeed, failure sometimes teaches us more than success ever could. It is fully within the Narrator’s rights and capability to award achievements for significant failures, losses, or even things the players don’t *want* achievements for. These “bad” achievements can lead to significant character color and major shifts in the way the players think of their characters, all of which drives good story telling. What’s more, spending these achievements during leveling codifies these failures into interesting, evocative attributes that have the potential to take a cool character and make them astounding. Success is great, but failure adds dimension. Don’t overlook it!

Roleplay Coins

Roleplay coins (or RP coins) are a special gameplay mechanic that helps encourage meaningful roleplay. RP coins are a lot like an achievement in that they are awarded to players during play by the Narrator. They are *unlike* achievements in that they are awarded for acts of particularly evocative and true-to-character role playing. Where achievements measure and express impressive actions on the part of the character, RP coins measure and reward roleplaying that makes the session more realistic or interesting, whether its effects in-game are impressive or not.

Each coin has a name describing the act that earned it, and a player can have up to two coins at a time. If a player already has two coins and would earn another, they may choose to discard one of their existing coins to make space for the new one, or they can choose to forgo gaining the new coin. A RP coin can be earned from the same event that generates an achievement, so long as the parameters for both are met in the same action.

Once it is earned, RP coins can be spent during play to gain a narrative advantage. The nature of this advantage has to do with how closely the *themes* of the situation where the coin is spent match the themes of the situation where it was earned. The closer the match, the more advantage is granted. On a weak match or no match at all, a contextual modifier of 2-4 is awarded to the roll of the player’s choosing. On a moderate match, the Narrator twists the narrative in the player’s favor slightly, “tilting” the story towards the result the player desires. On a strong to perfect match, the player is granted a minor amount of narrative control over the world, within reasonable limits. Achievements may be earned as the result of spending an RP coin, but additional RP coins cannot.

Example Roleplay Coins

One More Body In This Grave
Earning : Braug stares down a charging illusory troll in a hallway of a deep catacomb, calling its bluff.
Usage : Braug threatens to collapse an entire room of the catacombs on top of the party if they do not leave a hypnotic, evil relic undisturbed.
Match : High, since both cases have to do with Braug’s hardline view of the world and horrifying nihilism. Also it has to do with leaving more bodies in the grave they find themselves in.
Result : Braug’s player gets to narrate a situation where the lever he’s holding on to creaks. This lever will collapse the entire room on top of the party if pulled. Several pebbles fall from the ceiling onto the hypnotized party members- just enough to snap them partway out of the artifact’s thrall. The rest of the party seizes the opportunity, yanking the victims out of the room and free from the artifact’s clutches.

Wounded Soldier
Earning : Rian’s arm is nearly wrent completely off in the midst of a fight with a great beast. He knew he couldn’t win, but he took it on to save his friends the misery.
Usage : Taken hostage, Rian tries to identify with his captors whose friend is lying wounded in their camp.
Match : Medium. Rian is capable of identifying with the “woundedness” of the downed man, but not his motivations. Rian sacrificed his health out of love for his friends. This man sacrificed his health out of obligation to his boss.
Result : The Narrator picks that moment for the wounded man to stir, crying out in pain. This gives Rian a chance to advise the captors on how to relieve his pain. While this is going on, Rian’s friends are able to work on escaping because their captors are distracted.

Appendix B: Suggested Systems

Inventory Management

For this inventory system, inventory is measured in slots. A slot is essentially a measurement of size or volume. Each slot is relatively small- roughly what a character can hold in a hand or two. This size is flexible on purpose. An item that takes up one slot is essentially "small", three slots is "medium", and six slots is "large". Items can have slot counts between these values or exceeding six slots, as well as "fewer" than one slot- this is handled by designating how many of that item fit in a "stack", which itself has a size of one slot (i.e. a stack of arrows may contain 8 arrows in one slot).

Any item that increases a character's carrying capacity (packs, bags, suitcases, etc) is said to be a "container" and designates a number of slots it provides. An item cannot be split between containers, but a container can, of course, contain multiple items. A fully clothed character is assumed to have three slots besides their hands, allocated as one container of two slots and one container of one slot. Sheaths, holsters, and other special-made containers don't have any slots and can only hold items of the type they are designed for or attached to. Items that have straps, belts, or other things that enable carrying without a container do not require slots.

Slots	Description	Examples
1	Small	Handgun, box of shells, baseball, knife, blackjack, tinderbox, encyclopedia
3	Medium	Bedroll, baseball bat, buckler, short sword, tea service, shotgun, hunting rifle
6	Large	Portable battering ram, tower shield, greatsword, encyclopedia britannica

NOTE : A slot has no specific weight. This isn't to say that weight is not taken into account during play. Rather, since weight is only likely to be significant in extreme (and thereby easily notable) cases, it is handled narratively. Dragging the contents of the dragon's hoard out of the deep cave is a story unto itself. We should tell it!

Magic and Casting

Magic and casting are very common in tabletop RPGs, but since there is no real-world analog for these systems, they can be difficult to implement. Your setting may have all manner of different approaches to magic, but the system below can be used as a simple numerical framework for how magic works in-game. That said, if you want to make up your own magic system, go for it! The system below is simply an example.

Base DC

Spells, like all other tasks in Fools and Corpses, require a check to cast. Depending on the difficulty of casting in your setting, the base DC for spellcasting could be quite low or very high. We suggest a base DC of 0.

The Base DC for a spellcaster is a starting point to calculate the DC required to cast any magic with significant effects. The Narrator should add to or subtract from this DC to account for the spell's complexity and overall power.

As a general guideline, a spell with Base DC should have an effect which is approximately as powerful as a mundane attack, like firing a crossbow or chopping with an ax. Effects more powerful or complex than this, targeting more enemies, having a longer range, or having a longer duration should have their DC increased to compensate.

Scaling Difficulty

Magic is a finite resource, and becomes more difficult to cast without resting and recovering between spells. Depending on your setting and the narrative element of your magic system, this rest might take the form of physical rest, meditation, communion with a deity, or gathering spell components.

To account for these factors, we use Scaling Difficulty. Each time a character tries to cast a spell, whether they succeed or fail, the character receives one negative rank in the Magical Strain status effect. Magical Strain works against the character when casting further spells. Since this status effect is cumulative, the fifth spell a character casts between rests would have its DC increased by 4. Magical Strain typically resets when a character finishes a Rest Cycle (see Resting), but there could be other ways to reset it in your world.

In a narrative sense, Magical Strain could represent many things, depending on your setting. Are the characters drawing from a well of raw magic within themselves? Do they drain energy from the environment around them? Are they running out of their magical components? Do their fingers get tired from playing their magic lute? Is their chosen deity getting cranky at having their power used all the time? Describe (and possibly rename) Magical Strain, and let your players reset it, in a way that makes sense in your world.

A Narrator may rule that Minor Magical Effects (lighting a single torch, generating a small sound, changing the color of one's eyes, etc.) do not require checks, and do not contribute to a caster's Magical Strain.

Casting Checks

Any time a spell is cast, the caster must make a Casting Check to see if they are able to achieve their intended effect. The caster adds ranks in relevant skills, just like any other check. A Wizard casting a teleportation spell, for example, would add all of their ranks in Wizardry-related attributes, plus any ranks that would apply specifically to teleportation. If the caster has a magical focus, such as a staff, wand or orb, they could potentially add that item's ranks as well, if appropriate. The Attributes, Items and Statuses that contribute to spellcasting ability depend entirely upon your setting and the nature of the magic being wielded.

The Casting Check is made against the spell's DC, which is calculated by summing the Base DC, the caster's current Magical Strain (applied positively), and any situational modifiers the GM deems appropriate for the spell. Situational modifiers for this DC should account for the power and complexity of the intended effect.

Spell Components

Depending on your setting, spellcasting may be able to incorporate rare or valuable components. These components should have ranks assigned to them when they are collected by the party, or conceived of by the Narrator. For example, a Flawless Ruby (4) could be consumed in the casting of a particularly potent fire spell, adding 4 to the caster's Casting Check and to the spell's damage, or improving some more qualitative aspect of its effect. The character should declare their intent for the component's amplification when they cast the spell. In some settings, these components may be required to cast the spell *at all.* Regardless, any such materials used are consumed during the casting process, whether the spell succeeds or fails.

Spells with Duration and Focus

Many spells' effects continue for a time after the spell is cast. Things like buffs, debuffs, damage-over-time effects, or spatial effects (ground fire, conjured chasm, hailstorm, etc.) to name a few. The desired duration of these effects should be taken into account when deciding on a spell's DC.

In addition, maintaining a powerful effect could require **Focus** beyond the initial casting of the spell. A character may choose to spend one or more energy each round to maintain Focus on a spell's effect, allowing it to continue to the next round. A character could focus on up to three magical effects at the same time, but would use up all of their energy to do so. Some spells may not require Focus, but should instead be given a fixed duration, after which their effects end.

Focus should generally be treated as an optional way to enhance a spell's effect, and the caster should decide whether to Focus on the spell at the time it's cast, improving its potency.

Focus can be interrupted when the caster takes damage or is otherwise distracted (e.g. blinding flashes, loud explosions, being shoved or buffeted). The Narrator could rule that this happens automatically, or require the caster to make a check to maintain Focus. An easy rule of thumb is that the caster makes a Focus check, adding any attributes which would contribute to their mental fortitude, and the DC is equal to the damage they took. If damage did not prompt the Focus check, the Narrator should choose a DC based on the intensity of the distracting effect.

Damage, Statuses, and Resisting Spell Effects

Some magics will be cast at an unwilling target, and thus will require that target to actively or passively resist the spell's effects. This is done in exactly the same way as physical combat, though which Attributes apply to resisting a spell's effects, actively *or* passively, will often be quite different to those which apply in physical combat. The narrative should guide you in determining which attributes to apply.

For spells which deal damage, the Casting Check doubles as the Landing Check (See Combat), and the damage for spells is calculated in the same way as it is calculated for mundane attacks. Choose a weapon tier that fits the magical effect, add that tier's Base DC to the Casting Check DC, and use the tier's Base Damage as a starting point for the spell's damage. Spells can be used to execute Special Shots, so long as the Special Shot is narratively possible given the spell effect.

Some spells will place statuses on their targets, positive or negative. These statuses will have a number of ranks which can be applied to future checks made by those targets. The nature of the status, number of ranks, number of targets affected, and the duration of the status should be accounted for in the spell's DC. Narrators should take care to establish a consistent application of the power level of such spells within their world.

If the target is actively defending, they might react by casting a counterspell, leaping forward to interrupt the caster's incantation, dodging the spell's effect, or simply readying their mental defenses. In cases like these, the Casting Check becomes a Check with Opposition.

Learning and Creating Spells

The process of learning and preparing spells varies enormously between settings and magic systems. That said, below is a simple example of how you might handle this process:

For the sake of simplicity, Spellcasting should be treated as a Skill all its own, which represents a character's dedication to their magical craft. A character's ranks in Spellcasting should be added to checks made to cast their spells, maintain Focus on a spell, identify spells cast by others, or any other casting-related check.

Each time a character gains one or more ranks in their Spellcasting Skill, they may choose one Magical Effect to learn per rank they gained. These effects could be Fiery Blasts, Short-Distance Teleportation, Conjure Small Objects, Heal Wounds, Control Wind, Command Animals, Domination, Charming, or any other fairly narrow category of magical effect you can imagine. Once they've learned these effects, they can attempt to use them thereafter to complete any task for which the effect could reasonably be used. For example, Control Wind could be used to speed up a sailing vessel, blow an enemy off a cliff, or, with sufficient skill, could even allow the caster to fly.

While a spell's DC is determined by its intended effects, that *does not* mean that every category of magical effect will have possible applications at very low DCs. Take, for example, the magical effect Raise Dead. Depending on the campaign setting, it's likely that Raise Dead is an *extremely difficult* magical effect to invoke, regardless of how "minimal" you make it (e.g. raising a single small animal may still be significantly complex). In these cases, it's usually good to talk with your Narrator and establish a "floor" DC, under which no spells for that magical effect are possible. This floor DC should be set on a case-by-case basis and is not necessary to establish for every magical effect, especially those common in your setting.

Helpful Questions for Custom Casting

We have found that there are a few key questions that need answered when implementing a casting system to ensure maximum consistency while still allowing for the flexibility that Fools and Corpses attempts to preserve :

- Can you cast without training? Are spells something you must learn the proper steps to, or can your raw emotion and passion translate into magical effects?
- How hard is casting in the setting? Is it something only a select few know how to do as the result of rigorous training and study, or is it fairly common in the world?
- What is used or consumed during casting? Are there materials that are necessary for each spell, or do spells use physical or mental energy?
- How much can the caster affect the outcome of the spell? Can increased focus or energy cause the spell to be more potent? Does increased talent allow the caster to change a spell's effect?
- When a cast fails, does it have adverse side-effects, or does it just "fizzle"?
- How does a caster regain or renew the resource they use to cast?
- Do all spells cost something, or are some of them "free"?

This might seem like a lot to think about up front, but if these questions are left unanswered they will have to be answered during play. That's perfectly fine if that's how you want to run your game, but it might result in setting a precedent that makes play harder (or less fun) down the road.