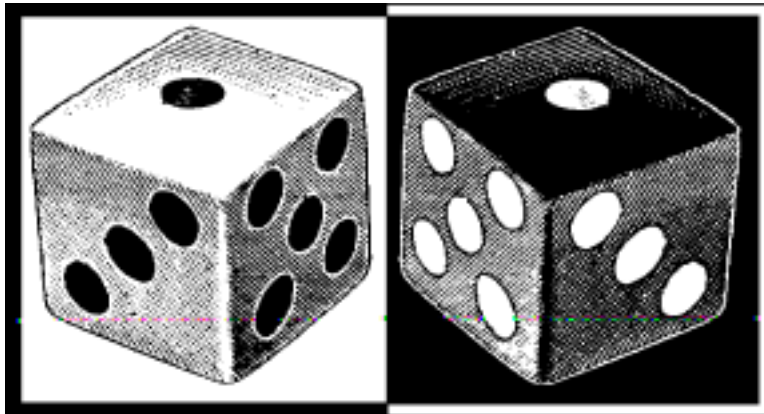# Osprey

Open Source Paper Roleplaying Engine Y*

Chris Goodwin (Project Lead)   Ron Hale-Evans (Writer)
Marty Hale-Evans (Editor)
and the rest of the Osprey Team

February 4, 2000

DRAFT

---

*Y? Because we like you.

ranty of MERCHANTABILITY or FITNESS FOR A PARTICU-
LAR PURPOSE. See the GNU General Public License for more
details.

You should have received a copy of the GNU General Public Li-
cense along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

For the purposes of the GNU General Public License, the Pro-
gram is a compiled version of Osprey such as a PDF file, and
source code is a human-readable and modifiable version of Os-
prey, such as ASCII text, HTML, or LaTeX source.

1 Introduction

1.1 What is Osprey?

Osprey is the **Open Source Paper Roleplaying Engine Y**. That means
it's a freely-distributable, freely-modifiable construction kit for creating role-
playing games that will not be mediated by computer.

If you have never played a role-playing game before, you may find your
first experience of one to be hauntingly familiar. A role-playing game, or
RPG, is essentially a grown-up version of "Cowboys and Indians" or "Cops
and Robbers". Players of the game take on roles, or *characters*, and take
part in a *campaign*, or series of connected adventures in a self-consistent
world imagined by a *gamemaster*, or *GM*. GMs resolve disputes, describe
the scenery of the world to the players, handle behind-the-scenes action of
which the players may know nothing, and portray all the incidental *non-
player characters*, or *NPCs*.

One difference between RPGs and "Cops and Robbers" is that the game
action usually takes place in your head, not your backyard. Another dif-
ference is that there are often strict rules for determining exactly what
happened in situations where the outcome is likely to be contested, such as
combat. These rules eliminate tiresome "Bang! I got you!" "Did not!" "Did
too!" arguments. The determination often takes the form of rolling dice
and comparing the results to a table in the rulebook, but *diceless RPGs* are
becoming popular, and Osprey can be played without dice too.[1]

Now that you know what a role-playing game is, let's examine Osprey in
particular, by considering the terms in the acronym "Osprey".

Roleplaying Engine: Osprey is not a single role-playing game, but a mech-

---

[1]Thanks to Guy W. McLimore and his essay "What is a Role Playing Game?"
at <http://plainlabel.galstar.com/WhatRPG.html> for the "Cops and Robbers"
metaphor.

anism for creating and playing many different ones. It contains a set of rules that can be used to create games in various genres. Going back to the "Cops and Robbers" analogy, you might say that "Cowboys and Indians" and "Cops and Robbers" are really variations on a single game called "Let's Pretend"; one is an RPG in the Western genre, the other in the crime genre. There are other possible variations on "Let's Pretend": comic books, fairy tales, science fictional space adventures, medical melodramas, and so on. As a child, the author's wife played a game called "Gods and Goddesses" based on Greek mythology. All of these are variations on "Let's Pretend", which is not itself a role-playing game, but instead a *role-playing engine*—like Osprey.

Paper: Osprey is usually played largely on paper, face-to-face around a table in a gaming club or someone's living room, rather than on a computer screen or over a network. Maps of your environment in the game (dungeon, forest, city, country, planet, or galaxy) are likely to be drawn on paper. The rulebook you are using (this one) is probably printed out on paper, the description of your character is probably written on a piece of paper called a *character sheet*, and so on.

It is possible to play Osprey over the Internet—for example, you could play over IRC or in a chat room—but much of what you do will still be done with paper while you sit in front of the computer, and your GM will be a human being with pencil and paper in front of another computer, rather than the computer itself. In this way Osprey is different from, and infinitely richer than, computer games billed as RPGs but refereed by computer, such as DOOM or the Ultima series of games.

Open Source: Osprey is *free software*. It is *software* because it can be loaded into your computer; although you may have received your copy of Osprey on paper from your gamemaster, you can also download it off the Internet as a PDF or PostScript file. It is *free* because you are free not only to copy the Osprey game and give it to anyone you like, but also to modify the game in any way and redistribute your modified copies, provided that you give appropriate credit to the original authors and agree that everyone who receives the modified copies of Osprey is free to do the same. The word "free" is used here to imply "freedom", as in the term "free speech", not as in "free beer"—although free software is often free in both senses.

The term *open source* is widely used as a synonym for this meaning of "free", but many people, including the author of this document and others in the Osprey Project, prefer the term "free software" to that of "open source software". One good reason to do so is that calling a program "free software" emphasises the freedom to use, copy, modify, and redistribute it without paying tribute to some corporate overlord. Unfortunately, Osprey has been so named for years, and although there

have been several attempts to rename the project, none has stuck.[2] So while the name of the project uses the term "open source", this rulebook will use the term "free software" everywhere else. You can call *your* version of Osprey whatever you like!

Osprey is distributed under the GNU General Public License, and is, to the best of our knowledge, the first game of its kind to be so distributed. We hope that this will change. Perhaps one reason so few games have been "GPLed" is the misunderstanding that the GPL applies only to executable computer software. But, as mentioned above, anything that can be loaded into a computer is software. The GPL states that source code must be made freely available for software released under the GPL. In Osprey's case, the compiled or executable version of the software is a PDF file that can be read by xpdf or Adobe Acrobat Reader, and the modifiable source code is a LaTeX file that can be rewritten in any text editor, such as GNU Emacs on a GNU/Linux system or BBEdit on a Macintosh. And no one is stopping you—least of all us!—from creating a new version of Osprey in some format you find more convenient, such as plain ASCII or HTML.

2 Character Generation

2.1 Introduction

This section describes how to create your character, your counterpart in the imaginary world created by your gamemaster. Your character may be an outgrowth of your real personality, or may be nothing like you at all; some people find playing characters who are very different from themselves to be extremely rewarding. You will decide what kind of character you want to play, but character creation is not a free-for-all—the choices you will make will be constrained by the rules of Osprey, as described below.

2.1.1 Characteristics

A large part of your character's description will come solely from your imagination. For example, you might decide that your character was born in Hotzeplotz, New Jersey, is 39 years old, and is named Tom. Certain aspects of your character, however, will be governed by the rules of Osprey.

---

[2]One popular suggestion for a new name was GNURPS, the GNU Role-Playing System, emphasising the GNU General Public License and following in the tradition of naming free software after its non-free counterparts; consider bison ≈ yacc, gawk ≈ awk, and Linux ≈ UNIX. Predictably, Steve Jackson Games thought it was too close to the trademark for their game GURPS, and "GNURPS" sank into the waters of Lethe. Such is the fate of many good ideas in a world of intellectual property.

To see why this should be so, consider a game where everyone could play whatever sort of character they wanted. They might all decide to play characters who were a million miles tall, able to fly at billions of times the speed of light, and strong enough to throw suns off their courses. Such a campaign, in which everyone was allowed to live out juvenile power fantasies without any rules about what is possible and what impossible, would quickly become a boring mess as the ultra-powerful squabbled with the super-duper-powerful (and the merely super-powerful) over who could beat whom up.

Then again, even people with more reasonable expectations can be stymied when trying to determine the results of a game action, without definite guidelines. If Joe is "kinda strong" and Bill is "sorta strong", who wins an arm-wrestling contest between Joe and Bill?

To create a situation in which such issues can be quickly resolved, role-playing games often use *characteristics*, or numbers representing facts about game characters. There are several different kinds of characteristics in Osprey, not all of which you may end up using:

Attributes are characteristics shared by everyone in the game, and often mimic real-life characteristics. For example, you are probably aware that not everyone has the same degree of intelligence, but everyone does have some degree of intelligence, ranging from stupid to stellar. Similarly, everyone has some degree of strength, some degree of dexterity, and so forth. These real-life characteristics are often mirrored in Osprey campaigns by the attributes of Intelligence, Strength, Dexterity, and so on.

Skills are characteristics that some people have and some don't, but that almost everyone can gain to some degree with practice, such as Computer Programming and Auto Repair. Osprey characters often have access to additional skills that people don't have in real life, such as Android Programming and Starship Repair.

Advantages are characteristics that some people have and some don't, but that usually happen as an accident of birth or life circumstances, and can't be learned as skills can. Examples include having a rich uncle, being disease-resistant, and having an eidetic memory. There can be disadvantages in Osprey too, such as having only one kidney or being married to someone who beats you.

Powers are characteristics that some people have and some don't, that cannot be learned as skills can (although they can sometimes be improved), and that give the possessor extraordinary, superhuman abilities. These might include becoming invisible, breathing water, or shrinking to subatomic size like Particle Man. Almost all Osprey campaigns have attributes, skills, and advantages, but fewer have powers.

(Not all campaigns need them. Consider an historical game set aboard the *Titanic*; as far as we know, there were no superheroes on the *Titanic*.)

The following pages will walk you through the creation of a character, Norbert Frankfurter, and generating his attributes, skills, advantages, and powers.

If you have played role-playing games before, you will probably notice one unusual thing about Osprey. Whereas most RPGs use static numbers for characteristics, such as 5, 17, or 99, Osprey uses expressions based on dice, such as 4d8+6. (See the section *Notation* below.) This convention represents the tendency of people to have good and bad days, have momentary losses of ability, make dumb mistakes, and so forth. Someone with a Strength of 4d8+6 can have a Strength as low as 10 on a "bad day" and as high as 38 on a "good day".

It is still possible to create a character with static numbers for characteristics using standard Osprey rules, but since characteristics must be "bought", it is not always cost-effective.

2.1.2 Notation

In the examples below, as in any Osprey game, the notation *x*d*y+z* indicates that you should roll *x* dice having *y* sides each and add *z* to the result.

> **Example:** *3d6+2 means you should roll 3 familiar 6-sided dice and add 2 to the result. If you roll three 6-sided dice and get 4, 5, and 1, then you should add 4 + 5 + 1 + 2 for a total of 12.*

2.2 Allotting Character Points

Characters start with a certain number of points per attribute (called *attribute points*), multiplied by the number of attributes they have. Average "couch potato" characters start with 14-15 points per attribute, but your gamemaster will probably start your characters somewhere in the "heroic" range, with about 25 points per attribute. If characters in such a campaign have 6 attributes, then they start with 150 *attribute points* (6×25). If they are part of a simple campaign with only 3 attributes (for example, a gladiatorial combat game), they start with 75 attribute points (3×25).

You will also receive a number of *open pool points* that can be applied to buying any of the four kinds of characteristics: attributes, skills, advantages, and powers (if available). If your game has powers, you will also receive a number of *power points* with which to buy powers. You should

start your character with the same number of open pool points as attribute points. If your character has powers, then start with the same number of power points as well.

> **Example:** *The Osprey character Norbert Frankfurter is a hard-hittin', hard-lovin' hacker of the old school. The campaign in which he is playing is a modern-day setting without superpowers. Since there are 5 character attributes in this campaign—Body, Intelligence, Will, Dexterity, and Social—and the characters are starting in the "medium heroic" range (25 points per attribute), Norbert receives 125 attribute points and 125 open pool points to start (but no power points).*

2.3   Buying Attributes

Next, buy attributes with the attribute points your character received. Attributes are represented by *attribute dice*. These are the dice that you roll when you are trying to accomplish a task with the attribute—for example, juggling bagels employs your Dexterity attribute. (If your campaign had a Juggling skill, however, you would use that instead, since it was more directly relevant.)

The *value* of an attribute is its minimum roll plus its maximum roll. Since the lowest that can be rolled on 3d6 is 3 and the highest is 18, an attribute of 3d6 has a value of 21 attribute points (3 + 18 = 21).

In most cases, the *cost* of an attribute (which is the number of attribute points that must be spent to buy it) is equal to its value. For example, an attribute of 2d6 has both a value and a cost of 14 attribute points. However, the part of an attribute in excess of 30 points costs double. For example, an attribute of 5d6 does not cost the expected 35 points, but 40, because after the 30 point limit, the extra 5 points of value cost twice as much.

> **Example:** *Norbert's player, Fred Bloggs, decides to allot Norbert's 125 attribute points fairly evenly among his 5 attributes. Fred decides to buy Norbert 3d6 attribute dice in every attribute. Skill dice of 3d6 cost 21 attribute points, so that leaves 20 points to buy extra skill dice. (5 attributes × 21 points = 105 points. 125 starting points - 105 points = 20 points.)*
>
> *Fred says a hacker should be "wicked smart", so he gives Norbert an extra 2d6 in Intelligence at a cost of 19 (remembering that 5d6 of Intelligence costs 40, not 35, because of the double-after-30 rule).*
>
> *That still leaves 1 attribute point, so he buys Norbert a +0.5 dice modifier in Social, knowing that it will round up in most*

situations to +1 (numbers always round in the player's favour in Osprey).

This is what Norbert's attributes look like now:

```
Body          3d6
Intelligence  5d6
Will          3d6
Dexterity     3d6
Social        3d6+0.5
```

2.4 Buying Skills

Next, buy skills with *skill points* derived from the values of the relevant attributes, buying points for Dexterity-based skills with points derived from your character's Dexterity attribute, and so on. For example, someone with a Dexterity of 3d6 has 21 skill points to spend on Dexterity-based skills, because the value of 3d6 is 21. Remember to buy skills with an attribute's *value*, not its *cost*. For example, an attribute of 5d6 buys 35 skill points (from its value), not 40 (from its cost). Skill points buy *skill dice*, just as attribute points buy attribute dice.

Many skills have a *default value*, which represents how well someone not trained in that skill can be expected to perform. For example, the Handgun skill has the relatively high default value of 6, since anyone who has seen an American movie can point a gun and pull the trigger. The much trickier Fast Draw skill, however, starts at 1 point.

Add the skill's default value to the number of skill points you are paying for the skill; the result is the value of the skill. (Remember to double the cost of the skill after 30 points.)

> **Example:** *Fred decides he'll buy Norbert's skills starting with Dexterity. Fred decides Norbert belongs to Geeks With Guns, so he's a pretty good shot. Norbert's Dexterity is 3d6 (a value of 21) and the default value of the Handgun skill is 6, so Fred can afford to buy Norbert a Handgun skill of 2d6 (a value of 14) by spending just 8 skill points (8 skill points + a default value of 6 = 14, the value of 2d6).*
>
> *Spending 8 skill points from Dexterity leaves 13 points to spend on other Dexterity skills. Fred decides Norbert spent a lot of time crawling the steam tunnels of his alma mater with his buddies, learning to pick a few locks along the way, so he buys Norbert's Lockpicking skill up to 2d6+0.5 (a value of 15) by spending his remaining 13 Dexterity points (13 points + a default value of 2 for Lockpicking = 15). Remember, 2d6+0.5 rounds up to 2d6+1 in practice.*

Norbert has a Body attribute of 3d6, giving him 21 skill points to spend on Body skills. He spends 11 of his 21 points on the Endurance skill, allowing him to spend long all-nighters at the keyboard fueled by nothing but BoboBars and CrankyCola. He buys an Endurance of 2d6+1 with his 11 points, adding it to the default value of 5 for a total value of 16. And since Norbert is a hard-hittin' hacker, Fred buys him a Brawling skill of 2d6 (14 points) by adding Norbert's remaining 10 Dexterity points to the skill's default value of 4.

Norbert has a Will of 3d6 (worth 21 skill points). Norbert has a high Endurance, and the high Pushing skill Fred buys him allows Norbert to force his body even beyond that. After all, Norbert is not only hard-hittin', but a love machine as well. (Norbert's motto: "I must be running GNU/Linux 'cause I just don't STOP!") Fred takes Norbert's 21 Will points and buys him 3d6+2 of Pushing skill (25 points of value, with a default value of 4 for the skill).

Norbert has a Social skill of 3d6+0.5 (worth 22 skill points). In keeping with the "love machine" theme, Fred buys Norbert a Seduction skill of 2d6 (14 points of value), with 10 of Norbert's 22 Social skill points and a default value of 4 for the skill. Fred's plan is that Norbert will become a hacker superspy, seducing high-ranking executives at his arch-nemesis, the Tinylimp Software Corporation, and learning their dark secrets. So that Norbert can gain information from them without arousing suspicion, Fred gives him a Conversation skill of 2d6 by adding Norbert's remaining 12 Social points to the Conversation skill's default value of 2, for a total of 14 points.

Norbert has an Intelligence of 5d6. Since skill points do not come from an attribute's cost (40 in this case, because of the double-after-30 rule), but from its value (35), Fred has 35 points to spend on Intelligence skills. Fred decides he will buy Norbert the Computer Science skill (of course). He spends 18 Intelligence points on top of the skill's default value of 3 to buy Norbert 3d6 of Computer Science (a value of 21). He also decides to buy Norbert a high Deduction skill, finely honed by years of software debugging. He pays the remaining 17 Intelligence points on top of a default of 2 points for 3d6-1 of Deduction (a value of 19 points).

Fred has now used all of Norbert's skill points from attributes. Norbert is turning out to be a sort of hacker Father Brown, with detection skills based on his hacking abilities, so Fred decides he'll buy Norbert a high Perception skill later when he spends Norbert's open pool points.

Here are Norbert's current stats:

```
Body 3d6
```

```
      Brawling 2d6
      Endurance 2d6+1
Intelligence 5d6
      Computer Science 3d6
      Deduction 3d6-1
Will 3d6
      Pushing 3d6+2
Dexterity 3d6
      Handgun 2d6
      Lockpicking 2d6+0.5
Social 3d6+0.5
      Conversation 2d6
      Seduction 2d6
```

2.5 Allocating Open Pool Points

Next, distribute your open pool points. Although we will spend open pool points on skills only in the example below, you can also spend open pool points on advantages, powers, or raising your attributes; in all cases, open pool points work the same way as skill points, advantage points, power points, or attribute points, respectively.

> **Example:** *It's time for Fred to distribute Norbert's open pool points. Norbert starts with 125 open pool points, the same number as his attribute points.*
>
> *First, Fred decides to increase a few of Norbert's attributes that seem a little low. Norbert's Handgun skill is at 2d6; Fred spends 7 open pool points to buy it up to 3d6. Similarly, Fred spends 7 points to buy Norbert's Conversation skill from 2d6 up to 3d6, and 9 points to buy Norbert's Computer Science skill up to 4d6+1 (a total value of 30 points, the maximum possible for a normal human—raising it any higher would cost double).*
>
> *Fred remembers that he wanted to buy Norbert a Perception skill, the rationale being that Norbert is a superb debugger and can spot tiny clues. He spends 17 points (on top of the default of 4) to buy Norbert a Perception of 3d6 (21 points of value). While buying Norbert detective and spy skills, Fred decides to give him a Persuasion skill of 3d6 (at a cost of 18 points on top of the default 3, for a total of 21)—all the better to bend the Tinylimp Software executives he has seduced to his will.*
>
> *Norbert has gained some useful skills from his hobbies, too. Fred says that like many good geeks, Norbert is an avid wargamer and role-playing gamer, and therefore has developed a high Tactics skill. Fred spends 18 points (plus the skill's default of 3) to buy Norbert a Tactics of 3d6 (21 points). Norbert also happens to*

be a Penn and Teller fan; he has studied their books and gone on to other stage magic texts, so he has a Sleight of Hand skill of 2d6 (Fred spent 12 points on top of the default of 2 to buy Norbert a total of 14 points of this skill). This skill will come in handy when trying to palm the microfilm or slip incriminating "evidence" into a Tinylimp VP's pocket. . .

As you might imagine from his interest in stage magic, Norbert is something of a show-off—he asserts he's the fastest gun in the Pacific Northwest. Well, he's pretty fast, anyway—Fred buys him 3d6 of the Fast Draw skill at a cost of 20 open pool points (the default for the skill is 1 point).

Finally, Fred decides Norbert is a hardware hacker as well as a software one. He buys Norbert 3d6 of Electronics skill (17 points spent + 4 points default = 21 points total).

Fred has now spent all of Norbert's 125 open pool points. Here are Norbert's new stats:

```
Body 3d6
    Brawling 2d6
    Endurance 2d6+1
Intelligence 5d6
    Computer Science 4d6+1
    Deduction 3d6-1
    Electronics 3d6
    Perception 3d6
    Tactics 3d6
Will 3d6
    Pushing 3d6+2
Dexterity 3d6
    Fast Draw 3d6
    Handgun 3d6
    Lockpicking 2d6+0.5
    Sleight of Hand 2d6
Social 3d6+0.5
    Conversation 3d6
    Persuasion 3d6
    Seduction 2d6
```

2.6 Buying Open Pool Points with Disadvantages

Disadvantages are aspects of a character that cause problems. It is not necessary to buy disadvantages, but they tend to make a stronger and more flavourful character. They can provide the Game Master with hooks to customise the game to fit the characters, and they earn the player extra points

to spend. These points are regular open pool points, and can be spent in the same way as any other open pool points.

In order to avoid creating characters that might have flopped or flapped their way out of a circus freak show, it is recommended that the Game Master limit the disadvantages a character can buy to approximately 5 points in quirks and 20 points in other disadvantages (for a total of 25 points). (On the other hand, in a campaign that really was set in a circus freak show, the gamemaster might want to allow players *hundreds* of points of disadvantages...)

There are three types of disadvantages.

Quirks are extremely small disadvantages, primarily meant to provide colour for the character. Examples: always wearing red, hating vegetables, being named Dweezil. Quirks are only worth 1 open pool point each.

Minor disadvantages cause inconvenience to the character but are not significantly harmful and are never deadly. Examples: correctible nearsightedness, missing a finger, slightly bad luck. Minor disadvantages are worth 5 open pool points each.

Major disadvantages are potentially quite harmful and may limit the character's freedom in some way. Examples: being hunted by the King's Guard, unfamiliarity with Earth culture in a campaign based on Earth, blindness, missing a leg, *really* bad luck. Major disadvantages are worth 10 points each.

> **Example:** *Fred decides he will max out Norbert's disadvantages so as to have as many points as possible to spend on advantages.*
>
> *Fred decides Norbert is nearsighted and needs glasses (no matter how suave, Norbert is still a geek; perhaps he wears contacts sometimes). Nearsightedness is a minor disadvantage, worth 5 points.*
>
> *We already know Norbert is a gamer, but Fred says that he's a compulsive MUDder as well. Norbert is on MUDs, MOOs, MUCKs, and all manner of other online RPGs until the wee hours of the morning (and sometimes the big ones). He's often bedraggled the next day and sometimes sleeps through an important call to action. A Compulsion is a minor disadvantage, worth 5 points.*
>
> *Norbert believes that one day, when nanotechnology is everywhere, free software will make Rudy Rucker's "porkchop bushes and fritter trees" possible. Giving Norbert a sense of duty toward the Free Software Movement means that he marches to the beat of a different MIDI metronome, most likely one under the GNU*

General Public License. In the materialistic milieu of Norbert's 20th-century campaign, High Ideals are a minor disadvantage, worth 5 points.

A good brawl is one thing, says Norbert, but opening fire is another. In fact, Fred decides Norbert will never, ever shoot (or strike someone) except in self-defense. Self-Defense Pacifism is a minor disadvantage, worth 5 points. (True Pacifism is a major disadvantage, worth 10 points.)

That makes 20 points of minor disadvantages—the maximum. Fred adds flavour to Norbert with the following quirks, worth 1 point each:

- *He is a devout, if somewhat unorthodox, Bahá'í.*

- *He is vegetarian (but will eat eggs and cheese).*

- *He is an animal lover and brings his pets nearly everywhere (Fred will play this quirk as a real nuisance).*

- *He fancies himself an Edwardian gentleman and carries a cane, although he can walk perfectly well.*

- *As a Gene Wolfe fan, Norbert is addicted to the Wellerism, a figure of speech even more obnoxious than the pun. If a waiter says, "Here are your eggs, sir,", Norbert might be forced to quip, "That's what the surrogate mother said when she met the sperm donor!"*

With 20 points in minor disadvantages and 5 points in quirks, Norbert has 25 open pool points to spend however he likes. He decides to buy 25 points of advantages.

2.7   Buying Advantages

Advantages are traits of your character that are either inborn or gained in some other way than learning. Good luck is an advantage, as is being well-heeled or well-connected. It is not necessary to buy advantages, but they make life easier for your character.

It is possible to buy advantages with the open pool points your character got when your character was first created, but sometimes it is a good idea to buy all advantages with points earned only from selecting disadvantages. This practice can help create a more balanced character, as in the example below.

There are three types of advantages, corresponding to the three types of disadvantages: miniscule advantages (which correspond to quirks), minor advantages, and major advantages.

Miniscule advantages mainly add colour and flavour to one's character, but do not help the character out of many difficulties. For example, looking good in red, being good in bed, loving vegetables. Miniscule advantages cost 1 point each.

Minor advantages are convenient for the character but are rarely helpful enough to save the character's life. Examples: having a flexible body, excellent eyesight, or slightly good luck. Minor advantages cost 5 points each.

Major advantages are potentially life-saving and can give your character freedom that ordinary characters do not possess. Examples: being Chief of Police, being rich or politically well-connected, or having *very* good luck. Major advantages cost 10 points each.

> **Example:** *Fred decides he will buy Norbert a few advantages with the 25 open pool points he gained from disadvantages, then wrap up the character generation process.*
>
> *First, Fred buys Norbert the Animal Magnetism advantage. This means that Norbert gets a bonus on his rolls for skills like Seduction, Persuasion, and so on. His GM rules that a +3 modifier to his rolls is reasonable. Norbert, you're a lady (gentleman?) killer. Animal Magnetism is a minor advantage and therefore costs 5 points.*
>
> *Speaking of animals, we already know Norbert is a vegetarian and has a lot of pets. Fred decides to give Norbert the Good With Animals skill in light of these traits. This means, among other things, that Norbert has an increased chance to quiet a wild or feral animal, or, knowing its ways, avoid running into it in the first place. Good With Animals is a minor advantage and costs 5 points.*
>
> *Norbert has a high Endurance and probably keeps his body in pretty good shape in between his occasional hackerly BoboBar and CrankyCola binges, so Fred buys him the minor advantage of being Healthy at a cost of 5 points. Norbert is now resistant (not immune) to disease, a fact that will almost certainly help him on his adventures.*
>
> *Since Fred has bought only minor advantages so far, he decides to splurge on a really powerful major advantage. And what more generally useful advantage could Fred buy for his protégé than luck? That is, Luck—really good luck, a major advantage at a cost of 10 points. Things are really going to go Norbert's way...*
>
> *That's it for Norbert! Fred is going to take Norbert's character sheet home, mull over Norbert's character a bit more, and get ready to play Osprey next Friday night. (Fred's GM has promised a surprise.)*

2.8 Buying Powers

Oh no! Friday night has rolled around, and no sooner has Fred's game group started playing Osprey than the GM announces that a meteor of highly radioactive and magical Improbabilium has struck the Earth (in the game, that is), and all the players are now superheroes! What a dirty trick. Now Fred has to give Norbert superpowers. . .

Superpowers in Osprey are powers beyond those of ordinary humans. For example, if a character can see through a brick wall when she takes off her spectacles of leaded glass, or can fly faster than sound without benefit of a vehicle, that character has superpowers (*powers* for short.)

At the beginning of the character creation process (or in poor Fred's case, after the GM decides that a superhero-generating miracle occurs), the player is given a certain number of *power points* to distribute among her character's powers. It is also possible to gain power points with *power limitations*, which are similar to character disadvantages but only affect powers (for example, not being able to eat while invisible, lest the food appear half-digested in mid-air and give the character away). It is also possible to spend open pool points on powers.

Players are typically given the same number of power points as they have attribute points, but this varies from campaign to campaign. If you are a player, consult your GM on how many power points you receive. If you are a GM, see the Gamemaster section for hints on deciding how many points players should get.

In a Supers campaign (one with superheroes), one power point typically buys a player one point of power value; this works differently from attribute and skill value points, which cost two points each. However, in a campaign where players are portraying wizards, psionic mutants, aliens, or other beings who have powers but who are not superheroes in the traditional comic-book sense of the word, one point of power value costs two power points, as you might expect. This also varies from campaign to campaign.

Example: Fred's GM tells the players that their characters will each receive the same number of power points as they have attribute points. Since Norbert was created with 125 attribute points, he will receive 125 power points as well.

2.8.1 Power Functions

The part of a power that does something (such as flying or throwing a fireball) is called a *function*. Each power has two functions that are somehow

related to the power itself: a *major function* and a *minor function*. The major function is allotted the same number of points as the value of the power, while the minor function is allotted only half the value of the power.

There are seven basic categories of function that superpowered characters can have:

Attack *TEXT TO BE ADDED.*

Defense *TEXT TO BE ADDED.*

Alteration *TEXT TO BE ADDED.*

Increased Attributes *TEXT TO BE ADDED.*

Movement *TEXT TO BE ADDED.*

Senses *TEXT TO BE ADDED.*

Other *TEXT TO BE ADDED.*

> **Example:** *Fred decides that since Norbert has the Good With Animals advantage, has a lot of pets, is a vegetarian, and so forth, that when the meteor strikes, Norbert should receive a power that is somehow related to animals. He allots 70 power points to Norbert's Animal power. He decides to buy 35 more power points for this power with power limitations to be determined later (see below). He allots 10 of these 35 points to the power itself, for a total of 80; he will spend the remaining 25 points on power stunts for this power.*
>
> *Fred wants to give Norbert sharper senses as the major function of his new Animal power. Norbert is nearsighted, and it would be inconsistent with his character conception to give him the eyesight of a hawk, so Fred gives Norbert a bloodhound's ability to smell, and a dog's hearing as well. Since the Sense function contains two senses, each of them is at half the value of the function, or 40 points, which Fred decides to buy as 4d8+2.*
>
> *The Animal power also has a minor function, of course. Fred decides that Norbert should grow a tough animal hide, proof against knives, fists, bludgeons, and bullets—kinetic energy attacks, in other words. The Tough Hide is a Defense function, and has a value of 40 (half the value of the power itself, since this is a minor function). Fred decides to buy this as 4d8+2 as well.*
>
> *Fred decides that Norbert is fixated on his cane, and his superpowers should have something to do with the thing. Norbert will imbue the cane with a part of his psychic energy, and give it the ability to stun his enemies and sow confusion among them, in keeping with the brawl in the song* Finnegan's Wake *(it turns out that Norbert is a folkie):*

> Then the war did soon engage,
> Woman to woman and man to man.
> Shillelagh law was all the rage,
> And a roar and a ruction soon began!

All good superheroes need a battle cry. Norbert's is "Shillelagh law!".

Fred allots the 55 power points Norbert has left over to Norbert's cane, and decides to buy 25 more power points with power limitations, for a total of 80 points allotted to this power.

The major function of the cane will be its Confusion Attack. The roll for an attack function can be compared to a standard attribute of the target, like Dexterity or Body, or an unusual one, like Will or Intelligence. Since the Confusion Attack of the cane is targeted in a non-standard way (in this case, against the Intelligence of the target), the attack's value is halved, but since it does not directly damage the target, its value is also multiplied by 1.5. Thus, the value of the cane's Confusion Attack is $80 \times 0.5 \times 1.5$, or 60. Fred buys this as 8d6+2.

The minor function of the cane will be its Stun Attack. Fred decides that the Stun will work only when Norbert actually bludgeons the villainous target with the cane, so standard targeting rules apply and Fred need not halve the value of the function. But the Stun attack does not do direct damage, so the value of the function is multiplied by 1.5. The base value of the function is 40 (half the value of the power, since this is a minor function), and $40 \times 1.5 = 60$ points of value. Fred buys this 60-point attack as 8d6+2 as well.

Since all of Norbert's powers come from canines and canes, Fred decides that the day Norbert discovers his superpowers, he proudly takes the name of. . . Caniac!

Yes, Caniac, the Brainiac Maniac, fighting for truth, justice, and Free Software, with his trusty Shillelagh of Justice and a howl at the moon!

2.8.2 Power Stunts

TEXT TO BE ADDED.

2.8.3 Power Limitations

TEXT TO BE ADDED.

2.8.4   Role-Playing Powers

Powers offer a good opportunity for extra role-playing style. Each power has a name by which the character refers to it, rather than that by which the player refers to it. For example, Firedrake the Magician would not say "I'm casting my 5d8+2 Fire Attack at the goblins"; it wouldn't be in character, so his player shouldn't say it either. Instead Firedrake should say (via his player), "I call down the Mystical Flames of Hephaestos to strike down the goblins!"

3   Task Resolution and Combat

3.1   Task Resolution

When your character is trying to perform a task in Osprey that has a chance of failure, such as picking a lock or beating up a bad guy, *task resolution* rules help you and the GM determine the outcome. (Combat is a special type of task resolution, discussed below.) Tasks that have no chance of failure, such as the proverbial falling off a log, are assumed to be *automatic* and do not require task resolution. (There are always exceptions: falling westward off a log during a strong gale from the same direction would require task resolution.)

In an *unopposed action*, such as climbing a wall or programming a computer, you generally roll the dice in the skill or attribute your character has that is most closely matched to the task (for example, using your character's Firearms skill when shooting at something), and compare it to a static difficulty number for the task.

> **Example:** *Caniac is sniffing around Tinylimp Headquarters, trying to discover incriminating papers that he can leak to the attorneys in the ongoing Federal antitrust action. He encounters a locked door labelled "WARNING: MOON-SIZED DEATH RAY PLATFORM PLANS. NO ADMITTANCE." Caniac has reason to believe there's something useful behind this door, so he attempts to pick the lock.*
> *The GM has foreseen that Caniac would try to pick the lock (that's what it's there for), and he has secretly determined it's a 5-point lock. That means Fred must roll 6 or higher in order for Caniac to succeed in picking the lock. Caniac has a Lockpicking skill of 2d6+0.5. He rolls 2d6, coming up with a 4 and a 3: that's a roll of 7.5 when you add his 0.5 point dice modifier. Caniac's roll of 7.5 beats the 5-point lock quite handily.*

Sssnick! The lock opens and the door swings silently inward into a darkened room. Caniac switches on the light. . .

In an *opposed action*, such as combat or a friendly game of tiddlywinks, you must roll the dice in the relevant skill of your character and compare it to your opponent's roll in the relevant opposing skill. Sometimes, as in playing tiddlywinks, the skill or attribute your opponent uses is the same one that you use (in this case, the Tiddlywinks skill, assuming you were playing in a campaign where you could buy one). If your character were throwing snowballs at an opponent, however, you would most likely use your Throw skill and your opponent would use his Dodge skill. Regardless, as with unopposed actions, the higher number wins.

> **Example:** *. . . The lights come on. It's a trap! Standing before Caniac is none other than his arch-enemy Bill Bates (known as HeMaster to his 17,000 lackeys—but not to his face). "Ah, Caniac," he sneers. "Welcome to the Bates Motel. Insects like you check in, but they don't check out!"*
>
> *"Your metaphors are as buggy as your software, Bates!" growls Caniac. "I could smell you behind this, you cad! Little Norbert and I will give you a sound thrashing!" With a cry of "Shillelagh law!", Caniac swings into action.*
>
> *Fred decides Caniac will use his cane's Confusion attack to dazzle Bates into quick surrender. Bates, a long-standing* NPC *(non-player character), has always been overconfident; tonight he neglected to bring an anti-Confusion helmet, so the GM will have to roll Bates's raw Intelligence of 3d6 versus Caniac's Confusion attack of 8d6+2.*
>
> *Fred rolls for Caniac: 1, 1, 2, 1 (uh-oh), 6, 3, 1, 2, plus a dice modifier of 2. That's 19 points of Confusion attack.*
>
> *The GM rolls for Bates: 3, 5, and 1, plus no modifier. 9 points of resistance from Bates. (The GM could not have beaten an attack of 19 with only 3d6, the maximum roll of which is 18, but he rolled anyway in case of a critical failure; see below.)*
>
> *Bates swoons into a corner. You can practically see little birdies tweeting around his head. "When you wish upon a star..." he moans. Caniac now has a crucial ten or fifteen minutes to disarm Bates, rifle the room for the Death Ray plans, and bring Bates to justice.*

3.1.1   Critical Successes and Failures

TEXT TO BE ADDED.

<div align="center">

To be continued. . .

</div>

4 The GNU General Public License

GNU
GENERAL PUBLIC LICENSE

Version 2, June 1991

PREAMBLE

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If

the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

<div align="center">

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS
FOR COPYING, DISTRIBUTION AND MODIFICATION

</div>

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute

such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

    a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the orig-

inal licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

APPENDIX: HOW TO APPLY THESE TERMS TO YOUR NEW PROGRAMS

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> {one line to give the program's name and a brief idea of what it does.} Copyright ©19yy {name of author}

> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

> *signature of Ty Coon*, 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

5 Further Links

Osprey home page:
`<http://www.mindspring.com/~archer7/ospreycurr.html>`

Free PDF version of this document:
`<http://www.ludism.org/rpg/osprey.pdf>`

Free, modifiable LaTeX version of this document:
`<http://www.ludism.org/rpg/osprey.tex>`