

Univerzita Komenského v Bratislave
Fakulta Matematiky, Fyziky a
Informatiky

Tvorba príbehov v počítačových
hrách

Bakalárska práca

Univerzita Komenského v Bratislave
Fakulta Matematiky, Fyziky a
Informatiky

Tvorba príbehov v počítačových
hrách

Bakalárska práca

Evidenčné číslo: ...

Študijný program: Aplikovaná informatika

Študijný odbor: 9.2.9. aplikovaná informatika

Školiace pracovisko: Katedra Aplikovanej Informatiky

Školiteľ: RNDr. Jozef Šiška, PhD.

Bratislava 2017

Tomáš Bakoš

Obsah

1	Úvod	4
2	Východiská	6
2.1	Inšpirácia	6
2.2	Plánovanie	6
2.2.1	Fluent	6
2.2.2	Stav	6
2.2.3	Akcia	7
2.2.4	Riešenie/Plán	7
2.3	Výpočtová naratológia	8
2.4	Systémy generujúce príbehy	9
2.4.1	Novel Writer system	9
2.4.2	Talespin	10
2.4.3	Author	10
2.4.4	Universe	11
2.4.5	Minstrel	11
2.4.6	Mexica	12
2.4.7	Brutus	12
2.4.8	Fabulist	13
2.5	Goal Oriented Action Planning	13
2.6	Version Control System	15
2.7	Predošlé bakalárske práce	15

1. Úvod

Príbehy nás sprevádzajú celým životom, či ich už sú to naše vlastné, našich príbuzných, kamarátov alebo ľudí ktorých ani nepoznáme. Od malicka ich citáme a stretávame sa s nimi vyhľadávame ich. Z vlastnej skúsenosti viem, že človeka dokážu pohltiť, prieť ho k zamysleniu alebo jednoducho si pri nich človek oddýchne. Čo ich ale robí tak zaujímavými? Už starovekí gréci sa zaoberali písaním a štúdiom tvorby príbehov. Z naratologie, ktorej históriu datujeme až ku starovekým grékom vieme, že príbeh by mal mať nejakú dejovú štruktúru aby nás zaujal a príbeh by nemal mať veľa hlavných postáv, potom by sa mohlo stať že sa čitateľ v príbehu stratí. Navyše tvorba príbehov je komplexný proces, pri ktorom my ľudia veľa vecí robíme tak nejak automaticky.

Zaujímavá problematika nastáva, keď túto tvorbu príbehov dáme za úlohu počítaču. Keďže problém je komplexný a navyše ťažko definovateľný (veľa procesov robíme v hlave automaticky), vzniká pre programátora netriviálna úloha ako sa s týmto problémom popasovať. Takýmito úlohami sa zaoberá vedná disciplína výpočtová naratológia, ktorá študuje tvorbu príbehov z výpočtového hľadiska. Zameriava sa na algoritmické procesy ktoré vytvárajú a interpretujú príbehy a tiež na modelovanie štruktúry príbehu z hľadiska vypočítateľných reprezentácií. Povieme si o nej viac v ďalšej kapitole.

Počítačové hry sú mojou záľubou, ktorá ma sprevádza už od malička a teda mi bolo prirodzené zasadiť problematiku tvorby príbehov do počítačovej hry. Hra príbeh zinterktívni a autori, programatori príbehu majú k dispozícii viacero nástrojov ako hráča viac vtiahnuť do deja. Môžu použiť rôzne obrázky, zvuky, animácie, ktoré príbeh dotvárajú a vytvárajú určitú atmosféru. Kombináciou tvorby príbehov a tvorby hier si myslím že vzniká zaujímavá téma na študovanie.

Cieľom tejto práce je teda vytvoriť jednoduchú textovú hru, ktorá bude generovať dobre rozvinutý príbeh. Tento príbeh sa zakaždým vygeneruje nanovo, teda hru bude možné hrať znovu a znovu vždy s iným príbehom. Keďže ide o

hru tak posun v príbehu bude záležať od interakcie hráča ktorý hru hrá. Ďalším aspektom je, že príbeh buďe "zaujímavý", to znamená, že sa nebudú často opakovať tie isté akcie, akcie budú na seba logicky nadväzovať, budú zmysluplné z hľadiska deju atď. Spôsob akým sa dajú vytvárať takéto príbehy popíšeme v nasledujúcej kapitole.

2. Východiská

V tejto kapitole opíšem všetkú potrebnú teóriu, poznatky, metódy, ktoré sú potrebné alebo mi pomohli pri vypracovávaní témy zadania.

2.1 Inšpirácia

Už dlhší čas som mal chuť vytvoriť nejakú počítačovú hru. Rád som v pozícii hráča a tak som si chcel aj vyskúšať pohľad na hry z tej druhej strany a vyskúšať si aspoň v malom merítku s akými problémami sa vývojari stretávajú. V poslednej dobe ma začala zaujímať téma umelej inteligencie a tak téma tejto práce bola príjemným sklbením mojich dvoch záujmov.

2.2 Plánovanie

Plánovanie je problém alebo oblasť, ktorou sa zaoberá umelá inteligencia. Je to proces hľadania nejakých akcií, ktoré je potrebné aby sa vykonali na dosiahnutie vytýčeného cieľa. Je základným stavebným kameňom v problematike zadanej témy a teda upresním nejaké základné definície z tejto oblasti.

2.2.1 Fluent

Fluent vo všeobecnosti je niečo, čo dokáže plynúť ako tekutina. V našom ponímaní je to objekt, ktorý opisuje či niečo platí alebo nie a je schopný sa meniť časom. Je to vlastne niečo ako predikát v prvorádovej logike.

2.2.2 Stav

Stav je definícia, opis sveta alebo objektu s ktorým pracujeme. Príbeh bude vlastne tvorený menením stavov sveta, čo nás privádza k akciám.

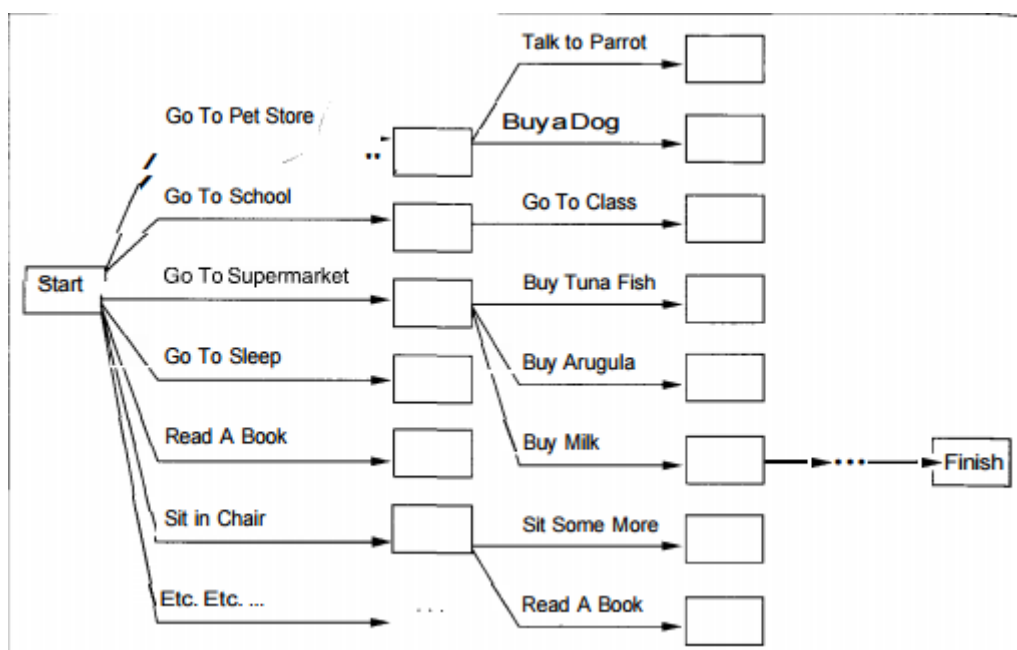
2.2.3 Akcia

Akcia obsahuje predpoklady a dôsledky. Predpoklady sú nejaké podmienky alebo stav, ktorý musí byť splnený aby sa daná akcia mohla vykonať. Dôsledky na druhej strane sú efekty, ktoré ovplyvnia aktuálny stav sveta alebo iného objektu na ktorom sa akcia vykonáva. Pomocou akcií sa bude posúvať príbeh a vyvíjať postavy.

2.2.4 Riešenie/Plán

Nakoniec k riešeniu (plánu) sa dostávame zadefinovaním počiatočného a koncového stavu (cieľa) a nasledným prechádzaním akcií, ktoré možno vykonať až kým nedosiahneme cieľa. Teda riešenie plánovania (plán) je neprerušená postupnosť akcií z počiatočného stavu do cieľa.

Jedna z reprezentácií ako definovať vzťahy medzi týmito pojmami je nejaká grafová štruktúra. V našom prípade to bude strom. Na obrázku môžeme vidieť stavy



Obr. 2.1: Ukážka reprezentácie stavov a akcií

definované ako vrcholy grafu a akcie ako hrany medzi nimi. Máme počiatočný stav Start a koncový stav Finish. V tomto prípade je našim cieľom získanie mlieka. A teda v stave Start už máme definované fluenty a počiatočný stav sveta. Potom vhodným grafovým prehľadávacím algoritmom vieme hľadať postupnosť akcií a stavov pomocou, ktorých sa dostaneme z počiatočného stavu do koncového. Teda

riešenie tohto plánovacieho problému je vlastne cesta v grafe začínajúca vo vrchole Start a končiaca vo vrchole Finish. Táto cesta nám teda hovorí, že na dosiahnutie nášho cieľa (získanie mlieka), musíme ísť do supermarketu a v ňom kúpiť mlieko. Riešení by mohlo byť aj viac ak by sme mali iný graf alebo doplnili do tohto nové akcie a stavy.

2.3 Výpočtová naratológia

Naratológiu ako pojem prvý krát zaviedol Tzvetan Todorov, bulharsko-francúzsky filozof, historik, sociológ a literárny kritik. Je to humanitná disciplína, ktorá sa zaoberá štúdiom narácie, teda príbehu alebo postupnosti udalostí. Ďalej skúma štruktúru, logiku, princípy a tiež aj reprezentáciu narácie. Zo začiatku dominovali štrukturálne prístupy štúdia z ktorých sa vyvynuli rôzne teórie, koncepty a analytické procedúry. Tieto koncepty a modely sú používané ako heuristické nástroje a naratologické teórie hrajú kľúčovú rolu v našej schopnosti vytvárať a spracovávať narácie vo všetkých možných formách. Čo nás privádza k umelej inteligencii a výpočtovej naratológii.

Výpočtová naratológia študuje tvorbu príbehov z pohľadu vypočítateľnosti a spracovávania informácií. Zameriava sa na algoritmické procesy ktoré vytvárajú a interpretujú príbehy a tiež na modelovanie štruktúry príbehu z hľadiska vypočítateľných reprezentácií. Sem patria aj spôsoby automatickej interpretácie a tvorby príbehov, ďalej aj prístupy k rozprávaniu príbehov pomocou umelej inteligencie v hrách. Teda výskumníci sa snažili vytvoriť systémy umelej inteligencie, ktoré by rozprávali príbeh ako ľudia a tiež sa pokúšali vytvoriť inteligentné počítačové prostredie na interakciu s naráciami. V rámci vývoja týchto systémov, výskumníci využili princípy z naratológie na vytvorenie výpočtových princípov a vysvetlili prepojenia medzi nimi. Jeden z princípov bolo využitie naratologického rozdelenia fabuly a sujetu. Kde fabula je zvyčajne charakterizovaná ako prirodzený sled udalostí celého príbehu v chronologickom poradí. Sujet na druhú stranu je umelecky realizovaná fabula, teda je to konštrukcia epickej alebo dramatickej fabuly. Výpočtová naratológia bola významne ovplyvnená aj lingvistikou napríklad gramatikami príbehov. Je to rýchlo rozvíjajúce sa odvetvie, hlavne vďaka zvýšenému

záujmu o interaktívne hry a príbehy, ktoré sa javia ako živé.

2.4 Systémy generujúce príbehy

Systémy generujúce príbehy vznikali ako odpoveďe na otázky výpočtovej naratológie. Hľadaním všeobecných výpočtových metód, ktoré by sa dali použiť na rôzne druhy narácie sa v 70-tych rokoch 20. storočia upriamila pozornosť na plánovanie. Odvtedy sa toto zameranie veľmi nezmenilo, ale plánovacie techniky sa vylepšili aby mohli poňať obsiahle problémy naratológie.

V problematike plánovania, na pochopenie príbehu je potrebné vyvodenie založené na Aristotelovom ponímaní mýtu, kde príčiny udalostí príbehu a ciele zapojených postáv su známe. V podstate zrekonštruovať z viet v sujete plán, ktorý reprezentuje súslednosť udalostí, ktoré dokážu počiatočný stav transformovať na cieľový. Takéto systémy, ktoré sa snažia pochopiť príbeh sa nedostali veľmi ďaleko z troch dôvodov. Po prvé, vyvodenie cieľov zainteresovaných postáv si vyžaduje obrovský priestor na prehľadávanie. Ďalej, ľudia využívajú obrovské množstvo znalostí na pochopenie aj tých najjednoduchších príbehov. Príkladom nám môže byť veta od anglického spisovateľa Edwarda Morgana Fostera: "Kráľ zomrel a kráľovná zomrela od žiaľu.", z ktorej nám ľuďom je jasné prečo bola kráľovná smutná, avšak definovať takúto dávku zdravého rozumu počítaču je náročné. Po tretie, niektoré aspekty jazyka, ktoré sú pre pochopenie príbehu dôležité, sa ťažko formalizujú ako napríklad humor, irónia a iné nepatrné lexikálne prostriedky. Na druhú stranu algoritmy, ktoré využívajú na generovanie príbehu plánovanie pomocou fabuli sa osvedčili oveľa viac aj preto že si autor môže výrazne obmedziť systém. O takýchto algoritmoch si v tejto sekcii povieme viac.

2.4.1 Novel Writer system

Prvý storytellingový systém, ktorý bol zaznamenaný je Novel Writer system od Sheldona Kleina. Vytváral príbehy o vražde na víkendovej oslave. Údajne vedel vygenerovať 2100 slovné príbehy o vražde, s hlbokou myšlienkou za menej ako 19 sekúnd. Ako vstup sa zadal opis sveta v ktorom sa mal príbeh odohrať a aj charakteristika postáv, ktoré vystupovali v príbehu. Vrah a obež zúvyseli

od charakterových črt tiež špecifikovaných ako vstup. Všetky možné motívy na vraždu boli obmedzené iba na chamtivosť, zlosť, žiarlivosť a strach. Príbeh bol vygenerovaný pomocou dvoch algoritmov. Prvým bola množina pravidiel, ktorá obsahovala všetky možné zmeny stavu sveta z aktuálneho stavu do nasledujúceho. Druhým bola postupnosť scén ktorá korešpondovala s typom príbehu, ktorý mal byť prerozprávateľ. Nevýhodou bolo, že množina pravidiel vysoko obmedzovala typ príbehu, ktorý sa vedel generovať. Rozdiely v príbehu spočívali len v tom, kto koho zabil s čím a prečo a v tom kto objavil mrtvolu.

2.4.2 Talespin

Neskôr bol vyvinutý systém TALESPIN, ktorý generoval príbehy o lesných tvoroch. Na tvorbu príbehu sa postave zadal cieľ alebo zámer a následne sa zkonštruoval plán ktorým bol dosiahnutý daný cieľ. TALESPIN ako prvý využil charakterové ciele ako spúšťače akcií. Taktiež tu bola po prvý krát využitá možnosť viacerých takýchto postáv, kde každá mala svoj zoznam cieľov, ktoré chce dosiahnuť. Čo dalo možnosť vymodelovať zložitejšie vzťahy medzi postavami, napr. súťaživosť, dominancia alebo vernosť. Tieto vzťahy potom slúžili ako predpoklady pre niektoré akcie a ako dôsledky pre ostatné akcie. Toto predstavovalo jednoduchý model motivácie postáv. Nakoniec charakterové črty postáv sa vygenerovali pomocou rôznych hodôt láskavosti, samolúbosti, úprimnosti a inteligencie.

2.4.3 Author

Ďalej si povieme o programe AUTHOR, ktorý mal simulovať myseľ autora ako vymýšľa príbeh. Premisa tohto programu bola, že prostredie príbehu je vytvorené zdôvodnením udalostí pre ktoré sa autor už rozhodol, že budú súčasťou príbehu. Autor sám môže mať, pri vytváraní príbehu, nejaké ciele, ktoré príbehom chce dosiahnuť. A aj ak nemá tak je všeobecne známe, že pri vytváraní príbehu existujú nejaké podvedomé obmedzenia a ciele tvorivého procesu. Pod týmto sa má na mysli napríklad, či je príbeh konzistentný, či je vierohodný, či postavy v príbehu sú uveriteľné alebo aj či pozornosť čitateľa je udržiavaná počas celého príbehu. Toto sa môže preniesť na nejaké podciele situácií v príbehu do ktorých chce autor zaviesť niektoré postavy z príbehu alebo do rolí ktoré jednotlivé postavy majú zahrať v

príbehu. Môžeme teda povedať, že príbeh sa dá chápať ako nejaký výsledok zložitej spleti autorových zámerov. Tieto zábery potom prispievajú ku štrukturovaniu príbehu a nejako riadia proces budovania príbehu. Avšak vo výslednom diele už niesu vidno.

2.4.4 Universe

O 2 roky neskôr prišiel systém UNIVERSE, ktorý generoval scenáre pre radu epizód do telenovely v ktorej hralo rolu veľa postáv a plynulo viaceo súčasne prebiehajúcich a prekrývajúcich sa dejov. UNIVERSE bol prvým storytelingovým systémom, ktorý venoval špeciálnu pozornosť na proces tvorby postáv. Komplexné dátové štruktúry boli použité na reprezentáciu postáv. Spolu s nimi bol aj použitý jednoduchý algoritmus, ktorý mal tieto štruktúry naplniť. Väčšina však ostala na používateľovi. Tento systém bol skôr prostriedkom na prieskum sériového generovania príbehu ako na generovanie príbehu so začiatkom a koncom. Pôvodne bol zamýšľaný ako pomôcka spisovateľom s nádejou na postupnú transformáciu na samostatný systém generovania príbehu. UNIVERSE priblížil otázku týkajúcu sa vytvárania fiktívneho sveta. Či by sa najprv mal vybudovať svet a potom do sveta vložiť príbeh alebo či by príbeh mal riadiť konštruovanie sveta, kde by sa postavy, lokality a rôzne objekty vytvárali podľa potreby. Tu nastáva rozdiel medzi UNIVERSE-om a AUTHOR-om. Tvorca systému UNIVERSE sa priklonil ku prvej možnosti a implementoval vytváranie postáv nezávisle od deja a na druhej strane bol tvorca AUTHOR-u ktorý sa prikláňal k druhej možnosti.

2.4.5 Minstrel

Ďalší v chronologickom poradí, ktorý si spomenieme je počítačový program MINSTREL. Tento program vytváral príbehy o Kráľovi Arturovi a jeho rytieroch okrúhleho stola. MINSTREL používal stavebné jednotky pozostávajúce z cieľov a plánov ako dosiahnuť dané ciele. Tieto fungovali na dvoch rôznych úrovniach: autorské ciele a ciele vystupujúcich postáv. Vytváranie príbehu fungovalo na základe procesu, ktorý mal dve štádiá, ktoré zahrnovali najprv plánovacie štádium a potom štádium riešenia problémov, ktoré navyše opätovne využívalo vedomosti z predošlých príbehov. Každý beh programu bol založený na nejakej vete, ktorá slúžila ako jadro

okolo, ktorého sa príbeh staval. MINSTREL vytváral príbehy dlhé pól strany až jedna strana.

2.4.6 Mexica

MEXICA je počítačový model, ktorý bol vytvorený na skúmanie kreatívneho procesu tvorby príbehov. Jeho úlohou bolo vytvárať krátke príbehy o pôvodných obyvateľoch Mexica. Autor tohto modelu sa riadil hypotézou tvorby príbehu na základe dvoch stavov. Prvý je stav, kedy spisovateľ je silno angažovaný do produkcie materiálu pre dané dielo. V tomto stave sa spisovateľ spolieha na predošlé skúsenosti iné mentálne schémy a zároveň sa vyhýba úmyselnému plánovaniu a rôznym dejovým štruktúram na vývoj diela. V druhom stave spisovateľ analyzuje a hodnotí to, čo doteraz napísal a uvážlivo to skúma a pozmieňa. A teda táto hypotéza hovorí, že cyklus medzi týmito dvoma stavmi tvorí dôležitú časť tvorby príbehu.

Podobne teda bol aj zkonštruovaný program tohto modelu. Pozostáva z dvoch hlavných súčastí. Prvá je založená na množine predchádzajúcich príbehov, ktoré sú definované používateľom pomocou textového dokumentu. Vytvorí sa skupina dátových štruktúr, ktoré reprezentujú abstraktné schémy príbehu. V druhej časti sa program odkazuje na štruktúry vytvorené v prvej časti a program prechádza dvoma stavmi z udanej hypotézy na vytvorenie príbehu. Tento model bol prevratný v tom, že využíval na poháňanie a hodnotenie príbehu aj emocionálne prepojenia a napätia medzi postavami.

2.4.7 Brutus

Ďalším zaujímavým programom, ktorý si spomenieme je BRUTUS. Tento program vytváral krátke príbehy o zrade. Zaujímavý bol v tom, že na tvorbu príbehu využíval logický model zrady. Tento model bol dosť rozsiahly na to, aby sa z neho dali vyvodzovať informácie, ktoré umožňovali tvorbu kvetnatých príbehov. Bol navrhnutý tak, aby vedel brať do úvahy aj veľké množstvo vedomostí o literatúre a gramatike. Príbehy, ktoré tento program vytváral obsahovali vlastnosti, aké by sme mohli nájsť v príbehoch vytvorených ľuďmi napr. rôzne literárne trópy a dialógy. Tvorcovia programu ale podotkli, že BRUTUS vôbec nie je tvorivý program je to skôr výsledok reverzného inžinierstva príbehu, na ktorom potom sledovali, či dokáže

vybudovať daný príbeh. Dalo by sa povedať, že to bolo skĺbenie kreativity a logiky.

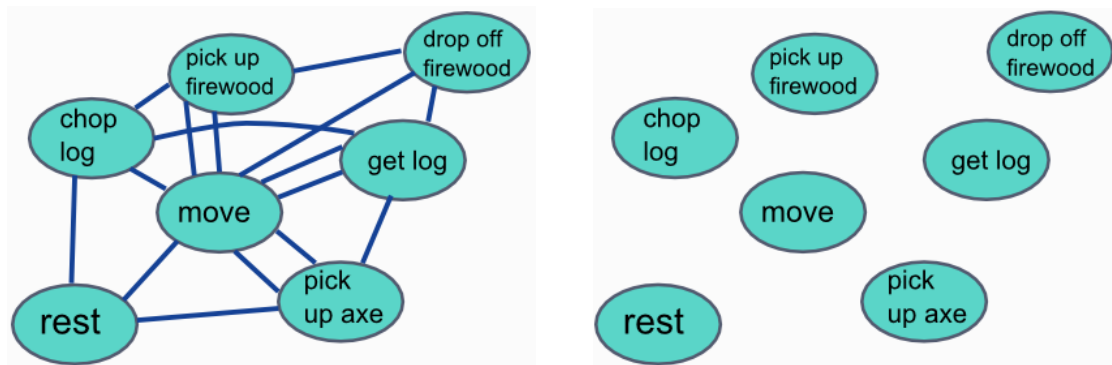
2.4.8 Fabulist

Najnovším spomedzi spomínaných algoritmov je architektúra FABULIST, ktorá automatizuje generovanie príbehu a jeho prezentáciu. Využíva techniku takzvaného upresňujúceho vyhľadávania na riešenie problému ohľadom generovania príbehu, na nájdenie zvučnej a uveriteľnej postupnosti akcií postáv, ktorá teda počiatočný stav pretransformuje na konečný. Túto techniku využíva IPOCL (Intent-based Partial Order Causal Link) plánovač, ktorý priniesli v tejto architektúre, ktorý okrem vytvorenia bežne znejúceho deja aj pojednáva o zámeroch postáv v príbehu identifikovaním možných cieľov pre danú postavu, ktoré vysvetľujú správanie postavy a následne vytvára plánovacie štruktúry, ktoré vysvetľujú prečo sa tieto postavy zameriavajú na ich cieľ.

2.5 Goal Oriented Action Planning

Goal Oriented Action Planning (GOAP) teda cieľovo orientované plánovanie akcií, je systém umelej inteligencie, ktorý ľahko zabezpečí možnosti a nástroje na rozumné rozhodovanie sa pre agentov bez toho, aby sme museli udržiavať veľký a zložitý stavový automat. Umožňuje agentom naplánovať postupnosť akcií, ktoré dosiahnu ich cieľ. Táto postupnosť nezávisí len od cieľa ale aj od aktuálneho stavu sveta a stavu agenta. To znamená, že ak by sme dali rovnaký cieľ dvom rôznym agentom alebo dvom agentom v rôznom stave sveta, môžeme dostať úplne odlišné postupnosti akcií, čo robí umelú inteligenciu viac dynamickú a realistickú.

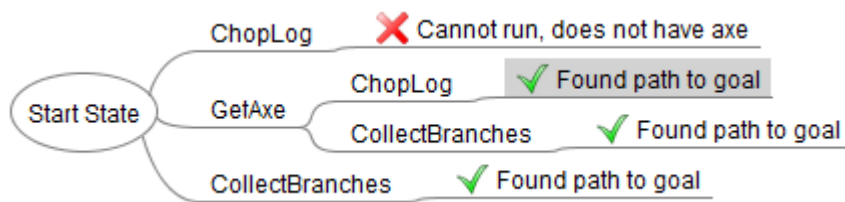
Tento systém vyniká v situáciách, kedy by nám vznikali pri riešení problému vznikaly veľké a komplikované stavové automaty, kde sa často stáva, že so dôjde do bodu, kedy už nechceme pridávať nové akcie lebo by to vyvolalo priveľa vedľajších efektov a medzier v umelej inteligencii.



Obr. 2.2: Rozdiel medzi reprezentáciou pomocou stavového automatu a GOAP-u

GOAP teda umožňuje z ľavého obrázka spraviť pravý, kde sú akcie samostatnou jednotkou a nezaoberajú sa vedľajšími efektami svojej činnosti. Tým, že sme jednotlivé akcie odpojilim, tak sa môžeme venovať každej zvlášť. Toto robí kód viac modulárnym a ľahšie sa testuje a udržiava. Ak by sme chceli pridať novú akciu, tak ju nám stačí pridať do našej reprezentácie nijako to neovplyvní ostatné akcie, čo sa o reprezentácii stavovým automatom povedať nedá. Navyše rôzne akcie môžeme agentom pridávať a odoberať za behu. Aby sme pomohli GOAP-u zistiť aké akcie chceme vykonať, každá akcia dostane svoju cenu. Akcia s vysokou cenou nevyberieme namiesto akcie s cenou nižšou. Keď akcie usporiadame do postupnosti, sčítame hodnoty jednotlivých akcií, aby sme nakoniec mohli vybrať postupnosť s najnižšou cenou.

O toto sa stará GOAP plánovač, ktorý sa pozerá na predpoklady a dôsledky akcií a vytvára frontu akcií, ktoré spĺňajú náš cieľ. Cieľ je poskytnutý agentom spolu s stavom sveta v ktorom sa agent vyskytuje a zoznamom akcií, ktoré agent môže vykonať. S týmito informáciami potom plánovač môže skúšať vykonávať akcie, uvidí, ktoré sa dajú vykonať a ktoré nie a potom sa rozhodne, ktoré akcie je najlepšie vykonať. Plánovač neskončí hneď ako nájde prvú postupnosť, ktorá nás privedie do cieľa ale prejde aj ostatnými akciami, pretože sa môže stať že existuje iná postupnosť, ktorá má nižšiu cenu ako tá prvá. Teda prejde cez všetky možnosti aby našiel tú najlepšiu. Pri plánovaní sa buduje strom. Vždy keď plánovač použije nejakú akciu, tak ju vyhodí zo zoznamu akcií aby sa nestalo, že sa zacyklíme a chceme vykonávať tú istú akciu dookola. Následne sa zmení stav podľa dôsledku akcie. Keby sme mali za úlohu napr. získať drevo na oheň, tak by plánovací strom mohol vyzeráť nasledovne:



Obr. 2.3: Plánovací strom GOAP plánovača

2.6 Version Control System

Je dobrou praktikou pri tvorení akéhokoľvek projektu, používať Version Control System. Jeden z takýchto systémov je Github. V tomto systéme vieme uchovávať rôzne verzie projektu, pracovať na vylepšeniach, ktoré môžeme testovať nezávisle od hlavného projektu, komentovať tieto vylepšenia a v prípade nejakej chyby nás zachráni staršie verzie. Systém je vhodný aj na prácu viacerých ľudí na tom istom projekte, pretože môžu pracovať nezávisle na sebe a následne vložiť novú funkcionality do hlavného projektu tak aby to napríklad spolupracovníci museli schváliť. Teda všeobecne takýto systém sprehľadňuje vývoj projektu.

2.7 Predošlé bakalárske práce

Pri uvedení do problematiky a tvorbe bakalárskej práce mi boli nápomocné dve predošlé bakalárske práce. A to Tvorba interaktívnych príbehov ako plánovanie od Martina Zvaru a Tvorba zápletiiek v dobrodružných hrách od Dominika Dobiáša.

Prvá spomenutá bakalárska práca je podobná mojej v tom, že tiež využíva plánovanie na generovanie príbehov. Ďalej ako základ používa zformalizovaný Proppov model ktorý je zapísaný ako plánovací problém a naprogramovaný ako logický program pomocou ASP. A na samotné generovanie príbehov bol použitý solver Clingo, ktorý bol spúšťaný pomocou Javy. Jeden z rozdielov v mojej práci je skutočnosť že tu sa bude využívať čisto Java a nie logické programovanie. Potom na riešenie plánovania sa použije vlastný solver. Taktiež v spomenutej práci sa nehľadí na zaujímavosť generovaného príbehu, čo sa v tejto práci riešiť bude. A nakoniec tvorba príbehov v tejto práci bude zasadená do jednoduchej textovej hry, kdežto v práci od pána Zvaru išlo čisto o program ktorý generuje príbehy.

Druhá bakalárska práca, ktorú som spomenul je mojej viac podobná ako prvá. A to napríklad v programovacom jazyku, tiež má vlastný solver a taktiež rieši problematiku zaujímavosti príbehu. Ale nepoužíva GOAP plánovač, a tiež generovanie príbehu nebolo vyjadrené v hre.