

data_types.py

```
"""Core data types, enums, and constants for the JRPG game."""
```

```
from __future__ import annotations
from dataclasses import dataclass, field
from enum import Enum, auto
from typing import Optional, List, Dict, Any
```

```
class WeaponType(Enum):
```

```
    SWORD = "Sword"
    SPEAR = "Spear"
    AXE = "Axe"
    DAGGER = "Dagger"
    BOW = "Bow"
    STAFF = "Staff"
```

```
class Element(Enum):
```

```
    FIRE = "Fire"
    ICE = "Ice"
    LIGHTNING = "Lightning"
    WIND = "Wind"
    EARTH = "Earth"
    LIGHT = "Light"
    DARK = "Dark"
    NONE = "None"
```

```
class JobClass(Enum):
```

```
    WARRIOR = "Warrior"
    KNIGHT = "Knight"
    PALADIN = "Paladin"
    BERSERKER = "Berserker"
    ASSASSIN = "Assassin"
    RANGER = "Ranger"
    HUNTER = "Hunter"
```

```
SPEARMAN = "Spearman"  
DRAGOON = "Dragoon"  
MONK = "Monk"  
CLERIC = "Cleric"  
PRIEST = "Priest"  
FIRE_MAGE = "Fire Mage"  
ICE_MAGE = "Ice Mage"  
STORM_MAGE = "Storm Mage"  
WIND_MAGE = "Wind Mage"  
EARTH_MAGE = "Earth Mage"  
DARK_MAGE = "Dark Mage"  
LIGHT_MAGE = "Light Mage"  
ARCANE_SAGE = "Arcane Sage"
```

```
class SkillType(Enum):  
    PHYSICAL = "Physical"  
    MAGICAL = "Magical"  
    HEAL = "Heal"  
    BUFF = "Buff"  
    DEBUFF = "Debuff"  
    SPECIAL = "Special"
```

```
class SkillTier(Enum):  
    BASIC = "Basic"  
    INTERMEDIATE = "Intermediate"  
    ULTIMATE = "Ultimate"
```

```
class SkillTarget(Enum):  
    SINGLE_ENEMY = "SingleEnemy"  
    ALL_ENEMIES = "AllEnemies"  
    SINGLE_ALLY = "SingleAlly"  
    ALL_ALLIES = "AllAllies"  
    SELF = "Self"  
    RANDOM_ENEMY = "RandomEnemy"
```

```
class StatusEffectType(Enum):
```

```
# Damage over time
POISON = "Poison"
BURN = "Burn"
BLEED = "Bleed"
VENOM = "Venom"
CURSE = "Curse"
# Action restriction
SLEEP = "Sleep"
STUN = "Stun"
FREEZE = "Freeze"
PARALYZE = "Paralyze"
PETRIFY = "Petrify"
# Debuffs
ATTACK_DOWN = "Attack Down"
DEFENSE_DOWN = "Defense Down"
MAGIC_DOWN = "Magic Down"
SPEED_DOWN = "Speed Down"
ACCURACY_DOWN = "Accuracy Down"
# Buff blockers
SILENCE = "Silence"
SKILL_SEAL = "Skill Seal"
ITEM_SEAL = "Item Seal"
HEAL_BLOCK = "Heal Block"
MANA_BURN = "Mana Burn"
# Persistent effects
REGEN = "Regen"
MANA_REGEN = "Mana Regen"
SHIELD = "Shield"
REFLECT = "Reflect"
COUNTER = "Counter"
# Special
CONFUSION = "Confusion"
CHARM = "Charm"
FEAR = "Fear"
BLIND = "Blind"
WEAKNESS_MARK = "Weakness Mark"
# Enhancements
BERSERK = "Berserk"
HASTE = "Haste"
FOCUS = "Focus"
```

```
GUARD_UP = "Guard Up"
MAGIC_BOOST = "Magic Boost"
# Advanced
DOOM = "Doom"
TIME_STOP = "Time Stop"
CURSE_MARK = "Curse Mark"
BLOOD_LINK = "Blood Link"
SOUL_DRAIN = "Soul Drain"
# Defend
DEFENDING = "Defending"
```

```
class AIType(Enum):
    AGGRESSIVE = "Aggressive"
    DEFENSIVE = "Defensive"
    BALANCED = "Balanced"
    SUPPORT = "Support"
    RANDOM = "Random"
    BERSERKER = "Berserker"
    TACTICAL = "Tactical"
```

```
class ItemType(Enum):
    RECOVERY = "Recovery"
    STATUS_CURE = "Status Cure"
    BUFF = "Buff"
    OFFENSIVE = "Offensive"
    ADVANCED = "Advanced"
    REVIVAL = "Revival"
```

```
@dataclass
```

```
class SkillEffect:
    status: Optional[StatusEffectType] = None
    duration: int = 0
    chance: float = 0.0
    stat_modifier: float = 1.0
    heal_percent: float = 0.0
    shield_amount: int = 0
```

```
@dataclass
class Skill:
    id: int
    name: str
    skill_type: SkillType
    power: float
    mp_cost: int
    accuracy: float
    element: Element
    hits: int
    target: SkillTarget
    tier: SkillTier
    description: str
    effect: Optional[SkillEffect] = None
    jobs: List[JobClass] = field(default_factory=list)
```

```
@dataclass
class StatusEffect:
    effect_type: StatusEffectType
    duration: int
    power: float = 1.0
    source_name: str = ""

    def tick(self) -> bool:
        """Returns True if effect is still active after tick."""
        if self.duration > 0:
            self.duration -= 1
        return self.duration > 0 or self.duration == -1 # -1 = permanent until removed
```

```
@dataclass
class Item:
    id: int
    name: str
    item_type: ItemType
    description: str
    effect_value: int
    target: SkillTarget
```

```
element: Optional[Element] = None
status_cure: Optional[List[StatusEffectType]] = None
stat_buff: Optional[Dict[str, float]] = None
buff_duration: int = 0
```

```
@dataclass
```

```
class Stats:
```

```
    hp: int
    mp: int
    patk: int
    matk: int
    pdef: int
    mdef: int
    spd: int
    eva: float
    acc: float
    crit: float
```

```
def copy(self) -> 'Stats':
```

```
    return Stats(
        hp=self.hp, mp=self.mp, patk=self.patk, matk=self.matk,
        pdef=self.pdef, mdef=self.mdef, spd=self.spd,
        eva=self.eva, acc=self.acc, crit=self.crit
    )
```

```
@dataclass
```

```
class GrowthRates:
```

```
    hp: float
    mp: float
    patk: float
    matk: float
    pdef: float
    mdef: float
    spd: float
    eva: float
    acc: float
    crit: float
```

```
RARITY_COLORS = {  
  3: "***",  
  4: "****",  
  5: "*****"  
}
```

Revision #1

Created 2026-03-18 16:22:06 UTC by Samuel Lee

Updated 2026-03-18 16:22:39 UTC by Samuel Lee