



Středoškolská technika 2013

Setkání a prezentace prací středoškolských studentů na ČVUT

POKÉMON

Martin Dvořák

Střední průmyslová škola elektrotechnická
a Vyšší odborná škola, Pardubice, Karla IV. 13

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: **01 Matematika a statistika**

POKÉMON

Autor: **Martin Dvořák**

Škola: **Střední průmyslová škola elektrotechnická
a Vyšší odborná škola, Pardubice, Karla IV. 13**

Konzultanti: **Kvasničková Alena Mgr.
Fišar Petr Ing.
Prof. Mgr. Radomír Halaš, Dr.
RNDr. Josef Rak, Ph.D.**

Pardubice 2013

„Prohlašuji, že jsem SOČ vypracoval samostatně a použil jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

Pardubice dne

.....

podpis

Anotace

Tato práce o hře Pokémon vychází z 4. generace originálních Pokémon her od Nintendo. Byla prozkoumána herní mechanika a informace poskládány dohromady. Práce podrobně analyzuje skryté hodnoty ve hře Pokémon. Byl vytvořen program pro efektivní výpočet IV při křížení Pokémonů. Dále byl analyzován generátor pseudonáhodných čísel v originálních hrách, aby bylo zjištěno, jestli je použit dokonale, nebo jestli jeho dokonalost vytváří nějaké přídavné vztahy pro hru. Optimální rozdělení obranných EV bylo řešeno jak analyticky, tak algoritmicky.

Klíčová slova

Pokémon; herní mechanika; IV; EV; RNG; LCG; optimalizace

Annotation

This work about Pokémon games is based on 4th generation of original Pokémon games made by Nintendo. Game mechanics were examined and put together. The work analyzes the hidden variables in Pokémon games in detail. A program for effective calculation of bred Pokémon's IVs was created. The work also analyzes the pseudo-random number generator in the original games in order to find out if the generator is perfect or if it has any additional impacts on the game possibilities. The optimal distribution of defensive EVs was solved both analytically and algorithmically.

Keywords

Pokémon; game mechanics; IV; EV; RNG; LCG; optimization

Obsah

1 Úvod	5
2 Herní mechanika	7
2.1 Vlastnosti Pokémona	7
2.2 Útoky	9
2.3 Pořadí efektů na konci kola	11
2.4 Počasí	13
2.5 Double battle	13
2.6 Rozdíly mezi verzemi	15
3 IV kalkulačka	16
4 RNG	18
4.1 LCG	18
4.2 Obrácení cyklu	20
4.3 Rozbor	21
5 Optimalizace obranných EV	24
5.1 Vysvětlení optimalizace	24
5.2 Optimalizace pro $p = 1$ v Defense a Special Defense	25
5.3 Optimalizace v obecném případě	27
5.4 Správnost povahy	29
6 Umělá inteligence pro hru Pokémon?	31
7 Závěr	32
8 Použitá literatura	32

1 Úvod

Tato práce pro SOČ je jednou z částí mojí maturitní práce na téma Strategie her.

Hry hrajeme především ze 2 důvodů: pro zábavu a pro rozvíjení různých schopností. Také slouží jako záminka k setkání se s jinými lidmi a ke studiu různých věcí. Mnoho lidí založilo svoji kariéru na vytváření, vymýšlení, organizování nebo hraní různých her. V maturitní práci jsem se nezabýval hrami sportovními, ale těmi ostatními, tj. hrami, kde hráč vybírá z určitých možností. Mohou to být stolní, karetní nebo počítačové hry.

Snaha o dosahování lepších a lepších výsledků nás vede ke zdokonalování postupů při hře i postupů pro přípravě na hru. Ambiciózní hráč potřebuje do svého arzenálu přidávat řadu vědeckých poznatků.

Moje maturitní práce není referátem mapujícím stav současného poznání teorie her, ale souborem informací různého typu, které se týkají výkonnostního hraní. Hraju soutěžně 2 hry: bridge a Pokémon. V práci jsem spojil zájem o tyto hry se svým zájmem o matematiku, programování a hry všeobecně. Cílem mojí práce bylo ukázat různé přístupy k hledání optimální strategie. V práci jsem se pokusil shrnout, co jsem se naučil při hraní těchto her a co jsem zjistil ve svém soukromém výzkumu. Některé části byly v určité podobě již zveřejněny na internetu. Důležitou součástí průběhu vytváření bylo procvičení programátorských schopností a nabytí dalších kompetencí, které by mi mohly pomoci k dalšímu rozvoji v tomto oboru. Vzhledem k šířce rozsahu práce ji nelze chápat jako učebnici algoritmiky ani studijní materiál k perfektnímu ovládnutí uvedených her. Může sloužit ale jako úvod do této problematiky a hráči zmíněných her mohou zjistit, jakým směrem se ubírat. Práce si v žádném případě neklade za cíl, aby její přečtení nahradilo zkušenosti získané hraním. Také jejím cílem není pojmoutí celé šíře postupů, které lze najít ve specializované literatuře.

Ze všech her se nejdéle zabývám videohrou Pokémon. Tato hra má in-game spadající pod žánr RPG. Při něm hráč prochází vymyšlený svět, v němž se odehrává určitý příběh, chytá a trénuje Pokémony a získává různé věci. Samotné souboje probíhají jako současná tahová strategie. Zatímco u stolních a karetních her je potřeba před prvním hraním znát pravidla, v případě hry Pokémon herní konzole řídí průběh hry. Pokud se ale chceme alespoň malinko vážně zabývat jejím hraním, je potřeba začít zjištěním, jak vlastně hra funguje. A proto začneme vysvětlením herní mechaniky. Poté budeme řešit problémy, které mohou být důležité a zajímavé hlavně pro soutěžní hráče.

Celé to začalo 27. 02. 1996 vydáním japonských verzí her Pokémon Red a Pokémon Green pro Game Boy. Momentálně rozdělujeme hry do 5 generací. Vždy v každé celé generaci šli hrát všichni Pokémoni, útoky, schopnosti i předměty z libovolné hry dané generace. Proto nelze jejich herní mechaniky, ani metagame, oddělit. Nová generace vždy s sebou přinesla nové Pokémony a útoky a novou herní mechaniku. Herní mechanika prodělala největší změny s příchodem 3. generace, tj. při vydání her Pokémon Ruby a Pokémon Sapphire v Japonsku 21. 11. 2002. V ní také došlo k jedinému navenování kompatibility, Pokémony z her prvních dvou generací nelze posílat do her novějších. V současné době můžeme v 5. generaci používat Pokémony získané ve hrách 3., 4. a 5. generace.

Tuto práci budu psát podle 4. generace her. Tato generace je momentálně nejnovější ukončenou generací. Kdybych psal o 5. generaci, mohly by být některé informace v brzké době neúplné. Přestože v 5. generaci přibylo nejvíce Pokémonů ze všech generací her, herní mechanika 5. generace neprodělala žádné významné změny od 4. generace, a proto bude většina poznatků platná i pro ni. Čtvrtou generaci her znám výborně a účastnil jsem se v ní nejvíce soutěží.

Účastnit se turnajů ve hře Pokémon je možné buď fyzicky s herními konzolemi, s oficiálními hrami prostřednictvím internetu nebo na neoficiálních simulátorech. Simulátory je třeba neplést si s emulátory. Nejedná se o pirátské kopie oficiálních her, ale o počítačové programy, které vychází z herní mechaniky a možností oficiálních her. Hráč si může v simulátoru naeditovat tým, který by mohl získat v reálné hře, a hrát přes internet s jinými hráči takto naeditovaným týmem. Hráč si tak může rychle vyzkoušet nejrůznější strategie, aniž by ztrácel čas trénováním Pokémonů v RPG in-game prostředí. Simulátory by měly být pomůckou soutěžních hráčů, ale nikdy by neměly nahrazovat hraní skutečných her.

V USA, Japonsku, Velké Británii, Německu, Francii, Španělsku, Itálii a Korei organizuje společnost TPCi (The Pokémon Company International) Video Game Championships (VGC) sérii, která vrcholí mistrovstvím světa, které se již několik let pořádá společně s mistrovstvím světa v PTCG, která má kvalifikace ve více zemích, viz [1]. V roce 2013 se bude mistrovství světa konat v Kanadě, Vancouver. Soutěž se dělí do 3 věkových kategorií rozdělných podle roku narození, jejichž hranice se mění vždy na začátku nové sezóny, tj. v září. Kategorie jsou Juniors (do 10 let), Senior (10 až 15 let) a Masters (od 15 let). Těsně před mistrovstvím světa probíhá Last Chance Qualifier (LCQ), který umožňuje několika hráčům v každé kategorii, kteří nepostoupili z národních kvalifikací, dostat se na mistrovství světa. Samotné mistrovství světa se skládá ze základní části, která se hraje švýcarským systémem, a TOPky, která se hraje vyřazovacím systémem na 2 vítězné zápasy.

2 Herní mechanika

Tento článek pojednává o principech zápasů ve hrách Pokémon Pearl, Pokémon Diamond, Pokémon Platinum, Pokémon Heart Gold a Pokémon Soul Silver. Vzniknul na základě střádání informací z neoficiálních webů a internetových diskuzí a testováním herní mechaniky v reálných hrách. Při experimentování bylo využito zápisu dat, vytvořených v počítači, do originálních herních cartridge.

2.1 Vlastnosti Pokémona

Každý Pokémon přísluší k 1 nebo 2 typům, ovládá až 4 útoky, může držet předmět a má 1 schopnost (většina druhů Pokémonů může mít 2 různé schopnosti, ale jen jednu z nich má ten určitý Pokémon). Schopnost i předmět se aktivují automaticky, útoky volí hráč sám během kola. Jinou možností je výměna Pokémona a pokud se jedná o tzv. in-game battle (při procházení hry, tj. ne při zápase mezi fyzickými hráči), tak hráč může používat i předměty.

Dále je každý Pokémon charakterizován 6 statistikami: HP (Hit points – životy), Attack (útočná síla), Defense (obrana), Special Attack (útočná síla pro speciální útoky), Special Defense (obrana proti speciálním útokům) a Speed (rychlost). Tyto statistiky záleží zejména na druhu Pokémona, ale i různí Pokémoni v rámci jednoho druhu se statistikami liší. Tyto odlišnosti se odvíjí od povahy Pokémona, IV (genů) a způsobu tréninku.

HP: Určuje počet životů. Když klesne na 0, Pokémon je vyřazen.

Attack: (Att) Čím vyšší, tím větší zranění Pokémon udílí fyzickými útoky .

Special Attack: (SpA) Čím vyšší, tím větší zranění Pokémon udílí speciálními útoky .

Defense: (Def) Čím vyšší, tím menší zranění Pokémonovi udělí fyzické útoky.

Special Defense: (SpD) Čím vyšší, tím menší zranění Pokémonovi udělí speciální útoky.

Speed: (Spe) Na rychlosti záleží pořadí útoků v kole. (viz útoky)

Statistika kromě HP se vypočte podle vzorce:
$$X = \left(\frac{L \cdot \left(2 \cdot Z + IV + \frac{EV}{4} \right)}{100} + 5 \right) \cdot p$$

Počet HP se vypočte podle vzorce:
$$H = \frac{L \cdot \left(2 \cdot Z + IV + \frac{EV}{4} \right)}{100} + 10 + L$$

X ... Attack, Special Attack, Defense, Special Defense nebo Speed

H ... počet HP

L ... level (maximálně 100)

Z ... základ pro danou statistiku, určeno druhem (popřípadě formou) Pokémona

IV ... hodnota „genu“ pro danou statistiku, určená při chycení, vylíhnutí nebo obdržení Pokémona

EV ... proměnlivá hodnota pro danou statistiku určená tréninkem Pokémona

p ... hodnota určená povahou Pokémona, nabývá hodnot {0,9 ; 1 ; 1,1}

Statistiky mohou být v průběhu zápasu měněny. Nejsou však měněny přímo, ale u každé statistiky kromě HP se mění jejich stupeň, v rozsahu od -6 do 6. Původní hodnota tohoto stupně je 0, pokud je Pokémona vystřídán a vstupuje znova do zápasu, tento stupeň se vrací na 0. Každému stupni odpovídá určitý koeficient, kterým se násobí daná statistika. Tyto koeficienty určuje následující tabulka:

stupeň	koeficient
-6	0,25
-5	0,29
-4	0,33
-3	0,4
-2	0,5
-1	0,67
0	1
1	1,5
2	2
3	2,5
4	3
5	3,5
6	4

Dále existují další 2 statistiky: **Accuracy** (přesnost) a **Evasion** (výhybnost). Tyto statistiky nemají výchozí hodnotu, pouze stupeň. Čím vyšší je přesnost útočníka vůči výhybnosti soupeře, tím větší má útok šanci na zásah. Koeficienty těchto statistik jsou méně strmě odstupňované:

stupeň	koeficient
-6	0,33
-5	0,38
-4	0,43

-3	0,5
-2	0,6
-1	0,75
0	1
1	1,33
2	1,67
3	2
4	2,33
5	2,67
6	3

2.2 Útoky

Útoky se dělí na fyzické, speciální a na ty, které vůbec neubírají HP. Každý fyzický nebo speciální útok má svoji sílu, typ, podstatu (podle ní se dělí na fyzické a speciální), přesnost, určení cíle útoku a může mít druhotný efekt (ten pak má taky určitou šanci).

Síla útoku se obvykle spočítá následujícím způsobem:

Sílu fyzického útoku vyjadřuje vzorec:

$$K = \frac{\left(\left(\frac{2 \cdot L}{5} + 2 \right) \cdot A \cdot U \right)}{50 \cdot D} + 2 \cdot f \cdot T \cdot R \cdot 100$$

Sílu speciálního útoku vyjadřuje vzorec:

$$K = \frac{\left(\left(\frac{2 \cdot L}{5} + 2 \right) \cdot B \cdot U \right)}{50 \cdot E} + 2 \cdot f \cdot T \cdot R \cdot 100$$

K ... kompletní síla útoku (absolutní)

L ... level (maximálně 100) útočícího Pokémona

U ... základní síla útoku

A ... fyzický útok (Attack) útočícího Pokémona

B ... speciální útok (Special Attack) útočícího Pokémona

D ... fyzická obrana (Defense) bránícího se Pokémona

E ... speciální obrana (Special Defense) bránícího se Pokémona

S ... rychlost (Speed)

f ... součín vlivu slabostí a odolností typu útoku na jednotlivé typy zasaženého Pokémona
za **T** dosadíme číslo 1,5 při útoku typu shodném s typem útočnicka ; při jiném typu

dosadíme číslo 1

R ... náhodné číslo od 85 do 100, které se náhodně určuje při každém útoku, na všechny hodnoty není stejná pravděpodobnost, nejčastěji nabývá hodnot 92~99.

Sílu útoku ještě ovlivňují další koeficienty určené předměty, předchozími útoky a schopnostmi Pokémonů. Pokud dojde ke Critical Hitu, tak je síla útoku dvojnásobná a navíc se nepočítají změny statistik, které by mohly sílu útoku snížit.

Všechny výpočty se v Pokémon hrách počítají se zaokrouhlením dolů, ale pokud vyjde číslo menší než 1, tak se zaokrouhlí nahoru. Proto je vždy důležité vědět, co se počítá. Například útok Super Fang nepodléhá tomuto vzorci, ale rovnou sníží soupeřovo HP (životy) o polovinu předchozího množství. Zranění se při lichém počtu HP zaokrouhluje dolů, ale pokud má soupeř jen 1HP, tento útok ho zničí.

Přesnost útoku se spočítá tímto způsobem:

$$P = \frac{P_1 \cdot AccuracyM}{EvasionM}$$

P ... výsledná přesnost útoku

P₁ ... přesnost uvedená ve vlastnostech útoku

AccuracyM ... koeficient odpovídající stupni přesnosti útočnicka

EvasionM ... koeficient odpovídající stupni výhybnosti bránícího se Pokémona

Přesnost útoku ještě ovlivňují další koeficienty určené předměty, předchozími útoky a schopnostmi Pokémonů. Důsledkem změn přesnosti nebo výhybnosti tedy může minout i útok se základní přesností 100%. Nicméně existují útoky, které neminou nikdy (jako třeba Aerial Ace, Aura Sphere...) nebo za určitého počasí nikdy neminou (Thunder při dešti, Blizzard při krupobití).

Rychlost útoku je vlastnost každého útoku, která má přednost před rychlostí Pokémona. Pokud hráči v jednom kole zvolí útoky se stejnou rychlostí, tak je první použit útok, který používá rychlejší Pokémon. Vzhledem k tomu, že rychlost většiny fyzických i speciálních útoků je 0, tak se rychlost celkem často projeví. Útoky, které nemají rychlost rovnou nule jsou uvedeny v této tabulce:

rychlost	útoky + jejich efekty
5	Helping Hand (v double battlu zvýší sílu partnerova útoku o 50%)
4	Magic Coat (obrací proti soupeři neúčinné útoky), Snatch (používá za soupeře neúčinný útok)

3	Detect, Protect (na jedno kolo chrání uživatele), Endure (na jedno kolo zabraňuje zničení uživatele zraněním z útoku) (pokud použití útoku Protect, Detect nebo Endure předcházelo úspěšné použití libovolného z těchto útoků, tak mají jen poloviční šanci na provedení), Follow Me (v double battle stáhne na uživatele útok jdoucí na partnera)
1	Aqua Jet, Bide, Bullet Punch, Extremespeed, Fake Out, Ice Shard, Mach Punch, Quick Attack, Shadow Sneak, Sucker Punch, Vacuum Wave (rychlé útočné útoky)
-1	Vital Throw (pomalý jistý útok)
-3	Focus Punch (povede se pouze tehdy, nebyl-li Pokémon v průběhu kola zasažen)
-4	Avalanche, Revenge (základní síla 70, mají 2x větší sílu, pokud byl útočník v průběhu kola zasažen)
-5	Counter, Mirror Coat (vrací s dvojnásobnou silou fyzické, resp. speciální útoky)
-6	Roar, Whirlwind (prohodí soupeřova Pokémona)
-7	Trick Room (Tento útok obrací na 5 kol pořadí Pokémonů podle rychlosti, takže rychlejší útoky stále útočí před pomalejšími, ale pokud 2 Pokémoni zvolí útoky stejné rychlosti, tak jako první útočí pomalejší)

Všechny ostatní útoky mají rychlost 0. Mezi ně patří naprostá většina útoků, které udílí zranění.

2.3 Pořadí efektů na konci kola

Efekty na konci kola se vyhodnocují v tomto pořadí (vysvětleno níže):

1.0 konec Reflect

1.1 konec Light Screen

1.2 konec Safeguard

1.3 konec Tailwind

2.0 Wish

3.0 Hail, Rain, Sandstorm nebo Sun

4.0 Dry Skin, Hydration, Ice Body, Rain Dish

5.0 Gravity

6.0 Shed Skin, Speed Boost

- 6.1 Ingrain
- 6.2 Black Sludge, Leftovers
- 6.3 Leech Seed
- 6.4 Burn, Nightmare, Poison, Poison Heal
- 6.5 aktivace Flame Orb nebo Toxic Orb
- 6.6 Curse
- 6.7 Bind, Clamp, Fire Spin, Magma Storm, Sand Tomb, Whirlpool, Wrap
- 6.8 konec Outrage, Petal Dance, Thrash
- 6.9 Magnet Rise
- 6.10 Yawn
- 6.11 Sticky Barb
- 7.0 Doom Desire, Future Sight
- 8.0 Perish Song
- 9.0 Trick Room

Tento seznam byl zveřejněn v [2] a tuto mechaniku jsem otestoval.

Na konci každého kola se nejdříve vyhodnotí efekty 1.x, potom efekty 2.x ... až 9.x. U efektů se stejným prvním číslem rozhoduje rychlost Pokémona. Pokud jsou v aréně 2 Pokémoni se stejnou momentální rychlostí, jeden se na celý konec kola vylosuje jako rychlejší. V rámci celé skupiny efektů se stejným prvním číslem se provedou nejdřív všechny týkající se rychlejšího Pokémona a pak všechny týkající se pomalejšího Pokémona. Pořadí efektů se stejným prvním číslem u každého Pokémona záleží na druhém čísle.

Ve 3. generaci her byli vyřazení Pokémoni nahrazováni okamžitě. Ve 4. generaci již nezáleží na tom, v jaké části kola je Pokémon vyřazen a nasazení nových Pokémonů probíhá až jako poslední událost v každém kole (takže pokud se rychlejší Pokémon zničí svým útokem, útok druhého Pokémona vyjde do prázdna). Pokud jsou v jednom kole zničeni oba Pokémoni, tak oba hráči nejprve vyberou Pokémona, kterého chtějí nasadit, a pak teprve vidí, koho zvolil soupeř. Nově nasazení Pokémoni jsou pak považováni, jako by vstoupili do arény zároveň (takže pokud mají oba Pokémoni schopnost Intimidate, působí na oba Pokémony; pokud ale oba hráči v jednom kole dobrovolně vymění Pokémony, pořadí jejich výměn záleží na aktuální rychlosti).

Pokud dojde k tomu, že jsou v jednom kole vyřazení poslední Pokémoni obou hráčů, pak se konec hry vyhodnocuje následovně: Konec hry se vyhodnotí po provedení každého útoku (může vzniknout i remíza). Pokud jsou oba Pokémoni vyřazení efektem na konci kola a tyto efekty mají stejné první číslo různé od 6, pak hra končí jako remíza. Pokud by

měli být oba Pokémoni vyřazeni efektem na konci kola s různým prvním číslem, nebo oba efekty mají číslo 6, pak se po vyřazení jednoho Pokémona okamžitě ukončí hra a jeho trenér prohrál.

2.4 Počasí

V zápasech mezi hráči se vyskytují následující 4 počasí: Intense sunlight (silný sluneční svit), Heavy rain (prudký déšť), Sandstorm (písečná bouře) a Hailstorm (krupobití). V rámci jednotlivých her se ještě vyskytuje Shadow Sky (stínová obloha) a Fog (mlha). Každé nové počasí vždy ruší předchozí. Počasí se může spouštět buď útoky (pak trvá 5 kol, popřípadě 8 s patřičným předmětem) nebo schopnostmi (pak trvá neomezeně dlouho).

Intense sunlight lze spustit útokem Sunny Day nebo schopnostmi Drought. Zesiluje ohnivé útoky o 50 %, ale zeslabuje vodní útoky o 50 %. Solarbeam se nemusí nabíjet. Snižuje přesnost útoku Thunder ze 70 % na 50 %. Moonlight, Synthesis, a Morning Sun doplňují 2/3 HP (místo 1/2). Účinek útoku Sunny Day lze prodloužit předmětem Heat Rock.

Heavy rain lze spustit útokem Rain Dance nebo schopnostmi Drizzle. Zesiluje vodní útoky o 50 %, ale zeslabuje ohnivé útoky o 50 %. Solarbeam útočí jenom s poloviční silou. Thunder se vždy trefí. Moonlight, Synthesis, a Morning Sun doplňují pouze 1/4 HP. Účinek útoku Rain Dance lze prodloužit předmětem Damp Rock.

Sandstorm lze spustit útokem Sandstorm nebo schopnostmi Sand Stream. Na konci každého kola zraňuje ve výši 1/16 HP Pokémony, kteří nejsou kamenného, ocelového nebo zemního typu. Zvyšuje speciální obranu kamenných Pokémonů o 50 %. Solarbeam útočí jenom s poloviční silou. Moonlight, Synthesis, a Morning Sun doplňují pouze 1/4 HP. Účinek útoku Sandstorm lze prodloužit předmětem Smooth Rock.

Hailstorm lze spustit útokem Hail nebo schopnostmi Snow Warning. Na konci každého kola zraňuje ve výši 1/16 HP Pokémony, kteří nejsou ledového typu. Blizzard se vždy trefí. Solarbeam útočí jenom s poloviční silou. Moonlight, Synthesis, a Morning Sun doplňují pouze 1/4 HP. Účinek útoku Hail lze prodloužit předmětem Icy Rock.

2.5 Double battle

Double battle je druh zápasu, kde oba hráči zároveň používají 2 Pokémony. Poprvé se objevil ve hrách Pokémon Ruby a Sapphire. V těchto hrách jich bylo ale málo, což jim možná ještě přidalo na zábavnosti. Vyskytovaly se i ve hrách LeafGreen a FireRed, ale nejvíce jich bylo v Pokémon Emerald, kde se na double battle spojili libovolní 2 trenéři, které hráč potkal zároveň. V in-game double battle měl každý soupeř maximálně

3 Pokémony a mohl používat pouze svoje (tudíž nemohli být v aréně 2 Pokémoni stejného soupeře), ale hráč ovládal oba svoje Pokémony, které libovolně vybíral.

V soubojích mezi hráči se zápas 2 na 2 objevil ve dvou verzích: Double Battle a Multi Battle. Double battle je formát pro 2 hráče, kde každý ovládá 2 Pokémony. Multi battle je formát pro 4 hráče, kde jsou hráči spojeni po dvojicích a každý ovládá 1 Pokémona. Každý hráč má v Multi Battlu maximálně 3 Pokémony a žádná dvojice těchto Pokémonů se nemůže v aréně objevit zároveň, ani kdyby jeho partner měl všechny Pokémon vyřazené ze souboje. V Multi battle vyhrává vždy celá dvojice hráčů, nehledě na to, jestli jeden z hráčů skončil nebo neskončil bez bojeschopného Pokémona.

Některé útoky se projeví jen při Double battle. Navíc existence Double battle rozdělila všechny útoky podle cíle na tyto skupiny:

1. útoky, které je možno zaměřit (skoro všechny fyzické a speciální útoky), ty je možno zaměřit i na partnera
2. útoky, které se zaměřují samy (buď náhodně (Outrage) nebo podle předchozích útoků (Counter)), ty se na partnera nezaměří
3. útoky, které působí na oba soupeře (Heat Wave), síla těchto útoků byla ve 3. generaci her 50 % síly, ve 4. generaci her mají 75 % síly, kterou by měly v single battle (ale pozor, pokud jeden hráč má pouze 1 aktivního Pokémona, tento útok má 100% sílu a to i v případě, kdy byl jeden Pokémon zničen tímto útokem a na druhého tento útok útočí plnou silou)
4. útoky, které působí na oba soupeře i partnera (Earthquake), síla těchto útoků byla ve 3. generaci stejná jako v Single battle , ve 4. generaci mají tyto útoky jen 75 % sílu
5. útoky, které působí na uživatele (Calm Mind)
6. útoky, které působí na uživatelovu stranu (Tailwind)
7. útoky, které působí na soupeřovu stranu (Spikes)
8. útoky, které působí na celou arénu (Sunny Day)

Pokud jsou v Double battle vyřazeni 2 Pokémoni jednoho hráče a 1 Pokémon druhého hráče během jednoho kola, pak si oba hráči zároveň nasadí 1 Pokémona a poté, co první hráč uvidí, koho nasadil soupeř, si první hráč nasadí 2. Pokémona.

Pokud jsou v Double battle vyřazeni všichni 4 Pokémoni, pak oba hráči nasadí 1 Pokémona, podívají se, jakého Pokémona nasadil soupeř, a oba zároveň nasadí 2. Pokémona.

První nasazený Pokémon vždy vstupuje na levou stranu. Některé efekty přecházejí na všechny Pokémony jednoho hráče, kteří jsou na určité straně, například příjem HP od Leech Seed.

2.6 Rozdíly mezi verzemi

Pokud hrají proti sobě 2 hráči a každý má jinou verzi, je zápas vyhodnocován ve hře toho hráče, kdo při propojení zvolil možnost „Become leader“. Vzhledem k tomu, že ve 4. generaci nelze zápasit se hrami jiných generací, nemělo by to činit velké potíže. Nicméně v průběhu vzniku jednotlivých her 4. generace došlo k pár odlišnostem.

V Pokémon Diamond a Pokémon Pearl měl útok Hypnosis přesnost 70. V Pokémon Platinum, Pokémon Heart Gold a Pokémon Soul Silver má již přesnost pouze 60 %.

Pokud byl v Pokémon Diamond nebo Pokémon Pearl vybrán útok, který měl díky počasí jistotu zásahu (Thunder při dešti, Blizzard při krupobití), ale normálně bylo (před vyhodnocením této vlastnosti) vyhodnoceno, že tento útok minul, bylo jeho netrefení se vždy přepsáno na trefení se, a to i v případech, kdy k tomu nemělo dojít, například když bránci se Pokémon použil Protect. V Pokémon Platinum, Pokémon Heart Gold a Pokémon Soul Silver k tomuto jevu již nedochází.

Útok U-Turn je 100% fyzický hmyzí útok o základní síle 70, po jehož použití se uživatel vymění za jiného Pokémona (zbývá-li ještě nějaký). Pokud Pokémon v Pokémon Diamond nebo Pokémon Pearl zvolil útok U-Turn a vyměnil se za Pokémona, který také uměl tento útok a měl předmět, který ho nutil opakovat stále stejný útok (Choice Band, Choice Specs nebo Choice Scarf), tak nově nasazený Pokémon byl donucen také použít U-Turn. V Pokémon Platinum, Pokémon Heart Gold a Pokémon Soul Silver si nově nastupivší Pokémon může vždy znovu zvolit, jaký útok použije.

V Japonské verzi Pokémon Diamond a Pokémon Pearl docházelo k tomu, že když Pokémon držel předmět, který ho nutil opakovat stále stejný útok, a úspěšně provedl Pursuit na vyřazení stahujícího se Pokémona, tak mohl v dalším kole vybrat jakýkoliv útok. V anglické verzi Pokémon Diamond a Pokémon Pearl a ve všech verzích Pokémon Platinum, Pokémon Heart Gold a Pokémon Soul Silver musí útočník znovu opakovat Pursuit.

3 IV kalkulačka

Vytvořil jsem program IV_stat.exe , ve kterém může hráč zjistit IV (hodnoty genů pro jednotlivé statistiky) Pokémona na základě levelu, základů, povahy a aktuálních hodnot statistik. Dále v tomto programu může hráč vidět, jaká bude hodnota statistik na vyšším levelu. Zde můžete vidět okno programu ke konci výpočtu.

	Base Stat		Real Stat		IV Estimation / IN	IV min / none	IV max / none
HP	70	-	61	+	30	29	31
Attack	110	-	56	+	25	23	27
Defense	80	-	39	+	21	19	24
Special Attack	55	-	23	+	3	1	5
Special Defense	80	-	39	+	21	19	24
Speed	105	-	45	+	3	1	5

Level: - 19 +

Nature: Adamant

Buttons: Calculate Stats from IVs, Save Intervals, Calculate IVs from Stats, Reset Possibilities, Load, Save

Summary: HP: 31~31, Att: 24~24, Def: 19~21, SpA: 1~1, SpD: 19~21, Spe: 2~2

Program jsem navrhnul tak, aby co nejvíce urychlil zdlouhavé výpočty IV při křížení Pokémonů. Hráč typicky vylíhne velké množství vajíček. Některé z nich okamžitě vyloučí podle povahy nebo schopnosti. Hráči zbyde stále dost Pokémonů a chce mezi nimi najít toho nejlepšího. S ohledem na čas chce hráč co nejrychleji vyloučit Pokémony se špatnými geny a k tomu výborně slouží právě tahle kalkulačka.

Na rozdíl od konkurenčních programů můj „Stat <-> IV calculator“ neumožňuje výpočet IV po získání EV, ale umožňuje rychle vypočítat IV netrénovaného Pokémona, na kterého je opakovaně používáný předmět Rare Candy ke zvýšení levelu bez získání EV. Po této proceduře hráč hru vypne bez uložení, takže o Rare Candies nepřijde, level jeho Pokémona se taky nezvýší, ale z hodnot statistik průběžně zapisovaných do tohoto programu získá informaci o jeho IV, které se po restartu nezmění, a hráč může použít Rare Candies na odhalení genů dalšího Pokémona.

Vzhledem k tomu, že ve vzorci pro výpočet statistik dochází k zaokrouhlení dolů po dělení, odpovídá většině zadání celý interval možných IV, který je tím širší, čím nižší je level. Vzhledem k zaokrouhlení po započítání povahy dokonce ve statistikách posílených povahou nemůže nikdy vyjít číslo dávající zbytek 10 po dělení 11, přestože hodnotě o 1 vyšší i nižší odpovídají neprázdné intervaly možných IV.

Když uživatel zadá level, základy, povahu a IV (do sloupce označeného „IV Estimation / IN“), po stisku tlačítka „Calculate Stats from IVs“ se zobrazí hodnoty statistik (ve sloupci „Real Stat“). Když místo neznámých IV zadá hodnoty statistik (zobrazené ve hře vždy po získání nového levelu nebo po otevření „SUMMARY“) a stiskne tlačítka „Calculate IVs from Stats“, program zobrazí střední, minimální a maximální hodnotu IV.

Po zadání první šestice statistik pro první hodnotu levelu a stisknutí tlačítka „Save Intervals“ si program uloží možné hodnoty a provede odhad IV jako průměr mezi příslušného intervalu. Po druhém a každém dalším zadání se odhad počítá z průniku všech platných intervalů, není to tedy již střední hodnota IV zobrazená ve sloupci „IV Estimation / IN“ podle posledního zadaného levelu.

Po přepsání levelu nebo stisknutí tlačítka „+“ program rovnou nabídne předpokládané hodnoty statistik pro nový level podle odhadu IV. Uživatel pak nemusí nic zdlouhavě vypisovat a pokud nabídnutá hodnota neodpovídá hodnotě zobrazené ve hře, upraví hodnotu tlačítkem „+“ , nebo „-“. V každém případě stačí pro každou statistiku stisknout jedno z tlačítek maximálně jednou a takováto úprava vždy vede ke zúžení intervalu možných hodnot. Po každém stisknutí tlačítka „Save Intervals“ dojde k průniku intervalů a aktualizaci intervalů dole.

Protože základy statistik jsou dané druhem Pokémona (a případně formou), není je nutné při každém spuštění programu znova zadávat, ale je možné načíst je z XML po zadání jména Pokémona, pod kterým byla daná šestice uložena.

Pro Pokémony s odlišnými statistikami pro rozdílné formy je vhodné za jméno napsat pomlčku a zkratku formy. Forma Pokémona může záviset na verzi hry, akci ve hře, prostředí posledního zápasu, drženém předmětu, příležitosti získání, ročním obdobím, použití předmětu, slučovaném Pokémonovi, použitém útoku nebo na počasí.

4 RNG

Počítačový program nedokáže vytvořit opravdovou náhodu. Může ale obsahovat funkce, které vytvářejí hodnoty, jež se zdají být náhodné. Generátor pseudonáhodných čísel je klíčovou komponentou originální hry Pokémon od Nintendo. Ale je ten generátor opravdu dokonalý nebo z jeho nedokonalosti plynou nějaké přídavné vztahy pro hru?

4.1 LCG

Hry Pokémon obsahují tzv. lineární kongruentní generátor pseudonáhodných čísel (LCG). Ten má daný určitý koeficient a , inkrement c , modulo m a počáteční hodnotu X_0 . LCG vytváří řadu čísel v rozmezí od 0 do $m-1$ podle předpisu:

$$X_{n+1} = (aX_n + c) \bmod m$$

Tento generátor má určitou periodu, protože možných hodnot je konečně mnoho a po určité hodnotě vždy nastává stejná následující hodnota. Jeho perioda může být nanejvýš n a tuto maximální periodu (nazývanou full cycle) má právě tehdy když:

- 1) c a m jsou nesoudělná
- 2) $a-1$ je dělitelné všemi prvočiniteli m
- 3) $a-1$ je dělitelné 4, pokud je m dělitelné 4

Třetí pravidlo je velice důležité, protože s ohledem na výkon programu na reálných procesorech se obvykle volí m v hodnotě mocniny 2.

To s sebou ovšem nese nepříjemný vedlejší efekt, že pokud bychom sledovali posledních b bitů čísla, budou mít i při full cycle periodu jen 2^b , viz [4].

Zdůvodnění:

$$x \equiv y \pmod{m} \wedge d|m \Rightarrow x \equiv y \pmod{d}$$

Vznikne vlastně nový generátor daný předpisem $X_{n+1} = (a_d X_n + c_d) \bmod d$, kde $a_d = a \bmod d$ a $c_d = c \bmod d$. $d = 2^b$; c_d je nesoudělné s d , protože c nemohlo být násobkem 2; $a_d - 1$ je dělitelné 4, protože $a - 1$ bylo dělitelné 4.

Nový generátor opět splňuje podmínky pro full cycle, ale střídá jen $d = 2^b$ hodnot.

Například se neustále střídají sudá a lichá čísla (a to vždy, když je m sudé). Proto se obvykle berou z každého volání generátoru jen počáteční bity, které se zdají být „náhodnější“. Pak už neplatí to, že by se po určité hodnotě vždy objevovala stejná

následující hodnota, protože každá hodnota prvních několika bitů se vyskytuje na $\frac{m}{d}$

různých místech původní periody.

LCG je i přes svoji jednoduchost poměrně kvalitní, ale není vhodný pro všechny aplikace. Například není vhodný pro metody Monte Carlo, kvůli korelacím. Dále není vhodný pro náhodná data do šifrovacích programů. V kryptografii je totiž požadován takový generátor pseudonáhodných čísel, který generuje libovolně dlouho pseudonáhodná čísla taková, že pokud známe prvních k bitů čísel, nedokážeme určit $(k+1)$ -ní bit s větší než 50% úspěšností žádným polynomiálním algoritmem.

LCG ve hrách Pokémon 4. generace má následující parametry. Čísla v uvozená znakem \$ jsou zapsaná v hexadecimální soustavě (s použitím cifer 0 až F). Volím toto vyjádření, protože pak každé dvojčíslí odpovídá jednomu bytu. Tyto informace ze hry byly zjištěny disassemblerem. Já jsem provedl pouze analýzu vznikajících hodnot.

$$a = \$41C64E6D \text{ (1 103 515 245)}$$

$$c = \$6073 \text{ (24 691)}$$

$$m = 2^{32} \text{ (\$100000000)}$$

$X_0 = \$?$ (počáteční hodnota se inicializuje podle času spuštění hry a dalších parametrů)

výstup (V_n): prvních 16 bitů X_n

$$X_{n+1} = (X_n \cdot \$41C64E6D + \$6073) \bmod 2^{32}$$

$$V_n = X_n \div 2^{16}, \text{ kde } \div \text{ označuje celočíselné dělení.}$$

Při správném použití je tento generátor perfektní. K zodpovězení naší otázky ovšem musíme nejprve rozebrat, jakým způsobem se jeho hodnoty zpracovávají. Nebudeme se zabývat jeho prací uvnitř samotného souboje, kde je volán například při použití nepřesných útoků, protože tam se zdá být jeho funkce dokonalá (a to je taky vysoce nutné). Zaměříme se na generování počátečních vlastností nově potkaného Pokémona. Budeme si klást otázku, jestli princip generátoru umožňuje získat všechny kombinace počátečních vlastností Pokémona, a pokud to neumožňuje, tak jaké vztahy pro ně platí. Protože je celá záležitost velmi složitá, využijeme počítačové simulace a budeme analyzovat její výsledky. V Delphi jsem vytvořil řadu programů, které by měly odhalit konkrétní vlastnosti. Začneme ale teorií.

Vlastnosti Pokémona, které mají být náhodné, jsou dány 4 hodnotami vygenerovanými pomocí LCG, viz [3]. První 2 hodnoty nám dávají PID, ze třetí a čtvrté hodnoty se počítají IV. Z PID se určuje pohlaví a schopnost, ty závisí na druhu Pokémona a PID. Dále se z PID určuje povaha, ta není závislá na druhu Pokémona a zajímá nás nejvíce, protože společně s IV vytváří statistiky Pokémona.

První výstup LCG tvoří druhou polovinu PID. Druhý výstup LCG tvoří první polovinu

PID. Pokud jsou čísla zapsaná hexadecimálně (nebo binárně), pro získání výsledné hodnoty je stačí zapsat za sebe ve správném pořadí. Obecně PID získáme tak, že k prvnímu výstupu LCG přičteme 2^{16} násobek druhého výstupu. PID mod 25 nám udává povahu, která má 25 možností (zvyšuje libovolnou ze statistik kromě HP ; snižuje libovolnou statistiku kromě HP a statistiky, kterou zvyšuje ; to dává 20 možností ; + 5 neutrálních povah).

Hodnoty IV se z výstupů LCG počítají následujícím způsobem. Každý výstup má 16 bitů. První bit je zbytečný a po něm následují 3 pětičky bitů, které nám dávají IV v rozmezí od 0 do 31 (\$1F).

x xxxxx xxxxx xxxxx	x xxxxx xxxxx xxxxx
- DefIV AttIV HP IV	- SpDIV SpAIV SpeIV

Ve hrách 3. generace byly při generování Pokémonů (pouze divokých) použity jiné metody, které volaly LCG 5x a jednu z nekrajních hodnot nevyužívaly. Protože Pokémony získané ve hrách 3. generace lze poslat do her 4. generace prostřednictvím Pal Parku, je potřeba si uvědomit, že nejsou vygenerovány na základě za sebou jdoucích výstupů LCG. Ale protože je možné divoké Pokémony křížit, množinu řešení nám to nerozšiřuje. Zásadní je způsob generování legendárních Pokémonů (jež není možné křížit), který je ve hrách 3. i 4. generace stejný a odpovídá výše uvedenému předpisu.

4.2 Obrácení cyklu

Když známe parametry generátoru a hodnotu X_n , může nás zajímat hodnota X_{n-1} . Mohli bychom samozřejmě zkoušet všechny hodnoty generátoru a počítat pro ně následující hodnotu podle předpisu $X_{n+1} = (aX_n + c) \bmod m$, až bychom narazili na hodnotu, po níž následuje X_n , ale to je příliš zdlouhavé. Je možné vytvořit předpis pro určení předcházející hodnoty vyjádřený ve stejném tvaru jako původní generátor. Podmínkou pro takový předpis je full cycle, který je zároveň i podmínkou nutnou k tomu, aby ke každé hodnotě bylo možné přiřadit hodnotu předcházející.

$$X_{n-1} = (a_r X_n + c_r) \bmod m$$

Kde a_r je inverzní prvek k a v modulu m , pro nějž platí $a \cdot a_r \equiv 1 \pmod{m}$. Ten můžeme spočítat rozšířeným Euklidovým algoritmem nebo s využitím Eulerovy funkce podle vzorce $a_r = a^{\phi(m)-1} \bmod m$ (full cycle zajišťuje, že a je nesoudělné s m); pro m v hodnotě mocniny 2 lze vzorec zjednodušit na $a_r = a^{\frac{m}{2}-1} \bmod m$. Aby výpočet byl efektivní, tak umocňujeme podle rekurentního předpisu $a^0 = 1$, $a^{2n} = (a^n)^2$, $a^{2n+1} = (a^n)^2 \cdot a$ a modulo počítáme po každém umocnění.

Pro inkrement platí $c_r = a_r \cdot (-c) \bmod m$.

Pro LCG použitý ve hrách Pokémon nám vychází:

$$a_r = \$EEB9EB65$$

$$c_r = \$A3561A1$$

$$X_{n-1} = (X_n \cdot \$EEB9EB65 + \$A3561A1) \bmod 2^{32}$$

4.3 Rozbor

Nejprve jsem vytvořil program, který pro každou hodnotu zobrazí hodnoty předchozí a následující. Program dokáže zobrazit hodnoty v desítkové a šestnáctkové soustavě.



Dále jsem vytvořil program, který ověřil, že hodnoty X_n tvoří periodu o délce 2^{32} . Poté jsem vzal náhodné dvojice hodnot V_n a zjišťoval, kde se v této periodě vyskytují za sebou. Nejčastěji se daná uspořádaná dvojice vyskytovala v periodě jednou, ale některé tam nebyly vůbec a některé vícekrát.

Vytvořil jsem program, který počítal Pearsonův korelační koeficient pro výstupní hodnoty

podle vzorce $\rho = \frac{C(V_n, V_{n+1})}{\sigma(V_n) \cdot \sigma(V_{n+1})}$. Čím delší část periody jsem použil, tím nižší byla $|\rho|$.

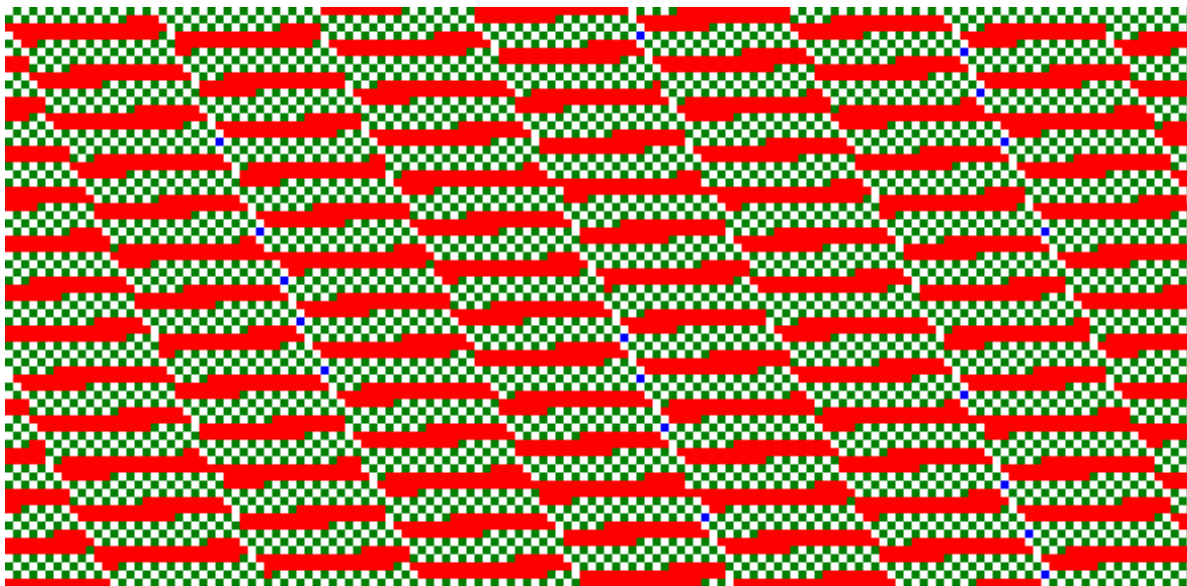
Pro 10^8 hodnot vyšlo -0,000 075. Rozhodl jsem se, že touto statistickou oblastí se dál nebudu zabývat, protože z ní nedokonalost LCG nezjistíme.

Jaký je vztah mezi povahou (PID mod 25) a šesticí hodnot IV?

Podle způsobu získání IV víme, že každá uspořádaná šestice IV odpovídá 4 uspořádaným dvojicím V_n . Každé pozici v hlavní periodě odpovídá určitá hodnota PID. Některá hodnota PID se ale v periodě vůbec nevyskytuje, jiná se objevuje vícekrát. Dvěma PID odpovídají v 96 % případů dvě různé povahy.

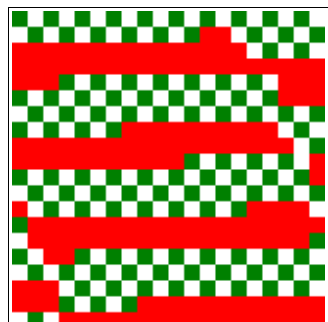
Vytvořil jsem program, který graficky zobrazuje korelace dvou za sebou jdoucích hodnot

V_n . Červený bod vyjadřuje, že daná dvojice za sebou šla 1x, zelený bod 2x, modrý bod 3x a pokud se daná dvojice vůbec nevyskytla, je bod bílý. Celý výstup má rozměry 65 536 x 65 536 a toto je jen malý výřez, který ukazuje, jakou mají výstupní hodnoty strukturu.

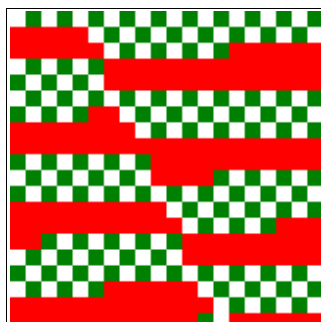


Celý výstup můžeme rozdělit na 4 čtverce: 00 s levým dolním rohem v $[0 ; 0]$, 0Y s levým dolním rohem v $[0 ; 2^{15}]$, X0 s levým dolním rohem v $[2^{15} ; 0]$ a XY s levým dolním rohem v $[2^{15} ; 2^{15}]$.

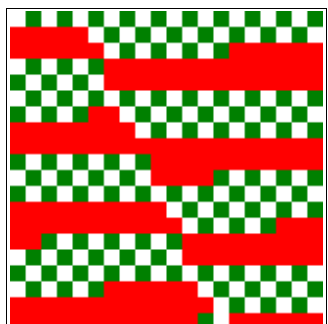
Toto jsou 4 detaily o rozměrech 20 x 20 z levého dolního rohu všech 4 čtverců.



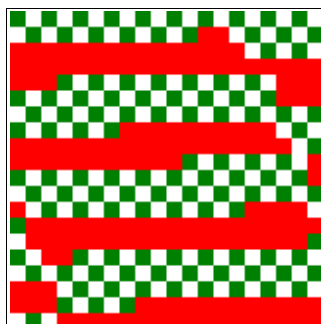
0Y



XY



00



X0

Můžeme vidět, že čtverec 0Y se shoduje s X0. Čtverec 00 se shoduje s XY. To je ekvivalentní se skutečností, že pokud se hodnota V_n zvýší/sníží o 2^{15} , hodnota V_{n+1} se zvýší/sníží o 2^{15} . Důvodem je, že a je liché a m je sudé.

S ohledem na to, jak se IV získává, každé uspořádané šestici odpovídají 4 body v těchto korelacích, každý je v jednom čtverci a všechny na stejném místě ve svých čtvercích. Protože se vždy dvojice čtverců shodují, každé uspořádané šestici IV přísluší sudý počet umístění v periodě. Dále si všimněme, že kdybychom čtverce dali na sebe, nikdy se nebudou překrývat bílými body, tudíž můžeme získat každou kombinaci IV. Programem jsem určil, že 216 760 320 uspořádaných šestic IV se vyskytuje v periodě 2x, 640 221 184 se jich vyskytuje v periodě 4x a 216 760 320 se jich vyskytuje v periodě 6x. Modrých bodů je málo a všechny by se při překrytí kryly s bílými body.

Závěr mojí práce je, že jsem nenašel exaktní postup pro popis vztahu mezi generovanými hodnotami, ale vztah mezi nimi existuje, takže generátor není použit dokonale. Vytvořil jsem dvojici programů, které hrubou silou procházejí periodu LCG. Program RNG4_PID_to_IV.exe určuje vlastnosti Pokémona podle PID. Druhý program RNG4_IV_to_PID.exe najde, jaké povahy (a případně další parametry vyplývající z PID) může mít Pokémon se zadanými IV. Kterýkoliv z těchto programů lze použít k ověření, že statistiky nějakého Pokémona nebyly nacheatovány pomocí podvodných zařízení jako Action Replay.

Tímto programem jsem zjistil, že při maximálních IV může mít legendární Pokémon pouze následující povahy: Docile, Calm, Modest a Timid. To je výhodnější pro Pokémony útočící především speciálními útoky, kterých je stejně mezi legendárními Pokémony většina, hlavně těch psychických. Dále jsem experimentoval s jinými typy zadání. Za zmínku stojí, že pokud požadujeme 5 konkrétních hodnot IV a 6. nám je jedno (typicky Attack nebo Special Attack), většinou je možné mít vybranou povahu, ale ne vždy, například pro IV: 15/15/15/15/15/x nemůžeme mít povahu Timid.

5 Optimalizace obranných EV

H označuje HP.

D označuje Defense.

E označuje Special Defense.

X označuje obecnou statistiku.

5.1 Vysvětlení optimalizace

Maximální level je 100. V naprosté většině případů se hraje maximální povolený level, tj. většinou 100, budeme proto rozebírat tento (jednodušší) případ.

Povaha Pokémona p určuje koeficienty pro statistiky kromě H . Jsou 2 skupiny povah. První skupina má ve všech statistikách koeficient 1. V této skupině je celkem 5 povah, které jsou pro nás shodné. Druhá skupina má v jedné statistice koeficient 1,1, v jiné statistice koeficient 0,9 a ve zbylých 3 koeficient 1. Existuje celkem všech 20 možností. Kromě Pokémonů získaných na speciálních eventech je možné získat exemplář s libovolnou povahou.

IV nabývá hodnot 0 až 31. Je určeno při chycení, vylíhnutí nebo obdržení Pokémona. Většinou je teoreticky možné získat Pokémona s všemi IV maximálními, proto budeme předpokládat hodnotu 31. Výjimkou jsou legendární Pokémoni ve hrách 3. a 4. generace. Vzhledem k principu použitého generátoru pseudonáhodných čísel a způsobu získání hodnot je možné získat maximální IV pouze pro některé povahy. Není to však moc významné, protože buď Attack nebo Special Attack je nám obvykle jedno a zbylé je již možné získat maximální.

EV nabývá hodnot 0 až 255. Ve všech statistikách dohromady je možné dosáhnout součtu EV maximálně 510. Body do EV jsou Pokémonovi dávány během tréninku podle poražených Pokémonů a případně podle drženého předmětu.

To nás vede k důležité a často obtížné úloze: Jak EV do statistik rozdělit? Před zahájením tréninku bychom měli mít rozmyšlenou celou sestavu Pokémonů, abychom věděli, jaké vlastnosti od daného Pokémona chceme. Protože se EV celočíselně dělí 4, budeme pochopitelně do jednotlivých statistik aplikovat násobky 4. S neutrální povahou nám aplikace 508 EV zvýší statistiky celkem o 127, zbylé 2 EV jsou k ničemu.

Při maximálním levelu a IV, neutrální povaze a nulovém EV dosáhneme kvaziminimálních hodnot, které označíme dolním indexem M .

$$X_M = 2 \cdot Z + 36 \quad H_M = 2 \cdot Z + 141$$

Bonus za EV označíme dolním indexem B . $X_B = \frac{EV}{4}$

Skutečná hodnota statistiky je potom X (popřípadě H).

$$X = X_M + X_B$$

Dále definujme G .

$$G = D_B + E_B + H_B$$

S G budeme pro optimalizaci obranných EV pracovat jako se vstupem (nebo chcete-li parametrem). Řekněme, že jsme již rozhodli investici EV do útočných statistik (Attack, Special Attack, Speed), což jsou zcela jistě násobky 4. Potom zbývá hodnota G , o kterou je možné v součtu vylepšit HP, Defense, Special Defense.

$$G = 127 - A_B - B_B - S_B$$

Cílem optimalizace obranných EV je, aby našemu Pokémonovi způsobovaly útoky co

nejnižší zranění ve srovnání s jeho HP, tedy aby výraz $\frac{K}{H}$ byl minimální. Pro zjednodušení zkusíme 2 věci. Zaprvé ve vzorci pro zranění z fyzických útoků smažeme

druhé přičtení 2 a získáme
$$K = \frac{(\frac{2 \cdot 100}{5} + 2) \cdot A \cdot U}{50 \cdot D} \cdot f \cdot T \cdot R = \frac{21 \cdot A \cdot U \cdot f \cdot T \cdot R}{2500 \cdot D}$$
, pro

speciální útoky analogicky. Část vzorce, o které nerozhodujeme, popíšeme konstantou

$c = \frac{21 \cdot A \cdot U \cdot f \cdot T \cdot R}{2500}$ (pro speciální útoky analogicky). Zadruhé budeme nebezpečí

fyzických a speciálních útoků považovat za rovnocenné. Relativní síla (vyjadřující jakou

část HP nám útok ubere) fyzického útoku pro nás tedy je $\frac{c}{H \cdot D}$ a speciálního útoku

$\frac{c}{H \cdot E}$. V součtu nám fyzický a speciální útok ubere relativně $\frac{c \cdot (D + E)}{H \cdot D \cdot E}$. Naším

cílem je tedy maximalizovat obrannou funkci $y = \frac{H \cdot D \cdot E}{D + E}$.

5.2 Optimalizace pro $p = 1$ v Defense a Special Defense

Součástí optimálního řešení zcela jistě bude využití všech volných EV , tedy

$$H_M + D_M + E_M + G = H + D + E$$

. Není ovšem pravda, že hodnoty H , D a E jsou určeny jen součtem jejich kvaziminimálních hodnot a počtem volných EV , EV totiž nemůžou jít

do záporu ani přes maximální hodnotu pro danou statistiku. Proto:

$$H \in \langle H_M ; H_M + \min(63, G) \rangle$$

$$D \in \langle D_M ; D_M + \min(63, G) \rangle$$

$$E \in \langle E_M ; E_M + \min(63, G) \rangle$$

Řekněme, že H je dané. Jaké D a E je potom optimální? Součet D a E je daný, označme ho F . Hledáme pak maximum obranné funkce:

$$y = \frac{H \cdot D \cdot (F - D)}{D + (F - D)} = \frac{H \cdot D \cdot (F - D)}{F} = H \cdot D - \frac{H \cdot D^2}{F}$$

Zderivujeme ji podle D .

$$y' = H - \frac{2 \cdot H \cdot D}{F}$$

A najdeme maximum.

$$H - \frac{2 \cdot H \cdot D}{F} = 0$$

$$D = \frac{F}{2}$$

Pro dané H je tedy optimální, aby $D = E$.

Nyní řekněme, že víme, že $D = E$ a hledáme hodnotu H (ze které vyplyne D a E). Součet

H , D a E , který je daný, označíme Z . Potom $D = E = \frac{Z - H}{2}$. Obranná funkce pak je:

$$y = \frac{H \cdot \left(\frac{Z - H}{2}\right) \cdot \left(\frac{Z - H}{2}\right)}{\left(\frac{Z - H}{2}\right) + \left(\frac{Z - H}{2}\right)} = \frac{H \cdot (Z - H)}{4}$$

$$y' = \frac{Z - 2H}{4} = 0$$

$$H = \frac{Z}{2}$$

Došli jsme tedy k závěru, že optimální poměr $H : D : E = 2 : 1 : 1$. Toto rozložení aplikujeme přesně, pokud je Z dělitelné 4. V opačném případě hodnoty zaokrouhlíme dolů a přebývající EV přidělíme následovně: Pokud Z dává po dělení 4 zbytek 1, zvětšíme H o 1. Pokud dává Z po dělení 4 zbytek 2, je jedno, jestli zvětšíme H o 2 nebo jestli zvětšíme D a E o 1. Pokud dává Z po dělení 4 zbytek 3, je nejlepší zvýšit H , D a E o 1.

Pokud se nám optimální H dostane mimo dané rozmezí, přiřadíme mu hodnotu příslušné krajní meze a jednoduše položíme $D = E$ pro získání optimálního rozložení (případně libovolnou zvětšíme o 1, pokud je $Z - H$ liché).

Nejsložitější situace nastává, pokud se D nebo E dostane mimo dané rozmezí. Dané statistice přidělíme hodnotu příslušné krajní meze a otázkou je, jak vyvážit zbylé dvě. Řekněme, že se mimo rozmezí dostalo D a musíme vyvážit E a H (druhý případ analogicky).

Hledáme maximum obranné funkce:

$$y = \frac{H \cdot D \cdot (Z - D - H)}{Z - H} = \frac{H \cdot D \cdot Z - H \cdot D^2 - H^2 \cdot D}{Z - H}$$

Zderivujeme ji podle H .

$$y' = \frac{(D \cdot Z - D^2 - 2 \cdot D \cdot H) \cdot (Z - H) - (H \cdot D \cdot Z - H \cdot D^2 - H^2 \cdot D) \cdot (-1)}{(Z - H)^2}$$

Opět najdeme maximum.

$$(D \cdot Z - D^2 - 2 \cdot D \cdot H) \cdot (Z - H) = H^2 \cdot D + H \cdot D^2 - H \cdot D \cdot Z$$

$$D \cdot Z^2 - D \cdot Z \cdot H - D^2 \cdot Z + D^2 \cdot H - 2 \cdot D \cdot H \cdot Z + 2 \cdot D \cdot H^2 = H^2 \cdot D + H \cdot D^2 - H \cdot D \cdot Z$$

$$D \cdot H^2 - 2 \cdot D \cdot Z \cdot H + D \cdot Z^2 - D^2 \cdot Z = 0$$

$$H^2 - 2 \cdot Z \cdot H + Z^2 - D \cdot Z = 0$$

$$H_{1,2} = \frac{2 \cdot Z \pm \sqrt{4 \cdot Z^2 - 4 \cdot Z^2 + 4 \cdot D \cdot Z}}{2} = Z \pm \sqrt{D \cdot Z}$$

Kořen H_1 leží mimo interval, kořen H_2 je řešení.

Do HP dáme $(H - H_M) \cdot 4$ EV.

Do Defense dáme $(D - D_M) \cdot 4$ EV.

Do Special Defense dáme $(E - E_M) \cdot 4$ EV.

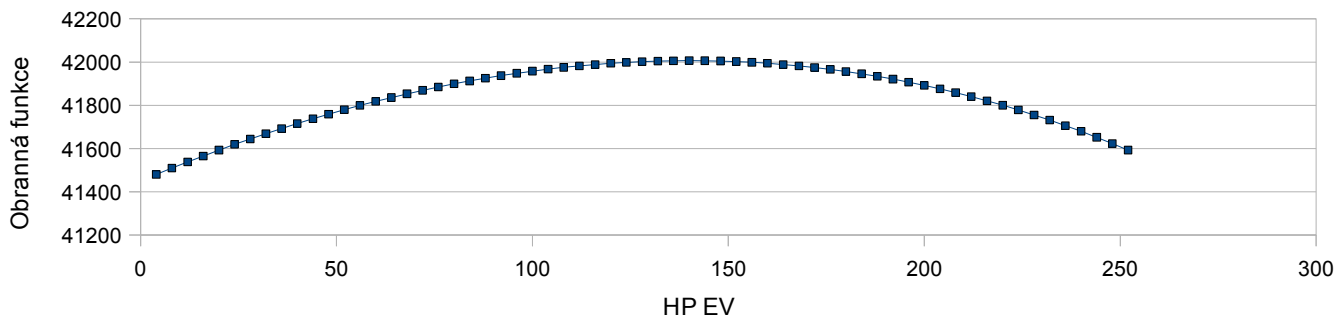
5.3 Optimalizace v obecném případě

Vyřešit optimalizaci obranných EV pro jakoukoliv povahu je už mnohem těžší, protože dochází k zaokrouhlení dolů po započítání povahy. Optimální řešení toho diskrétního problému tedy už neodpovídá řešení nalezenému pro problém deklarovaný spojitě tak, jako jsme toho využívali pro obranně neutrální povahy.

Rozdíl mezi řešením problému pro obranně neutrální povahu a pro povahu zvyšující jednu z obran demonstrují následující grafy. Mějme Pokémona se základy 120 v HP, 50 v Def a 70 v SpD. Pak $H_M = 381$, $D_M = 136$ a $S_M = 176$. V obou případech se vyplatí dát 252 EV do Defense (zadání je tak zvoleno záměrně, ale nebyl na demonstraci potřeba 3D graf).

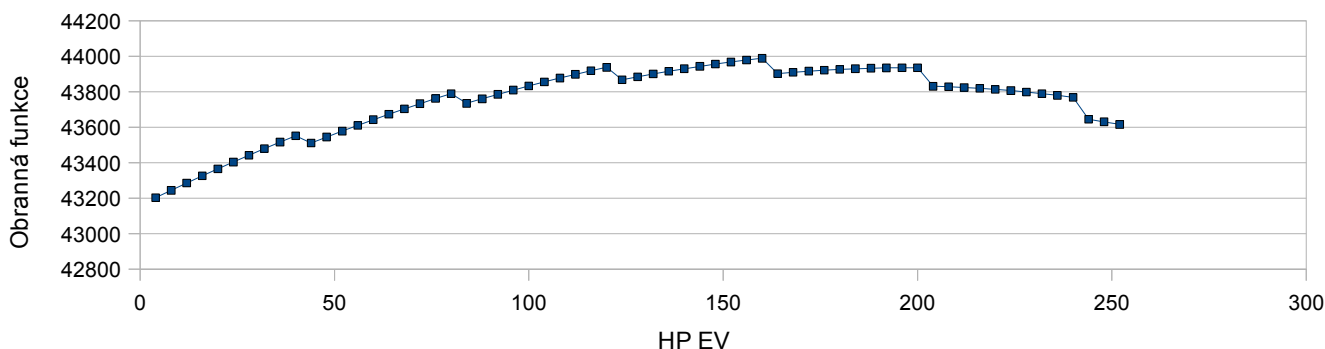
Závislost obranné funkce na EV v HP ; neutrální povaha

Def EV = 252 ; SpD EV = 256 - HP EV



Závislost obranné funkce na EV v HP ; povaha zvyšující SpD

Def EV = 252 ; SpD EV = 256 - HP EV



Z grafu je patrné, že pro jinou než neutrální povahu při výpočtu nevíme, jestli se maximum nachází na okraji intervalu, v bodě odpovídajícímu spojitému řešení nebo na vrcholu „zubu“ daného tím, že jsme v posílené statistice před vynásobením 1,1 dosáhli násobku 10 a statistika se tu zvýšila o 2 oproti předchozí hodnotě (místo častějšího zvýšení o 1).

Proto jsem vytvořil program DefensiveEV.exe , který řeší optimalizaci obranných EV pro jakoukoliv povahu hrubou silou.

Uživatel zadá základy pro HP, Def a SpD a počet volných EV (což je hodnota $4G$). Program za H_B postupně dosazuje čísla od 0 do menšího z čísel 63 a G a pro každou hodnotu H_B zkouší za D_B dosadit čísla od 0 do menšího z čísel 63 a $G - H_B$. Pak spočítá S_B jako $63 - H_B - D_B$. Následně program vypočítá hodnoty statistik a poté hodnotu obranné funkce. Pokud nalezne řešení lepší než dosavadní nejlepší a S_B při tom nepřevyšuje 63, tak novou možnost uloží jako nejlepší řešení. Nakonec nejlepší řešení vypíše ve tvaru počtu aplikovaných EV do jednotlivých statistik, výsledných statistik a hodnoty obranné funkce.

	Base	EVs	Ideal	Stat
HP:	120	160		421
Defense:	50	252		199
Special Defense:	70	96		220
EVs left	510			

Coefficient (Nature)
 None
 + Defense
 - Defense
 + Special Defense
 - Special Defense

Defensive function result: 43988,9737470167

5.4 Správnost povahy

Pro optimální rozdělení EV (nejen těch obranných) musí být také zvolena správná povaha. Vybrat statistiku, která může být povahou snížena, bývá většinou snadné. Obvykle se vybírá povaha posilující nejvyšší ze statistik, které je žádoucí zvýšit. Jindy by nějaké nižší statistice nestačila maximální investice 252 EV, takže se ještě zvýší povaha. Je ale zcela neoptimální, aby byla povahou posílena jakákoliv nižší statistika, ve které je tolik EV, že by bonus od povahy nebyl potřeba pro dosažení dané hodnoty při více EV, zatímco v nějaké statistice s vyšší hodnotou jsou investovány EV, které by mohly být nižší při správné povaze. V takovém případě by volbou správné povahy mohly být dosaženy stejné hodnoty při nižším součtu spotřebovaných EV.

Příkladem objektivně chybné volby povahy může být Walrein zveřejněný v [5]. Byla u něj doporučena povaha Modest zvyšující SpA na úkor Att s následujícími EV: 224 HP, 216 Def, 48 SpD, 20 Spe. Je tak dosaženo Defense 270 a Special Attack 248. Defense je tak vyšší, ale povahou je posílen Special Attack. S povahou Bold, která posiluje Def, by bylo dosaženo stejných hodnot při investici pouze 120 EV do Def a 88 EV do SpA, takže by ještě zbylo 8 (10) EV na zvýšení jakékoliv statistiky o 2 body.

Argument lze chápat logicky, ale odvození je nasnadě. Ve statistikách X a Y chceme dosáhnout určitých hodnot při spotřebování co nejméně EV.

Když povahou posílíme statistiku X , bude platit: $X = (X_M + X_{B1}) \cdot 1,1$ a $Y = Y_M + Y_{B1}$.

S povahou zvyšující statistiku Y bude platit: $X = X_M + X_{B2}$ a $Y = (Y_M + Y_{B2}) \cdot 1,1$.

V prvním případě budou spotřebovaná EV: $G_1 = X_{B1} + Y_{B1} = \frac{X}{1,1} - X_M + Y - Y_M$,

v druhém případě $G_2 = X_{B2} + Y_{B2} = X - X_M + \frac{Y}{1,1} - Y_M$.

První povaha je výhodná právě tehdy, pokud $G_1 < G_2$.

$$\frac{X}{1,1} + Y < X + \frac{Y}{1,1}$$

$$X + 1,1 Y < 1,1 X + Y$$

$$0,1 Y < 0,1 X$$

$$X > Y$$

Pokud tedy při obou povahách leží EV v rozmezí od 0 do 252, je výhodnější povaha zvyšující statistiku s vyšší požadovanou hodnotou.

6 Umělá inteligence pro hru Pokémon?

Originální hry Pokémon od Nintendo obsahují velmi základní umělou inteligenci, která hraje za soupeře i divoké Pokémon v in-game. Nicméně nerozhoduje se vůbec efektivně a pro soutěžní hraní by byla nepoužitelná. Kdyby existovala kvalitní umělá inteligence pro hru Pokémon, mohli by ji použít soutěžní hráči k procvičování hry, analýze odehraných her a především k testování tým na základě tisíců odehraných her samotným počítačem.

Jaké metody by mohla umělá inteligence pro hru Pokémon používat? Hra Pokémon je pro počítač nesrovnatelně složitější než třeba šachy, protože se tu vyskytuje náhoda, skryté informace a také proto, že je to současná hra. Není tu tedy možné vycházet z algoritmu minimax. Nicméně jeho princip je stále možné aplikovat v případě, že byl hráči vyřazen Pokémon a vybírá dalšího (nebo používá U-Turn / Volt Switch po útoku soupeře).

Základem umělé inteligence pro hru Pokémon by byla bezvadná implementace herní mechaniky originálních her, stejně jako v případě simulátorů. Program by bez větších problémů dokázal dosáhnout kvalit jako bezvadný výpočet zranění z jednotlivých útoků pro konkrétní zadání a výpočet pravděpodobností z jasných zadání, i různě kombinovaných a podmíněných. Program by se nikdy nezmyšlil v počítání PP, kol od vyvolání nějakého efektu, ani by nikdy neopomenul existenci různých schopností.

Program by asi stěží dokázal využít psychologii soupeře, ale samotný program by zase byl odolný proti psychologickým taktikám. Velkým problémem AI by bylo využívat nepřímou předaných informací. Například to, že když soupeř nenasadil před x koly tohoto Pokémona proti tomu našemu Pokémonovi, nemá u něj pravděpodobně tento útok, ale spíš tenhle útok. Ale celkové trendy v sestavování týmu by mohly být analyzovány s využitím statistik metagame a počítač by mohl soupeře taky škatulkovat tím, že by po několika odehraných kolech, v čase, který nepotřebuje k rozhodování, zařazoval soupeře s určitou pravděpodobností k nějakým skupinám hráčů používající s určitými pravděpodobnostmi některé jevy při sestavování týmu.

Pokud by byla kvalitně zvládnutá umělá inteligence pro zápasy, bylo by možné s využitím statistik metagame konkrétních formátů velice efektivně aplikovat evoluční algoritmy k hledání dobrých týmů. Jedincem by zde byl celý tým 6 Pokémonů. Mohlo by se začít od náhodných týmů (nebo nějaké množiny vytažené ze statistik dané metagame) a postupně by byly vytvářeny dědičností a mutacemi týmy nové. Fitness jedince by zde určovalo procento vyhraných zápasů. Fitness týmů by měla být postupně aktualizována, zatímco by byly vytvořené týmy testovány pro dalším a lepším týmům. Takovýto způsob testování týmů by vyžadoval obrovský výpočetní výkon, ale teoreticky by mohl přinést optimálnější

výsledky, než testing prováděný hráči. Nicméně i kdyby potom hráč s takovým týmem odehrál mnoho her, mohl by zjistit, že se mu s takovým týmem moc nedaří.

7 Závěr

Byly řešeny problémy, které jsou důležité zejména pro soutěžní hráče. Problémy jsou za současných podmínek přesně a uspokojivě vyřešeny. Byla důkladně otestována a následně vysvětlena herní mechanika. S využitím programu na výpočet IV lze Pokémony křížit rychleji než s využitím konkurenčních programů. Na základě analýzy RNG byla zjištěna nejen teoretická omezení hry, ale také je ji možno použít k ověření pravosti Pokémonů. Podle optimalizace obranných EV je možno sestavovat týmy s ambicemi na vyhrávání turnajů. Optimalizace byla řešena pro 4. generaci her, ale má stejnou platnost i pro 5. generaci.

8 Použitá literatura

- [1] THE POKÉMON COMPANY INTERNATIONAL. *Pokémon* [online]. 2013 [cit. 2013-03-11]. Dostupné z: <http://www.pokemon.com/us/>
- [2] End-of-Turn Effects. In: STONE, David. *Smogon University* [online]. 2007 [cit. 2013-03-11]. Dostupné z: <http://www.smogon.com/forums/showthread.php?t=28946>
- [3] The Process of PID and IV Creation of Non-Bred Pokemon. In: FARRUGIA, Alexander. *Smogon University* [online]. 2009 [cit. 2013-03-11]. Dostupné z: http://www.smogon.com/ingame/rng/pid_iv_creation
- [4] KNUTH, Donald E. *The Art of Computer Programming: Volume 2 – Seminumerical Algorithms*. USA: Addison-Wesley, 1969. ISBN 0-201-03802-1.
- [5] MAKKOTO, Jigen. Flavor of the Month: Hail. *The Smog* [online]. 2009, č. 5 [cit. 2013-03-11]. Dostupné z: <http://www.smogon.com/smog/issue5/flavor>