

companions.py

```
"""Companion character database: 20x3★, 10x4★, 5x5★."""
from __future__ import annotations
import random
from data_types import *
from character import Character, JOB_BASE_STATS, JOB_GROWTH
from skills_db import SKILL_DB, JOB_SKILL_POOL

def _make_companion(name, job, weapon, element, rarity,
                    stat_mult=1.0, growth_mult=1.0) -> Character:
    base = JOB_BASE_STATS[job].copy()
    g = JOB_GROWTH[job]

    # Apply rarity scaling
    base.hp = int(base.hp * stat_mult)
    base.mp = int(base.mp * stat_mult)
    base.patk = int(base.patk * stat_mult)
    base.matk = int(base.matk * stat_mult)
    base.pdef = int(base.pdef * stat_mult)
    base.mdef = int(base.mdef * stat_mult)
    base.spd = int(base.spd * stat_mult)

    # Scale growth rates
    scaled_g = GrowthRates(
        hp = min(0.99, g.hp * growth_mult),
        mp = min(0.99, g.mp * growth_mult),
        patk = min(0.99, g.patk * growth_mult),
        matk = min(0.99, g.matk * growth_mult),
        pdef = min(0.99, g.pdef * growth_mult),
        mdef = min(0.99, g.mdef * growth_mult),
        spd = min(0.99, g.spd * growth_mult),
        eva = min(0.99, g.eva * growth_mult),
        acc = min(0.99, g.acc * growth_mult),
        crit = min(0.99, g.crit * growth_mult),
    )
```

```
pool = JOB_SKILL_POOL[job]
basics = [sid for sid in pool if SKILL_DB[sid].tier == SkillTier.BASIC][:4]
inters = [sid for sid in pool if SKILL_DB[sid].tier == SkillTier.INTERMEDIATE][:4]
ultims = [sid for sid in pool if SKILL_DB[sid].tier == SkillTier.ULTIMATE][:2]

chosen_basic = random.sample(basics, min(2, len(basics)))
chosen_inter = random.sample(inters, min(2, len(inters)))
chosen_ultim = random.sample(ultims, min(1, len(ultims)))

skill_pool = chosen_basic + chosen_inter + chosen_ultim
unlocked = chosen_basic[:]

return Character(
    name=name, job=job, weapon=weapon, element=element, rarity=rarity,
    base_stats=base, growth=scaled_g,
    skill_pool=skill_pool, unlocked_skills=unlocked
)
```

```
# =====
# COMPANION TEMPLATES (factories – called fresh each game)
# =====
```

```
# stat_mult: 3★=0.90~0.95, 4★=1.10~1.20, 5★=1.35~1.50
# growth_mult: 3★=0.85~0.90, 4★=1.05~1.15, 5★=1.25~1.40
```

```
THREE_STAR_TEMPLATES = [
    # name, job, weapon, element, stat_mult, growth_mult
    ("Rook", JobClass.WARRIOR, WeaponType.SWORD, Element.FIRE, 0.90, 0.88),
    ("Gara", JobClass.KNIGHT, WeaponType.SWORD, Element.EARTH, 0.92, 0.87),
    ("Tifa", JobClass.MONK, WeaponType.AXE, Element.WIND, 0.91, 0.88),
    ("Sera", JobClass.CLERIC, WeaponType.STAFF, Element.LIGHT, 0.93, 0.88),
    ("Rex", JobClass.RANGER, WeaponType.BOW, Element.ICE, 0.90, 0.87),
    ("Bram", JobClass.BERSERKER, WeaponType.AXE, Element.FIRE, 0.89, 0.86),
    ("Nira", JobClass.WIND_MAGE, WeaponType.STAFF, Element.WIND, 0.92, 0.87),
    ("Dag", JobClass.SPEARMAN, WeaponType.SPEAR, Element.LIGHTNING, 0.91, 0.88),
    ("Ella", JobClass.FIRE_MAGE, WeaponType.STAFF, Element.FIRE, 0.90, 0.87),
    ("Wynn", JobClass.HUNTER, WeaponType.BOW, Element.EARTH, 0.93, 0.88),
    ("Bart", JobClass.EARTH_MAGE, WeaponType.STAFF, Element.EARTH, 0.91, 0.86),
```

```

("Yuna", JobClass.PRIEST, WeaponType.STAFF, Element.LIGHT, 0.92, 0.87),
("Kage", JobClass.ASSASSIN, WeaponType.DAGGER, Element.DARK, 0.90, 0.86),
("Coral", JobClass.ICE_MAGE, WeaponType.STAFF, Element.ICE, 0.91, 0.87),
("Miles", JobClass.WARRIOR, WeaponType.AXE, Element.WIND, 0.90, 0.86),
("Dora", JobClass.CLERIC, WeaponType.STAFF, Element.FIRE, 0.92, 0.87),
("Oryn", JobClass.STORM_MAGE, WeaponType.STAFF, Element.LIGHTNING, 0.90, 0.86),
("Fenn", JobClass.PALADIN, WeaponType.SWORD, Element.LIGHT, 0.93, 0.88),
("Zell", JobClass.DRAGOON, WeaponType.SPEAR, Element.WIND, 0.91, 0.87),
("Mira", JobClass.DARK_MAGE, WeaponType.STAFF, Element.DARK, 0.90, 0.86),
]

```

```
FOUR_STAR_TEMPLATES = [
```

```

("Cael", JobClass.KNIGHT, WeaponType.SWORD, Element.LIGHT, 1.12, 1.10),
("Lyra", JobClass.ARCANE_SAGE, WeaponType.STAFF, Element.LIGHTNING, 1.15, 1.12),
("Vance", JobClass.BERSERKER, WeaponType.AXE, Element.DARK, 1.18, 1.14),
("Shira", JobClass.ASSASSIN, WeaponType.DAGGER, Element.ICE, 1.12, 1.10),
("Bael", JobClass.DRAGOON, WeaponType.SPEAR, Element.FIRE, 1.14, 1.11),
("Noel", JobClass.PRIEST, WeaponType.STAFF, Element.LIGHT, 1.15, 1.12),
("Kira", JobClass.STORM_MAGE, WeaponType.STAFF, Element.LIGHTNING, 1.16, 1.13),
("Roan", JobClass.PALADIN, WeaponType.SWORD, Element.LIGHT, 1.14, 1.10),
("Zara", JobClass.HUNTER, WeaponType.BOW, Element.WIND, 1.12, 1.10),
("Drax", JobClass.MONK, WeaponType.AXE, Element.EARTH, 1.18, 1.14),
]

```

```
FIVE_STAR_TEMPLATES = [
```

```

("Seraph", JobClass.LIGHT_MAGE, WeaponType.STAFF, Element.LIGHT, 1.45, 1.38),
("Abyss", JobClass.DARK_MAGE, WeaponType.STAFF, Element.DARK, 1.42, 1.35),
("Titan", JobClass.BERSERKER, WeaponType.AXE, Element.EARTH, 1.50, 1.40),
("Oracle", JobClass.ARCANE_SAGE, WeaponType.STAFF, Element.LIGHTNING, 1.45, 1.40),
("Valkyrie", JobClass.PALADIN, WeaponType.SWORD, Element.LIGHT, 1.48, 1.38),
]

```

```
COMPANION_GACHA_POOL = {
```

```

    3: THREE_STAR_TEMPLATES,
    4: FOUR_STAR_TEMPLATES,
    5: FIVE_STAR_TEMPLATES,
}

```

```
GACHA_RATES = {3: 0.65, 4: 0.30, 5: 0.05}
```

```
def summon_companion(target_level: int = 2) -> Character:
    """Roll a random companion via gacha, leveled to target_level."""
    roll = random.random()
    cumulative = 0.0
    rarity = 3
    for r, rate in GACHA_RATES.items():
        cumulative += rate
        if roll < cumulative:
            rarity = r
            break

    templates = COMPANION_GACHA_POOL[rarity]
    tpl = random.choice(templates)
    name, job, weapon, element, stat_mult, growth_mult = tpl

    char = _make_companion(name, job, weapon, element, rarity,
                           stat_mult, growth_mult)

    # Level the character up silently
    for _ in range(1, target_level):
        char._level_up()
    char.level = target_level
    char.current_hp = char.max_hp
    char.current_mp = char.max_mp

    return char
```

Revision #1

Created 2026-03-18 16:21:31 UTC by Samuel Lee

Updated 2026-03-18 16:21:54 UTC by Samuel Lee