

OOP Semestrální Práce - AI for RPG Combat

Vojta Harašta, Jan Holan, Mikuláš Hrdlička, Ondra John

Shrnutí

Zadání

Vytvořit AI pro RPG soubojový systém typu Fire Emblem.

Naše řešení

Rozhodli jsme se vytvořit velmi jednoduchý RPG soubojový systém, a pro něj napsat AI stylem, který je popsán dále. Zvolili jsme typ AI, které nemá přístup k informacím o protivnících, a vše si musí odhadovat samo. Zároveň jsme také zvolili AI pro zábavu, kterému nejde o to vyhrát, ale mít osobnost a být zábavné. To jsme provedli skrze ideu jistých “traitů”, což jsou archetypální vzory chování, které kromě osobnosti dodají AIčku i strategii, které se drží. Např. tedy Zmatený Mág je opravdu zmatený, a jeho strategie je náhodně kouzlit náhodná kouzla. Nenáviděč Mágů velmi nesnáší mágy, a utočí pouze na ně. No a třeba Nejlepší Kamarád má svého oblíbeného člena skupiny, a celou dobu se ho snaží chránit. Takovýchto vlastností jsme navrhli a implementovali několik, a vše jsme napojili na systém, který podle akcí protivníka odhaduje co má za vybavení, jaké má vlastnosti, a jak se chová (jestli jako mág, zloděj nebo ochránce skupinky).

Instalace

Pro instalaci stačí následovat tyto jednoduché kroky:

- Za použití nástroje Iceberg v Phrau naklonovat tento repositář:
<https://gitlab.fit.cvut.cz/johnondr/OOP-rpg-ai-project>
- Z repositáře nahrát package OOPRPG-AI

Použití

Po nainstalování stačí otevřít Playground, a napsat a spustit tento příkaz:

```
Battle newGame.
```

Architektura

Naše práce je rozdělena na 4 hlavní části: UnitResolver, Battle, GameUI a Unit.

UnitResolver

UnitResolver je hlavní třída zodpovědná za to, že naše AIčko má dostatečné informace ke všemu co potřebuje vědět. Účelem této třídy je sledovat bitvu, a tiše soudit a sbírat data o všech zúčastněných. Jakmile si o nich posbírá dostatek dat, tak je schopná našeptávat našemu AIčku, a vybírat pro ně cíle. Pokud tedy například naše AI potřebuje zjistit kdo je mágem v hráčově skupince, UnitResolver to během pár kol zjistí, a informaci AI předá.

UnitResolver je psán formou Singletonu, kde kterému má všechno přístup. Co se ve hře děje je řešeno formou Move message, která je posílána Resolveru pokaždé, když někdo provede nějakou akci.

Battle

Soubojový systém, a samotná "hra". Tato třída zajišťuje správný chod bitvy, zajišťuje aby vše probíhalo ve správném pořadí, a také si uchovává informace i týmech a živých členech na té které straně. Konkrétní detaily soubojového systému jsou popsány v další části této dokumentace.

GameUI

Jedná se pouze o kabátek který dodává nějakou grafickou formu Battle třídy, a přidává výrazně příjemnější Interface na ovládání všeho co se ve hře děje. Vzhledem k tomu že ani Battle, ani GameUI nebyli cílem této práce, nebyla jim věnována příliš velká péče.

Unit

Třída která v sobě obsahuje jednu z nejdůležitějších součástí této práce. Spolu s UnitResolverem tvoří splnění zadání naší práce, neboť přesně toto je naše AIčko. Třída Unit reprezentuje postavu ve hře, a to jak Hráčovu, tak AI. Jediné čím se od sebe Hráčova a AI postava liší, je třídou která je přiřazena, za použití Strategy patternu, do delegátu volaného každé kolo bitvy, který zajišťuje způsob jakým se bude jednotka chovat. V případě AI má každá jednotka přiřazený jeden z Traitů. Jak přesně Traity fungují je popsáno dále, v podrobném design dokumentu, v části věnované Umělé Inteligenci.

Podmínky semestrální práce

Návrhové Vzory

- Polymorfismus
 - Polymorfismus se v naší práci vyskytuje například ve třídách Spell a Effect, kde každý Effect má rozdílné chování, ale všechny dědí stejnou třídu Spell, a mají stejné rozhraní.

- GTInspector extension
 - V naší semestrální práci jsme použili GTInspector extension pro třídu Unit, kde je možné zobrazit staty a effecty jednotky v Inspectoru.
 -
- Singleton
 - Singletonem je v naší práci třída UnitResolver. Vzhledem k tomu že nepočítáme s více než jednou bitvou, a bylo potřeba mít k této třídě pohodlný přístup odkudkoliv, zvolili jsme pro něj tento pattern.
- Strategy
 - Strategy pattern je použit v systému Traitů. To, co v podstatě Trait provádí je, že přiřadí jednotce Strategy algoritmus jakým má provádět své kolo. Vše je řešeno skrze obdobu funkčních delegátů.

Podrobný Design Dokument

Soubojový systém

Náležitosti UI

- Tlačítka pro výběr akcí uživatelem.
- Konzoli pro přehledný výpis toho, co se vlastně stalo.
- Přehledy postav obou týmů. Více informací o týmu živého hráče.
 - O cizím týmu je známý jen počet životů jednotlivých postav a iniciativa.
- Přehled iniciativy.

Atributy postav

- HP - životy
- MP - mana na sesílání kouzel
- ATK - Útok, odvozuje se od něj šance trefit i způsobené zranění
- DEF - Obrana
- MPWR - Magic power, odvozuje se od ní účinnost kouzel
- SPD - ~~Strana přímé demokracie~~ Rychlost, určuje pořadí tahu
- AP - Action Points, určují počet akcí, které postava každé kolo provede

Obecné znalosti

- HP
- Iniciativa
- Jaký DMG útok udělil
- Jestli byl útok critical či ne
- Jaký efekt byl přidán/odebrán
- Kolik HP/Many bylo vyléčeno

Iniciativa

Každé kolo se vyrobí nový array hrajících postav. Vezmou se všichni živí hráči, seřadí se dle SPD. Pokud nějaká postava umře dřív, než se dostane na tah, musí být poté z arraye iniciativy vyřazena.

Akce k dispozici

- Attack
 - Manuálně se vybere konkrétní cíl z živých protivníků.
- Guard
 - Přidá na libovolného spojence (i sebe) Guard Effect - přičte na jedno kolo DEF hlídací postavy k DEF hlídané postavy.
- Spell
 - Stojí MP
 - Implementovaný přes efekty

Kouzla

- Meditate - Dobití MP (nesmí se dát opakovat do nekonečna)
- First Aid - Přidání HP
- CureDebuff - Vyléčí debuff z kamaráda. Debuffem je Slow nebo Stun.
- Buff Attack - Zvýší DEF kamaráda o MPRW kouzelníka.
- Buff Deffense - Zvýší ATK kamaráda o MPWR kouzelníka.
- Poison - Otráví protivníka, který každé kolo dostane několik dmg.
- Protect - Ochrání kamaráda. Veškeré zranění které dostane kamarád místo něj dostane kouzlitel, na jedno kolo.
- Slow - Zpomalí protivníka na 1 AP.
- Stun - Protivník vynechává další kolo.
- Heal - Přidání HP, silnější ale dražší.
- Karma - Zakouzlí se na protivníka. Veškeré zranění které dostane kouzlitel místo něj dostane cíl kouzla.

Průběh útoku

- Útočník si hodí tolika k6, kolik má postava ATK
- Každá padnutá 5 a 6 je úspěch.
- Obránce si hodí tolika k6, kolik má DEF.
- Odečte se počet úspěchů DEF od počtu úspěchů ATK.
- Pokud je toto číslo větší než 0, obránce je zraněn za tolik dmg.
- Pokud je toto číslo menší rovno nule, je zražen za jeden dmg.

Umělá Inteligence

<https://gamedevelopment.tutsplus.com/articles/how-to-build-a-jrpg-a-primer-for-game-developers--gamedev-6676>

“Simple RPGs rely more on giving enemies cool-looking and overpowered abilities while leaving them with a fatally flawed strategy. The fun of that kind of game is finding the flaw in the enemy’s strategy and exploiting it to beat them.”

“To adapt for your problem: A monster would look to assign a score to each enemy. Pass the enemy to each of a collection of behaviour classes, which each return an integer score. Add all these scores together, and assign the final score to the enemy. Attack the enemy with the highest score.

This has the scope for different monsters having different behaviours, evaluating targets in different ways, and the personality extension I mentioned in the other answer could be modified at run time to alter target choice based on the monster’s situation: maybe he becomes more aggressive as his health lowers?”

“You could keep a sticky target by remembering the score of the last target, and any new target would have to beat that score by some extra %.”

Znalosti AI

Na začátku ví o oponentovi jen počet HP. Během hry zkusí dopočítat následující:

- SPD - dle pořadí v iniciativě
- AP - dle toho, kolik akcí daná postava provede
- ATK, MPWR a DEF - odhady podle toho, jak moc silný útok se podaří/nepodaří.

Co dělá AI zábavnou?

- Nepřátelé můžou vyhrát.
- Chování nepřátel není úplně náhodné.
- Nepřátelé vykazují nějakou osobnost.

Optimální strategie - první možnost

Uvažuje ve smyslu celé skupiny (Hrají 3 moje postavy před jednou hráčskou a zvládnou ji do té doby nejspíš zabít). Počítá si Aggro jako nebezpečnost protivníků a používá ho pro rozhodování, když není jasné, jaký cíl si má vybrat. Žádná úroveň náhody. Pravděpodobně tvrdě nezábavný oponent. Focus na nepřítele s největším poměrem způsobeného dmg ke zbývajícím životům.

- HP postavy či společníka pod určitou hranicí = Léčení
- HP nepřítele pod určitou hranicí = Doraž ho
- Má určitý nepřítel na sobě Guard Effect? = Neútoč na něj

Zábavná strategie

AI si na začátku postavy dle svých statů rozřadí do “class”. Každá “classa” bude mít o něco rozdílnější chování a vliv bude hrát také náhoda. Návrh tříd:

- Mage
- Tank
- DPS

Na začátku poté dle classy dostane každá postava náhodně přiřazený "trait", který bude definovat její chování.

3 kolekce traitů

Mage

- **Offensive** - Náhodné útočné kouzlo na náhodný cíl. Když nemá dost many, medituje.
- **Healer** - Každé kolo léčí jednotku s nejméně HP. Pokud je to potřeba, umí jako jediný seslat cureDebuff, pokud má někdo ze spřátelených jednotek na sobě nějaký efekt.
- **Golem Maker** - Každé kolo léčí jednotku s nejvíce HP a tak z ní vytváří ultimátního golema. Životy totiž nemají definovanou maximální hodnotu.
- **Apprentice** - Má jen 40% šanci, že se mu podaří seslat kouzlo. Jinak provede normální útok.
- **Buff** - Umí meditovat a používá kouzla na zvýšení atk a def spojenců.
- **Support** - Krom buffů umí používat i karmu, neumí ale meditovat.
- **Cautious** - je opatrný a každé kolo sesílá Protect a nebo Karmu. Když mu dojde mana, tak chrání sám sebe a nebo útočí, pokud je poslední ze svého týmu.

DPS

- **Unpredictable** - Útočí každé kolo na náhodný cíl.
- **Fair** - Každé kolo musí útočit na jiný cíl.
- **Focused** - Vybere si cíl a na ten útočí, dokud není mrtvý.
- **Magic-hater** - Jeho prioritou je vždy ten nepřítel, co poslední seslal kouzlo
- **Tryhard** - Jeho prioritou je healer, pak mage.
- **Bully** - Útočí na nepřítele s nejmenším počtem HP.
- **Clumsy** - Má třetinovou šanci, že praští sám sebe.

Tank

- **Vanguard** - Každé kolo brání náhodného člověka za svého týmu. Když je poslední naživu, útočí.
- **Coward** - každé kolo guarduje sám sebe.
- **Best Bro** - Každé kolo guarduje jednoho kamaráda, kterého si v prvním stepu náhodně vybere. Poté co kamarád umře přejde Focused na toho, kdo mu zabil kamaráda.
- **Guardian** - Každé kolo chrání jiného člena party.
- **Protector of the weak** - Každé kolo chrání člena party s nejméně HP.
- **Defender of Knowledge** - Chrání kamaráda s největší MPWR.
- **Trynotsohard** - Přepíná se každé kolo mezi Vanguardem a Tryhardem.

Velké plus traitů je to, že dávají možnost pro generování jmen typu “Vendelín The Wobbly Warrior”.