OpenD10SRD

Zeshan Rajput

2023-08-08

Table of contents

1 Welcome

OpenD10 is available as an "Open Access" HTML version on this site (insert GitHub pages link here) in addition to the usual print and e-book formats. This edition was initially published in August 2023 and will have errata fixed periodically. If you encounter any errata, please report them here (insert GitHub issues link here).

 Physical copies of this book are not yet available, but when it's near a "First Edition" we will work on that.

2 Introduction

OpenD10 is an open-source, freely available tabletop roleplaying game (TTRPG). OpenD10 is designed to be simple and easy to learn, while still providing a lot of depth and flexibility. It uses a d100 system to determine the outcome of actions, and it features a wide range of attributes and skills that players can use to customize their characters.

Here are some of the features that make OpenD10 unique:

- Open-source: Anyone can contribute to the development of OpenD10 by submitting feedback, testing the game, and creating content.
- Simple and flexible: OpenD10 is easy to learn, but it still provides a lot of depth and flexibility for players to create their own unique characters and stories.
- Supports a wide variety of genres: OpenD10 can be used to play a wide variety of genres, from fantasy to science fiction to horror.
- Free: This system reference document is maintained by the community for all to enjoy for free, as long as you adhere to the terms of the License.

We invite you to join us in the development of OpenD10! You can get involved by:

- Submitting feedback on the SRD
- Testing the game and reporting issues on GitHub
- Creating content for OpenD10, such as new races, classes, and spells, as pull requests

2.1 License

2.2 Inspiration

This work is influenced by the many, many TTRPGs the we have played in the past. This includes, but is not limited to Dungeons and Dragons (3.5 and 5th Editions), GURPS, Shadowrun, West End Games Star Wars d6, OpenD6, Rifts, and more. While we are indebted

to those works and thankful for the many hours of joy they have brought us, we endeavor to make OpenD10 a completely distinct effort that is free to all.

2.3 How to Use this SRD

This SRD is intended to be the template for your own implementation of OpenD10. When you're ready to create your own game based on OpenD10, you will:

- Create a branch of the OpenD10 SRD project on GitHub,
- Re-write the non-core chapters (like this one!) to match fit your game, and
- Add any content needed to flesh out your game.

The following are the core chapters of OpenD10:

- Chapter 03 - Mechanics
- Chapter 08 - Attributes

3 Mechanics

OpenD10 is a tabletop roleplaying game (TTRPG) that uses two 10-sided dice (d10s), or a d100 if you prefer. Players roll the d10s to determine the outcome of a task. The first d10 represents the tens digit of the result, and the second d10 represents the units digit. The player adds the two results together to get a random number between 1 and 100.

In addition to the d10s, OpenD10 also uses a system of attributes and skills. Attributes represent a character's abilities, with 0 representing the lowest possible ability and 100 representing the highest possible natural ability. Skills represent a character's learned abilities, such as Medicine, Athletics, and Persuasion. Skills are on the same 0 - 100 scale.

3.1 Checks

To determine if a character is successful at a task, Game Masters (GMs) assign a difficulty rating from 1 to 300, where 1 is impossible to fail and 300 is impossible for an individual to succeed. When assigning a difficulty rating, the GM considers factors such as the difficulty of the task, the environment, and the circumstances.

Here are some examples of difficulty ratings:

Difficulty Level	Description	More Detail
50	Very Easy	Untrained people will succeed most of the time. No checks below this.
75	Easy	average untrained person can do this half of the time
100	Moderate	average professional can do this half of the time
125	Difficult	the best an untrained person can do without help
150	Very Difficult	the best a professional can do without help

Difficulty Level	Description	More Detail
300	Heroic	the best anyone can possibly do without help

The character's relevant attribute and skill ratings are subtracted from the difficulty rating to yield the target number. The player then rolls a d100. If the result is equal to or above the target number, the character succeeds. Otherwise the character fails at that task. The target number represents the minimum number that the player must roll on the d100 in order to succeed at the task.

If a player rolls a number that is 25 or more points above the difficulty number, they may achieve exceptional success. This means that they succeed at the task with ease, and they may also have some additional benefits, such as finding a hidden compartment or gaining a magical blessing. If a player rolls a number that is 25 or more points below the difficulty number, they may suffer an exceptional failure. This means that they fail at the task spectacularly, and they may also have some negative consequences, such as being injured or cursed.

Finally, if a player rolls two zeros on the d100, this may be considered an automatic success, divine intervention, or anything else that the GM arbitrates. If a player rolls a 01 on the d100, this may be considered an automatic failure, the hand of fate, the Devil's curse, or anything else that the GM deems necessary.

The GM has a lot of flexibility when it comes to arbitrating exceptional successes, failures, and divine intervention. They can decide how much of an impact these events have on the game, and they can also add some flavor to the events to make them more memorable.

OpenD10 is a simple and easy-to-learn TTRPG that is perfect for players of all experience levels. It is also a versatile system that can be used to create a wide variety of fantasy and science fiction worlds.

4 Character Creation

4.1 Basic Overview

1. Choose a class.
2. Sketch-out your character concept.
3. Select a race, background, and other features (optional).
4. Assign starting attribute points to attributes.
5. Assign starting skill points to skills.
6. Purchase gear and/or abilities.
7. Complete character form.

4.2 Choosing a Power Level

Implementations of OpenD10 will vary widely in power level. Some games will be about novice students learning the ways of the world, while others may feature demigods battling over the fate of the cosmos. Implementers set the power level by assigning a number of points at character creation. The following table suggests some limits for an average implementation:

```
# Derive Tiering Table
#| label: tbl-tiering
#| fig-cap: "The packages for characters starting in each tier."
#| echo: false

from sysdef import *

tiers = ["NPC", "Novice", "Veteran", "Expert", "Hero"]

attributes = []
max_attributes = []
skills = []

for i in range(len(tiers)):
  attributes.append(starting_attributes + i * attributes_per_tier)
```

```
  max_attributes.append(starting_max_attribute + i * max_attributes_per_tier)
  skills.append(starting_skills + i * skills_per_tier)

tiering = {
    "Tier": tiers,
    "Starting Attribute Points": attributes,
    "Maximum Possible Attribute Value": max_attributes,
    "Starting Skill Points": skills
}

tier_table = pd.DataFrame(tiering)

tier_table["XP Value of Tier Package"] = tier_table["Starting Attribute Points"] * xp_per_

tier_table["XP per Event"] = round((tier_table["XP Value of Tier Package"] - tier_table["X

display(md(tabulate(tier_table)))
```

0	NPC	200	30	50	2100	nan
1	Novice	225	35	125	2500	80
2	Veteran	250	40	200	2900	40
3	Expert	275	45	275	3300	27
4	Hero	300	50	350	3700	20

This table assumes default values for experience points per attribute point, experience points per skill point, and so on. If you'd like to see or change these weights, see the system definitions python file on GitHub. This table assumes that all experience points are spent on attributes and skills. If you want your players to start with features, consider changing the point value of the tier package or lowering the starting numbers for skills and attributes.

4.3 Classes, Races, and Backgrounds

Most implementations of OpenD10 will have a set of classes that fit the theme of that implementation. These may be mages and knights in a fantasy setting, spies and soldiers in contemporary settings, spaceship captains and temporal engineers in science-fiction settings, and so on. Classes should convey a sense of that character's role on the team and be complementary with other classes to complete a wide variety of challenges.

Classes are implemented as packages of skills and features. For example, a mage might have bonuses to magic skills and features that allow them access to supernatural (i.e. magical)

abilities. Classes should be balanced against each other, so that no one class is significantly stronger or weaker than the others. To balance classes, ensure that the point values for the attributes, skills, and features they grant are equal.

Races are optional additional packages describing a character's heritage that may be offered as part of an implementation. A fantasy or space opera setting may have many races, while a contemporary setting would only have one or a very few. Races should be used to convey natural talents (such as darkvision or sensitivity to gravity) but not stereotypes (such as 'Dwarves usually don't do magic').

Backgrounds are another type of package that may allow a player to select from thematically appropriate origins for their character. Both races and backgrounds are implemented as packages of skills and features. These need not necessarily be balanced - instead deduct a number of points from their character creation pool equal to the cost of the race or background.

4.4 Attributes

All characters in OpenD10 share the same eight attributes regardless of implementation.

4.5 Skills

All characters in OpenD10 become skilled as a result of their experiences. These skills represent areas of particular expertise.

4.5.1 Optional Rule: Proficiency

It is possible to have a proficiency-based skill system in an OpenD10 implementation. To do so:

- Allow players to select a number of skills at character creation, and divide the available skill points among those skills.
- At character progression milestones, equally increment all selected skills.

4.6 Features

Features are the ways in which characters are exceptional. Some features are beneficial and cost character creation points, while others are detrimental and refund points. Some implementations may only allow features to be gained via class, background, and racial choices.

4.7 Gear and Abilities

Gear is the generic name for a character's possessions and represents tools that they have to enable them to solve certain problems more easily (or at all). A character's starting gear is purchased by converting points from the character creation pool into credits. An example of a piece of gear could be a laser gun.

An ability is an inherent piece of 'gear' - some way that a character has to solve certain problems more easily (or at all). A character that could fire a laser beam from their eyes has an ability to do so. Abilities are purchased using experience points.

The cost of a piece of gear is normally derived from factors such as:

- The bonus it confers to a check (the 'quality' of the gear),
- How often that check will occur,
- Whether the gear is consumed when used,
- How available the gear or its constituent parts are, and so on.

The cost of a piece of gear is derived the same way as the cost for an ability, except that gear is discounted for the ability to be stolen, broken, etc. In OpenD10 GMs must be cautious about the amount of loot players obtain, as this can equate to unexpected experience point gain and impact game balance.

4.8 Complete the Character Form

Some of a character's parameters, like height and weight, are derived from their attributes. Others (such a defenses) come up so often that pre-calculating them can greatly accelerate game play.

5 Classes

6 Races

7 Backgrounds

8 Attributes

Attributes reflect a character's natural talent or capabilities, though they may increase slowly over time. The average value of an attribute for a nomal, non-heroic human being is 25.

Physical Attributes:

1. Strength: Measures a character's physical power, ability to deal significant damage in combat, and perform feats of strength.
2. Agility: Determines a character's speed, reflexes, and nimbleness, affecting their ability to dodge attacks and perform acrobatic maneuvers.

Mental Attributes:

1. Intelligence: Represents a character's ability to learn, solve puzzles, and comprehend complex information. It influences their proficiency with magic and strategic thinking.
2. Wisdom: Reflects a character's intuition, perception, and common sense. It affects their ability to make sound judgments and resist mental manipulation.

Social Attributes:

1. Charisma: Indicates a character's charm, persuasiveness, and leadership skills. It helps them influence others and build alliances.
2. Manipulation: Measures a character's cunning, deceit, and ability to manipulate others for their own gain.

Meta Attributes:

1. Stamina: Determines a character's physical endurance and resistance to fatigue. It directly affects their hit points, reflecting how much damage they can withstand in battle.
2. Willpower: Represents a character's mental resilience and strength of mind. It directly affects their mana points, reflecting their magical energy and ability to cast spells.

9 Skills

The following is a list of common skills which implementations of OpenD10 may adapt or override to fit their theme. Example specializations are listed in parentheses.

- Unarmed Combat (Brawling, type of Martial Art)
- Melee Combat (Knives, Swords, Clubs, Staves)
- Ranged Combat (Pistols, Rifles, SMGs, Shotguns, Energy Weapons, Thrown Weapons)
- Dodge
- Stealth (Sleight of Hand, Sneaking)
- Athletics (any sport, lifting, rock climbing, jumping/parkour, marathon running, sprinting, swimming)
- Perception (Visual, Auditory, Olfactory, Insight)
- RTI (resistance to interrogation)
- Languages (specialization in a language grants fluency)
- Crafting/Engineering (Civil, Chemical, Electrical/Computer, Biomedical, Automaton, Demolitions, Smith, Carpenter, Alchemist)
- Medicine (First Aid, any medical subspecialty, Combat medicine/Emergency medicine)
- Drive (cars, trucks, tanks, motorcycles, trains)
- Pilot (freighters, sailing vessels, speed boats, cargo ships, starfighters, space trawlers/transports, cruise ships, capital ships)
- Persuasion (bargain, debate, oration, elocution, storytelling, flirt/seduction)
- Social Engineering (disguise, fast talk, acting, forgery)
- Intimidation
- Etiquette (Corporate, Streetwise, Government)
- Scholar (particular field of study)
- Survival (type of terrain, astrogation, naval navigation)
- Performance (paint, song, dance, poetry)
- Animal Handling (Taming, training, riding)
- Investigation (Interview, Search, Research, Examine)
- Supernatural (see the supernatural section)
- Business
- Skill 1

10 Features

The following list of features is a suggestion which may be expanded upon or overridden in an OpenD10 implementation.

```
# Create Features Table
#| label: tbl-features
#| tbl-cap: Features
#| echo: false

from sysdef import *

features_table = pd.DataFrame({
        'Unskilled': ["(Character Creation only). You've spent more time in life honing yo
                      f"Gain {xp_per_skill} times the number of skill points surrendered."
        'Untalented': ["(Character Creation only). You're not as naturally as gifted as yo
                       f"Gain {xp_per_attribute} times the number of skill points surrende
        'Supernatural': ["You are one of the rare individuals that has supernatural abilit
                         f"Costs {xp_per_supernatural_class * len(supernatural_classes)}."
        'Honorbound': ["You have sworn to follow a code of honor. Failure to do so may res
                       "Gain 3x the value of the code of honor."],
        'Unregistered': ["You have no identity in most common societies. You will need a f
                         "Gain 10."],
        'Ghost / AI': ["You exist on a different plane of existience. You will need some w
                       f"Costs {xp_to_ghost}."],
        'Resilient': ["You are naturally hardier. You gain bonus hit points.",
                      f"Costs {0.5 * xp_per_attribute} per hit point gained."],
        'Exceptional Attribute': ["You are supernaturally talented in an attribute. You ha
                          f"Costs {2 * xp_per_attribute} per attribute point over the maximu
        'Specialized': ["You have specific training or experience within a skill. Confers
                        f"Costs {2.5 * xp_per_skill}."],
        'Wealthy': ["You have spent some of your time working or otherwise trading your sk
                    f"Costs 1 per {credits_per_xp} credits."],
        'Invested': ["You have access to ongoing dividends from some source. You may earn
                     f"Costs 1 per {0.2 * credits_per_xp} credits."],
        'Gifted': ["You are permanently bestowed with a supernatural sense, power, or abil
```

```
                    f"Costs {permanent_gifted_scalar} times the cost of the ability."],
        'Educated': ["You have spent some of your earnings improving yourself.",
                     f"Costs {credits_per_xp} credits per experience point gained."],
        'Darkvision': ['You can see in dim light up to 15 meters away. Objects appear in h
        }
    )

features_table = features_table.sort_index(axis=1).T

display(md(tabulate(features_table,
           headers=["Name", "Description", "Value"])))
```

Name	Description
Darkvision	You can see in dim light up to 15 meters away. Objects appear in hues of red and grey
Educated	You have spent some of your earnings improving yourself.
Exceptional Attribute	You are supernaturally talented in an attribute. You have an attribute score higher th
Ghost / AI	You exist on a different plane of existience. You will need some way to interact with t
Gifted	You are permanently bestowed with a supernatural sense, power, or ability. See the 'S
Honorbound	You have sworn to follow a code of honor. Failure to do so may result in a decrease in
Invested	You have access to ongoing dividends from some source. You may earn bonus credits
Resilient	You are naturally hardier. You gain bonus hit points.
Specialized	You have specific training or experience within a skill. Confers a +10 bonus on checks
Supernatural	You are one of the rare individuals that has supernatural ability. Allows the character
Unregistered	You have no identity in most common societies. You will need a fake identity for many
Unskilled	(Character Creation only). You've spent more time in life honing your natural talents
Untalented	(Character Creation only). You're not as naturally as gifted as your peers. You have f
Wealthy	You have spent some of your time working or otherwise trading your skills for cash.

```
**Design Note**: Because credits are a form of experience points in OpenD10, the Invested fea
```

10.1 Codes of Honor

Values are assigned on a scale of 1-5 where:

- One (1) rarely limits the character's ability to behave however they want, and
- Five (5) makes them incapable of doing anything other living in a controlled environment and assiduously adhering to their code.

Examples:

- Pirate's Code (Value 1)
- Assassin's Code (Value 2)
- Chivalry (Value 3)
- Bushido (Value 3)
- Ascetic's Code (Value 4)
- Isolationism (Value 5)

11 Gear

11.1 Melee Weapons

A character using a melee weapon adds 1/5 of their Strength to damage rolls. Some weapons may allow a player to add 1/5 of their Agility instead.

```
# Build melee weapons table
#| label: tbl-melee
#| tbl-cap: Melee Weapons
#| echo: false

from sysdef import *

melee_data = pd.DataFrame({
    'Dagger': [0,3],
    'Short Sword': [1,0],
    'Long Sword': [1,2],
    'Bastard Sword': [1,4],
    'Great Sword': [2,0]
    }
).rename(index={0: "dice", 1: "pips"})

# Design Note: Ranged weapons are more difficult to use than melee ones. Targets
# are farther: harder to see, small movements of the barrel mean missing by
# meters, etc. From a game balance perspective PCs at range have the advantage
# against enemies that can only use melee abilities. This is balanced in two
# places. When items are defined, the ability to use them at range is balanced
# at each range segment. Some weapons are easier to use up close (75 or 100),
# others almost impossible (a 300). In the combat balance section we'll address
# the XP cost of ranged abilities - this will cover the cinematic balance.
ranged_data = pd.DataFrame({
    'Thrown Dagger': [0,3,75,125,300,300],
    'Pistol': [2,0,100,75,125,250],
    'Shotgun': [4,0,75,100,125,300],
    'Assault Rifle': [3,2,150,100,100,150],
```

```
    'Sniper Rifle': [5,0,250,150,125,100]
    }
).rename(index={0: "dice", 1: "pips", 2: "PB", 3: "S", 4: "M", 5: "L"})

armor_data = pd.DataFrame({
    'Blast Vest': [5],
    'Blast Helmet': [2],
    'Combat Jumpsuit': [4],
    'Powered Assault Armor': [15],
    'Armored Business Suit': [3]
    }
).rename(index={0: "pips"})

melee_table = melee_data.T
melee_table["Credit Cost"] = round(melee_pricer(melee_table["dice"],melee_table["pips"]) *

display(md(tabulate(melee_table,
           headers=["Name","Dice Damage", "Pips Damage", "Credits Cost"])))
```

Name	Dice Damage	Pips Damage	Credits Cost
Dagger	0	3	4909
Short Sword	1	0	9000
Long Sword	1	2	12273
Bastard Sword	1	4	15545
Great Sword	2	0	18000

11.2 Ranged Weapons

Ranged weapons have difficulties associated with the ranges at which they can be used. In the following table:

- 'PB' = Point Blank Range,
- 'S' = Short Range,
- 'M' = Moderate Range, and
- 'L' = Long Range.

More information on range is presented in the Combat section.

```
# Build ranged weapons table
#| label: tbl-ranged
#| tbl-cap: Ranged Weapons
#| echo: false

ranged_table = ranged_data.T

ranged_table["Credit Cost"] = round(ranged_pricer(ranged_table["dice"],
                                                  ranged_table["pips"],
                                                  ranged_table["PB"],
                                                  ranged_table["S"],
                                                  ranged_table["M"],
                                                  ranged_table["L"])* credits_per_xp * (1

display(md(tabulate(ranged_table,
           headers=["Name","Dice Damage", "Pips Damage",
                    "PB", "S", "M", "L", "Credits Cost"])))
```

Name	Dice Damage	Pips Damage	PB	S	M	L	Credits Cost
Thrown Dagger	0	3	75	125	300	300	5556
Pistol	2	0	100	75	125	250	40669
Shotgun	4	0	75	100	125	300	40641
Assault Rifle	3	2	150	100	100	150	89391
Sniper Rifle	5	0	250	150	125	100	121331

11.3 Armor

```
# Build armor table
#| label: tbl-armor
#| tbl-cap: Armor
#| echo: false

armor_table = armor_data.T

armor_table["Credit Cost"] = round(melee_pricer(0,armor_table["pips"]) * (1 - gear_discoun

display(md(tabulate(armor_table,
           headers=["Name", "Armor Value", "Credits Cost"])))
```

Name	Armor Value	Credits Cost
Blast Vest	5	8182
Blast Helmet	2	3273
Combat Jumpsuit	4	6545
Powered Assault Armor	15	24545
Armored Business Suit	3	4909

11.4 Other Gear

```
display(md(f"For items that replicate a feature or ability, multiply the XP cost by {credi
```

For items that replicate a feature or ability, multiply the XP cost by 900.0.

```
display(md(f"For all other gear multiply the real world cost in US Dollars by {credits_per
```

For all other gear multiply the real world cost in US Dollars by 100.0.

12 Abilities

13 Completing the Character Form

13.1 Physical Characteristics

- Body Weight: 3x Strength kg.
- Height: 6-8x Charisma cm.
- Hit Points (HP): Constitution
- Mana Points (MP): Willpower

13.2 Movement

- Walking speed: 3x Agility meters per minute, 3/10 Agility meters per round.
- Sprinting speed: 4x (Agility + Athletics check) meters per minute, 4/10 (Agility + Athletics check) meters per round
- Cost: 5mp / round, requires Focus (see Supernatural section). Modified by negative (body weight + items carried weight - 75).

13.3 Defenses

- Block: Strength + Unarmed Combat -or- Strength + Melee Combat (when armed)
- Parry: Agility + Unarmed Combat -or- Agility + Melee Combat (when armed)
- Avoid: Agility + Dodge
- Armor: Sum of armor ratings of worn equipment
- Mental: Willpower + Supernatural (if known) or Willpower + RTI

14 Non-Combat Interactions

Non-combat interactions provides rules on how characters may interact with their environment by building on the mechanics. Basic interactions with the environment are governed by checks, as described in Checks.

14.1 Creating and Repairing Items

The Engineering skill is used to repair, modify, and create items. The GM assigns a difficulty rating based on how damaged the item is, how difficult parts are to obtain, the availability of tools and a workspace, and so on.

Gear can be modified a finite number of times. Each modification is increasingly more difficult, and heavily modified items are more expensive to obtain.

14.1.1 Engineering / Repair Rolls:

- Critical Success: If making a modification, the number of times the items has been modified does not increase. If repairing an item, the number of times the item has been modified reduces by one.
- Success: The modification or repair is sucessful.
- Failure: The modification or repair is unsuccessful and the item is broken. The item must be repaired before future attempts to modify it are made.
- Critical Failure: The item is destroyed. Destroyed items cannot be repaired or salvaged.

14.1.2 Example Modifications

- Damage: +1 pip
- Range/Targeting: -25 difficulty to one range segment (or add at 250 DL if previously —)
- Fan/Beam Splitter/Cleave: +1 target per attack, multiply cost to use by number of targets.
- Network enabled/Enchanted/Spiritual (once only): can monitor weapon status instantly, can be fired and manipulated from a near plane, can fire into a near plane

- Efficient: Reduce cost to use ability by 10%, stacking additively. The cost could be represented as ammunition, energy cell usage, or mana points.
- Armor Piercing: Reduces victim's mitigation rating by 10% per time this modification is made, stacking additively.
- Augment: Improves any benefit or detriment to a check an ability provides by 5.

```
# Derive Modification Table
#| label: tbl-mod
#| tbl-cap: A table showing the difficulty to successively modify an item.

from sysdef import *

mod_difficulty = []
time_to_mod = []
mod_price = []

for i in range(1 + item_modification_limit):
  mod_difficulty.append(100 + (i * mod_difficulty_per_mod))
  time_to_mod.append(pow(mod_time_scalar,i))

for i in range(1 + 2 * item_modification_limit):
  mod_price.append(pow(mod_cost_scalar,i))

mod_data = {
    "Difficulty to Mod/Repair": mod_difficulty,
    "Time to Mod/Repair (in minutes)": time_to_mod,
}

mod_price_data = {
    "Black Market Purchase Cost (times base price)": mod_price
}

mod_table = pd.DataFrame(mod_data)
mod_price_table = pd.DataFrame(mod_price_data)

display(md(tabulate(mod_table,
  headers=["Times Modified", "Difficulty to Mod/Repair", "Time to Mod/Repair (in mins)"]))
```

Times Modified	Difficulty to Mod/Repair	Time to Mod/Repair (in mins)
0	100	1
1	110	5

Times Modified	Difficulty to Mod/Repair	Time to Mod/Repair (in mins)
2	120	25
3	130	125
4	140	625
5	150	3125

```
# Derive Modification Table
#| label: tbl-modprice
#| tbl-cap: A table showing the price of goods already modified.
#| echo: false

display(md(tabulate(mod_price_table,
  headers=["Times Modified", "Black Market Purchase Cost (times base price)"])))
```

Times Modified	Black Market Purchase Cost (times base price)
0	1
1	5
2	25
3	125
4	625
5	3125
6	15625
7	78125
8	390625
9	1.95312e+06
10	9.76562e+06

14.2 Investigation Scenes

These scenes reflect the characters' efforts to obtain information. This could be getting intel for a heist, researching an enemy's diplomatic strengths and weakness, or anything else. GMs will generally have a list of clues or tidbits that the characters can discover through a variety of means. Initiative (or scene order) is determined by having the players roll a check adding their character's highest mental attribute.

```
# Derive Investigation Table
#| label: tbl-investigation
#| fig-cap: "The cost of various ranks of investigation resources in experience points."
```

```
#| echo: false

investigation_ranks = [
    "Public",
    "Restricted",
    "Elite",
    "Confidential",
    "Top Secret"]

## Investigation Balance
weight_per_rating = 10
# This affects how much a PC having contacts or resources contributes to the
# progress of investigation scenes. Lower values highlight the PCs' skills more.

rating = []

for i in range(len(investigation_ranks)):
  rating.append((i + 1) * weight_per_rating)

investigation = {
    "Description": investigation_ranks,
    "Bonus": rating
}

investigation_table = pd.DataFrame(investigation)

md(tabulate(investigation_table, headers=["Rating", "Name", "Bonus to Checks"]))
```

Rating	Name	Bonus to Checks
0	Public	10
1	Restricted	20
2	Elite	30
3	Confidential	40
4	Top Secret	50

14.2.1 Common Investigation Actions

- Recall: You attempt to recall something pertinent to your situation.
- Research: You attempt to discern information using a library or similar resource available to you. Gain a bonus equal to the ten times the rating of your investigation resource.

- Call in a Favor: You call a contact to see if they know something relevant. Gain a bonus equal to ten times the rating of your investigation resource.

14.2.2 Investigation Resources

- Public (Rating 1): A resource that is freely available to anyone. The public library, city hall.
- Restricted (Rating 2): A resource that is available to anyone who has a membership. University libraries.
- Elite (Rating 3): A resource that few know about, but anyone who tries hard enough can eventually get access to. Hacker havens, magical lodges.
- Confidential (Rating 4): Resources that are not normally available. Military archives, secret research laboratories.
- Top Secret (Rating 5): Highly confidential resources available to a very few. Classified government archives, corporate secret dossiers.

14.3 Social Scenes

Social scenes are when the characters must interact with one or a group of non-player characters (NPCs) to progress. Social scenes do not involve combat, though they can (and often do) shift rapidly into combat scenes. Initiative (or scene order) is determined by having the players roll a check adding their character's highest social attribute.

14.3.1 Common Social Actions

- Garner Insight: You attempt to read body language or interpret some other tell to gain an advantage.
- Speak: You attempt to perform, persuade, decieve, intimidate, explain, or in some other way verbally interact with NPCs in the scene.
- Exposit Fact: Using your experience or previously gained knowledge, you attempt to explain a fact relevant to the scene.

15 Combat

Initiative / Scene Order: Agility check. One round is six seconds.

On your turn, in any order:

- One movement
- One action
- Speak up to 3 seconds

On others' turns: One reaction (once until your next turn)

15.0.1 Common Combat Actions:

- Attack: Make an unarmed, melee, or ranged attack. Difficulty rating per weapon stats or defender's defense rating (whichever is highest).
- Dodge: Spend your turn avoiding damage. Double your Avoid defense rating.
- Grapple: Grapple an enemy combatant within reach. Strength or Agilty + Unarmed Combat check vs defense. If successful, see Grapple sub-section.
- Hold Action: You hold an action to occur on someone else's turn. On your turn you must clearly describe what action you are holding and what will trigger your action to occur. When the trigger occurs, you use your reaction to execute your held action. If your turn comes around before your held action is triggered, you lose your held action.
- Use Item: You use an item either on your person or in the environment you can access.

15.0.2 Grapple

Defender's move is zero while grappled. Attacker's move is halved. Attacker must move defender to an adjacent square during any movement or release the grapple. Both attacker and defender have a -50 penalty to attacks against targets other than each other while the defender is conscious and resisting. You may not grapple something more than two scales larger or smaller than you.

15.0.3 Ranged Attacks

Character Scale Ranges:

0-2m (point blank), 3-10m (short), 11-24m (medium), 25+ meters (long).

15.0.4 Scale

Combat advantages derived from differences in size are represented using scale. Examples below are based on default size weighting.

Mechanics:

- Smaller scale combatants have a bonus to hit and a penalty to damage.
- Larger scale combatants have a penalty to hit and a bonus to damage.
- If an attacker is using a weapon of a different scale than theirs, use the weapon's scale instead.
- The bonus or penalty to hit is equal to 10 to the power of the difference in the combatant ratings.
- PCs suffer this same penalty when using a weapon not their scale (unless there is a control module attached, such as a gun on a capital ship).
- The bonus or penalty to damage is equal to a number of D10s equal to the difference in combat ratings.

15.0.4.1 Scale Ratings:

0: Tiny. Pixie, infant, rodent. Gear almost impossible to find.

1: Small. Child, dwarf, gnome, cat, dog. Gear harder to find.

2: Medium. Human, Elf, Orc.

3: Large. Troll, Ogre, Cyclops, Motorcycle. Gear harder to find.

4: Huge. Elephant, roc, young dragons, cars, wagons, dingies. Gear almost impossible to find.

5: Gargantuan. Whale, adult dragons, semi truck. Highest possible character scale.

6: Star Fighter. Star Fighters, Tanks.

7: Small Ships. Gunboats.

8 - 9: Medium Ships. Cargo Freighters, Star Freighters.

12: Large Ships. Cargo Container Ships, Cruise Ships, Capital Ships.

16: Planet.

22: Star.

```
# Derive Scale Table
#| label: tbl-scale
#| fig-cap: "A table showing the average size of an organism at each scale."
#| echo: false

from sysdef import *

size = []
size.append(smallest_scale)

for i in range(25):
  size.append(size[i] * scale_scalar)

scale_data = {
    "Size (m)": size,
}

scale_table = pd.DataFrame(scale_data)

display(md(tabulate(scale_table, headers=["Scale", "Size(m)"])))
```

Scale	Size(m)
0	0.25
1	0.5
2	1
3	2
4	4
5	8
6	16
7	32
8	64
9	128
10	256
11	512
12	1024
13	2048
14	4096
15	8192

Scale	Size(m)
16	16384
17	32768
18	65536
19	131072
20	262144
21	524288
22	1.04858e+06
23	2.09715e+06
24	4.1943e+06
25	8.38861e+06

15.0.5 Restoration

Mana Points are restored with food and rest. 1 MP is restored with consuming 100 calories of food and resting 5 minutes. For healthy organisms, after eating a full meal and getting a normal night of rest all MP are restored on awakening.

Hit Points restored naturally return at a rate of 1 HP per day of rest.

16 Supernatural

16.1 Realms

This SRD assumes a unified view of other planes of existence with the Material Plane being the point of origin. Implementations of OpenD10 may override this view, in which case adjust the remainder of this chapter to suit.

In the unified view of planes, the Material Plane is adjacent to 'near planes'. Near planes resemble but are not the same as the Material Plane. For example, a church on the Material Plane might exist as a beacon of radiant light on the spiritual plane, or as a digital haven on the Net. Near planes connect to 'far planes'. Far planes do not resemble the material plane, but expectations like the laws of physics will generally hold. Far planes are much more dangerous than near ones. Beyond the far planes are the 'Far Realms'. These furthest reaches are completely alien to the material plane. No assumption can be made for any expectation based in reality, and the sanity of unprepared travelers is at risk.

Examples:

Axis	Near Plane	Far Plane	Far Realm
Material	A Planet	Space	Subspace
Mental	Subconscious	Psyche	Dreamscape
Technological	The Net	Resonance	Deep Resonance
Magical	Astral Plane	Astral Sea	Deep Realm / Abyss
Spiritual	Spirit World	Fey Realm	Heaven, Hell, Valhalla, etc.

16.2 Abilities

Abilities are supernatural manifestations fueled by energy from one axis. An AI draws from the technology realm when they scry, while an angel draws from the spiritual realms when they smite their foes. Similarly material beings with supernatural abilities draw strength from one axis. When the 'Supernatural' feature is selected, the player must identify which axis their character draws strength from. When the character learns the 'Supernatural' skill, they must also immediately specialize in their axis.

16.2.1 Impact of a character's Supernatural Axis

Each possible supernatural axis has some strengths and weaknesses.

- Mental: Psychic abilities have no somatic, verbal, or material requirements. They are the most costly to learn but are difficult to interrupt. Mental abilities do not diminish in the planes of other axes. Core Attribute: Manipulation.
- Technological: Technological abilities may be paid for with credits instead of experience points. When chosen, the player may choose wetware or hardware. Hardware abilities, such as those from cybernetic implants, are discounted but cause a permanent reduction in mana pool. Wetware does not cause a reduction in mana pool but are more expensive. Core Attribute: Wisdom.
- Magical: Magical abilities require at least one somatic, verbal, and/or material component. As such, they are the easiest to interrupt or counter but are the most accessible supernatural axis. Core Attribute: Intelligence.
- Spiritual: Spiritual abilities derive from a patron deity or faction from the spiritual realms. Those patron deities may limit access to abilities, classes, or revoke abilities for behavior inconsistent with their goals. Core Attribute: Charisma.
- Material: Abilities powered by the material realm draw from the energy of the cosmos. Fusion reactors, chemical batteries, and gravity fall into this category. These abilities are accessed by purchasing gear which can be lost, stolen, or broken. The rules for purchasing gear are covered in Character Creation.

16.2.2 Optional Rule: Single Class / Limited Classes

A character may only have access to a few types of abilities from their axis, perhaps even as few as one. The 'Supernatural' feat is discounted by the ratio of selected classes over available classes. Optionally, certain supernatural axes may only have access to a subset of classes.

16.2.3 Optional Rule: Technowizardry

The technology to artificially imbue a person with supernatural ability exists under this optional rule. For example, a character may be able to access the Net by implanting a neural interface and connecting to a computer. The Supernatural feat is discounted and part of it may be paid for with credits as per the Technological Axis section.

16.2.4 Optional Rule: Restoring Humanity

Normally mana pool reductions caused by purchasing hardware abilities can never be restored. Furthermore once a character's mana pool is reduced it can never be increased again, even if a character's Willpower is increased. Under this optional rule, a character's mana pool

increases normally when their Willpower is increased even after purchasing hardware abilities. Optionally, a character can buy additional mana points at the same cost as the 'Resilient' feature.

16.2.5 Focus

Some abilities can be sustained for a duration, others indefinitely. Abilities may require Focus to sustain. A character can only focus on one ability at a time. While holding focus on an ability, characters may use other abilities or take actions that do not require focus. A character may drop focus at any time, at which point the ability immediately ends.

16.2.6 Opposing Abilities and Defenses

Some abilities can be resisted. Defense ratings are calculated during the 'Completing the Character Form' section of Character Creation. The user of the ability rolls a d100 check adding their axis' Core Attribute + Supernatural (axis) skill. If the roll exceeds the target's defense rating, the ability is successful.

16.2.7 Creating Abilities

Abilities can have one of four effects on the game universe:

- They can convert energy from one form to another (mana points),
- They can do work (such as moving an object from one place to another),
- They can restore or deplete integrity (hit points), and
- They can confer bonuses or penalties to skill checks.

Some abilities may have multiple simultaneous effects. Other abilities may be better modeled by a series of effects.

Abilities have two costs: an initial cost to acquire the ability and a recurring cost to use the ability. The initial cost is usually paid in experience points as the character learns the ability. Purchasing gear and gaining technology abilities may be paid for partially or wholly in credits. The recurring cost is paid using mana points. Some gear may require credits as the recurring cost, such as when buying ammunition or energy cells.

16.2.8 Improving Abilities

Abilities can be learned multiple times, and each copy of an ability can be modified per the 'Creating and Repairing Items' section. The 'Supernatural' skill is used to modify or repair supernatural abilities. Abilities are always considered 'Network enabled/Enchanted/Spiritual' to the axis they draw power from.

References