

Sistem za pametno generisanje protivnika u Soulslike igrama

Član Tima

- Dušan Lečić SV80/2021

Motivacija

Igre iz žanra [soulslike](#) zasnivaju se na pažljivo dizajniranom svetu. Ručno osmišljavanje svakog protivnika za svaku zonu oduzima vreme i otežava brz razvoj igre. Cilj je da pojedinac-dizajner, uz pomoć sistema baziranog na znanju, klikom definiše kontekst (npr. „noć u močvari, srednja težina“) i u roku od nekoliko sekundi dobije potpuno definisanog neprijatelja koji:

- odgovara lokaciji i narativu,
- nije kopija prethodno generisanih,
- automatski je balansiran prema jačini igrača.

Time se ubrzava ciklus dizajna i podiže raznovrsnost sadržaja.

Pregled problema

Trenutno ne postoje sistemi za automatsko generisanje protivnika u soulslike igrama bazirani na ekspertskom znanju. Proceduralna generacija koristi nasumičnost (roguelike pristup), ali gubi narativnu i balansnu kontrolu. Nedostatak postojećih rešenja:

- ne kombinuju **znanje o svetu igre** (biomi, lokacija, vreme dana) sa **znanjem o igraču** (lvl, statistike).
- ne postoji **razlog zašto** je baš određen protivnik u toj zoni – što ruši imerziju.

Moj sistem se razlikuje jer koristi **rule-based rezonovanje** (forward/backward chaining, zajedno sa template-ovima) da generiše protivnika koji ima smisla u datom kontekstu, koji je balansiran prema igraču i koji omogućava proširivanje baze znanja tokom korišćenja igre.

Metodologija rada

Ulazi u sistem:

- Tip regiona (močvara, planina, pećina, zamak, pustinja itd.)
- Težina regije (laka, srednje-laka, srednja, srednje-teška, teška)
- Informacije o igraču (nivo, klasa, korišćeno oružje)
- Vreme u igri i vremenski uslovi (dan, noć, kiša, vetar, lava, itd.)

Izlazi iz sistema:

Definisani protivnik sa sledecim osobinama:

- Tip protivnika (vojnik, čudovište, duh, goblin, ptica, vuk, boss, itd.)
- Sposobnosti (melee, ranged, magic, stealth)
- Statistike (život (HP), šteta (DMG), odbrana (DEF), otpornost (resistance))
- Ponašanje (agresivan, defanzivan, čeka u zasedi...)
- Dodatne karakteristike (otrov, led, vatra, status efekti)

Baza znanja:

- Pravila grupisana po regionu
 - Močvara – otrovno čudovište, duhovi
 - Zamak – vitezovi, čarobnjaci
 - Planina – ptice, gorske zveri
 - ...
- Pravila balansiranja (igrač niskog nivoaa → slabiji neprijatelji; igrač visokog nivoaa → jači neprijatelji)
- Šabloni (template) za različite vrste neprijatelja povezane sa tipom regiona.
- Mogućnost **proširivanja baze znanja tokom korišćenja** – dodavanje novih neprijatelja, regiona ili template-ova kroz interakciju sa sistemom

Primer rezonovanja (korak po korak)

Da bi dali primer rezonovanja potrebno je da imamo neki kontekst. Recimo da je kontekst (region= močvara, Težina regiona = srednje-teška, vreme = noć | magla, nivo igrača = 28 | katana + dex build.

1. Nivo 1 (region) - kandidati iz šablona za „močvaru“: *Swamp Witch*, *Swamp Goblin*...
2. Nivo 2 (balans na osnovu nivoaa regije) – zbog *srednje-teška*, prilagođava se HP i DMG za dobar balans igre
3. Nivo 3 (balans na osnovu nivoaa igrača) – zbog nivoaa 28, prilagođava se HP i DMG za dobar balans igre
4. Nivo 4 (balans na osnovu tipa igrača) – zbog *katana + dex klasa*, prilagođava se HP i DMG za dobar balans igre kao I odabrani tip protivnika
5. Nivo 5 (accumulate funkcija) – evaluira kandidate I bira najprikladnijeg

Forward Chaining

Primer ulančavanja forward chaining-a:

- provera regiona => provera težine => nivo igrača => klasa => oružje igrača => vreme

Backward Chaining

Backward chaining se koristi kada želimo **ciljni tip neprijatelja** i tražimo kontekst u kojem se on može pojaviti:

- Cilj: generiši protivnika za regiju pustinja koji je izazovan za dex klasa
- Sistem pretražuje region, težinu i vreme da bi pronašao validne kombinacije
- Pravila i template-ovi se ulančavaju da bi došli do optimalnog scenarija pojavljivanja

Primeri template-a

Template-ovi se koriste za generisanje gotovih protivnika.

Template *Castle Monster*:

- Region: Zamak
- Težina: srednje-teška
- Sposobnosti: melee
- HP: 3580 - 3700
- DMG: 900 – 1200
- DEF: 400
- Otpornost: Vatra
- Ponašanje: defanzivan, čeka u zasedi
- Dodatno: status efekat – poison

Pravila

Primeri pravila:

when difficulty = Easy

then DMG * 0.6, HP * 0.6

when difficulty = EasyMid

then DMG * 1, HP * 0.6

when difficulty = Hard

then DMG * 2.5, HP * 3

when playerLevel <= 10

then HP + HP * 0.1

when playerLevel >= 100 and playerLevel <= 150

then HP * 2.7

when playerClass = Dex

then enemyContains(poison, stealth) * 1.2

when playerClass = Strength

then enemyContains(range, magic) * 1.2

when playerClass = Mage and difficulty = Hard
then enemyResistance(magic)

when (playerWeapon = Greatsword or playerWeapon = katana) and playerLevel < 50
and difficulty = MediumHard
then enemyContains(magic, range)

when weather = Rain and enemyContainsStatus(fire)
then DMG – DMG * 0.8

when weather = Wind and enemyType(rangeRanged)
then DMG – DMG * 0.1

when playerWeaponContains(poison) and area = Swamp and difficulty = Hard
then enemyReceivesPoisonResistance()

when playerHP > 2000 or playerWeapon = Greatsword
then enemyBehaviour(defansive)

when region = Volcano and playerLevel >= 60
then bossFight(FireDragogn), HP * 5, DMG * 4, enemyResistance(fire), specialAttack(fireBall,
flameBreath)

when region = Swamp and playerLevel >= 30 and difficulty = Medium
then spawnBoss(PoisonHydra), HP * 2, DMG * 3.2, enemyResistance(poison)

when weather = Rain and bossType = FireDragon
then bossDMG – bossDMG * 0.4, bossHP – bossHP * 0.1