

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Editor pro tvorbu gamebooků**

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 4. května 2015

Jakub Sido

# Abstract

In gamebooks the reader is the one who controls the storyline. This work deals with the possibilities of transforming the classic gaming books into electronic form. It is focused on creating a game system, which is able to capture the complexity of those works, as well as the development of the program - editor where is possible to create, edit and then save those often large books for distribution. The first part analyzes the elements that are used in this field and describes the system created by me. The second part describes the design of the editor and the manner of its implementations, including verification of its functionality.

# Abstrakt

Herní knihy jsou takové knihy, ve kterých čtenář řídí dějovou linii příběhu. Tato práce se zabývá možnostmi přesunutí klasických herních knihy do elektronické podoby. Je zaměřena na tvorbu herního systému, který dokáže obsáhnout komplexnost těchto děl, dále pak na vývoj programu - editoru, ve kterém bude možné tyto mnohdy rozsáhlé knihy vytvářet, editovat a následně uložit pro distribuci. V první části jsou analyzovány prvky, které jsou v tomto oboru používány, následně je popsán mnou vytvořený systém. Druhá část popisuje návrh editoru a způsob jeho implementace včetně ověření jeho funkcionality.

# Poděkování

Rád bych poděkoval Ing. Ladislavu Pešíčkovi za jeho trpělivost a cenné poznámky, které mi velmi pomohly při jejím vypracování. Dále pak svým kamarádům a blízkým za jejich pohled na mou práci z jiné perspektivy a odstupem.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Studie Gamebooku</b>	<b>2</b>
2.1	Gamebook . . . . .	2
2.1.1	Role . . . . .	2
2.1.2	Příklady gamebookových knih . . . . .	3
2.2	Stolní hry na hrdiny . . . . .	4
2.2.1	Role . . . . .	4
2.2.2	Příklady stolních her na hrdiny . . . . .	5
2.3	Herní prvky . . . . .	5
2.3.1	Gamebook . . . . .	5
2.3.2	Hry na hrdiny . . . . .	7
<b>3</b>	<b>Dekompozice a tvorba herního systému</b>	<b>8</b>
3.1	Stav hry . . . . .	8
3.2	Holder . . . . .	9
3.3	Postava . . . . .	9
3.4	Akce a jiné herní mechanismy . . . . .	9
3.4.1	Prostá akce . . . . .	9
3.4.2	Náhodná akce . . . . .	9
3.4.3	Akce závislá na vlastnosti . . . . .	10
3.4.4	Bojová akce . . . . .	12
3.4.5	Akce Proměnné - Milník . . . . .	14
3.4.6	Counter . . . . .	15
3.4.7	Selector . . . . .	17
<b>4</b>	<b>Datové struktury Gamebooku</b>	<b>21</b>
4.1	Přehledy nejdůležitějších objektů . . . . .	21
4.1.1	Graph . . . . .	21
4.1.2	State . . . . .	21
4.1.3	Action . . . . .	22

---

<b>5</b>	<b>Návrh Editoru Gamebooků</b>	<b>25</b>
5.1	Data . . . . .	25
5.2	Řídící logika . . . . .	26
5.2.1	libGraph . . . . .	26
5.2.2	libAkce . . . . .	26
5.2.3	libEditor . . . . .	26
5.3	GUI - Grafické uživatelské rozhraní . . . . .	27
5.4	Start aplikace . . . . .	27
5.4.1	Třída GUI . . . . .	27
5.4.2	Třída Drawarea . . . . .	27
5.5	Ostatní okenní prvky . . . . .	29
5.5.1	Okna tvorby Selectorů . . . . .	30
5.5.2	Okno tvorby Counterů . . . . .	30
5.6	Lokalizace . . . . .	32
<b>6</b>	<b>Ukládání Gamebooku</b>	<b>33</b>
6.1	Struktura dat . . . . .	33
6.1.1	INFO . . . . .	34
6.1.2	GRAF . . . . .	34
<b>7</b>	<b>Simulátor</b>	<b>37</b>
7.1	Data . . . . .	37
7.2	Zobrazení . . . . .	37
7.3	Simulátor v Editoru . . . . .	38
<b>8</b>	<b>Testování</b>	<b>39</b>
8.1	Jednotkové testy . . . . .	39
8.2	Testování autory gamebooků . . . . .	39
8.3	Souhrn testovacích metod a výsledků . . . . .	40
<b>9</b>	<b>Závěr</b>	<b>41</b>

# 1 Úvod

Cílem projektu je poskytnout vhodný nástroj pro vývoj *gamebookové hry* a následně jej přesunout do moderního světa.

To zahrnuje analyzování existujících gamebooků, vytvoření vhodného herního systému a odpovídajících datových struktur pro jeho reprezentaci. Tyto datové struktury dále logicky propojit.

Díky přesunutí příběhu z papíru do zařízení s výpočetním výkonem je možné jisté mechanismy ze světa *stolních her na hrdiny* přidat i do světa gamebooku, bez újmy na charakteru původního typu zábavy. Je tedy také potřeba analyzovat tyto možnosti a zakomponovat je do vytvářeného systému.

Cílem bude dále navrhnout editor, který bude uživatelsky přívětivý, přehledný a dokáže tyto mnohdy rozsáhlé příběhy vyvářet a ukládat tak, aby bylo možné je dále distribuovat do různých platforem. Také umožní autorovi gamebooku příběh ověřit jeho procházením tak, jak to později budou dělat čtenáři.

## 2 Studie Gamebooku

Tato kapitola vysvětluje, co jsou gamebooky. Dále popisuje podobné hry mezi deskovými hrami.

### 2.1 Gamebook

Gamebooky jsou knihy popisující zpravidla příběh jedné hlavní postavy. Čtenář, hlavní postava, je uveden do světa, ve kterém se zrovna nachází a většinou je postaven do nějaké problematické situace, kterou musí řešit. V příběhu je spousta klíčových rozhodnutí, která příběh větví do více dějových linií. V takovém případě je na čtenáři, kterou dějovou linii bude následovat. O tom rozhodne tak, že vybere jednu z několika možností volby, kterou pro něj připravil autor knihy. Čtenář je poté přesměrován na jinou stránku, kde jeho příběh pokračuje.

#### 2.1.1 Role

Mezi základní role, které se účastní životního cyklu knihy, patří:

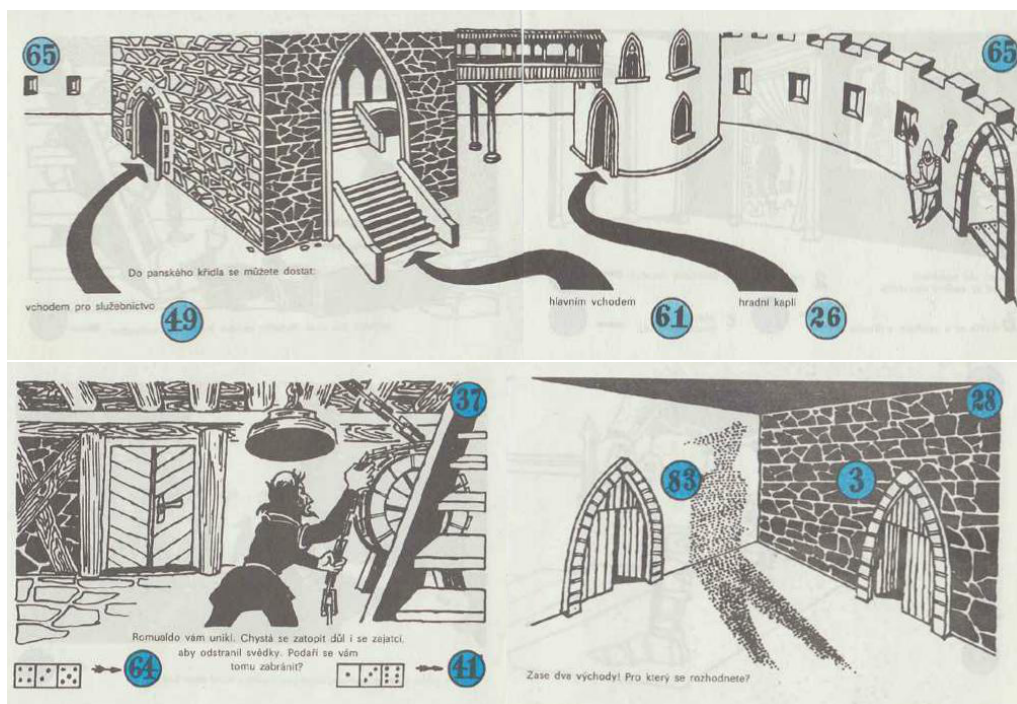
##### **Autor**

Je člověk nebo kolektiv, který utváří příběh. Je zodpovědný za to, jak je rozsáhlý a také kolik volnosti, při prožívání svého příběhu, čtenář dostane.

##### **Čtenář**

Pouze čte (prožívá) vytvořený příběh tím, že se pohybuje v jasně stanoveném, předem definovaném prostoru.





Obrázek 2.1: Ukázka z knihy Na rozkaz krále (1989, Ljuba Štíplová)

## 2.1.2 Příklady gamebookových knih

Z české literatury například:

- Na rozkaz krále (1989, Ljuba Štíplová) [2]  
Tato kniha je určena spíše pro mladší čtenáře. Král v úvodu žádá čtenáře o pomoc s vyřešením nekalostí, které se odehrávají na místním hradě. Kniha se skládá spíše z krátkých textů doplněných o rozsáhlé a věcné ilustrace (viz obr. 2.1 str. 3). Mnohdy za čtenáře rozhoduje náhoda.
- Kosmická hlídka (1991, Ljuba Štíplová)
- O Norikovi (Vladimír Chvátil)

Ze zahraniční literatury například :

- The Forbidden Towers (1985, Carol Gaskin)  
Do češtiny přeloženo roku 2004 [1]. Příběh se odehrává v pěti zapovězených věžích, kde hlavní hrdina, mladý elf, hledá způsob záchrany vesnice před nemocí. Tato kniha je typickým zástupcem nejjednodušších gamebooků. Ke hře

není potřeba kostka a hráč vždy pouze volí z několika možností. Příběh je dle mého názoru spíše zmatený a bez hlubšího smyslu. Je psaný tak, aby bylo jednoduché jej vytvořit, je toho docíleno tím, že hráč prochází jednotlivými místnostmi, které na sebe příliš nenavazují. Pokud je hráč v úzkých, má vždy na výběr "lousknout prsty". To ho přemístí do jiné místnosti většinou i jiné věže a jeho příběh pokračuje dál.

- Pretty Little Mistakes: A Do-Over Novel (2007, Heather McElhatton)
- Life's Lottery (1999, Kim Newman)
- If (2014, Nicholas Bourbaki)

## 2.2 Stolní hry na hrdiny

Tyto hry jsou velmi podobné klasickým gamebookům a jsou populární po celém světě už několik desetiletí. Jsou to společenské hry, při kterých se schází větší skupinky lidí, aby se společně ponořily do světa fantazie. Jeden z hráčů je vyčleněn pro tvorbu příběhu a je vlastně průvodcem celou hrou. Vymýšlí nebo interpretuje příběh a bere na sebe role postav, které hráči ve světě potkávají. Pokud je součástí hry boj, zastupuje protistranu. Přijímá názory a činy hráčů. Vyhodnocuje jejich následky dle svého nejlepšího uvážení, popřípadě je vyhodnocuje dle pravidel, která hráči přijali.

### 2.2.1 Role

Mezi základní role, které se účastní herního sezení, patří:

#### **Průvodce**

Předem připravuje, nebo interpretuje, již existující příběh. Během hry v reálném čase reaguje na námitky a přání hráčů a upravuje příběh tak, aby zajistil plynulost a spád.

#### **Hráč**

Účastní se hry, přejímá na sebe roli jedné konkrétní postavy a řídí její činnost. Pohybuje se ve volném světě. To znamená, že její činy jsou omezeny pouze schopnostmi postavy, kterou představuje.

## 2.2.2 Příklady stolních her na hrdiny

Jako příklady lze uvést:

- Dračí doupě  
Tuto hru vydává nakladatelství Altar[3]. Bylo vydáno několik knih, které definují pravidla - Pravidla pro začátečníky, Pravidla pro pokročilé, Pravidla pro experty a spousta dalších. Je to velmi sofistikovaný systém. Vyšlo také několik knih popisujících konkrétní dobrodružství, které mohou průvodci používat a hledat v nich inspiraci[10].
- Příběhy Impéria  
Tato hra zastřešená nakladatelstvím Mytango se odehrává v 19. století. V současné době jsou vydané tři knihy, které popisují herní systém. Zahrnují několik příkladových příběhů, ras, bytostí i ilustrací [11].
- Legendy Armandie  
Volně šiřitelný český herní systém vyvíjený od roku 1997 [9].

## 2.3 Herní prvky

Mnohdy se pro zpestření příběhů používá několik herních prvků, neboli herních mechanismů, jimiž se do příběhu dá přidat například role náhody. Nějaká akce může být závislá na dovednostech postavy, jiná na schopnostech a pozornosti čtenáře.

### 2.3.1 Gamebook

V následující kapitole jsou podrobněji rozděleny herní prvky v Gameboocích.

#### Postava

V každém gamebooku je čtenář postavou v herním světě. Někdy není nutné, aby byla specifikována něčím jiným, než jejím původem, sděleným čtenáři před začátkem dobrodružství. Jindy ale může být vyžadován podrobnější profil. Existují herní mechanismy, které jsou vyhodnocovány na základě konkrétních vlastností postavy. Takovými vlastnostmi může být například síla, inteligence, charisma, obratnost a spousta dalších. Některé gamebooky umožňují postavě nosit zbraně nebo jiné předměty a tím upravovat její vlastnosti.

## Vlastnosti v příběhu

Na vlastnosti postavy může být navázán herní mechanismus. Předpokládá, že je v postavě definován výčet vlastností, na jejichž základě se bude daná situace vyhodnocovat. Vyhodnocení probíhá většinou tak, že se numerická hodnota vlastnosti, které se daná situace týká, vynásobí nějakým koeficientem, popřípadě je k ní ještě nějaké číslo přičteno. Úspěšnost hráče se vyhodnotí podle toho, jestli je výsledné číslo větší nebo menší než daná konstanta. Také může být vyhodnocení ještě doplněné o hod kostkou, aby byla započtena ještě role náhody.

## Náhoda

V gameboocích je náhoda simulována hodem kostkou. Čtenář, pokud měla být vyhodnocena náhoda, kostkou vrhnul. Podle toho, co padlo, byl přesměrován na jinou stránku.

## Dovednost čtenáře

V průběhu příběhu mohla být čtenáři poskytnuta informace, kterou mohl později potřebovat. Pokud ji čtenář zapomněl, mohlo to mít následky. Pokud například v průběhu putování hráči nějaká postava sdělí informaci, že na konkrétním rozcestí musí jít vlevo, musí si tuto informaci hráč pamatovat. Pokud ji zapomene, může zabloudit.

## Boj

Boj je jistým způsobem modifikovaný mechanismus vyhodnocování vlastností postavy. Z vlastností postavy (někdy i z vybavení) jsou vypočítávány další odvozené vlastnosti, které už se přímo využívají k vyhodnocování boje. V různých gameboocích se tyto odvozené vlastnosti mohou počítat rozdílně. Většinou ale vedou k číslům s podobným významem. Jedním bývá číslo vyjadřující schopnost postavy zranit nepřítele a druhé číslo vyjadřuje možnost, že bude postava zraněna nepřítelem.

K těmto číslům se následně přičítají hody na kostkách obou stran v několika kolech, v jejichž průběhu jsou obě strany zraňovány protistranou.

Boj končí vítězstvím nebo prohrou, popřípadě může být zapojen útek z boje.

Prohra může znamenat smrt postavy, její upadnutí do bezvědomí, nebo zranění

a z něj plynoucí hendikep.

Výhra může být smrt protivníka, nebo i to, že protistrana je tak zaskočená silou svého oponenta, že se vzdá.

Význam výhry nebo prohry určuje autor příběhu.

### **2.3.2 Hry na hrdiny**

V následující kapitole jsou podrobněji rozděleny herní prvky ve stolních hrách na hrdiny.

#### **Postava**

Postava ve hře na hrdiny zahrnuje všechny prvky popsané výše. Protože je ale svět typicky větší a rozsáhlejší, přidávají se další charakteristiky k postavě. Například možnost výběru rasy, povolání, zaměření, vyznání atp.

#### **Intelligence a paměť průvodce**

Znalost historie postavy a schopnost nad ní logicky přemýšlet umožňuje průvodci tvořit reálnější příběh.

## 3 Dekompozice a tvorba herního systému

V následující kapitole je rozvržen mnou navržený herní systém, který je přizpůsoben pro interpretaci těchto her v moderním světě výpočetní technikou. Tento systém vznikl na základě mých zkušeností s těmito hrami a následnou zpětnou analýzou celistvosti systému (systém se skládá z primitiv, kterými jde namodelovat jakákoli situace, kterou by autor chtěl vytvořit).

Pro vyhodnocování některých mechanismů bude použito generování náhodného čísla. Pro snadnou interpretaci čtenáři jsou všechny náhodné prvky realizovány v rozsahu běžných hracích kostek. Pokud jsou taková čísla součástí vyhodnocení, bude uvedena pouze zkrácená funkce. Například  $F_n(6)$  znamená jeden hod kostkou o 6 stěnách. Obdobně  $F_n(10)$  je hod kostkou desetistěnnou.

Celý příběh je tvořen množinou textů, které se vztahují k jednotlivým situacím, a množinou akcí, pro které se může čtenář rozhodovat. Je tedy možné reprezentovat jej jako přechodový graf.

Jednotlivé uzly grafu budou představovat stavy hry a hrany budou Akce, pro které se čtenář může rozhodnout. Pokud čtenář zvolí z nabízených akcí a následně dojde k vyhodnocení jedné z nich, přesune se postava do nového stavu. Tuto událost budeme také nazývat příchodem postavy do nového stavu.

### 3.1 Stav hry

Představuje aktuální neúplný stav hry, tedy popis okolí a situaci, ve které se postava právě nachází (neobsahuje informace o postavě, proto jen neúplný stav hry).

Každý stav hry musí obsahovat text, který se čtenáři zobrazí.

Aby bylo možné implementovat všechny požadované herní mechanismy, musí stav umožňovat do postavy uložit informaci, která umožní následné vyhodnocení akce se znalostí toho, že se postava za dobu své existence v herním světě v tomto stavu již nalézala. (Akce vázaná na Counter nebo proměnnou)

Dále je nutné, aby uchovával výčet akcí, které budou čtenáři nabízeny.

## 3.2 Holder

Je speciálním případem stavu. Tento stav neobsahuje text, který se zobrazí čtenáři. Po příchodu postavy do Holderu se hráči nic nezobrazí a okamžitě se vyhodnotí akce vedoucí z Holderu (Proto je povolena pouze jedna akce vázající se k Holderu). Následně je hráč přesunut do dalšího stavu v závislosti na výsledku akce. Takto lze provést libovolný počet pro hráče transparentních přechodů. Tento zdánlivě triviální mechanismus umožní autorům vytvářet neomezené vyhodnocovací mechanismy z více na sebe plynule navazujících akcí.

## 3.3 Postava

Obsahuje informace o aktuálním stavu postavy, tedy počty životů, aktuální hodnoty vlastnosti, rasu atp. Stav hry společně s Postavou obsahuje úplný stav hry.

## 3.4 Akce a jiné herní mechanismy

Obecně je akce činnost, kterou čtenáři nabízíme v konkrétním stavu.

Každá akce musí obsahovat základní informace, a to hlavně popis, který se zobrazí čtenáři pro výběr akce, a orientační souhrnný popis pro autora.

### 3.4.1 Prostá akce

Tento typ nejjednodušší akce slouží pro přesun z jednoho stavu do druhého. Není třeba nic vyhodnocovat. Kromě základních informací, které musí obsahovat každá akce, má navíc pouze index stavu, do kterého se přesune hráč po výběru této akce.

### 3.4.2 Náhodná akce

Slouží k přidání prvku náhody při vyhodnocení nějaké akce. Je potřeba, aby uchovávala navíc informaci o pravděpodobnosti úspěchu vykonání dané akce. Také nestačí pouze jeden stav, ale je třeba uchovat index dvou stavů. Podle toho, jestli bude nebo nebude "mít štěstí", se hráč přesune do jednoho z nich.

### Shrnutí potřebných atributů

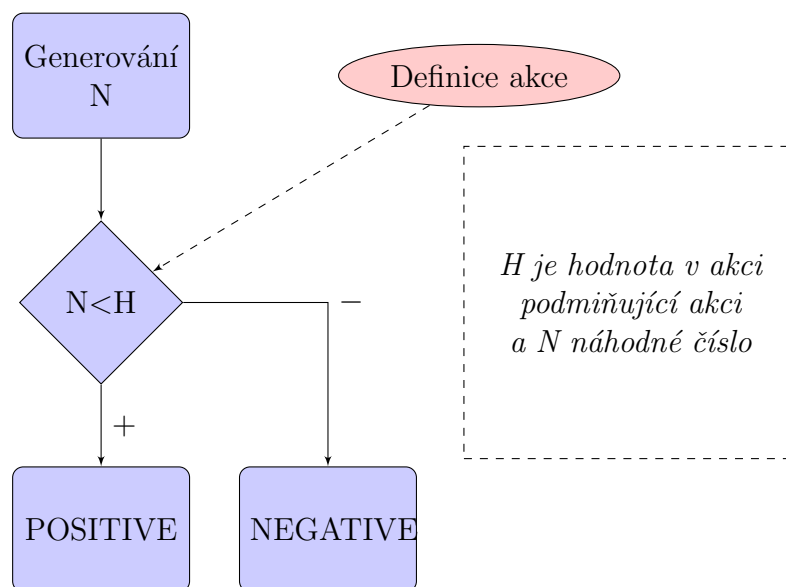
- pravděpodobnost úspěchu
- odkazy na dva stavy

### Vyhodnocení

Vyhodnocení probíhá porovnáním **náhodného čísla** v rozmezí 0-99 s **hranicí úspěšnosti** uvedenou v definici konkrétní náhodné akce. (viz obr. 3.1 str. 10)

#### Náhodné číslo vygenerované pro hráče

$$N = 10 \cdot F_n(10) + F_n(10) \quad (3.1)$$



Obrázek 3.1: Diagram vyhodnocení náhodné akce

### 3.4.3 Akce závislá na vlastnosti

Původní koncept používaný v gameboocích je těžké jednoduše reprezentovat autorovi tak, aby dokázal snadno navrhnout hranici úspěšnosti. Aby se zjednodušilo



nastavování takové hranice a abychom mohli oprostít autora od složité matematiky, budeme takové akce modelovat jakousi průměrnou úspěšností.

Je tedy nutné uchovat pouze procentuální pravděpodobnost na úspěch průměrně schopné postavy a vlastnost, na které je akce závislá. O konkrétní čísla použítá v dané situaci se postará matematický přepočít. Dále musí obsahovat opět informaci o dvou stavech. Jeden pro úspěch, druhý pro neúspěch.

### Shrnutí potřebných atributů

- pravděpodobnost úspěchu pro průměrně schopnou postavu
- vlastnost, na které je tato akce závislá
- odkazy na dva stavy

### Vyhodnocení

Vyhodnocení probíhá následovně: Nejdříve spočítáme **odchylku** konkrétní vlastnosti postavy od průměrné hodnoty

$$O = V_p - \frac{V_{max}}{2} \quad (3.2)$$

Kde  $V_p$  je konkrétní hodnota závislé vlastnosti postavy a  $V_{max}$  její teoretická maximální hodnota.

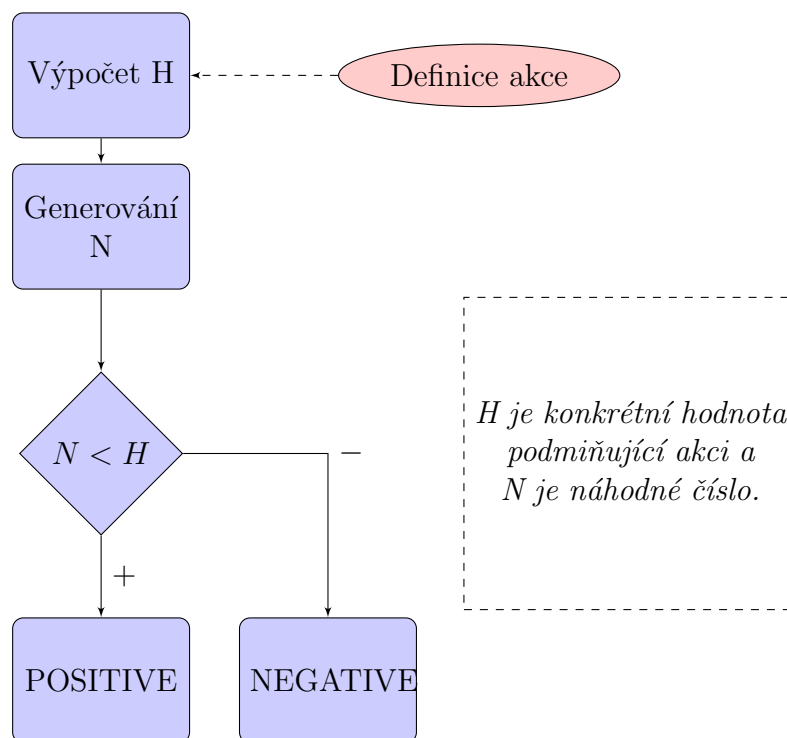
Následně přičteme **odchylku**  $O$  k **průměrné hodnotě**  $P$  uvedené v definici akce. Tím získáme konkrétní hodnotu hranice. Tu následně použijeme pro vyhodnocení porovnáním.

$$H = P + O \quad (3.3)$$

Dále vytvoříme **náhodné číslo** generováním čísel v intervalu 0 až 99, což odpovídá hodu dvěma desetistěnnými kostkami, kde jedna má význam desítek a druhá jednotek.

$$N = 10 \cdot F_n(10) + F_n(10) \quad (3.4)$$

Následným porovnáním čísel  $H$  a  $N$  provedeme samotné vyhodnocení (viz obr. 3.2 str. 12)



Obrázek 3.2: Diagram vyhodnocení akce závislé na vlastnosti

### 3.4.4 Bojová akce

Jak bylo již řečeno, v mnoha herních systémech se používá spousta způsobů vyhodnocení boje. Vždy ale v boji vystupuje útočné a obranné číslo účastníků obou stran. Aby bylo možné obsáhnout co nejširší spektrum, ale zároveň zůstal systém jednoduchý, jak pro autory, tak pro čtenáře, byl boj rozdělen na boj střelecký a boj zblízka. Obranné a útočné číslo se podle typu boje počítají různě.

Při tvorbě nepřátel ale nechceme autora nutit ke složitému počítání konkrétních čísel. Navíc pokud autor bude tvořit nepřátele pro boj, není třeba, aby nastavoval vlastnosti, které jsou pro boj nepotřebné. Není tedy vhodné, aby postavy, účastníci se boje jako nepřátelé, byly uloženy s konkrétními vlastnostmi. Autor tedy bude pouze modelovat základní tři složky, které jsou při boji potřeba.

- **Schopnost zranit protivníka:** čím je tato složka větší, tím větší pravděpodobnost má nepřítel zasáhnout hráče a způsobit mu zranění.
- **Pravděpodobnost být zraněn protivníkem:** s touto složkou se zvyšuje šance uniknutí hráčově ráně bez zranění.

- **Schopnost odolávat zraněním:** bude-li nepřítel zasažen ránou hráče a zároveň je tato složka malá, je velká pravděpodobnost, že bude zásah velmi zničující.

Boj může končit výhrou, prohrou, popřípadě útekem.

### Shrnutí potřebných atributů

- typ boje (zblízka/střelecký)
- výčet protivníků (včetně hodnot všech základních tří složek u každého z nich)
- odkazy na dva stavy (tři, pokud je zahrnuta možnost útěku)

### Vyhodnocení

Vyhodnocení probíhá po kolech (viz obr. 3.3 str. 15). V každém kole se vytváří **UC (Útočné číslo)** a **OC (Obranné číslo)** hráčovy postavy a protivníka (viz tab. 3.1 str. 13).

Na začátku každého kola se vygeneruje náhodné číslo pro každého ze zúčastněných. Ta určují pořadí hráčů v daném kole. Ten, kdo je na řadě, si vytvoří své **Útočné číslo** a obět' své **Obranné číslo**. A spočte se **zranění** :

$$Z = \min(OC - UC, 0) \quad (3.5)$$

Pokud je tedy UC větší než OC, jejich rozdíl je výsledné zranění. To se následně oběti odečte (resp. přičte). Na tahu je další hráč.

	Boj zblízka	Střelecký souboj
Útočné číslo (UC)	$V_{sila} + F_n(6)$	$V_{obratnost} + F_n(6)$
Obranné číslo (OC)	$V_{odolnost} + F_n(6)$	$V_{obratnost} + F_n(6)$

Tabulka 3.1: způsob tvorby Obranného a Útočného čísla

Konkrétní hodnoty **UC, OC** protivníků se počítají pomocí stejných vztahů, které byly použity pro výpočet u postavy. Hodnoty vlastností získáme pomocí koeficientů ve složkách modelujících nepřátele.

Pro všechny vlastnosti vystupující v boji platí stejný přepočítání. Nejdříve vypočítáme **bonus/postih**

$$P = \left[ (St_u - 2) \cdot \frac{K_{max}}{3} \right]; \quad (3.6)$$

Kde  $St_u$  je stupeň obtížnosti a  $K_{max}$  je maximální hodnota na použité kostce

$St_u$ (stupeň vlastnosti nepřítele)	0	1	2	3	4
P (rozdíl vzhledem k postavě)	-4	-2	0	2	4

Tabulka 3.2: Jednotlivé hodnoty postihu/bonusu pro stupně obtížnosti vlastností (podle vz. 3.6)

A následně spočítáme hodnotu konkrétní vlastnosti protivníka.

$$V_n = V_h + P \quad (3.7)$$

Kde je  $V_h$  je hodnota stejné vlastnosti hráče

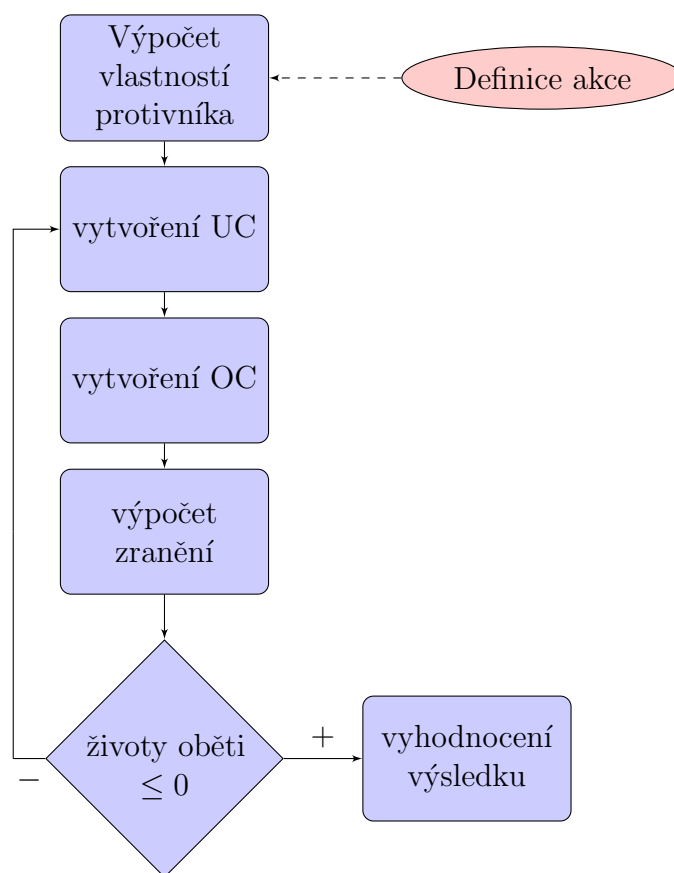
### 3.4.5 Akce Proměnné - Milník

Pro simulaci paměti průvodce byl zaveden jednoduchý mechanismus, který umožní do postavy uložit informaci, že byla přítomna za dobu své existence v nějakém stavu. Je to tedy jakýsi milník. Následně na základě existence/neexistence této informace v postavě rozvětví příběh.

Je třeba rozdělit tento mechanismus na dvě části. V první je do stavu přidána definice nějaké proměnné, která má být uložena do postavy. Druhou částí je akce, která se váže k nějaké dříve definované proměnné a podle její existence přesune hráče do jednoho, nebo druhého stavu.

#### Shrnutí potřebných atributů

- při definici klíčové události
  - identifikace proměnné, která identifikuje milník
- při opětovném rozdělení příběhu
  - identifikace proměnné, ke které se akce váže
  - odkazy na dva stavy



Obrázek 3.3: Diagram vyhodnocení boje

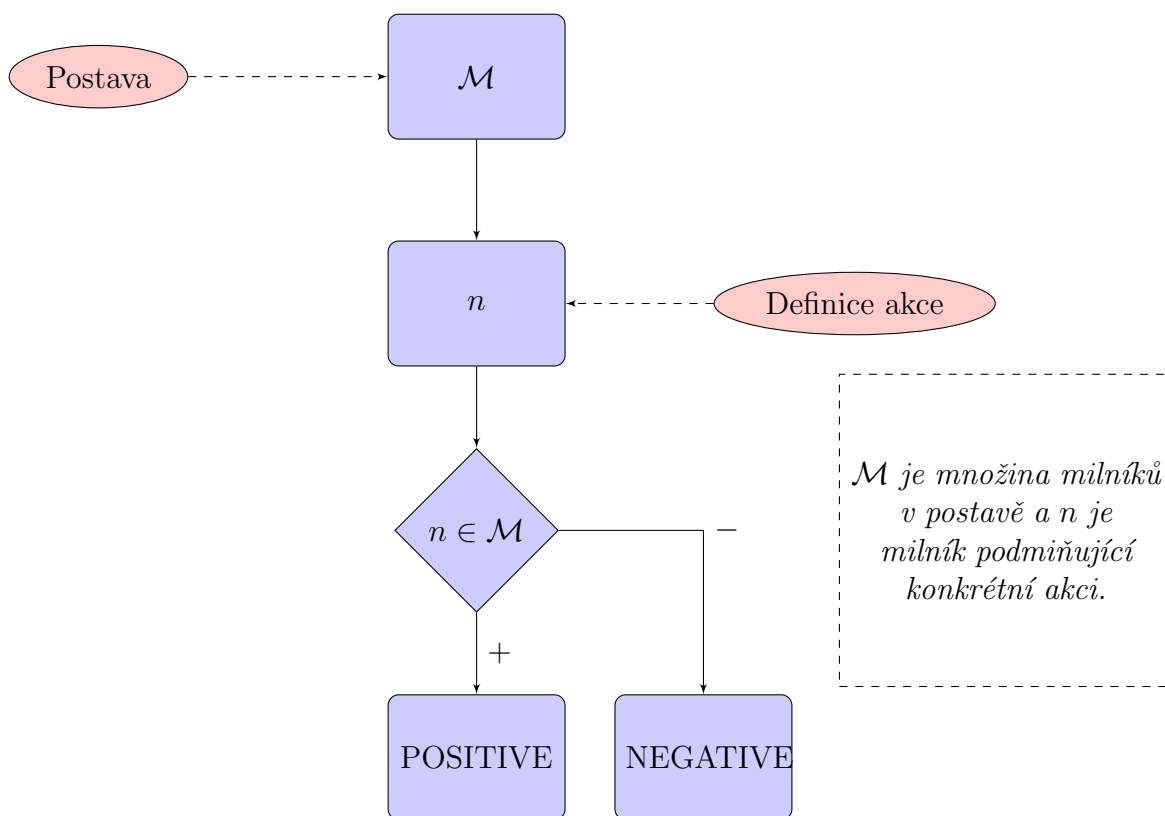
## Vyhodnocení

Vyhodnocení probíhá ověřením existence konkrétní proměnné vázané k akci v seznamu proměnných, kterou s sebou nese postava. Pokud konkrétní proměnná v seznamu je, hráč je přesunut do prvního z uvedených stavů. V opačném případě do druhého. 3.4

### 3.4.6 Counter

Pro další možnost simulace přítomnosti průvodce bylo navrženo počítadlo (Counter).

Proměnná, která bude zaznamenávat číslo, se uloží přímo do postavy. Autorovi tedy umožníme vytvářet proměnné pro jeho potřebu. Na začátku příběhu vytvoříme všechna počítadla, která bude autor používat. Pokud stav obsahuje informaci o tom,



Obrázek 3.4: Diagram vyhodnocení milníku

že má být nějaké počítadlo upraveno, po příchodu postavy do stavu nad ním provede matematickou operaci.

Na druhé straně je akce, která je reakcí na stav počítadla. Musí obsahovat navíc informaci o počítadlu, na které reagujeme, a jednoduchou podmínku. Dále dva indexy stavů. Jeden je pro případ, že podmínka je splněna, druhý pro případ, že není.

### Shrnutí potřebných atributů

- při tvorbě počítadla.
  - jméno pro počítadlo a jeho popis
  - počáteční hodnota
- při úpravě počítadla v průběhu příběhu.
  - identifikace počítadla
  - číslo a typ matematické operace
- při reakci na počítadlo
  - identifikace počítadla, které se k akci váže
  - vyhodnocovací podmínka
  - odkazy na dva stavy

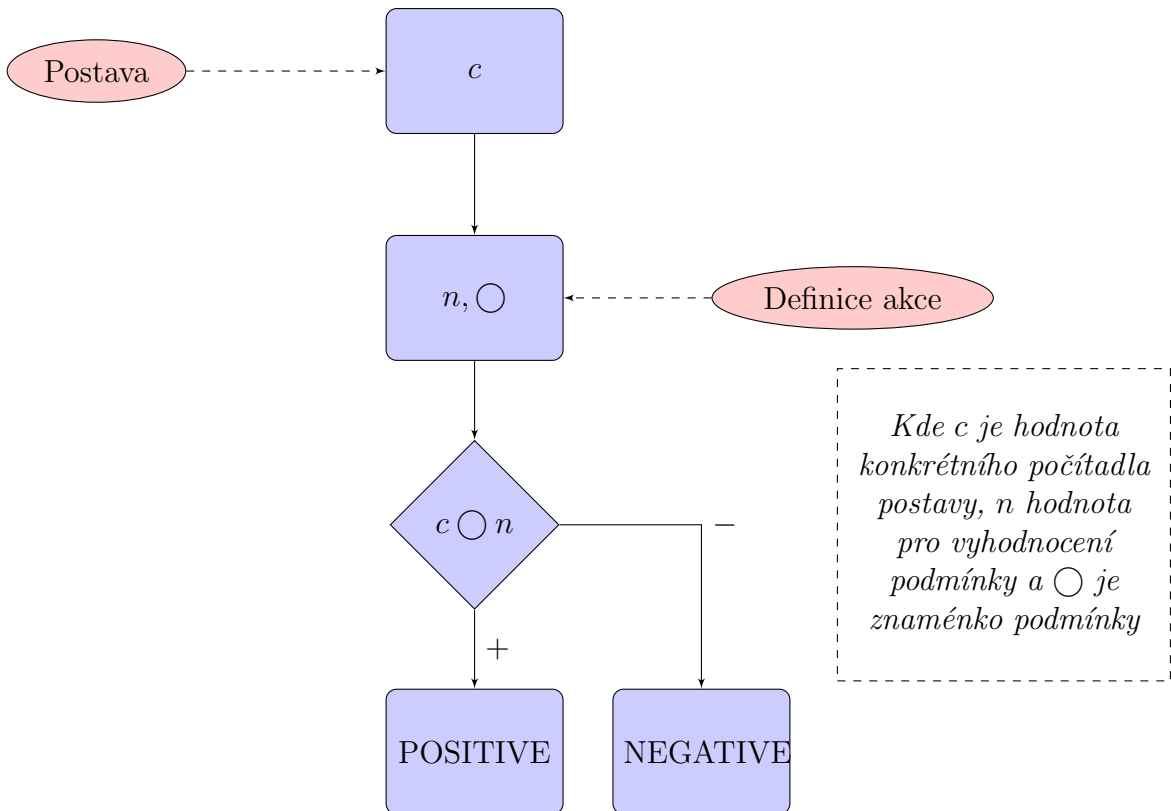
### Vyhodnocení

Vyhodnocení proběhne ověřením platnosti podmínky definované v dané akci. (viz obr. 3.5 str. 18)

### 3.4.7 Selector

Aby nebyl svázán autor ničím, je nutné umožnit vlastní definici ras, národů, nábožensví atp. Vytvoříme tedy abstrakci, kterou budeme moci vytvořit teoreticky neomezený počet těchto skupin.

Obecně je třeba vytvořit jednu nebo více výběrových povinných vlastností (dále jen Selector) s dvěma nebo více možnostmi (dále jen Selection). Ke každé takové možnosti se bude vázat Bonus, který může upravovat základní vlastnosti postavy.



Obrázek 3.5: Diagram vyhodnocení počítadla

Selector ale může být také použit pouze ke "značkování". Na základě každého selectoru může být totiž příběh větven.

Je tedy třeba uchovat název a popis výběrové vlastnosti a množinu voleb, pro které se může čtenář rozhodnout. U každé konkrétní volby opět uložit název a popis.

Před začátkem příběhu si pro každý Selector čtenář vybere jednu Selection. Pro rozšíření možností v herním systému bude zaveden atribut umožňující skrýt nějakou ze Selections. Umožní to tvořit skryté rasy nebo cokoli, pro co se čtenář nemůže rozhodnout na začátku příběhu, ale je s tím seznámen v průběhu své cesty.

Mělo by být také umožněno, aby se v průběhu hry dal Selector změnit. Aby herní systém neutrpěl na jednoduchosti a obecnosti, tato možnost bude přenechána pouze autorovi. Pokud bude chtít dát autor čtenáři možnost změnit své předchozí rozhodnutí, vystaví takový mechanismus pomocí základních primitiv.

Tento systém také umožní Selector přepsat na jinou hodnotu bez vědomí čtenáře. Zodpovědnost za logiku hry je tedy přenesena na autora. To přinese volnou ruku ve



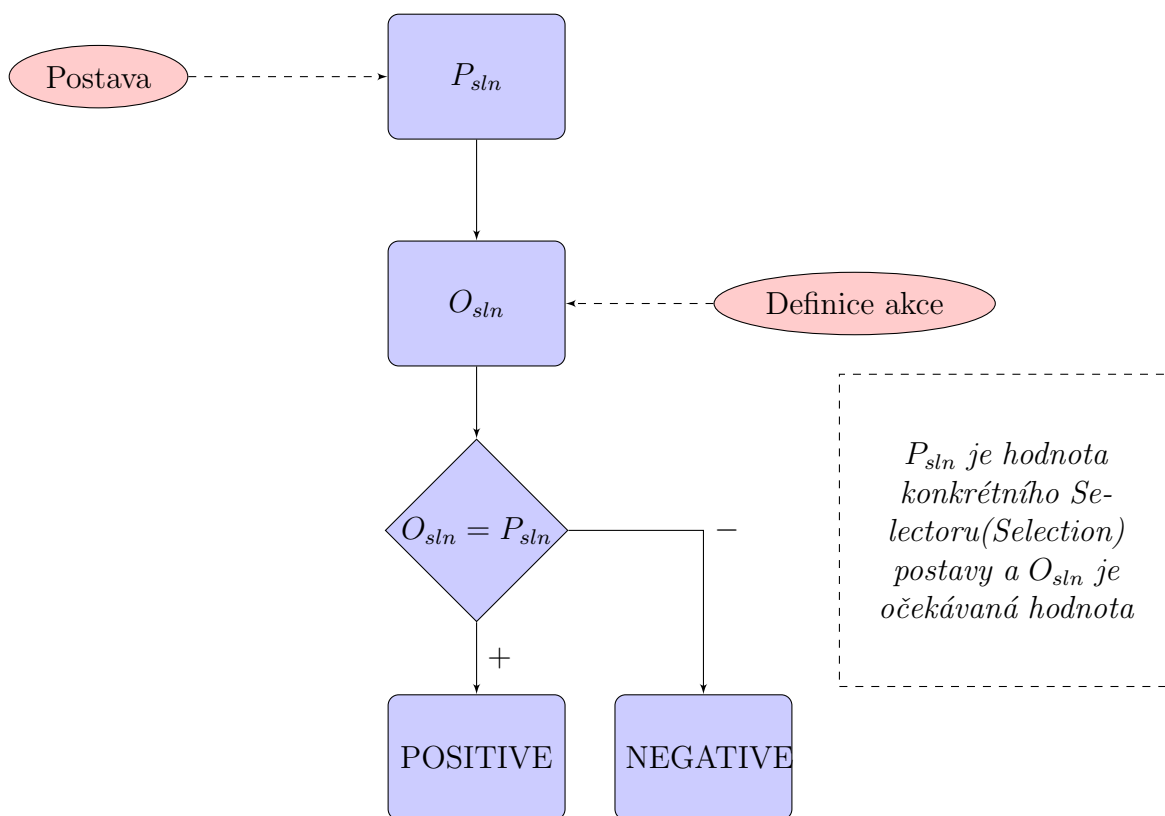
tvorbě světa autorům.

### **Shrnutí potřebných atributů**

- při tvorbě Selectoru
  - jméno a popis výběrové vlastnosti
  - výčet možností a jejich atributy (název, popis, bonus)
- při změně Selectoru
  - identifikátor Selectoru
  - identifikátor nové Selection
- při věvení příběhu
  - identifikátor Selectoru
  - identifikátor podmíněné Selection
  - odkazy na dva stavy

### **Vyhodnocení při větvení příběhu**

Vyhodnocení proběhne porovnáním čtenářem vybrané možnosti u konkrétního Selectoru v dané akci s hodnotou v definici akce. (viz obr. 3.6 str. 20).



Obrázek 3.6: Diagram vyhodnocení akce Selectoru

## 4 Datové struktury Gamebooku

Vzhledem k tomu, že předpokládáme interpretování gamebooku na různých, i mobilních, zařízeních. Je logické použít programovací jazyk s nejrozšířenější podporou. Bude tedy vhodné použít programovací jazyk Java, který je multiplatformní [4].

Také gamebookový klient je vyvíjený pro telefony s operačním systémem Android. Velkou výhodou tedy bude, že datové struktury, určené pro uchování gamebooku, bude možné použít i v tomto klientovi.

Protože celý příběh je orientovaný graf, musí být uložena množina vrcholů a množina hran. Pro uložení vrcholu byl vytvořen objekt **State**, pro hrany objekt **Action**. Protože akcí je několik typů, jsou všechny objekty reprezentovány potomky třídy **Action**.

Obě množiny jsou uchovány v instanci **HashMap**. Klíčem k objektu je vždy unikátní identifikační číslo. Obě tyto mapy jsou uloženy v objektu **Graph**.

### 4.1 Přehledy nejdůležitějších objektů

Následují popisy elementárních objektů, které jsou použity při tvorbě gamebooku.

#### 4.1.1 Graph

Obsahuje základní informace o gamebooku, jeho vývoji a autorovi. Shrnuje množiny všech stavů a akcí (viz tab. 4.1 str. 22).

#### 4.1.2 State

V tomto objektu jsou zahrnuty informace o jednom konkrétním stavu, včetně výčtu všech možných akcí, které nabízíme čtenáři. (viz tab. 4.2 str. 23).

Atribut	Datový typ	Bližší popis
inicialSetings	InitialSettings	Obsahuje informace o inicializačním nastavení (viz níže).
localID	int	Slouží k identifikaci konkrétní knihy v rámci počítače.
dateOfCreation	Date	Datum vytvoření knihy
edges	HashMap<Integer, Stav>	Mapa všech vrcholů grafu
connectors	HashMap<Integer, Akce>	Mapa hran v grafu
start	int	ID startovního stavu
titul	String	Jméno (titul) knihy
edice	String	Edice
autor	String	Jméno autora
abstrakt	String	Krátký informační popis
prolog	String	Prolog
epilog	String	Epilog
minutes	long	Počet minut, které autor strávil psaním.

Tabulka 4.1: Přehled atributů objektu Graph

### 4.1.3 Action

Uchovává podrobnosti ke konkrétní akci, kterou čtenáři nabízíme. Je to základní objekt, od kterého dědí všechny složitější akce. (viz tab. 4.3 str. 24)

Atribut	Datový typ	Bližší popis
infoPromennych	InfoPromennych	Obsahuje informace o proměnných konkrétního stavu.
counterUpdates	ArrayList <CounterUpdate>	Obsahuje informace o změnách Počítadel v konkrétním stavu.
selectorUpdates	ArrayList <SelectorUpdate>	Obsahuje informace o změnách Selectorů v konkrétním stavu.
akce	HashMap <Integer, Akce>	Všechny akce, které jsou na výběr v tomto Stavu. Akce je vložena vždy pod svým <i>id</i> .
koncovy	boolean	Pokud je stav poslední a příběh v něm končí, je nastaveno na <i>true</i> .
holder	boolean	Pokud je tento stav typu Holder, je nastaveno na <i>true</i> .
idStavu	int	Jedinečné identifikační číslo Stavů.
popis	String	Popis, který se zobrazí čtenáři.
jmeno	String	Krátký popis, který se zobrazí pouze autorovi v editoru.
x,y	int	Pozice v souřadnicovém systému. Slouží pro vykreslení v editoru.

Tabulka 4.2: Přehled atributů objektu State

Atribut	Datový typ	Bližší popis
mapaVysledkuAkce	MapaVysledkuAkce	Obsahuje informace o Stavech, do kterých může být hráč po vyhodnocení akce přeměrován (využívá se pouze pro složené akce).
typAkce	ETypAkce	Obsahuje konkrétní typ akce (enum).
x,y	int	Pozice v souřadnicovém systému. Slouží pro vykreslení v editoru.
idAkce	int	Jedinečné identifikační číslo akce.
pocatecniStav	int	Identifikační číslo stavu, ke kterému akce náleží.
koncovyStav	int	Identifikační číslo koncového stavu akce.
popisek	String	Popisek, který se zobrazí čtenáři pro výběr akce.
jmeno	String	Krátký popisek, který se zobrazí pouze autorovi v editoru.

Tabulka 4.3: Přehled atributů objektu Action

## 5 Návrh Editoru Gamebooků

Výsledná aplikace musí umožnit pohodlné a přívětivé vytváření gamebooku a dokázat mnohdy rozsáhlé příběhy, obsahující řádově desítky až stovky stavů, přehledně zobrazit a usnadnit tak orientaci.

Dále musí umožnit autorovi ověřit navržené modely situací tím, že si je zobrazí z pohledu čtenáře a bude je procházet tak, jak by to dělal on.

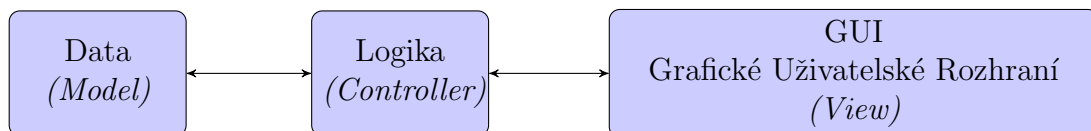
To zahrnuje tyto aktivity:

- Vytvářet stavy a editovat je
- Doplnovat metadata k vyvíjené knize
- Vytvářet a editovat prosté akce
- Vytvářet složené akce závislé na Vlastnostech, Náhodě, Selectorech, Counterech... (viz kapitola 3.4 str. 9)
- Vytvářený příběh přehledně organizovat tak, aby se v něm autor vyznal a zobrazovat jej v různém měřítku

Bude použita třívrstvá softwarová architektura MVC (Model-view-controller)[8]. Data budou oddělena od řídicí logiky, stejně tak jako od uživatelského rozhraní. (viz obr. 5.1 str. 25)

### 5.1 Data

Všechna data budou uchovávána v objektu *Graph*. Pracovat s nimi ale bude vrstva starající se o logiku (viz kapitola 4 str. 21).



Obrázek 5.1: Třívrstvá architektura

## 5.2 Řídící logika

O logiku používání operací s daty a zajištění integrity celého grafu příběhu se starají jednotlivé knihovny. Pokud jsou data změněna, zároveň nastavují potřebné flagy tak, aby editor například upozornil autora, že příběh není uložený před zavřením editoru.

Mezi nejvýznamnější knihovny patří:

### 5.2.1 libGraph

Tato knihovna zajišťuje například:

- **Ukládání gamebooku do souboru** - Tato akce je tvořena postupným zápisem jednotlivých složek grafu do souboru (viz kapitola 6 str. 33).
- **Přečtení verze herního systému** z existujícího uloženého gamebooku
- **Tvorba statistických dat**
  - Počet stavů
  - Rozsah čistého textu v normostranách
  - Průměrný počet akcí na stav
  - Datum vytvoření příběhu
  - Čas strávený nad psaním
- **Přidávání a odebírání Stavů**

### 5.2.2 libAkce

- Vytváření akcí a jejich úpravy dle parametrů
- Logické začlenění akcí do grafu

### 5.2.3 libEditor

- Načítání hodnot z konfiguračního souboru



## 5.3 GUI - Grafické uživatelské rozhraní

Pro pohodlnou práci je vytvořeno přehledné uživatelské rozhraní vyvíjené podle potřeb autorů gamebooků. Již při prvních krocích s nimi byl konzultován vzhled a ovladatelnost. Později autoři posloužili jako testéři a jejich poznámky výrazně pomohly přizpůsobit editor jejich potřebám, a tím se významně podíleli na přehlednosti a jednoduchosti editoru.

Aby bylo dosaženo sjednocení vzhledu dialogových oken, byla vytvořena knihovna *libGUI* pro jejich vytváření. Tato knihovna je používána také jako továrna na tlačítka používaná v celém editoru.

## 5.4 Start aplikace

Hlavní třída-*Editor*, po spuštění metody *main()*, nejdříve načte konfigurační soubor, ve kterém jsou informace pro logování chyb a běh editoru. Inicializuje datové struktury (objekt *Graph*) a vytvoří hlavní okenní prvek (instance třídy *GUI*). Také vytvoří instanci třídy *Timer*, pro zapisování statistik o časové náročnosti vytvářenému příběhu. (viz obr. 5.2 str. 28)

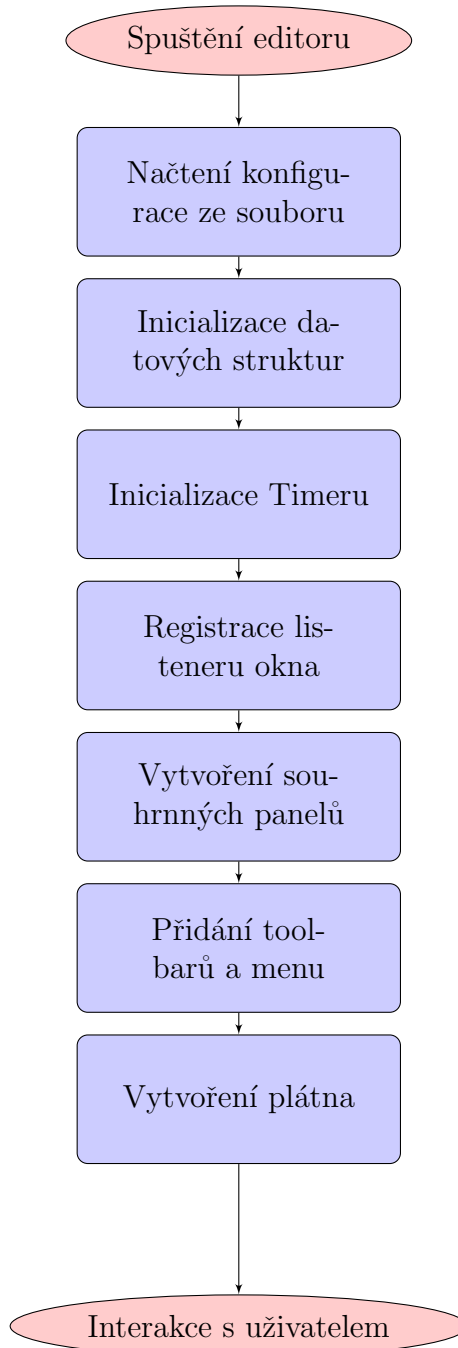
### 5.4.1 Třída GUI

Dědí od třídy *JFrame*. Obsahuje *JMenuBar* (horní menu) a dvě instance třídy *JToolBar* (nástrojové lišty). Jednu se základními nástroji pro vyrváření grafu (vlevo) a druhou pro pomocné nástroje (vpravo nahoře). Graficky se celý příběh vykresluje do komponenty *Drawarea*. (viz obr. 5.3 str. 29).

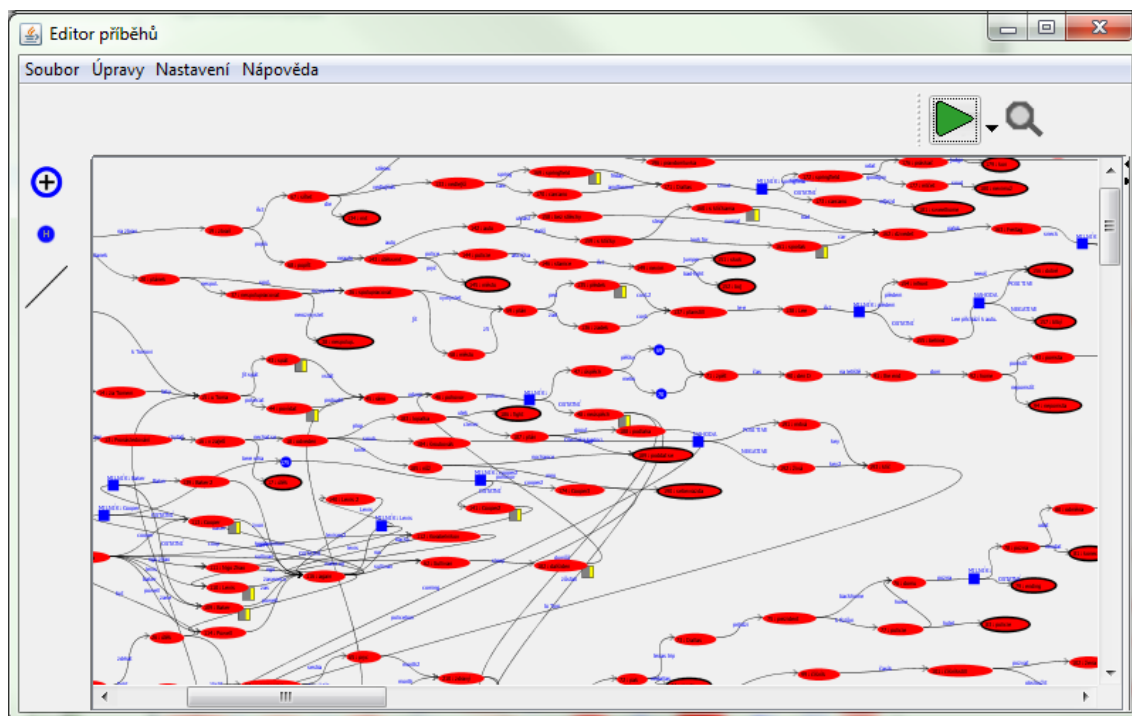
### 5.4.2 Třída Drawarea

Tato třída dědí od *JPanel*. Obsahuje několik *HashMap*, ve kterých jsou uloženy prvky, které mají být vykresleny na plátno. Jsou to mapy obsahující stavy a jejich infozáložky, akce, hrany grafu, obrazce pro zvýraznění, popisky a další. Pokud je třeba aktualizovat zobrazované informace, zavolá se metoda *prekresli()*, která se postará o aktualizování všech zobrazovaných objektů.

Plátno také implementuje *MouseListener* a *MouseMotionListener*, což umožňuje obsluhovat události vyvolané myší. Zobrazované objekty tedy lze přesouvat pomocí



Obrázek 5.2: Postup při spuštění aplikace



Obrázek 5.3: Okno editoru

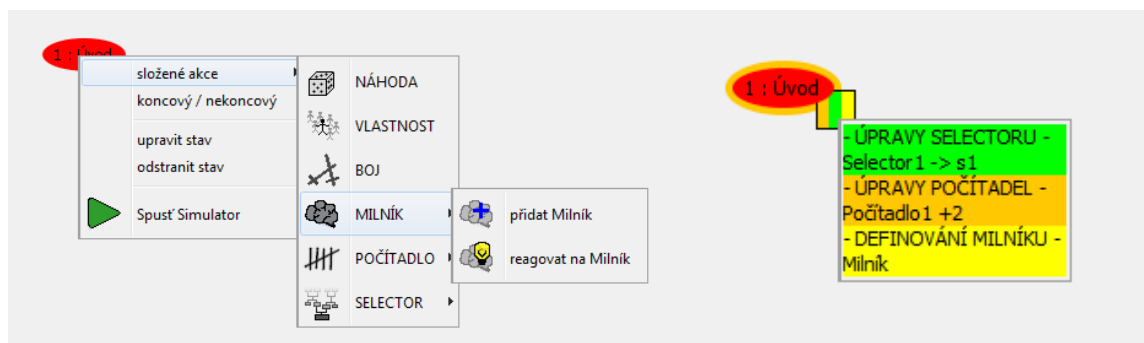
Drag&Drop, vyvolat kontextovou nabídku, popřípadě bližší informace rozbalením infoboxu (viz obr. 5.4a,b str. 30), který shrnuje změnu postavy ve stavu, například úpravu počítadla (3.4.6) nebo změnu Selectoru (3.4.7).

**Zoom** na plátně je implementován pomocí funkce *scale()*, kterou nabízí *Graphics2D*. Je tedy udržována proměnná typu *double*, která vyjadřuje přiblížení. Při každém kreslení se volá *scale()* a jako parametr je mu předáno požadované přiblížení.

## 5.5 Ostatní okenní prvky

Všechny okenní prvky jsou potomky rodičovské třídy **Okno**, nebo nějaké jeho alternativy (např. **OknoA**). Díky tomu mají všechna okna jednotný vzhled. Textová pole, určená pro dlouhé texty, ať už texty stavů nebo akcí, mají implementované Undo/Redo<sup>1</sup> pro příjemnější práci [7].

<sup>1</sup>Mechanismus umožňující vrátit úpravy v editoru o několik kroků zpět, popřípadě akce opět zopakovat



(a) Kontextová nabídka stavu

(b) Infobox stavu

Obrázek 5.4: Možnosti bližších informací stavu

### 5.5.1 Okna tvorby Selectorů

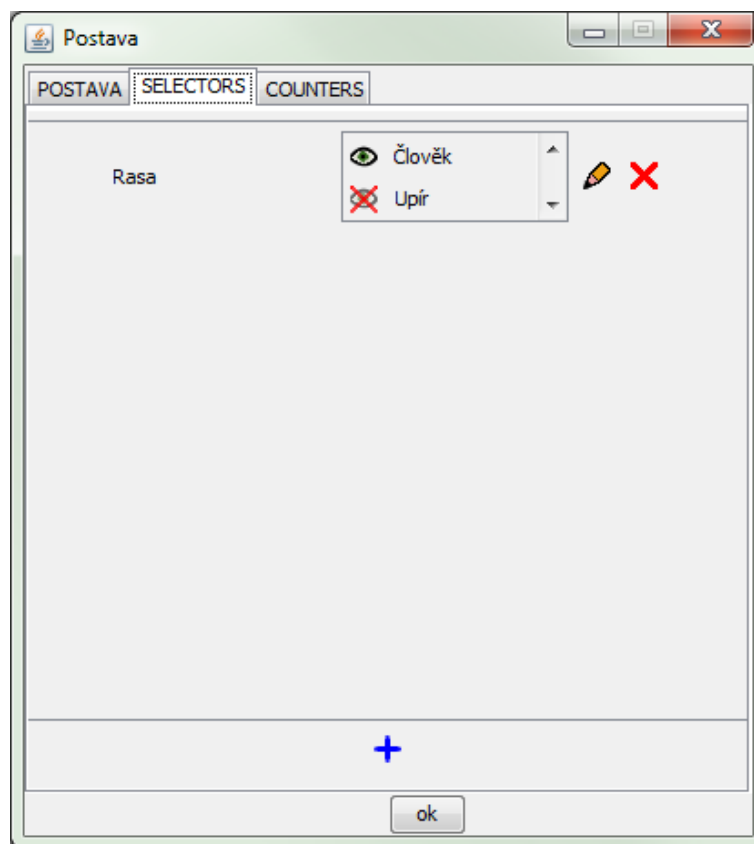
Tento okenní prvek zobrazuje základní informace o vytvořených Selectorech a umožňuje jejich úpravy. Jméno Selectoru je zobrazeno vlevo a v rámečku jsou vypsány k němu příslušné možnosti. (viz obr. 5.5 str. 31) Kliknutím na oko vlevo od Selection je možné ji znepřístupnit (znemožnit její výběr) čtenáři při začátku příběhu.

V detailním pohledu (viz obr. 5.6 str. 31) může autor nastavit bonusové hodnoty pro jednotlivé rasy. Pokud se ze čtenáře stane upír, bude mít například bonus k síle, ale jeho charisma utrpí. (viz obr. 5.7 str. 32)

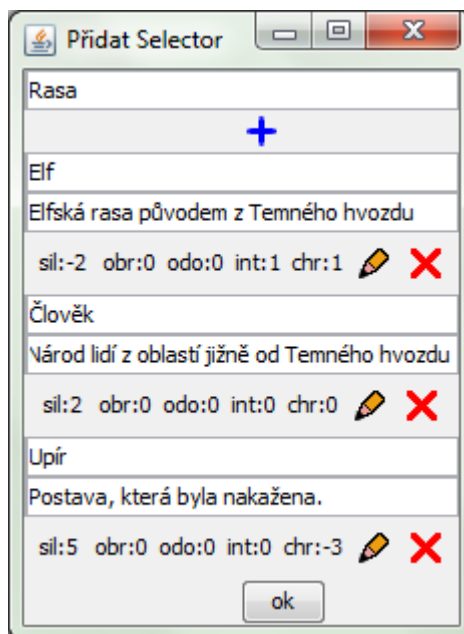
*Příkladem může být autorem napsaný příběh ze světa, ve kterém existují elfové, lidé a upíři. Čtenář si na začátku příběhu může vybrat, zda chce začínat jako člověk nebo elf, ale pro upíra se rozhodnout nemůže. Může se jím však stát v průběhu hry pomocí mechanismu, který pro něj autor připravil. Například se může někde nakazit. To autor může realizovat použitím změny Selectoru. (viz kapitola 3.4.7 str. 17)*

### 5.5.2 Okno tvorby Counterů

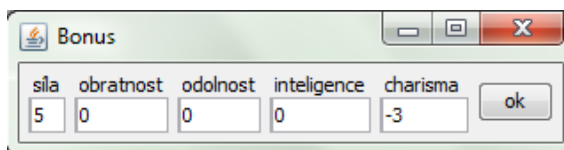
Tento okenní prvek zobrazuje základní informace o vytvořených Counterech a umožňuje jejich úpravy (viz obr. 5.8 str. 32). Vlevo je, stejně jako u Selection Selectoru, oko kterým lze počítadlo skrýt před čtenářem. Následuje jméno, stručný popis počítadla a jeho počáteční hodnota.



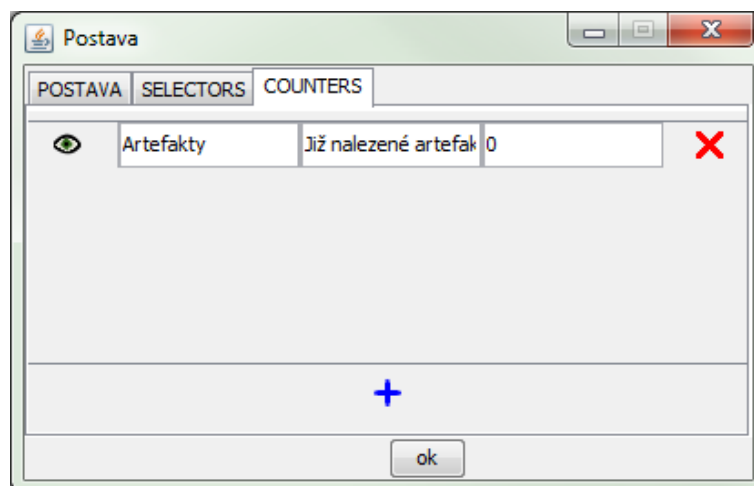
Obrázek 5.5: Okno přehledu Selectorů



Obrázek 5.6: Okno detailu Selectorů



Obrázek 5.7: Okno Bonus Selectorů



Obrázek 5.8: Okno Counterů

## 5.6 Lokalizace

Aby byla možná pozdější lokalizace, jsou všechny texty zobrazované v editoru uloženy v objektu *Bandle*. Pro lokalizaci stačí texty poskytované tímto objektem přeložit a načíst tento lokalizovaný objekt [13].

## 6 Ukládání Gamebooku

Pro dlouhodobé ukládání navržených příběhů a jejich následnou přenositelnost bude vhodné použít univerzální systém. Nabízí se formáty jako je XML nebo JSON.

**XML - Extensible Markup Language** Je zjednodušenou podobou staršího jazyka SGML. Umožňuje jednoduché vytváření značkovacích jazyků podle konkrétních požadavků. Existuje spousta knihoven pro zpracování XML v různých programovacích jazycích. Strukturu souboru tvoří jednotlivé elementy, které nesou informace týkající se přímo nich v podobě atributů. Jednotlivé elementy se do sebe mohou vnořovat. Vytváří tak struktury rodičovských elementů a jejich potomků[5].

**JSON - JavaScript Object Notation** K jeho výhodám patří hlavně úspora - oproti XML vytváří mnohem menší soubory. Rovněž je dostupná spousta knihoven pro práci s ním. Jedinou nevýhodou je, že pro člověka není tak přehledný.[14] Pro vývoj jsem zvolil jazyk XML, a to především proto, že je dobře čitelný nejen pro stroj, ale také pro člověka. Při vývoji usnadní čtení přímo souboru a tím umožní jednodušší hledání chyb. Pro ukládání bude sloužit knihovna, která bude standardní objekty, se kterými bude pracovat program, převádět do XML a ukládat do souboru.

Pokud by bylo později vyžadováno použití jiného způsobu ukládání, postačí pouze vyměnit nebo přidat knihovnu, a tím se rozšíří nabídka formátů pro export.

### 6.1 Struktura dat

Kromě standardní hlavičky, která se uvádí pro identifikaci XML, jeho verze a kódování, chceme uložit také informace o našich datech.

Předpokládejme, že pokud bude projekt dále vyvíjen, budou vycházet nové verze herního systému, které budou přidávat další možnosti. Proto bychom měli už od začátku ukládat verzi herního systému, pro který byl soubor vytvořen. Soubor obsahuje hlavní element ROOT, který má další dva potomky. Element INFO, kde jsou uložena data obecně používaná před zahájením samotného procházení příběhu a pro vytvoření postavy. Dále druhý element GRAF, ve které je uložen samotný příběh. (viz obr. 6.1 str. 36)

### 6.1.1 INFO

Obsahuje data, která se nevážou ke konkrétní situaci v příběhu, ale obecně popisují svět nebo postavu. Jeho rodičovský element je ROOT. Jeho potomky jsou elementy Figure, Selectors a Counters.

**Figure** Obsahuje data, která jsou nutná k vytvoření postavy. Jeho rodičovský element je INFO. Jako atributy jsou výchozí hodnoty vlastností postavy.

**Selectors** Obsahuje všechny Selectors a k nim příslušné Selections, které ve světě vystupují. Jeho rodičovský element je INFO. Potomky Selectors jsou elementy Selections.

U Selectors nesou atributy jejich názvy, popisy a unikátní identifikační čísla. Stejné údaje jsou uchovávány v attributech Selections, ty ale dále obsahují dalšího potomka Bonus, který uchovává bonusy a postihy k jednotlivým vlastnostem postavy.

**Counters** Obsahuje všechny Counters ve světě používané. Především jejich názvy, popisy a unikátní identifikační čísla, ale také informace o jejich viditelnosti pro čtenáře. Jeho rodičovský element je INFO.

### 6.1.2 GRAF

Obsahuje přechodový graf příběhu. Jeho rodičovský element je ROOT. Jeho potomky jsou Stav, které představují jednotlivé stavy příběhu.

Jeho atributy jsou informace o titulu a jeho autorovi (jméno autora, jméno titulu, edice, abstract, epilog a identifikační číslo knihy).

Dále obsahuje identifikační číslo startovního stavu. Tam je postava umístěna na začátku příběhu.

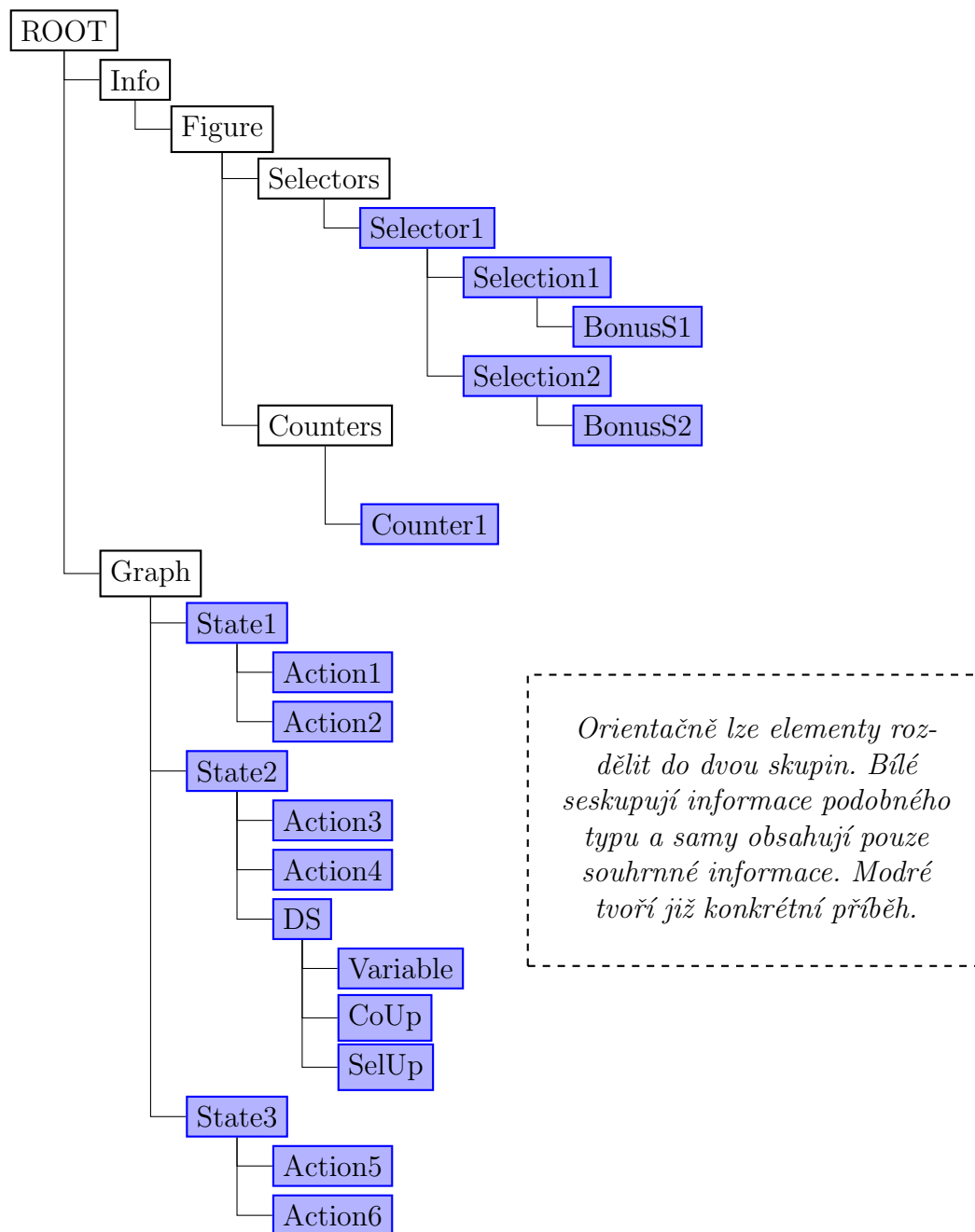
**Stav** Jeho atributy jsou jméno stavu, jeho popis a identifikační číslo. Dále atribut, který se nastaví na true, pokud je stav typu holder, a atribut, který označuje, jestli je stav koncový. Také obsahuje pozici v souřadnicovém systému, která je používána při vykreslení grafu. Jeho potomky jsou elementy DS a Akce.



**DS** Obsahuje informace o atributech postavy, které mají být změněny po jejím příchodu do tohoto stavu.

**Akce** Atributy akce jsou především jméno akce a její popis pro čtenáře. Dále pak její identifikační číslo a typ akce. Podle typu akce jsou potom určeny další atributy, které jsou potřebné ke kompletnímu popisu konkrétního typu akce.

- Pro náhodnou akci je to pravděpodobnost úspěšnosti.
- Pro akci závislou na vlastnostech je to konkrétní vlastnost a pravděpodobnost úspěšnosti.
- Akce proměnné (milník) je dále definovaná konkrétní očekávanou proměnnou.
- Akce rozhodující dle Selectors obsahuje identifikační čísla Selectoru a očekávané Selection.
- Závislost na Counters je vyjádřena identifikačním číslem konkrétního Counteru podmínkou.



Obrázek 6.1: Schéma XML souboru

## 7 Simulátor

Mnou navržený herní systém obsahuje postupy pro vyhodnocení konkrétních herních prvků. Tyto postupy jsou samozřejmě implementovány v knihovně herního systému.

Aby mohl autor ověřit správnost navržených modelů situací, byl vytvořen také Simulátor, který tyto postupy spojuje do jednoho celku, a tím vytváří rozhraní, které umožní procházení vytvořeného příběhu.

Tento celek umožňuje přívětivé vyhodnocování a procházení celého příběhu, který může používat všech mechanismů, které obsahuje navržený herní systém. Také je možné jej následně použít k tvorbě herních klientů.

Pokud chceme **Simulátor** použít, vytvoříme jeho instanci a předáme mu objekt s daty - **Graph**. Vznikne objekt Simulátoru, který se nachází v počátečním stavu. Následně jsou dostupné základní operace pro práci s ním:

- Získat text aktuálního stavu
- Získat výčet dostupných akcí
- Spustit konkrétní akci

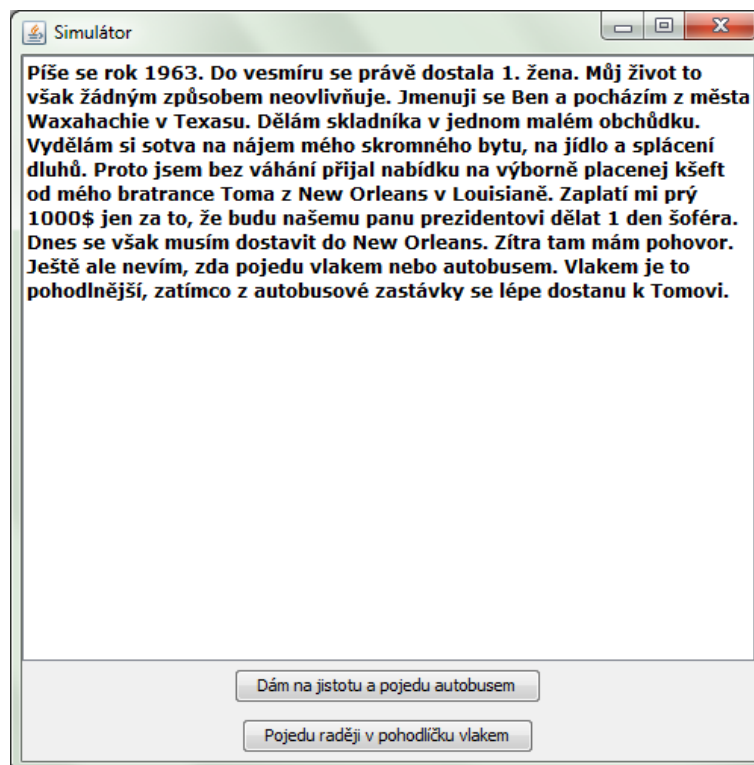
Pro maximální přenositelnost je simulátor implementován pomocí několika rozhraní.

### 7.1 Data

Při návrhu simulátoru jsem předpokládal, že mohou být data na různých zařízeních uložena jinak. Například na telefonech s Androidem je zvykem ukládat data aplikací do databáze SQLite[15]. V tom případě stačí doplnit pouze implementaci rozhraní pro danou platformu, které bude poskytovat potřebná data.

### 7.2 Zobrazení

Zobrazení výstupu ze simulátoru bude také pravděpodobně na každém zařízení jiné. Proto do simulátoru vstupuje také rozhraní, obsahující metody, jež jsou volány po dokončení nějaké akce, po které by měl následovat grafický výstup uživateli.



Obrázek 7.1: Okno Simulátoru

## 7.3 Simulátor v Editoru

Pro možnost ověření správnosti autorem navržených herních modelů byl implementován zjednodušený simulátor také v editoru (viz obr. 7.1 str. 38). Tento simulátor podporuje veškerou funkcionalitu kromě zobrazení průběhu boje. Boj se sice vyhodnocuje a skončí buď výhrou nebo prohrou, ale nevyžaduje žádnou autorovu iniciativu.

Pokud jsou vytvořeny nějaké Selectory, při prvním spuštění simulátoru je autor požádán o jejich vyplnění. Toto nastavení se uloží pro příští použití. Pokud chce autor volbu Selectoru změnit, může to udělat v rozšířené nabídce spuštění simulátoru.

Podrobnější práce se Simulátorem implementovaným v Editoru je popsán v uživatelské dokumentaci.

## 8 Testování

V průběhu vyvíjení celého systému se všechny stěžejní funkce testovaly JUnit testy [12]. Také byly použity alfa verze programu, které obsluhovali autoři knih, se kterými se úzce spolupracovalo po celou dobu vývoje. Jejich poznámky k funkcionalitě byly velkým přínosem. Během jejich testování vzniklo několik příběhů, které budou později připraveny ke zveřejnění. Několik dalších příběhů je v současné době rozpracováno. Někteří současní autoři jsou připraveni pokračovat ve svém psaní a rádi by publikovali několik dalších knih.

### 8.1 Jednotkové testy

Jednotkovými testy byla ověřena funkcionalita těchto navržených herních mechanismů podle běžných scénářů tak, jak jsou uvedeny v uživatelské dokumentaci:

- akce Prostá
- Holder
- akce Náhody
- akce Vlastnost
- akce Milník
- akce Boj
- akce Counter
- akce Selector

### 8.2 Testování autory gamebooků

Během vývoje byl vzhled, přehlednost a funkcionalita konzultována s autory gamebooků. Ti podávali pravidelně hlášení o nedostacích. V průběhu vývoje bylo vytvořeno několik knih. Souhrnné statistiky jsou uvedeny v 8.3.

## 8.3 Souhrn testovacích metod a výsledků

- cca 100 JUnit testů
- 3 alfa verze programu
- 3 autoři-testeři
- 2 knihy připravené pro publikování
- 2 knihy rozpracované
- více než 100 normostran textu vyprodukovaného v editoru
- přes 30 000 slov v cca 260 stavech
- více než 120 hodin strávených používáním editoru autory<sup>1</sup>

---

<sup>1</sup>Tato funkcionality byla do editoru přidána až v pozdějším stádiu vývoje. Skutečná hodnota tedy bude pravděpodobně mnohem vyšší

## 9 Závěr

Cílem bakalářské práce bylo vytvořit herní systém, který bude interpretovat gamebookové hry v moderním světě výpočetních technologií. To zahrnovalo nejdříve rozmyslet způsob, jak reprezentovat herní knihu včetně všech jejích prvků. Dále vytvořit vhodný a praktický editor, ve kterém bude možné tyto hry, mnohdy velice komplexní příběhy obsahující desítky až stovky stavů, pohodlně vyvíjet, dále je editovat a ukládat pro následnou distribuci.

Všechny tyto body byly splněny. Výsledkem je herní systém, který tyto komplexní herní knihy dokáže zachytit a rovněž umožňuje přidat, díky přesunutí hry z papíru na zařízení s výpočetním výkonem, některé z herních prvků stolních her na hrdiny.

Byl vytvořen rovněž editor, který používá tento systém k tvorbě těchto knih. Jeho funkcionalita byla ověřena standardními metodami používanými při softwarovém inženýrství a praktičnost a použitelnost autory gamebooků, kteří vývojem v tomto prostředí strávili spoustu času. Vytvořili v něm několik funkčních her tohoto typu, což dokládají statistiky nasbírané v průběhu jeho testování.

Dále byl vytvořen Simulátor, který tyto příběhy dokáže interpretovat v souladu s pravidly vyvinutého herního systému. Je používán v editoru a se stejným jádrem vzniká také klient pro mobilní zařízení založené na platformě Android.

Výsledkem práce je tedy komplexní systém, který je možné použít pro tvorbu a následnou distribuci herní knihy.

# Seznam použitých zkratek a výrazů

<b>Herní mechanismus/prvek</b>	Prvek, používaný při hře, přidává do ní další možnosti rozšíření. Například náhodu, roli schopností postavy atp.
<b>model situace</b>	Situace příběhu reprezentovaná dostupnými prvky herního systému.
<b>Životní cyklus knihy</b>	Proces, kterým prochází kniha, od autora až ke čtenáři.
<b>API</b>	Application Programming Interface - rozhraní pro programování aplikací
<b>XML</b>	Extensible Markup Language - rozšiřitelný značkovací jazyk
<b>JUnit testy</b>	Jednotkové testy psané v programovacím jazyce Java
<b>Alfa testování</b>	Testování, při kterém je vydána alfa verze softwaru, která je poskytnuta pouze v rámci malé skupiny lidí podílejících se na vývoji systému.
<b>SQLite</b>	Relační databázový systém psaný v jazyku C používaný v systému Android
<b>GUI</b>	Graphical User Interface - grafické uživatelské rozhraní
<b>Flag</b>	Příznak obsahující dva stavy



# Seznam příloh

Instalační příručka - Příloha A

Uživatelská dokumentace - Příloha B

Obsah CD - Příloha C

# Literatura

- [1] GASKIN, Carol. *Nekonečné příběhy*. 1. vyd. Překlad Stanislav Křížek. Praha: Fragment, 2004, 124 s. ISBN 80-720-0880-3.
- [2] ŠTÍPLOVÁ, Ljuba. *Na rozkaz krále: Hra pro chytré děti*. 1989. vyd. Praha: Panorama, 1989.
- [3] BUCHTA, Jiří. *Dračí doupě plus: fantasy hra na hrdiny : verze 1.0*. Ilustrace Roman Kýbus. Ostrava: Altar, c2005, 167 s. B (Altar). ISBN 80-859-7945-4.
- [4] *Java Standard Edition 7 API Specification* [online]. [cit. 2015-05-03]. Dostupné z: <http://docs.oracle.com/javase/7/docs/api/>
- [5] W3schools: web developer site. REFSNES DATA. [online]. [cit. 2015-05-03]. Dostupné z: [http://www.w3schools.com/xml/xml\\_dom.asp](http://www.w3schools.com/xml/xml_dom.asp)
- [6] *TEXample* [online]. [cit. 2015-05-03]. Dostupné z: <http://www.texample.net/tikz/examples/>
- [7] Java world: Add an undo/redo function to your Java apps with Swing. [online]. [cit. 2015-05-03]. Dostupné z: <http://www.javaworld.com/article/2076698/core-java/add-an-undo-redo-function-to-your-java-apps-with-swing.html>
- [8] Zdroják.cz: Prezentáční vzory z rodiny MVC. [online]. [cit. 2015-05-03]. Dostupné z: <http://www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc/>
- [9] *Legendy Armandie* [online]. [cit. 2015-05-03]. Dostupné z: <http://la.nokturno.net/>
- [10] Fantasyobchod: Dračí doupě. [online]. [cit. 2015-05-03]. Dostupné z: <http://www.fantasyobchod.cz/hra-na-hrdiny-draci-doupe>
- [11] *Příběhy Impéria* [online]. [cit. 2015-05-03]. Dostupné z: <http://imperium.mytago.cz/>
- [12] *JUnit: Project sites* [online]. [cit. 2015-05-03]. Dostupné z: <http://junit.org/>

- 
- [13] The Java™ Tutorials: Setting the Locale. [online]. [cit. 2015-05-03]. Dostupné z: <https://docs.oracle.com/javase/tutorial/i18n/locale/index.html>
- [14] W3schools: JSON. [online]. [cit. 2015-05-03]. Dostupné z: <http://www.w3schools.com/json/>
- [15] Android SQLite database and content provider - Tutorial. [online]. [cit. 2015-05-03]. Dostupné z: <http://www.vogella.com/tutorials/AndroidSQLite/article.html>

# Příloha A

## Instalační příručka

# 1 Instalace a spuštění

Zde je popsáno, jak zprovoznit gamebookový editor na vašem zařízení.

## 1.1 Instalace

Předpokladem pro bezproblémový běh aplikace je nainstalovaná Java ve verzi alespoň 1.7

Nejdříve prosím zkopírujte z přiloženého CD složku s programem na lokální disk.

## 1.2 Spuštění

Pro windows je připravený soubor editor.bat, kterým je možné aplikaci spustit s potřebnými parametry bez námahy. Pokud budete program spouštět ručně, nebo na jiné platformě, použijte k tomu následující příkaz.

```
javaw -jar -Dfile.encoding=UTF8 e6523.jar
```

Tím bude zaručeno, že bude použito správné kódování napříč všemi platformami. Sjednocení použitého kódování usnadní případnou pozdější distribuci.

# Příloha B

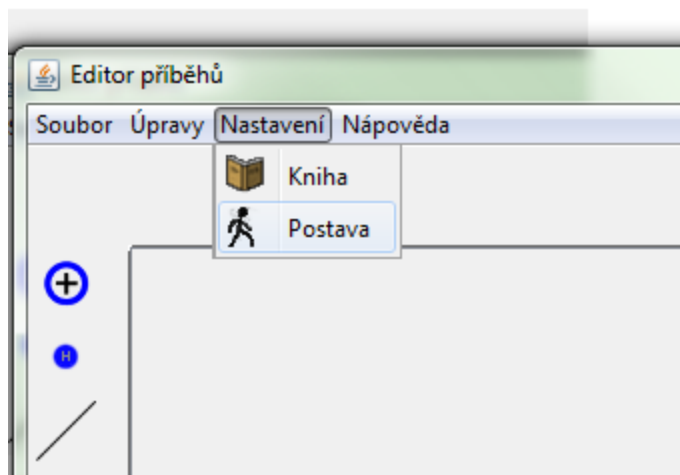
## Uživatelská dokumentace

# Obsah

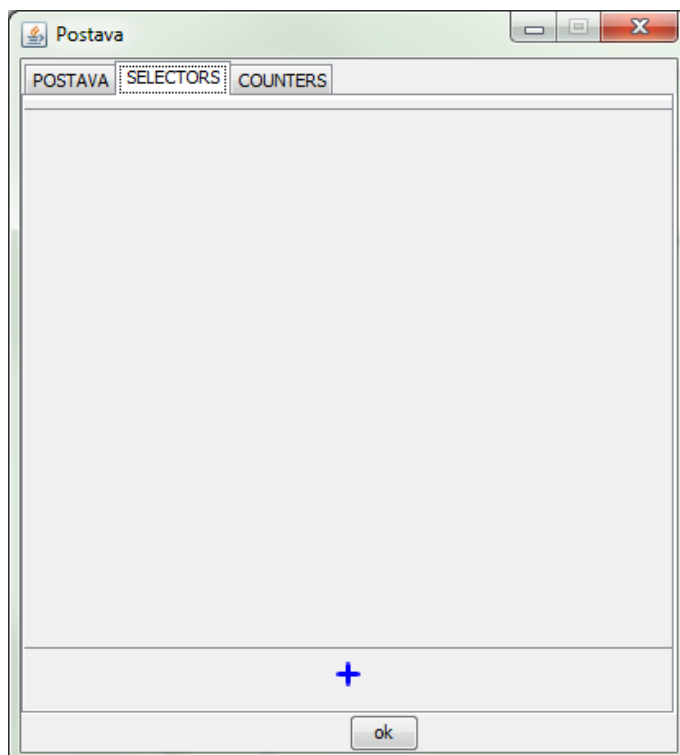
<b>1</b>	<b>Tvorba postavy</b>	<b>50</b>
1.1	Selector . . . . .	51
<b>2</b>	<b>Stavy a Jednoduché přechody mezi nimi (Prosté akce)</b>	<b>52</b>
<b>3</b>	<b>Složitější větvení příběhu - role náhody</b>	<b>61</b>
<b>4</b>	<b>Složitější větvení příběhu - role Vlastností</b>	<b>64</b>
<b>5</b>	<b>Složitější větvení příběhu - Používání milníků</b>	<b>67</b>
<b>6</b>	<b>Složitější větvení příběhu - Používání Selectorů</b>	<b>72</b>
6.1	Použití Selectorů na vytvoření různých začátků . . . . .	75
<b>7</b>	<b>Složitější větvení příběhu - Používání Counterů</b>	<b>77</b>
<b>8</b>	<b>Holder</b>	<b>85</b>
8.1	Holder - přesměrování jedné akce na druhou . . . . .	85
8.2	Holder - vytvoření oddělených startů . . . . .	86
<b>9</b>	<b>Simulátor</b>	<b>87</b>
<b>10</b>	<b>Ukládání a distribuce Gamebooku</b>	<b>88</b>
10.1	Nezbytné úpravy před distribucí . . . . .	88

# 1 Tvorba postavy

V každém příběhu vystupuje čtenář/hráč jako postava. Hrdina, který má svou historii, charakteristické vlastnosti i nějaké přesvědčení. Pohybuje se ve světě, ve kterém mohou žít různé rasy nebo národy. Někteří obyvatelé těchto světů se hlásí k různým náboženským skupinám, sektám či gangům. Proto i Vy ve svém světě máte možnost vytvořit množství těchto voleb. V menu Nastavení vybereme možnost Postava.



Obrázek 1.1: Menu nastavení : Postava

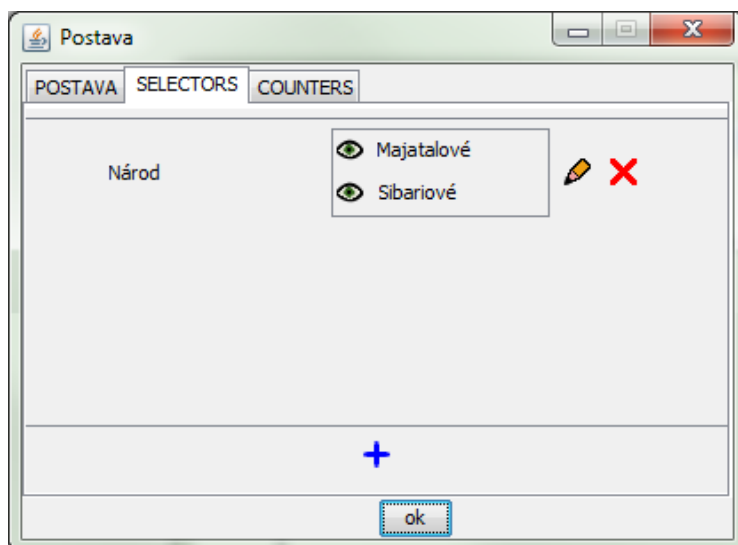


Obrázek 1.2: Nastavení postavy: Shrnutí Selectorů

První z možností je vytvořit takzvaný Selector.



## 1.1 Selector

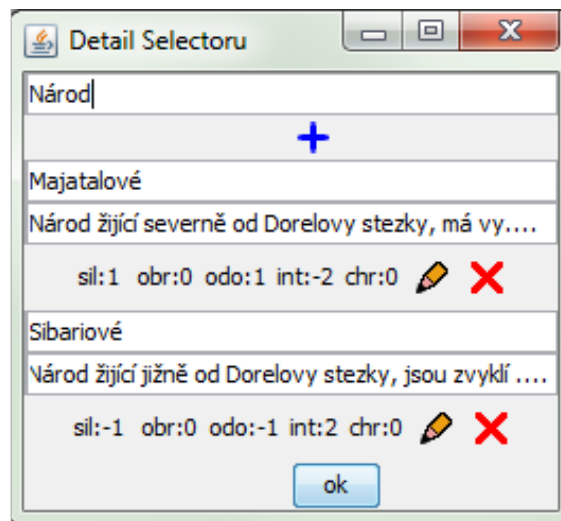


Obrázek 1.3: Vytvořený Selector: Národ (na výběr jsou dvě možnosti)

rod, protože je od přírody spíše slabší, spoléhá na svou nadprůměrnou inteligenci a důvtip. Pokud se tedy hráč rozhodne pro jeden z těchto národů, chceme upravit jeho vlastnosti. Jako severanovi mu přidáme bonus k síle a odolnosti, ale snížíme inteligenci. Jižanům nastavíme bonus v opačném smyslu.

Na základě součtu základních vlastností postavy a všech bonusů se vyhodnocují jednotlivé události ve Vašem příběhu. (viz Akce : role Vlastnosti) pozn.: je možné vytvořit libovolné množství národů, je ale nutné, aby bylo vždy z čeho vybírat, proto je minimální počet nastaven na 2. Jednou z vlastností postavy je tedy nyní národ a než začne čtenář hltat váš příběh, bude vyzván aby si vybral jeden z Vámi zadaných národů. Je vhodné zmínit se o obou národech v prologu, aby se o nich čtenář dozvěděl více a mohl se lépe rozhodovat. Námi vytvořený národ ale nemusí nutně bonusem upravovat vlastnosti postavy. Selector je možné použít také k větvení příběhu. (viz Akce: role Selectoru)

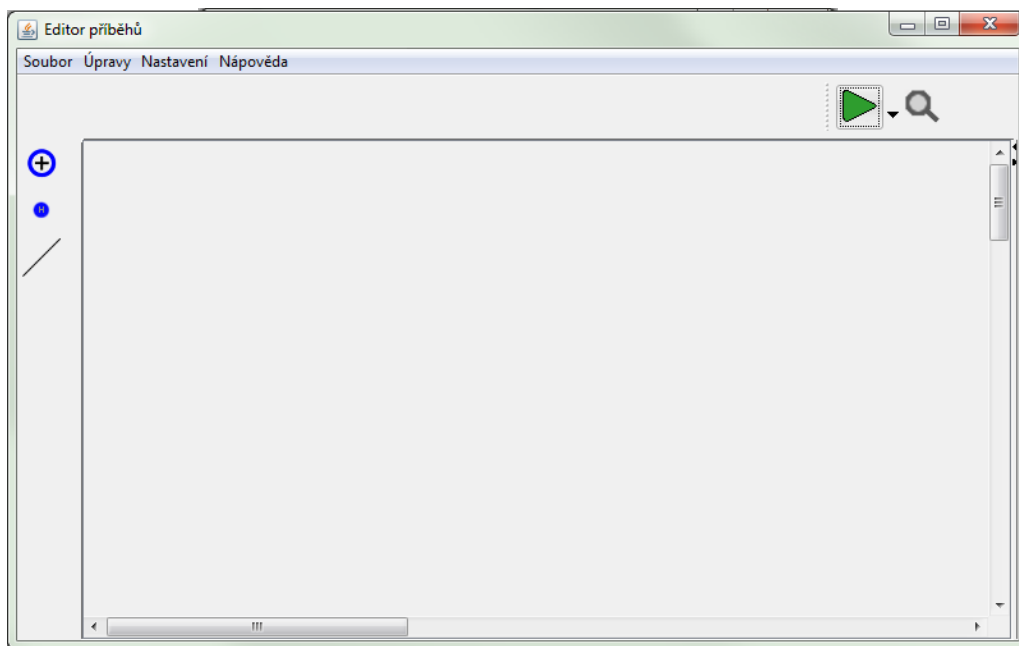
Pomocí tohoto mechanismu je možné přidat libovolnou "výběrovou vlastnost". Typickým příkladem takové vlastnosti mohou být již zmíněná náboženství, rody, rasy, gangy, národy... Řekněme, že v našem světě žijí národy dva. Jejich území jsou rozdělena hraniční stezkou a panuje mezi nimi jistá nevraživost už po několika staletí. Naše postava žije v příhraniční oblasti. Hráč si může vybrat, k jakému z národů přísluší. Řekněme, že severní národ je spíše válečný, statní muži a svalnaté ženy. Jižní ná-



Obrázek 1.4: Detail Selectoru Národ

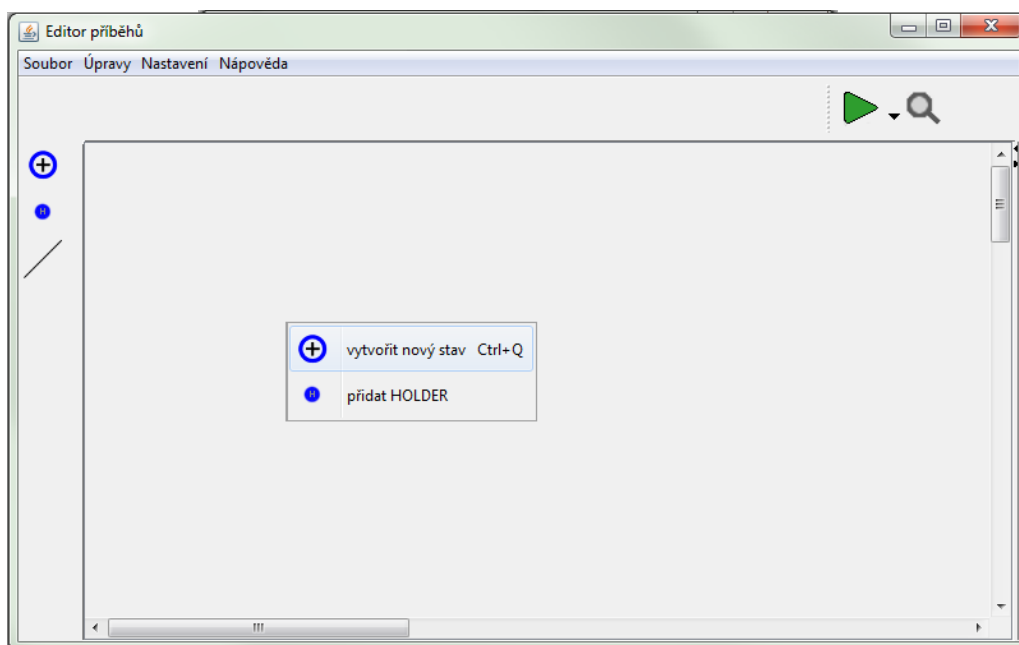
## 2 Stavy a Jednoduché přechody mezi nimi (Prosté akce)

Po otevření editoru můžeme začít tvořit vlastní příběh.



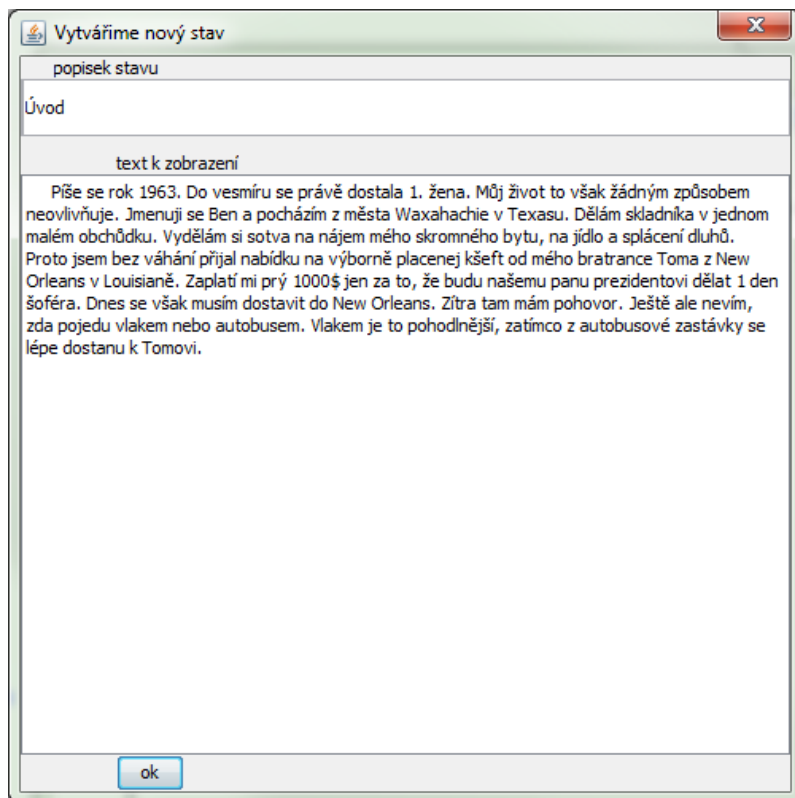
Obrázek 2.1: Editor před začátkem tvorby příběhu

Nový stav vytvoříme kliknutím pravým tlačítkem myši na prázdné plátno a výběrem z kontextové nabídky.



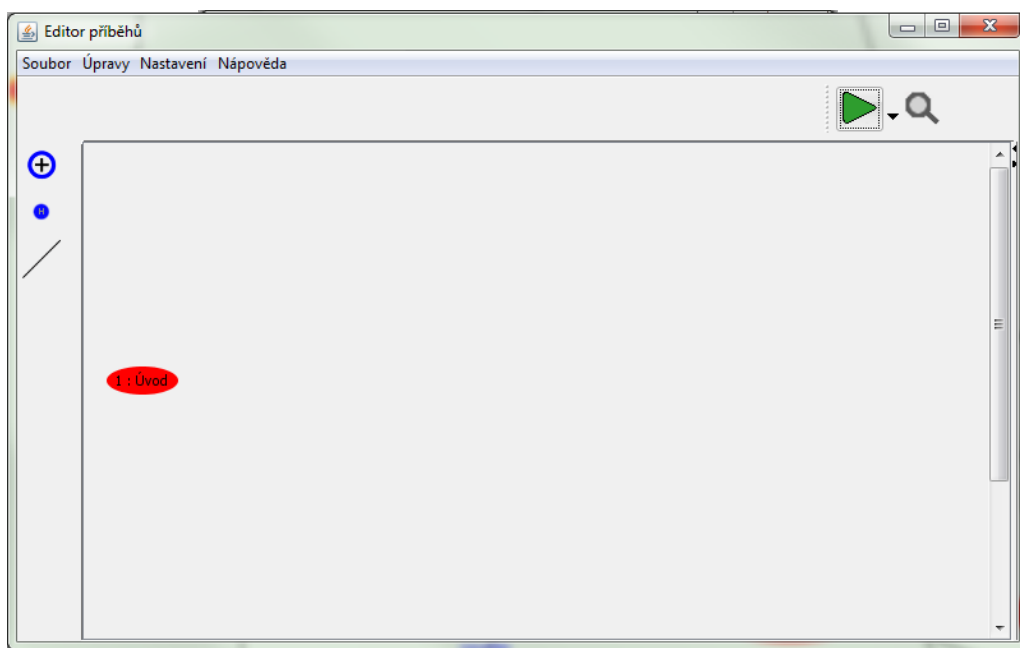
Obrázek 2.2: Rozbalená kontextová nabídka plátna

Objeví se okno, do kterého se vyplní pracovní popisek (je určen jen pro autora, čtenáři se nikde nezobrazí) a text stavu (ten se zobrazí čtenáři a popisuje mu, kde zrovna je, co vidí a co se mu honí hlavou).



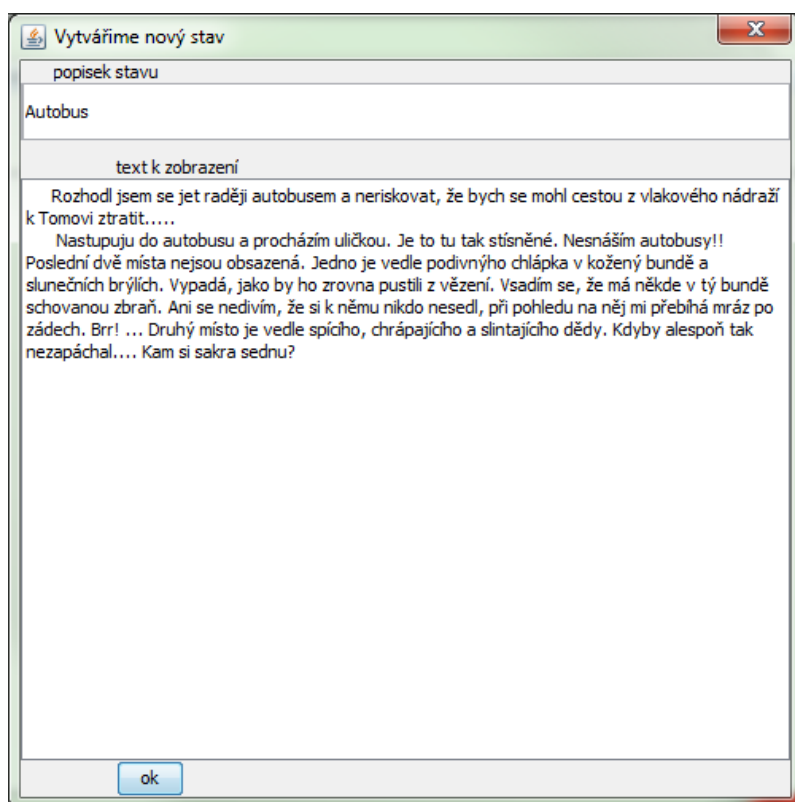
Obrázek 2.3: Okno tvorby nového stavu

Po klepnutí na tlačítko "ok" se zobrazí stav na plátně.

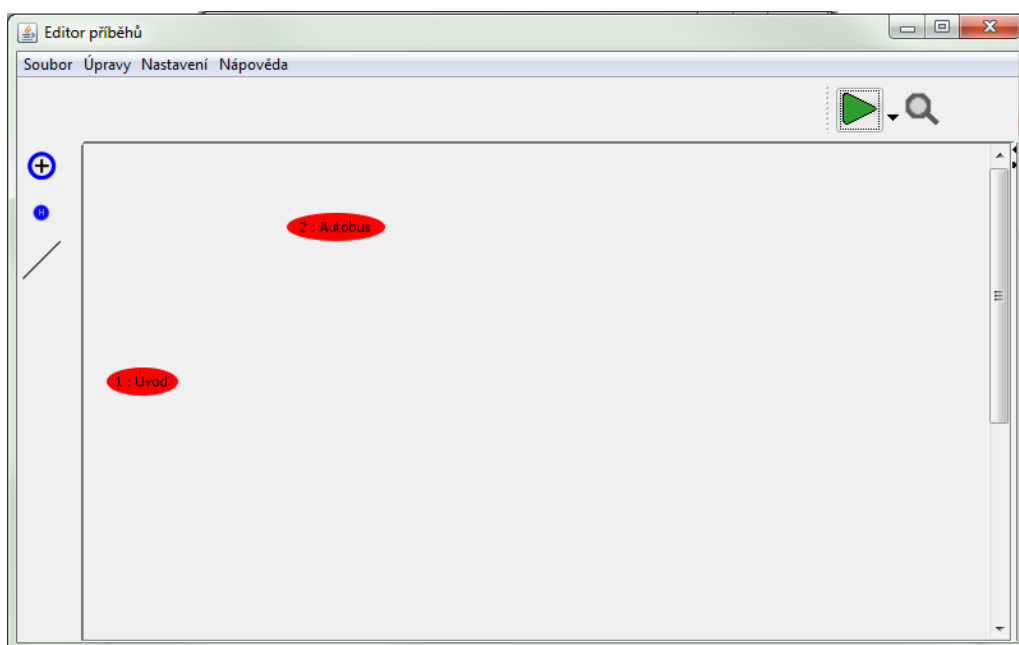


Obrázek 2.4: Okno editoru s prvním stavem

Stejný postup použijeme pro tvorbu druhého stavu, do kterého se hráč přesune z prvního.

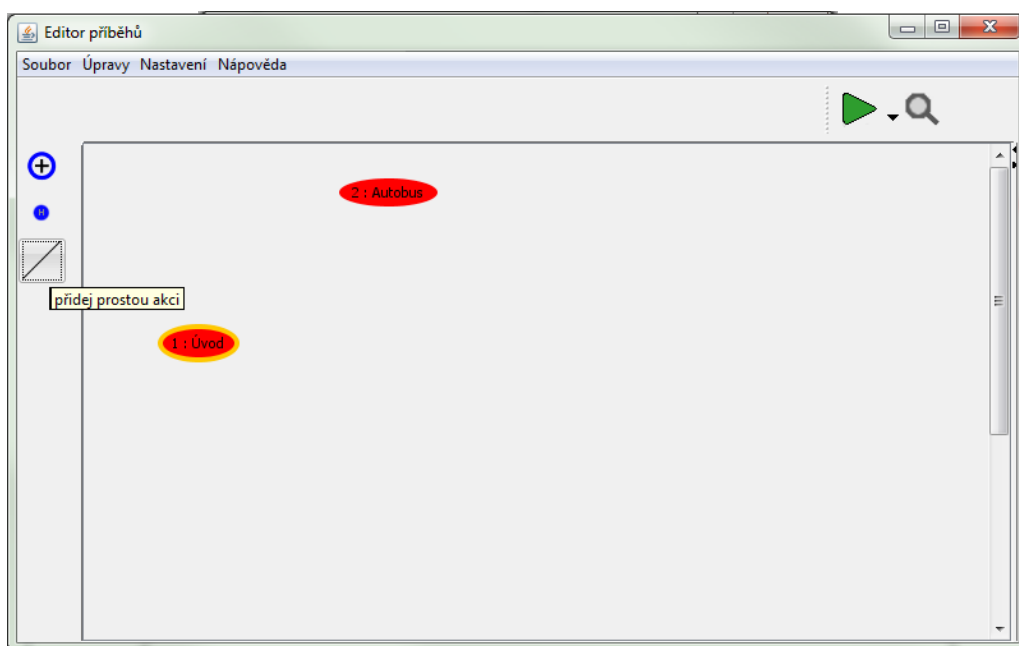


Obrázek 2.5: Tvorba druhého stavu



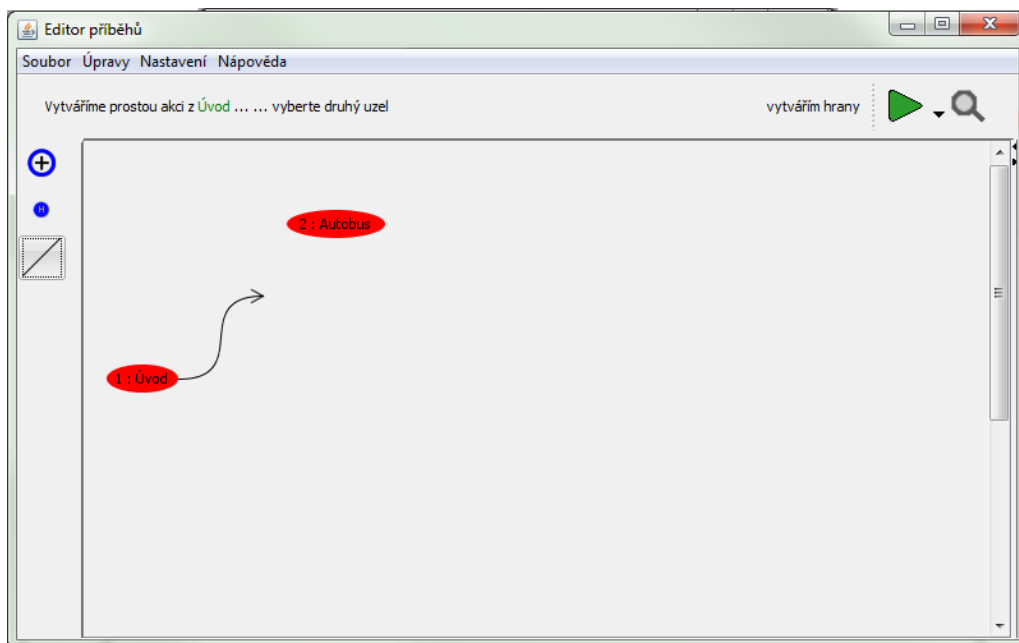
Obrázek 2.6: Změna se zobrazí na plátně

Máme hotové dva stavy a stačí přidat akci, po které se čtenář rozhodne jet autobusem. Klepneme na symbol přidávání prostých akcí.



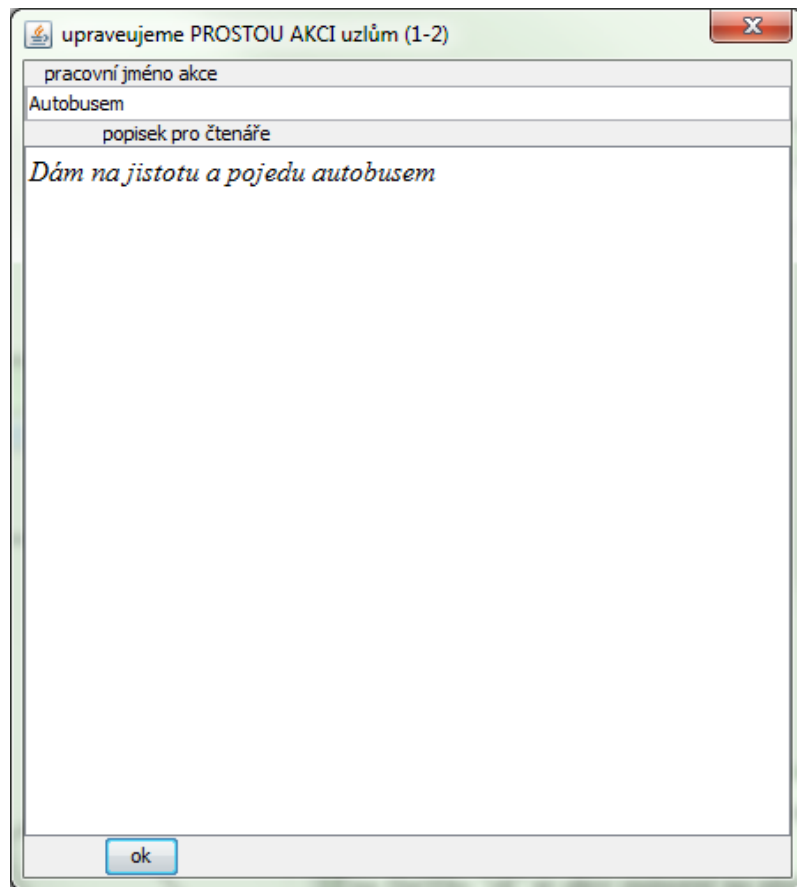
Obrázek 2.7: Tvorba prosté akce

Editor se přepne do režimu přidávání prostých akcí. Následně stačí kliknout na stav, ve kterém nabízíme čtenáři výběr akce.



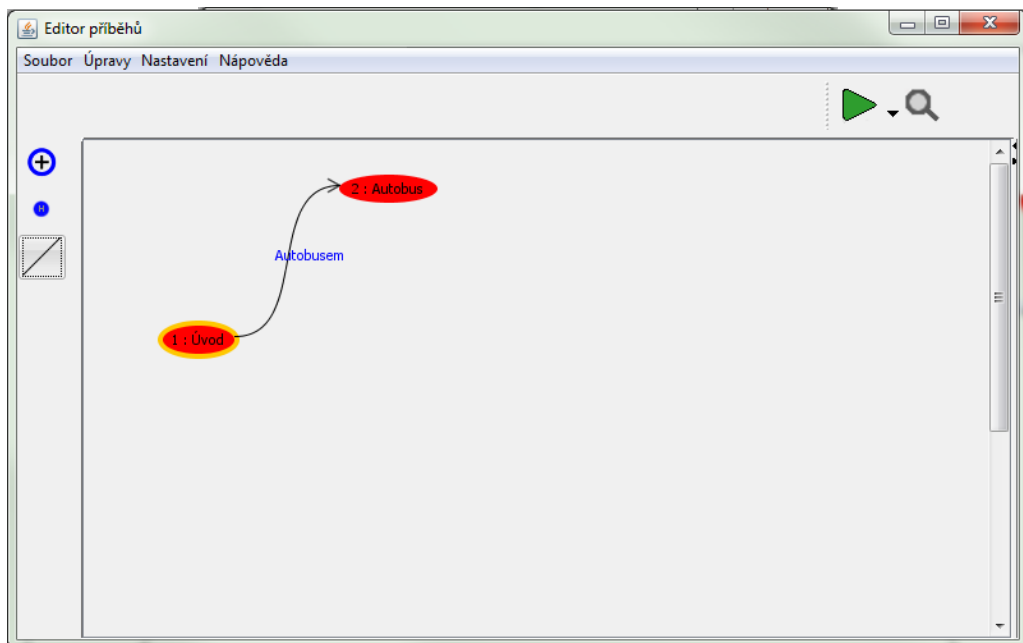
Obrázek 2.8: Natáhneme akci k druhému stavu

Natáhneme šipku k druhému stavu a klepneme na stav, do kterého nás akce přivede. Budeme vyzváni k vyplnění pracovního jména akce (jen pro Autora) a popisu akce pro čtenáře.



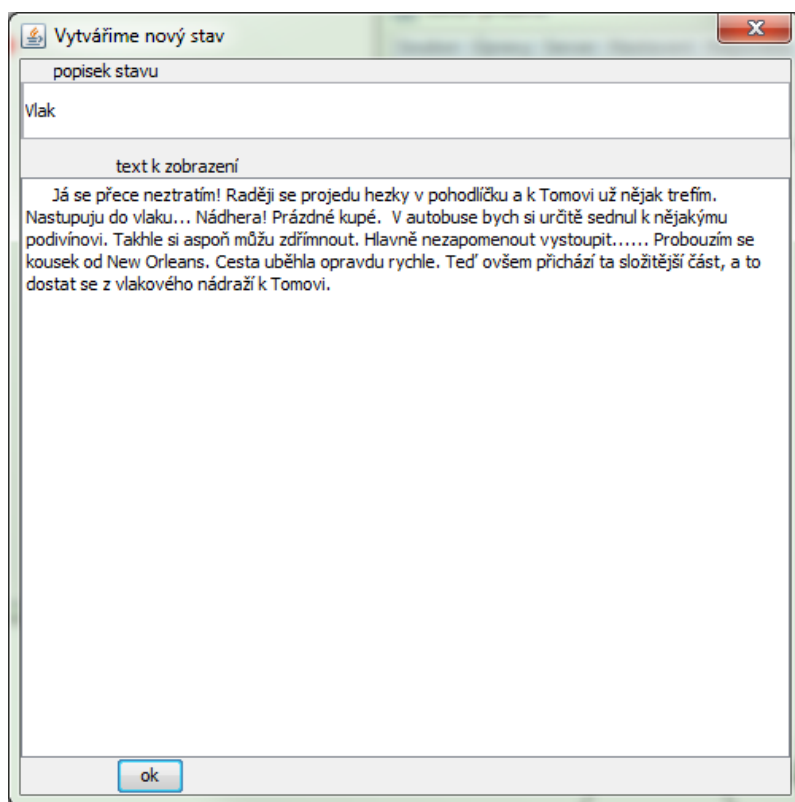
Obrázek 2.9: Vytvoříme popis pro výběr akce

Po klepnutí na tlačítko "ok" se akce znázorní na plátně.



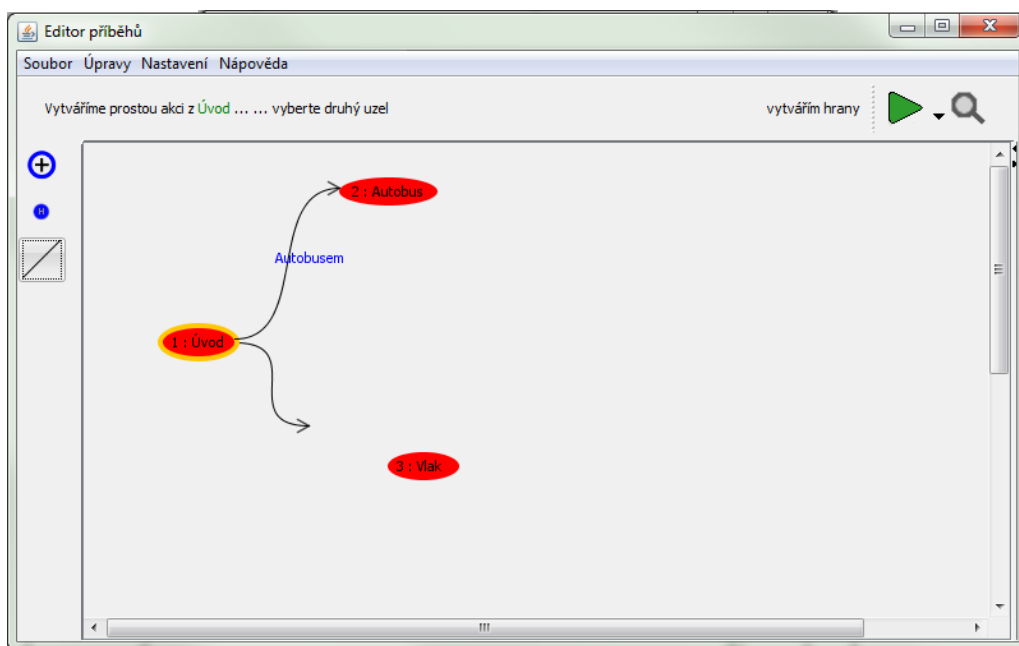
Obrázek 2.10: Vytvořená akce je znázorněna šipkou

Stejným způsobem postupujeme dále. Přidáme čtenáři druhou možnost výběru akce. Vytvoříme tedy druhý stav - Vlak.



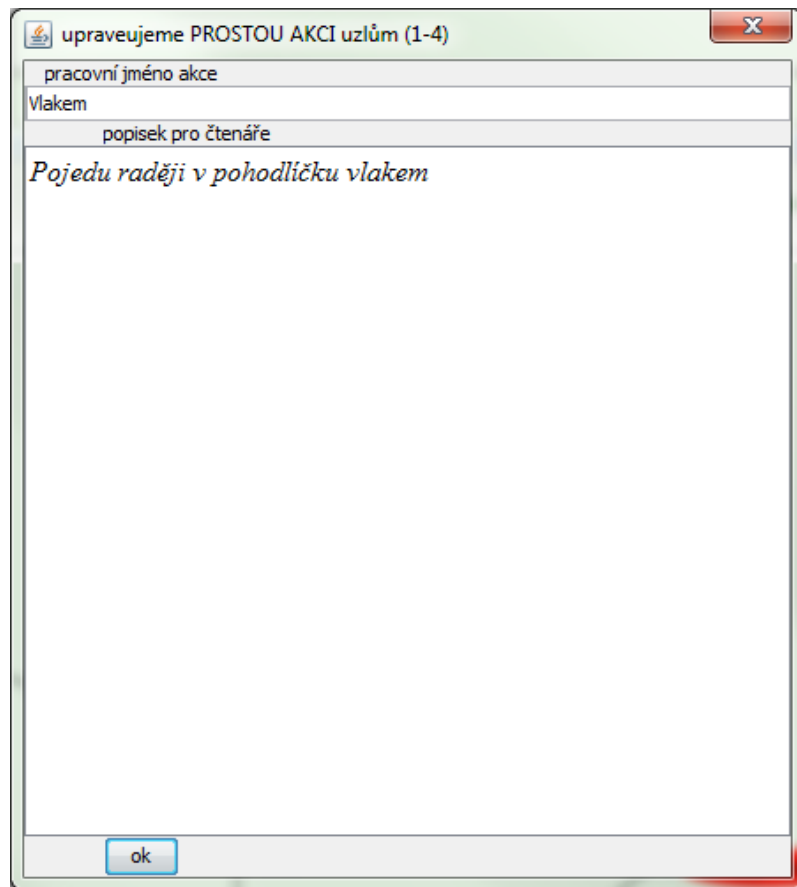
Obrázek 2.11: Vytvoříme druhý stav, ve kterém se Ben rozhodl pro cestu vlakem

A přidáme akci, kterou nabídneme čtenáři.

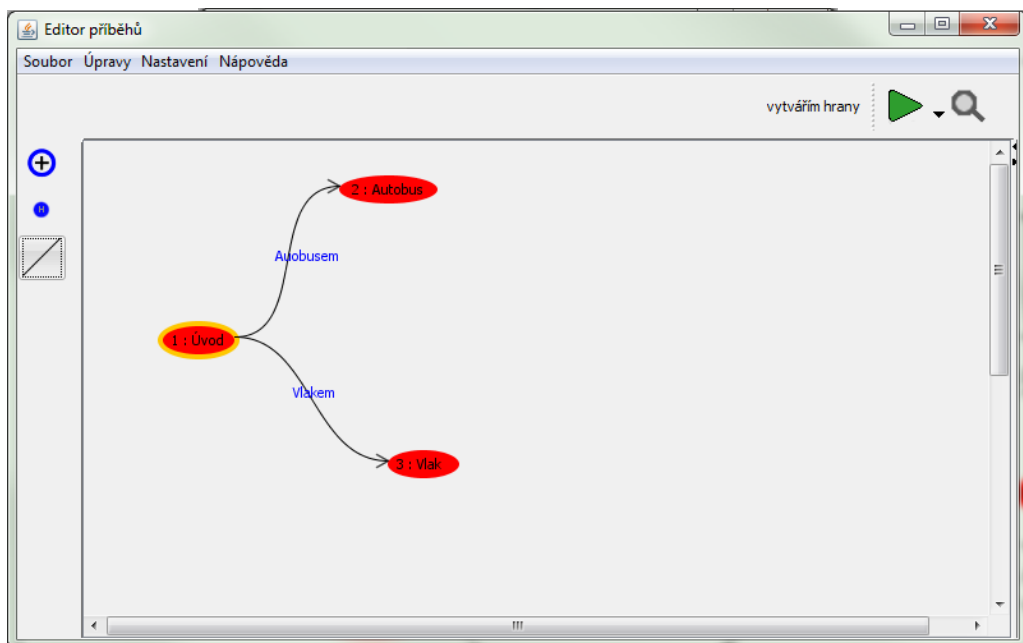


Obrázek 2.12: Vytvoříme prostou akci ze stavu Úvod do stavu Vlak



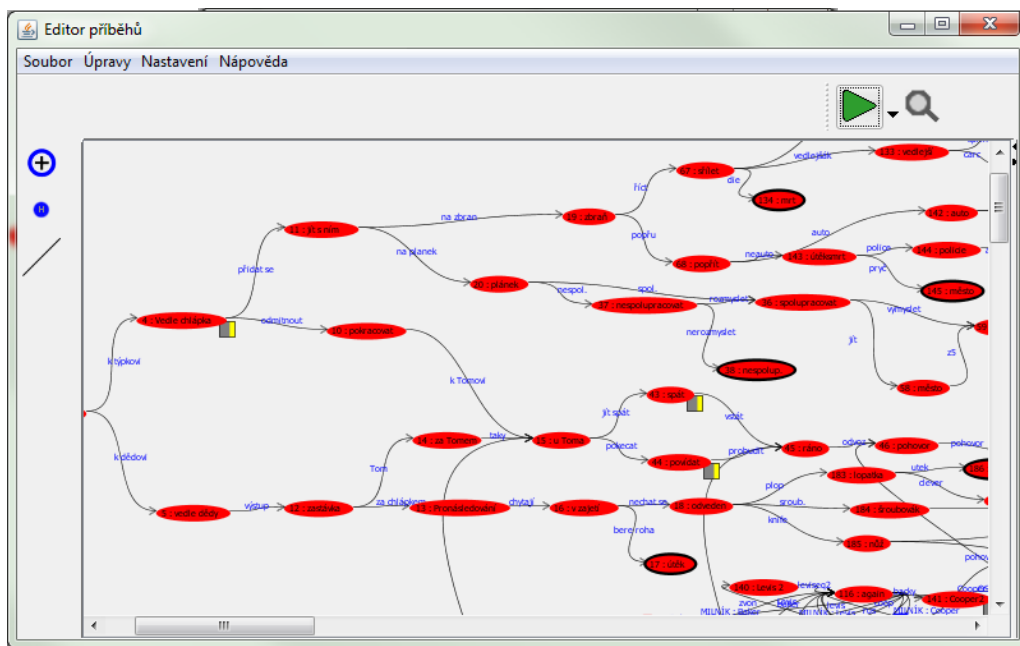


Obrázek 2.13: Vytvoříme popisek k výběru vlaku



Obrázek 2.14: Čtenář si nyní může vybrat cestu autobusem nebo vlakem

Tímto způsobem je možné příběh různě větvit a vytvářet i složité příběhy.



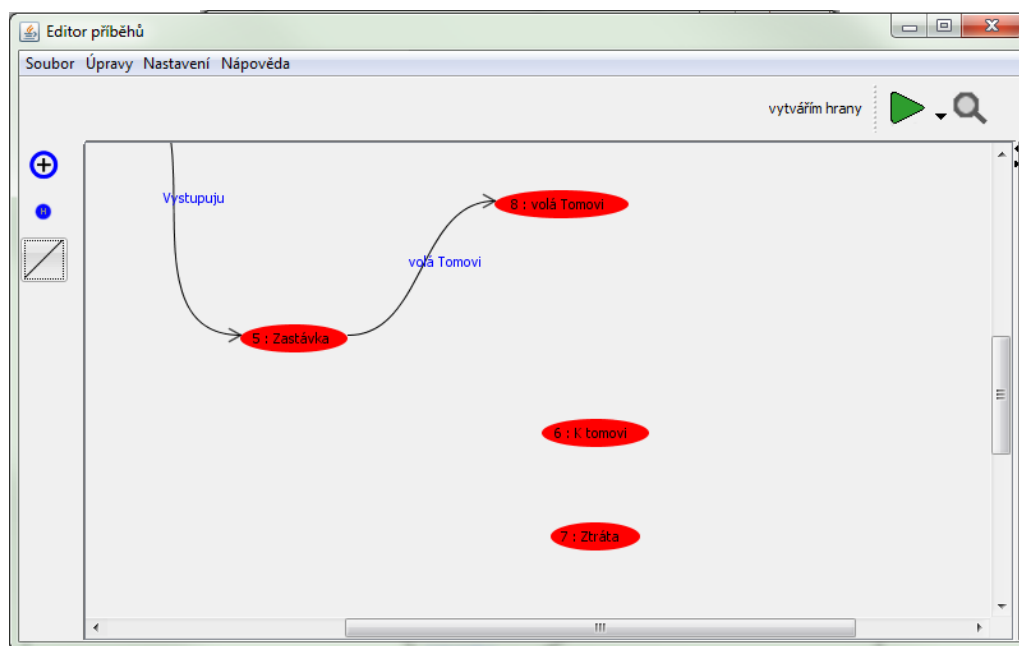
Obrázek 2.15: Ukázka větvení příběhu

## 3 Složitější větvení příběhu - role náhody

Aby byl příběh zajímavější a stejné rozhodnutí vždy nemuselo vést ke stejnému výsledku, je možné vložit místo Prosté akce Akci náhody. Když si čtenář zvolí nějaký čin, rozhodne náhoda, došlo-li k jedné akci nebo ke druhé. Příkladem může být situace, kdy Ben vystoupil z vlaku a musí se dostat k Tomovi domů. Pokud se rozhodne hledat cestu na vlastní pěst, ať náhoda rozhodne, jestli zabloudí nebo ne. Pokud zabloudí, střetne se s mafiány. Ben se tak může zaplést do něčeho, co původně neměl v plánu. Nebo se snad štěstí přikloní na Benovu stranu a dopadne to všechno dobře?

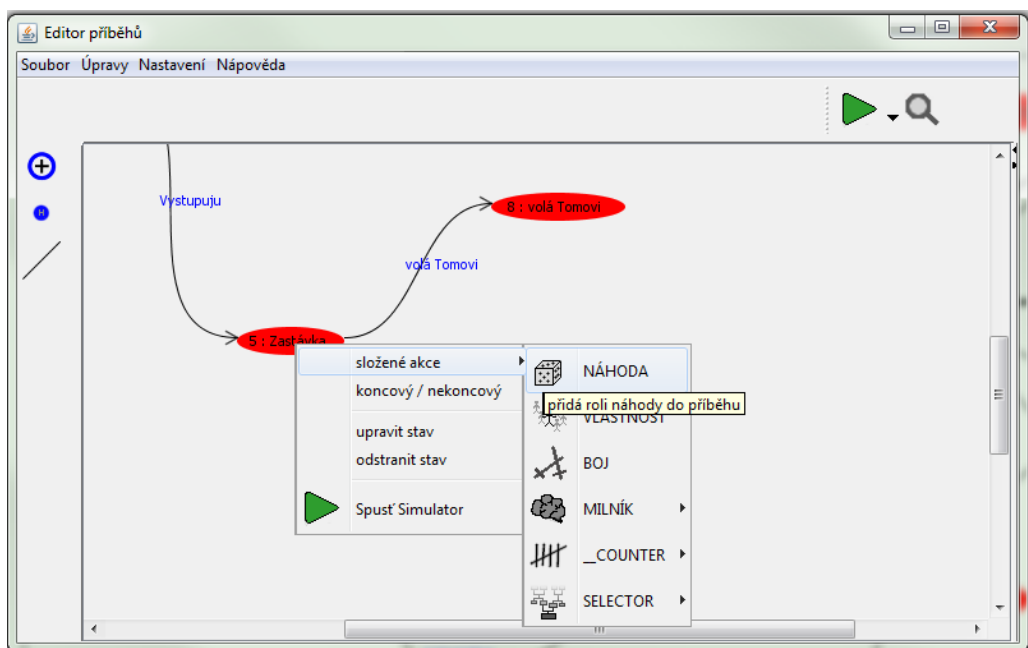
Tedy tedy jak na to.

Vytvoříme si tedy stav, ve kterém Ben vystoupil z vlaku a rozhoduje se, jestli otestuje svůj orientační smysl, nebo raději zavolá Tomovi.



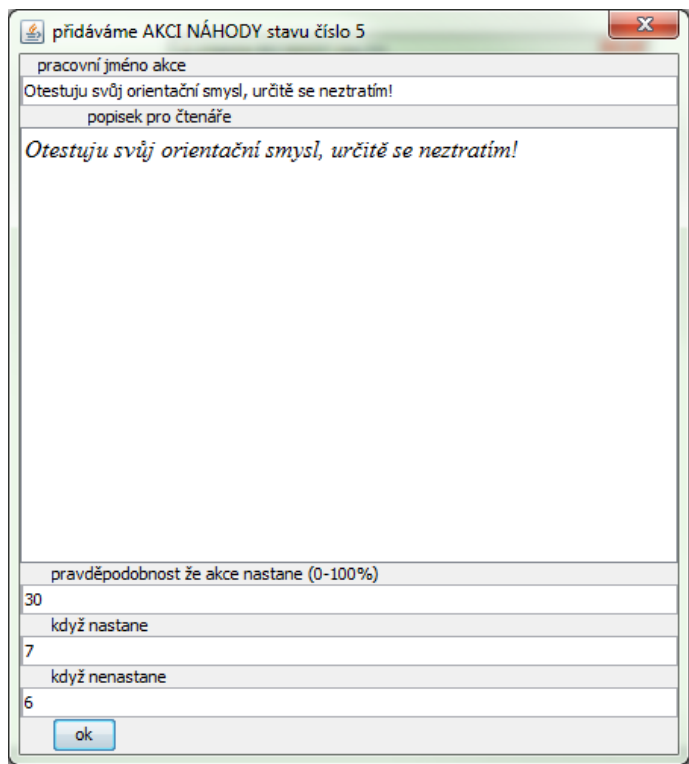
Obrázek 3.1: Situace, ve které chceme využít náhody

Pro přidání náhody klikneme pravým tlačítkem na stav, ve kterém se čtenář rozhoduje, a vybereme akci náhody.



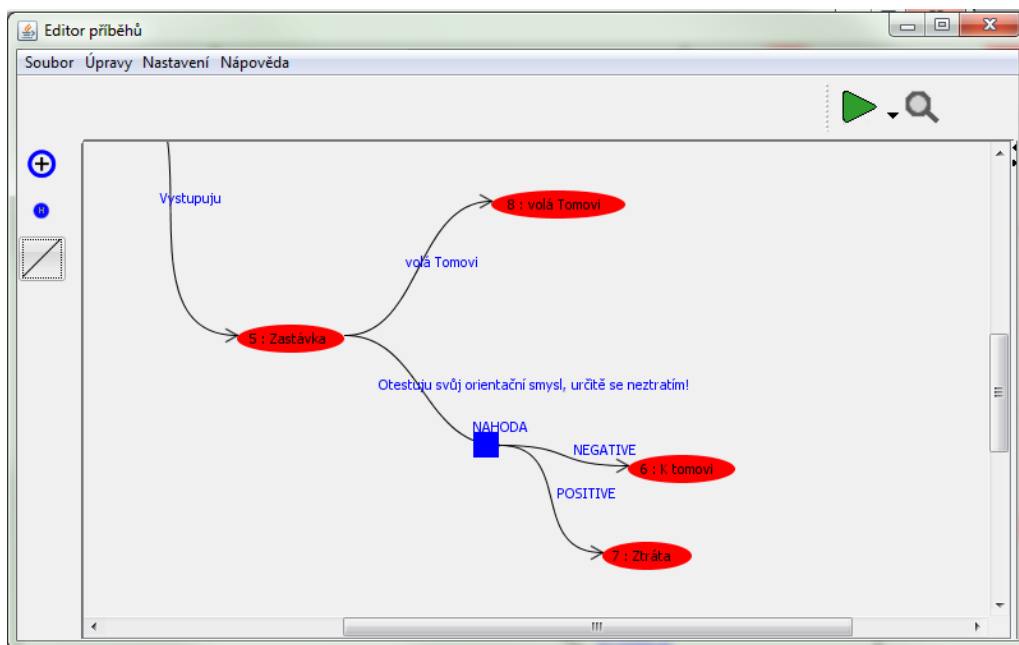
Obrázek 3.2: Z kontextové nabídky volíme náhodu

Zobrazí se okno podobné tomu u jednoduché akce. Vyplníme pracovní popisek i volbu pro čtenáře. Dále je potřeba určit procentuální pravděpodobnost toho, že akce nastane. V našem případě má Ben 30-ti procentní šanci, že zabloudí v ulicích města. Dále už stačí jen vypsát čísla stavů, do kterých se čtenář přesune (6-nezabloudí, 7-zabloudí).



Obrázek 3.3: Okno pro nastavení akce náhody

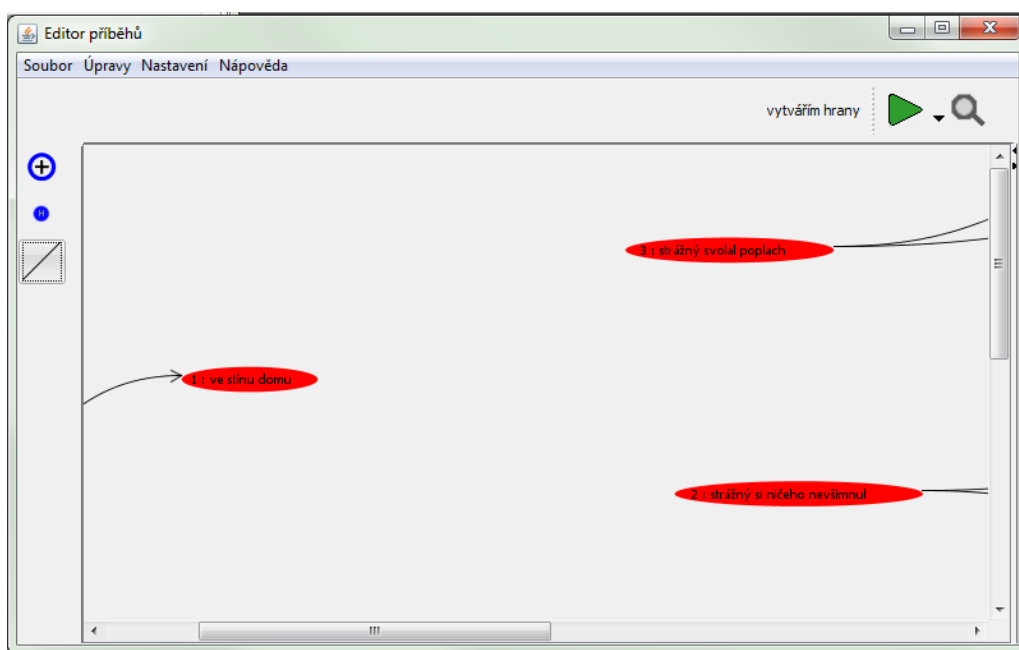
Akce se opět znázorní na plátně.



Obrázek 3.4: Příběh se zakomponovanou rolí náhody

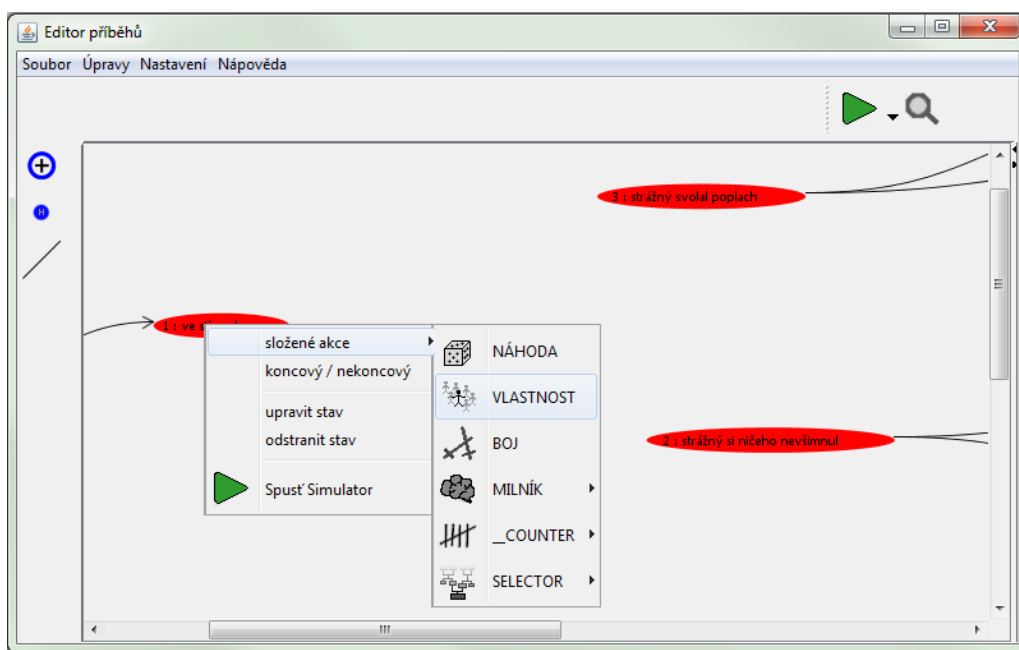
## 4 Složitější větvení příběhu - role Vlastností

Každý člověk je jiný, někdo je velký, jiný malý. Někdo chytrý, silný, nebo šikovný. Jiný už takový není. Každá postava, než začne prožívat svůj příběh, dostane něco do vínku (viz Postava). I ve Vašem příběhu takové vlastnosti mohou hrát roli. Pokud bude třeba porazit nepřítele v boji, povede si lépe silnější. Pokud bude muset prokličkovat mezi spícími, půjde to spíše obratnějším. Když se dostane na přemlouvání, odehraje svou roli charisma ... Představme si tedy situaci, kdy se náš hrdina potřebuje připlížit dost blízko ke strážnému a nepozorovaně mu sebrat svazek klíčů z opasku. Tento úkol můžeme navázat na obratnost postavy. Pokud se to povede, strážný si ničeho nevšimne, jinak vyhlásí poplach.



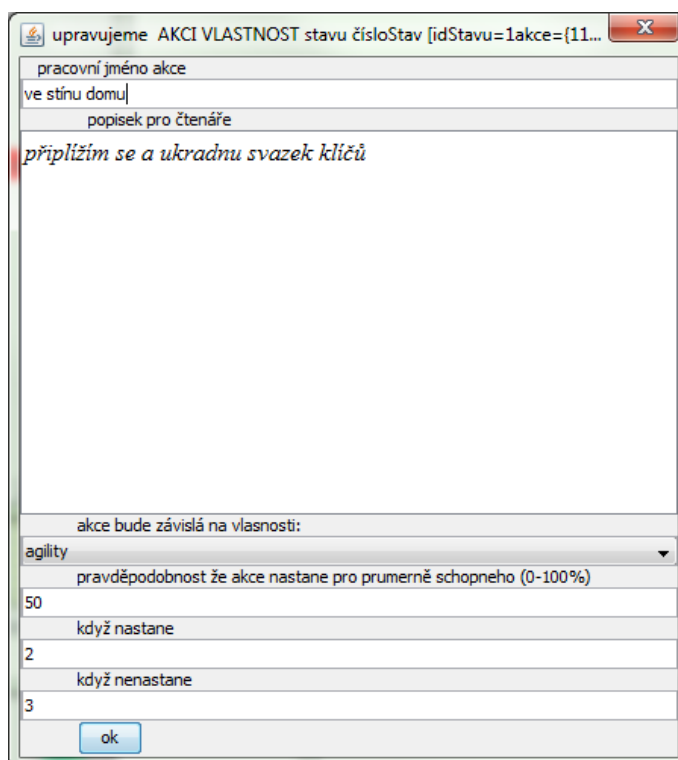
Obrázek 4.1: Fáze příběhu, ve které chceme využít role vlastnosti postavy

Pravým tlačítkem vyvoláme kontextovou nabídku a z akcí vybereme akci vlastnosti.

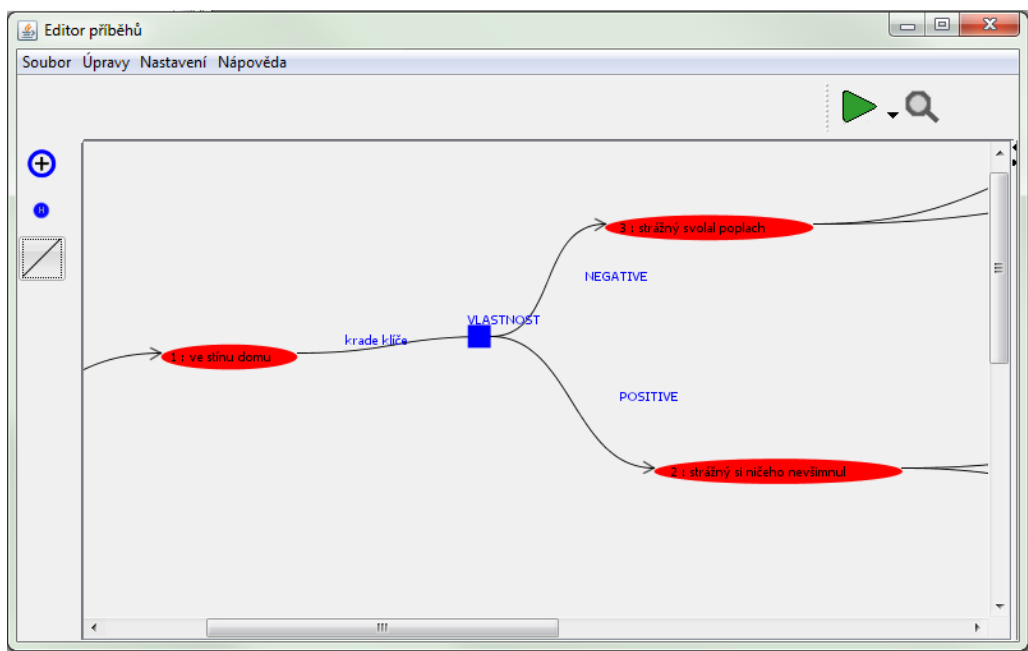


Obrázek 4.2: Rozbalená kontextová nabídka

Vyplníme pracovní popis akce a popis pro čtenáře jako vždy. Níže vybereme vlastnost, na které bude akce závislá. V našem případě je to obratnost. Jako další je třeba nastavit průměrnou obtížnost úkolu. Číslo vyjadřuje šanci v procentech, jakou by měl "průměrný" člověk.



Obrázek 4.3: Okno s nastavením akce závislé na vlastnosti postavy

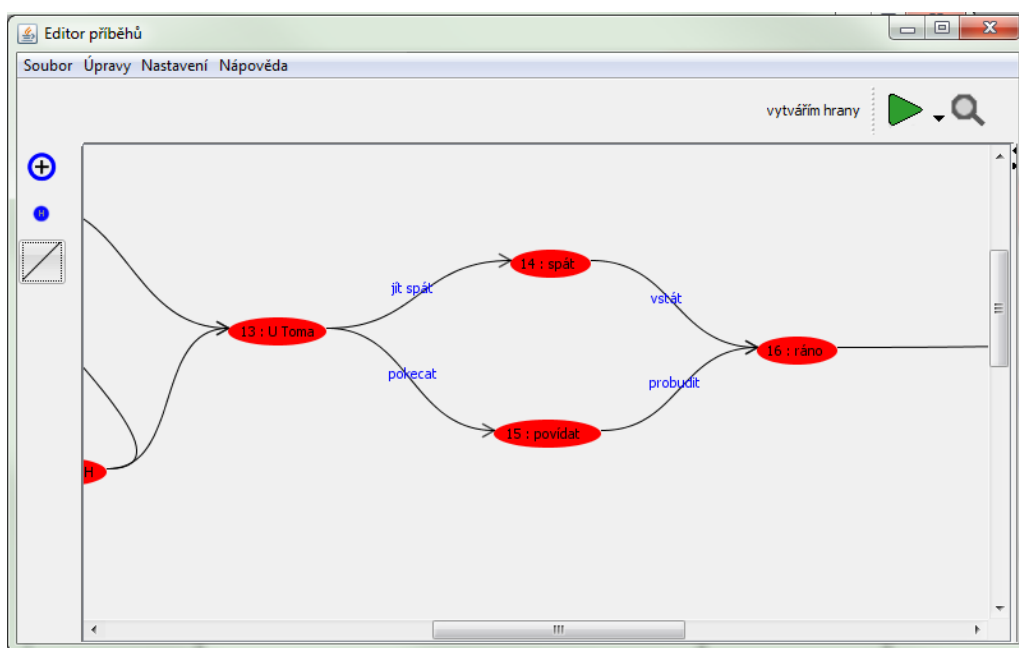


Obrázek 4.4: Příběh se zakomponovanou rolí vlastnosti postavy



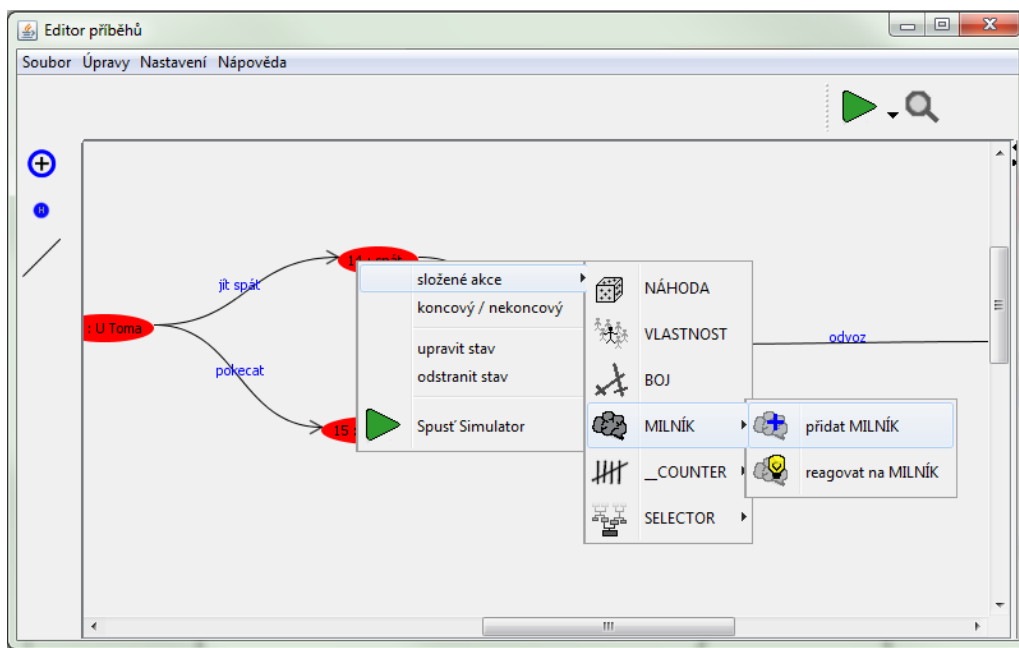
## 5 Složitější větvení příběhu - Používání milníků

Aby bylo možné příběh poté, co se rozvětví do více dějových linií, opět spojit, ale zároveň zachovat informaci o tom, čím si postava prošla, byl vytvořen jednoduchý mechanismus. Po zážitcích z cesty, ať už byly jakékoli, doráží Ben k Tomovi. Dlouho se neviděli, a tak Tom přemlouvá Bena, aby si spolu sedli a popili. Hráč se tedy rozhoduje, zda se nechá Tomem přemluvit, nebo půjde raději spát. Pokud se rozhodne jít spát, dobře se vyspí. Pokud raději bude pít až do časného rána, příliš se nevyspí. Ben se ale probouzí do stejného rána. Až na jeho únavu je vše stejné.



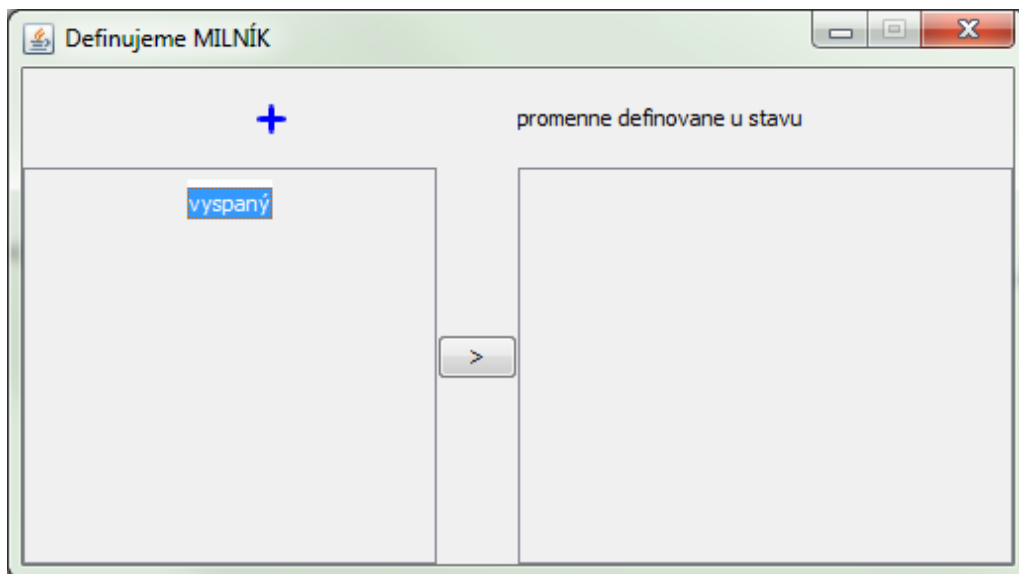
Obrázek 5.1: Část příběhu, která je důležitá pro budoucí vývoj příběhu

Pro tento případ tedy vytvoříme "milník". U stavu, kde se Ben v pořádku vyspal, vyvoláme pravým tlačítkem myši kontextovou nabídku a zvolíme přidat milník.



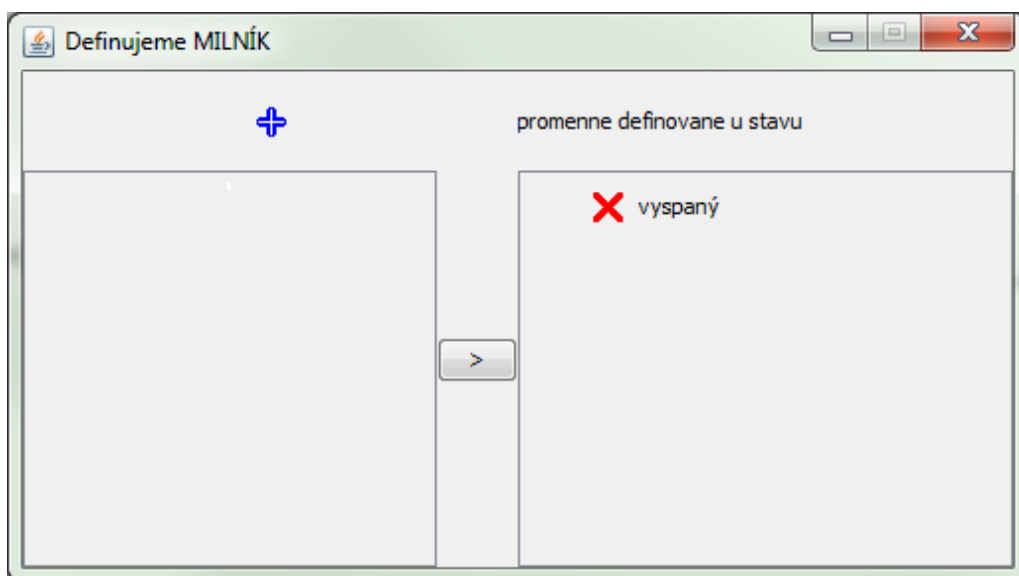
Obrázek 5.2: Rozbalená kontextová nabídka - přidat milník

Kliknutím na modré plus milník pojmenujeme a potvrdíme.



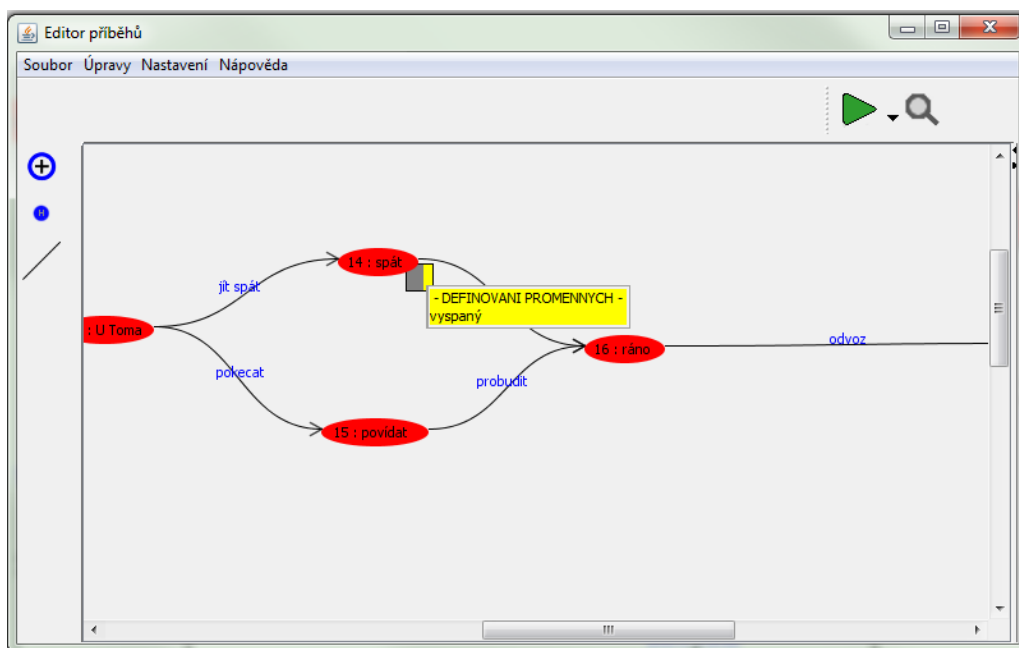
Obrázek 5.3: Vytvořený milník

Následně pouze dvojklikem, nebo označením milníku a kliknutím na šipku, přiřadíme náš milník do pravého sloupce.



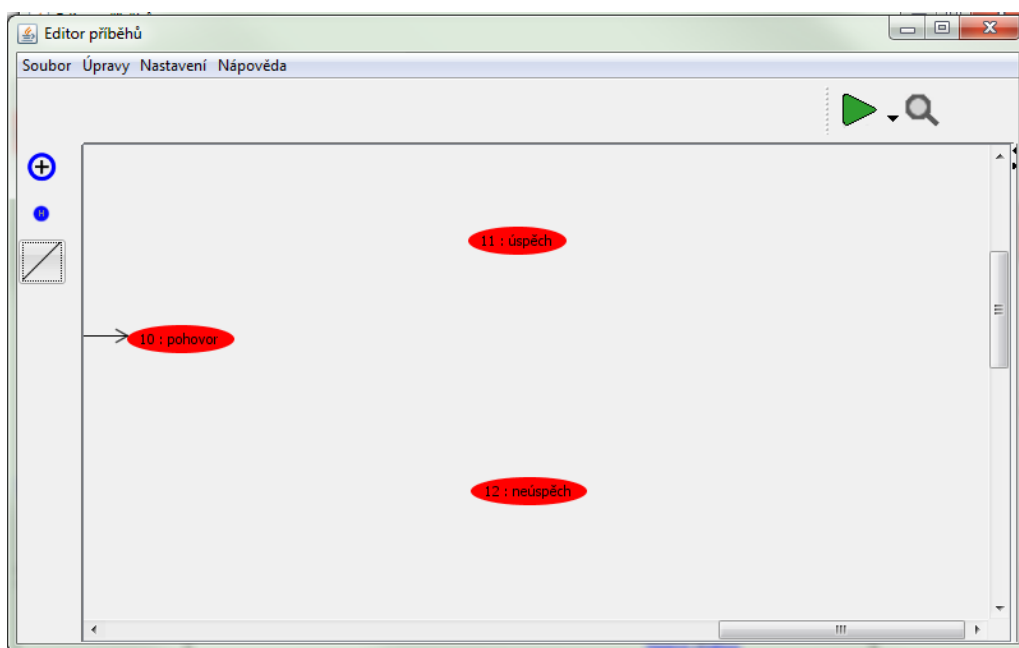
Obrázek 5.4: Milník přiřazený ke stavu

Nyní budeme vědět, jestli šel Ben spát, nebo se nevyspal i později, když se příběh opět spojí do jedné linie. U stavu se zobrazí infobox, ve kterém je vidět náš milník.



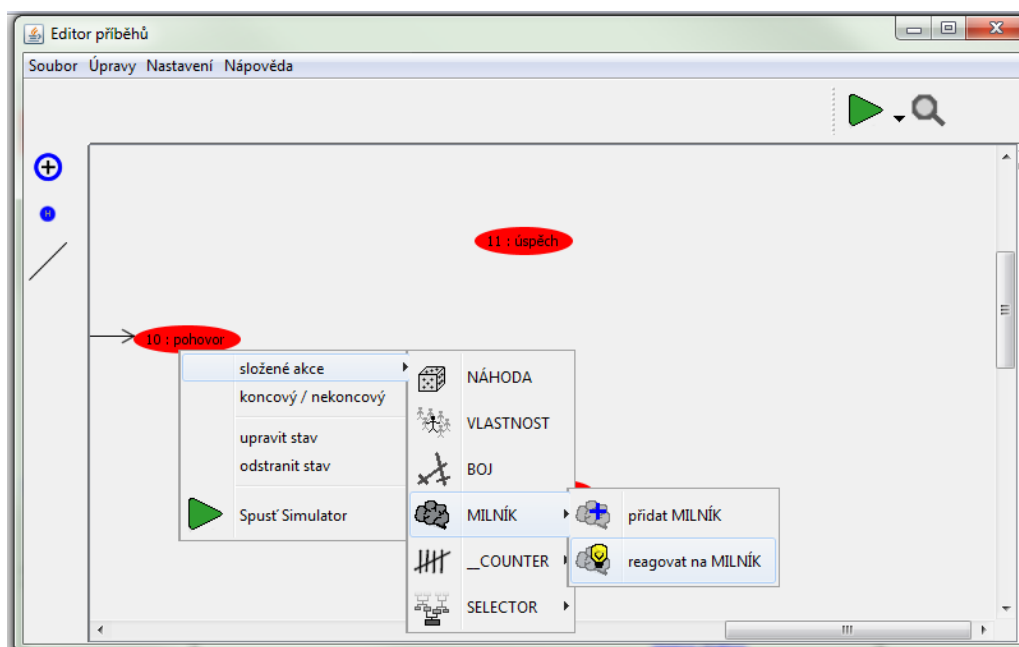
Obrázek 5.5: Milník zobrazený v infoboxu stavu

Ben tedy vyráží opět do města a prožívá svůj příběh. Později dorazí na pracovní pohovor.



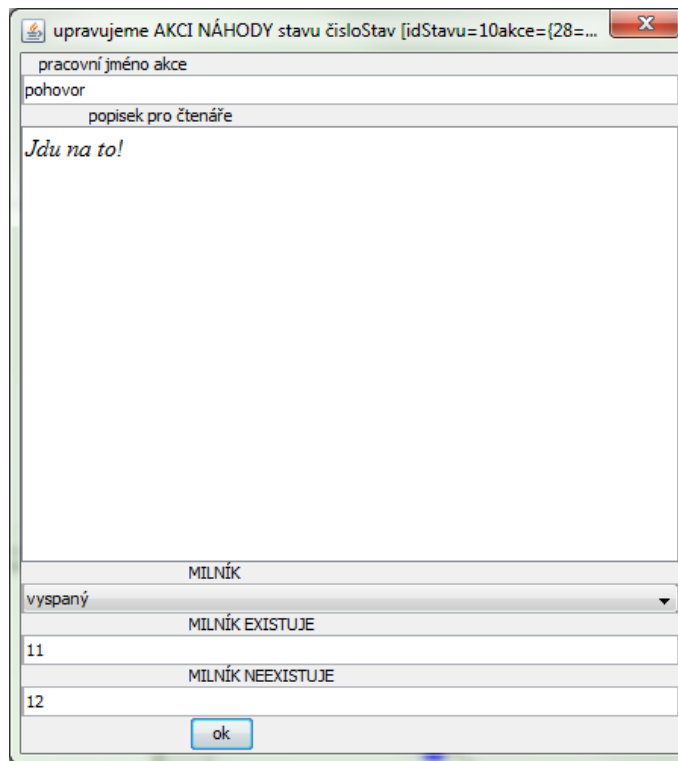
Obrázek 5.6: Situace, ve které hraje roli milník

Chceme, aby jeho úspěch závisel na tom, jak Ben strávil noc. Pokud je nevyspalý, nebude se soustředit a u pohovoru neuspěje. Pravým tlačítkem myši tedy rozbalíme kontextové menu a vybereme možnost reagovat na milník.

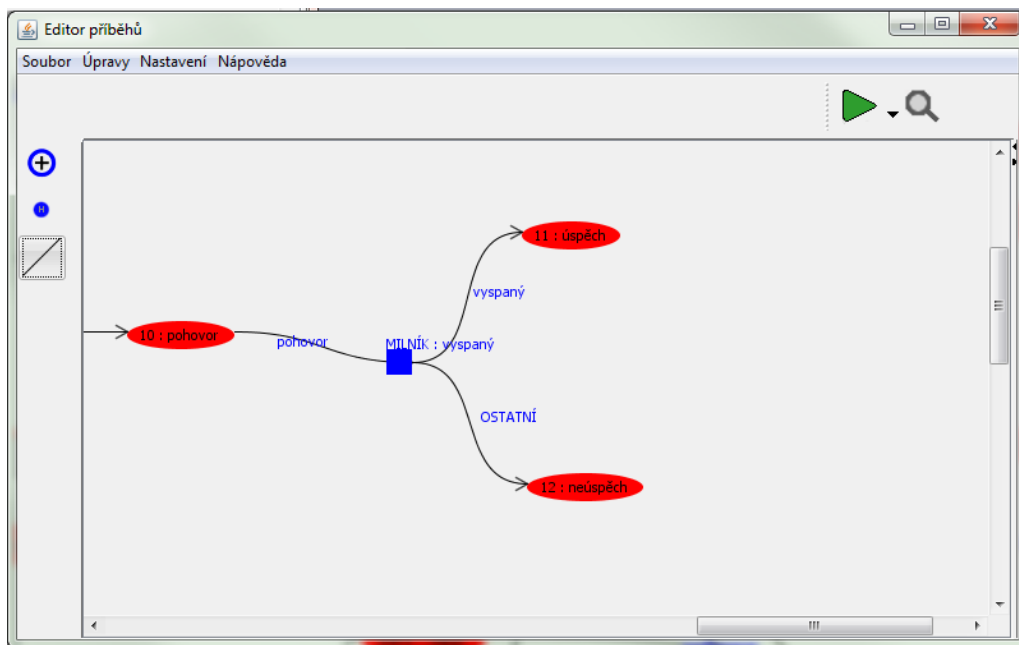


Obrázek 5.7: Rozbalená kontextová nabídka - reagovat na milník

V nabídce vyplníme jméno a popis pro čtenáře jako vždy a z nabídky zvolíme, na který "milník" chceme reagovat. Následně vyplníme čísla stavů, kam se postava přesune pokud tento "milník" v její historii existuje, Ben se vyspal - uspěje, nebo neexistuje, Ben je unavený - neuspěje.



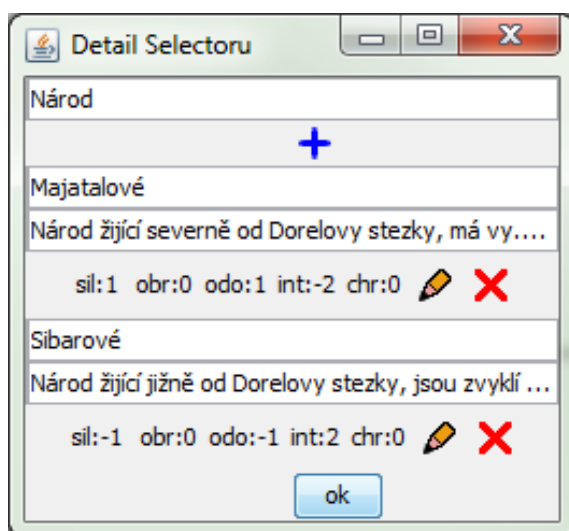
Obrázek 5.8: Okno nastavení reakce na milník



Obrázek 5.9: Reakce na milník zakomponovaná v příběhu

## 6 Složitější větvení příběhu - Používání Selectorů

Představme si, že se náš příběh odehrává v zemi, ve které žijí dva dominantní národy a každý národ má své území. Mezi nimi je ale malá oblast, která nikomu nepatří. Je tu jen několik vesniček. Náš příběh se částečně v těchto oblastech odehrává. Některé vesničky více tíhnou k jednomu či druhému národu. Proto se budou vesničané chovat přátelsky či nepřátelsky, když náš hrdina dorazí do jedné z vesnic, podle toho, ke kterému národu přísluší. Vezmeme tedy příklad dvou národů (viz tvorba postavy) a postavíme postavu do situace, kdy právě do jedné takové vesnice dorazila. Předpokládáme, že jsme už naše národy přidali v nastavení tvorby postavy.

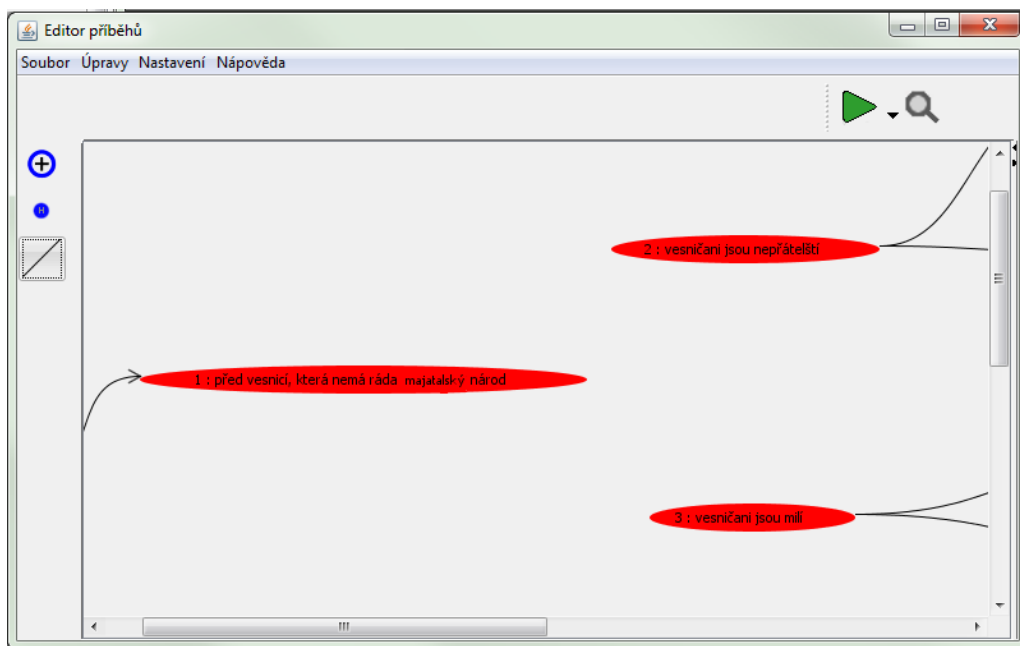


Obrázek 6.1: Vytvořený Selector - Národ s dvěma možnostmi výběru

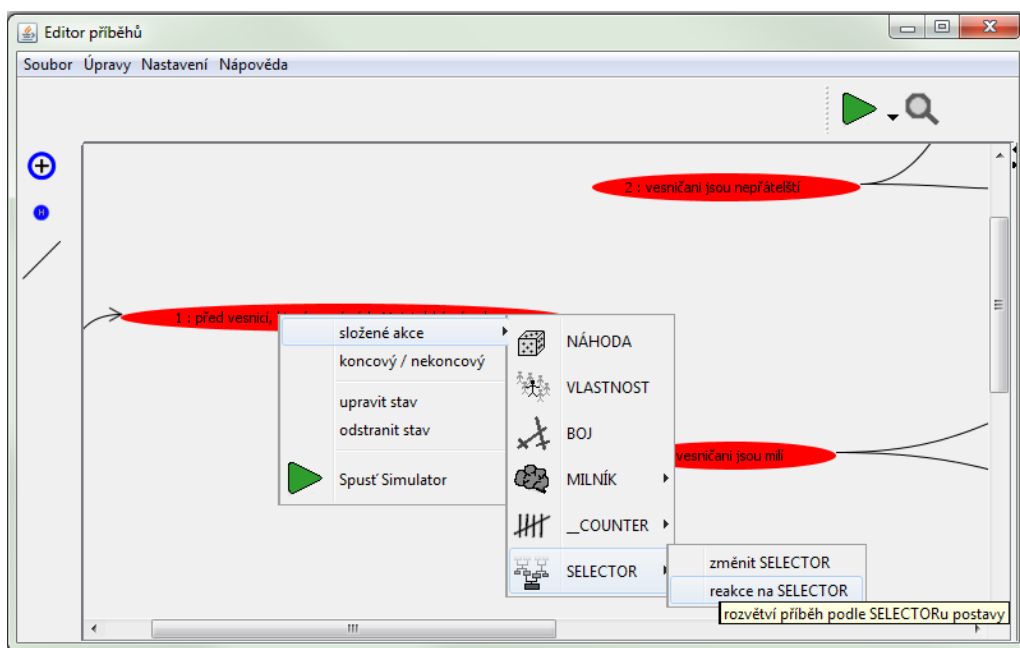
Vesnice je více přikloněná Sibariům a Majtaly nemá v lásce. Vytvoříme tedy stav, kdy postava stojí před vesnicí a rozhoduje se, jestli do vesnice vkročí, nebo ne. Následně připravíme stav, ve kterém se příběh odvíjí poté, co dorazila postava jako Sibari a druhý stav ve kterém je Majatal.

Pravým tlačítkem vyvoláme kontextovou nabídku a vybereme z akcí Selector.

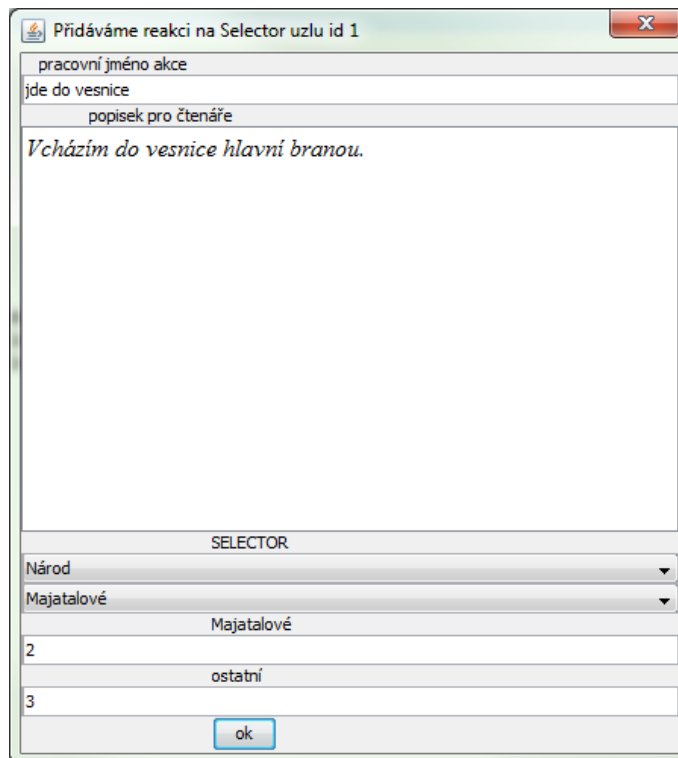
Mimo běžných věcí, jako je pracovní popis akce a popisku pro čtenáře, je třeba vyplnit, na jaký ze Selectorů chceme reagovat. Vybereme tedy námi stvořený národ. Níže vybereme, pro kterou možnost chceme přidat výjimku. Vesničané budou nepřátelští na člena majatalského národa, proto z nabídky vybereme Majatalové.



Obrázek 6.2: Situace, ve které chceme zakomponovat reakci na Selector

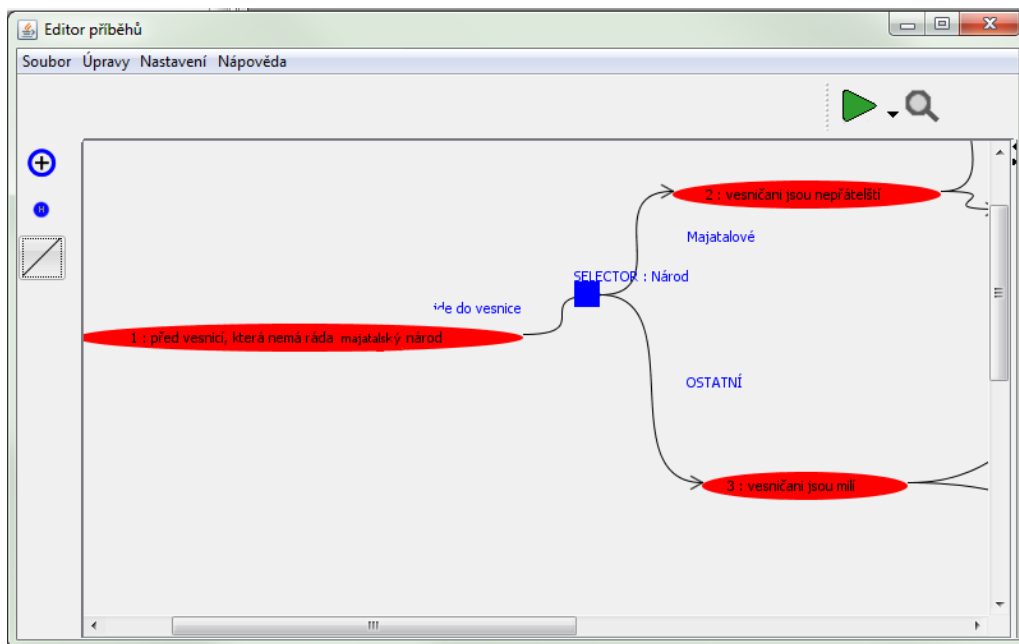


Obrázek 6.3: Rozbalená kontextová nabídka - reakce na Selector



Obrázek 6.4: Okno nastavení akce - reakce na Selector

Jako poslední musíme vyplnit čísla stavů, kam bude postava odeslána. Do stavu č.3 bude přesunut Majatal a do stavu č.4 všichni ostatní.

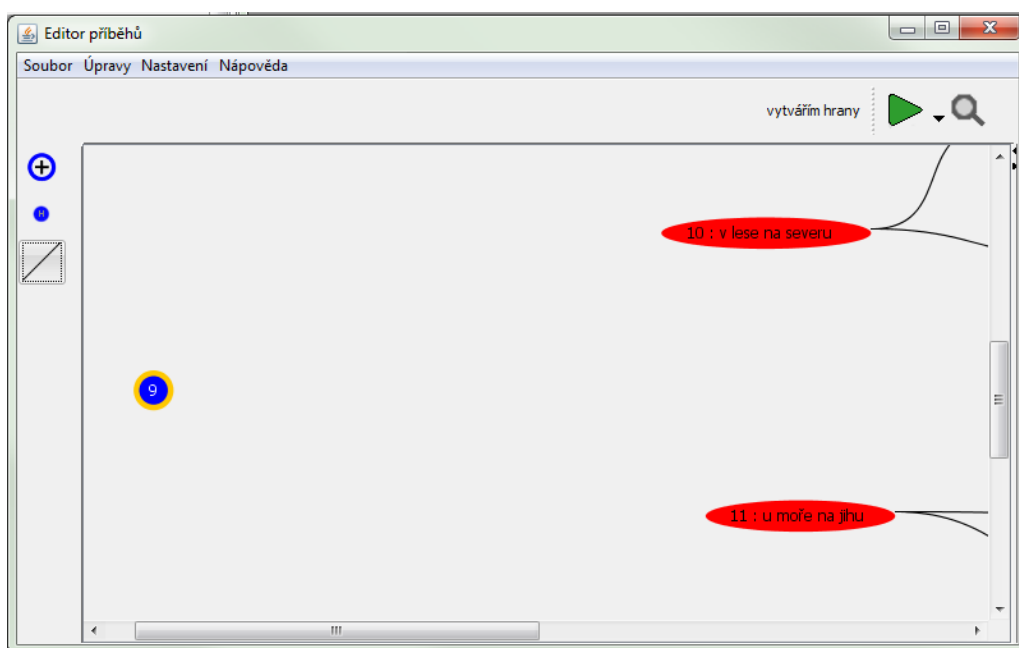


Obrázek 6.5: Příběh se zakomponovanou reakcí na Selector

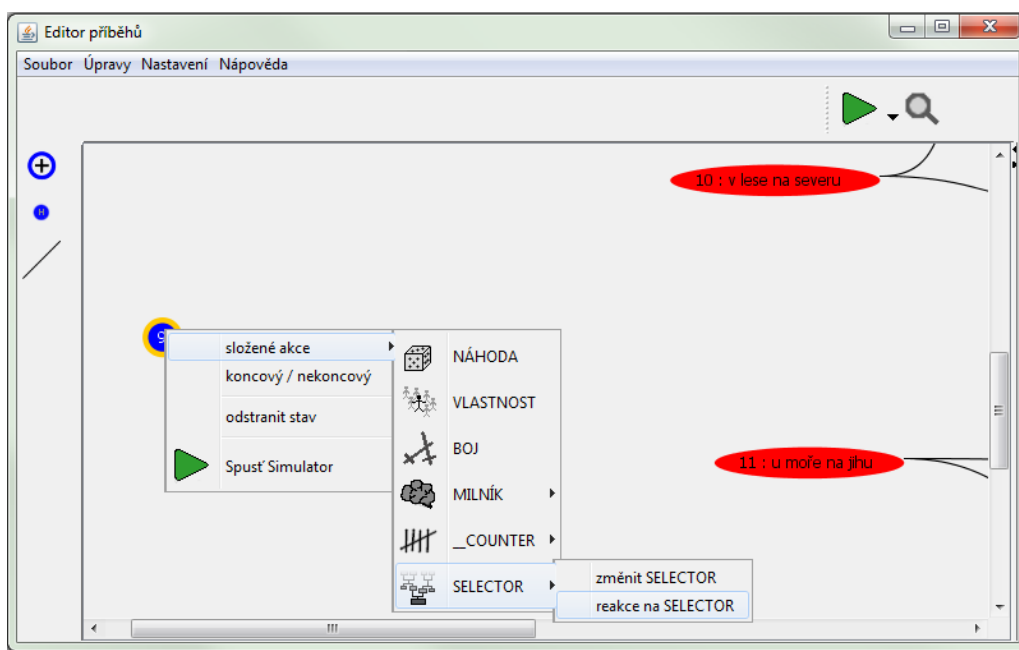


## 6.1 Použití Selectorů na vytvoření různých začátků

Selector se dá také použít k vytvoření rozdílných startů jednotlivých národů, gangů, ras, .... Pokud jako startovní stav v nastavení příběhu nastavíme holder (viz práce s Holdery) a z něj poté vytvoříme akci Selectoru, ihned po spuštění se postava ocitne buď v tom či onom stavu. Celý příběh se může odvíjet úplně jinak pro členy jednotlivých národů, v našem případě se liší jejich startovní pozice, Majatalové žijí na severu, ale Sibariové na jihu. Úkol hrdiny je ale stejný, vyřešit tento staletý problém nesnášenlivosti. Liší se tedy jejich cesta k hranici, kde se příběh odehrává, ale dále je podobná. Jen v některých klíčových momentech se opět rozdělí.



Obrázek 6.6: Dva rozdílné startovní stavy pro různé národy - Holder jako start



Obrázek 6.7: Reakce na Selector ihned při startu příběhu

## 7 Složitější větvení příběhu - Používání Counterů

V průběhu příběhu se může hodit něco počítat. Právě k tomu slouží Counter - počítadla. Mohou to být pouze informační sdělení pro hráče, například pokud chceme ukázat, kolik našel hledaných artefaktů, nebo kolik sebral magických bylin. Tato počítadla lze ale také použít pro větvení příběhu.

Představme si například, že hlavní postava, jeden z vězňů - Solo, se při útěku zmocnil revolveru. Má k dispozici 6 ran. Jak je hráč využije, je pouze na něm.

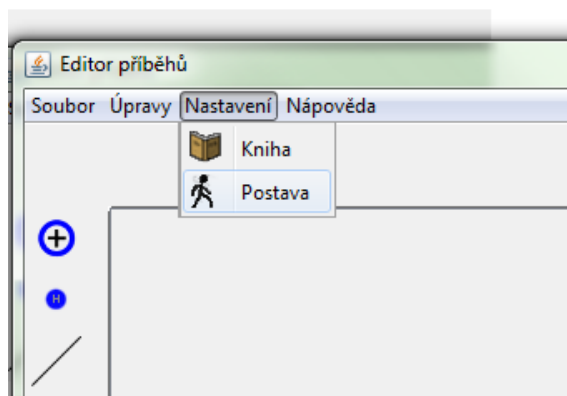
Postupně během příběhu můžeme počítat, kolik ran hráč vystřelil. Pokud bude během svého útěku plýtvat, budou mu chybět, až je bude nejvíce potřebovat. Zároveň, pokud ho chytí, jej budeme chtít potrestat za rozsévání smrti podle počtu zabitých strážných.

Budeme tedy potřebovat dvě počítadla:

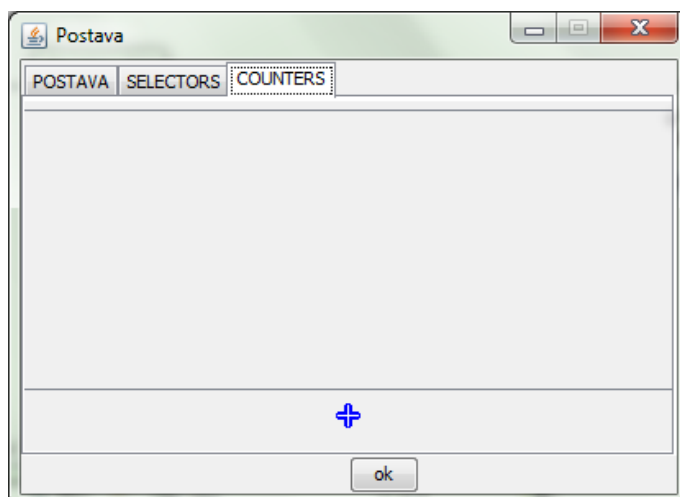
- vystřelené kulky
- zabití

V menu Nastavení->Postava vybereme záložku Counter (viz obr. 7.1 str. 77)

Kliknutím na modré plus přidáme dvě počítadla.

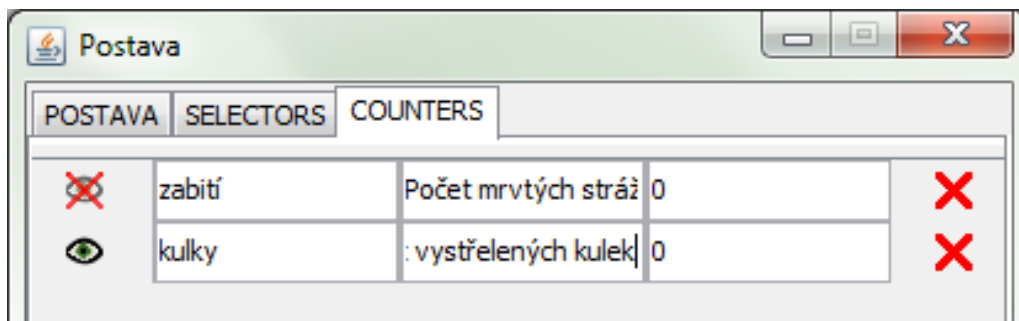


Obrázek 7.1: Menu Nastavení : Postava



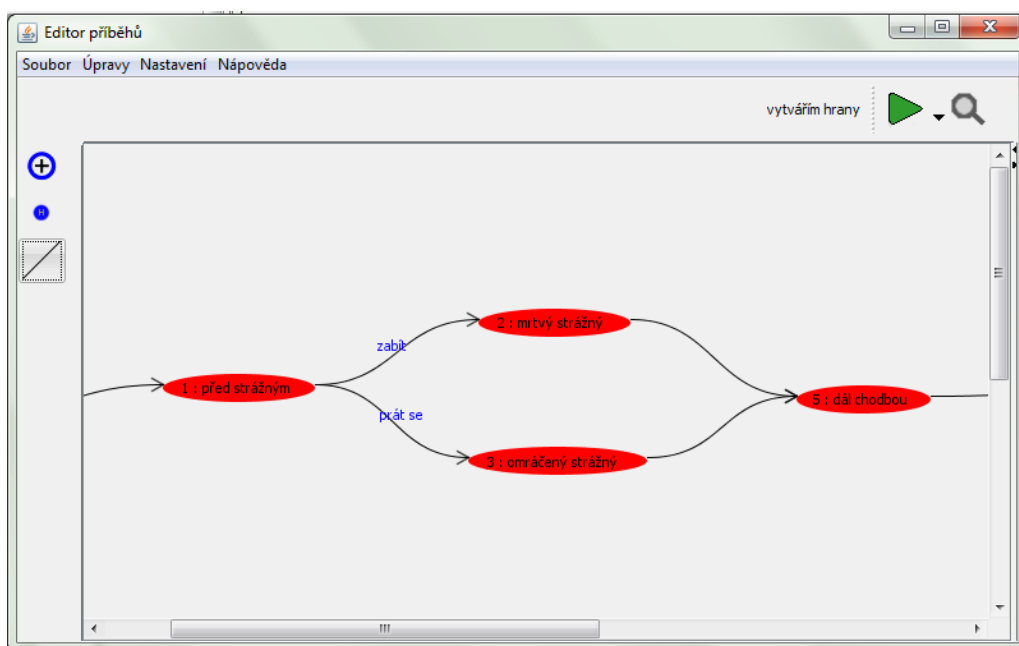
Obrázek 7.2: Okno souhrnu počítadel

Dále vyplníme jejich jména, krátké popisky pro čtenáře a počáteční hodnoty. Protože nechceme, aby čtenář věděl, že počítáme mrtvé strážné, nastavíme neviditelnost tohoto Counteru (škrtnuté oko).



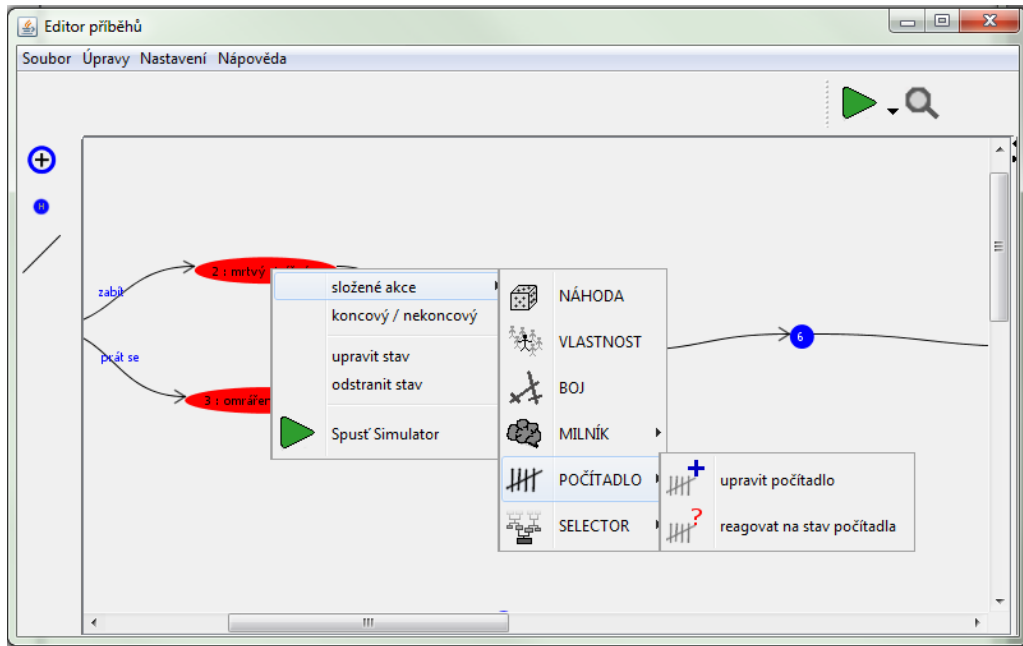
Obrázek 7.3: Přidaná počítadla, pouze jedno viditelné pro čtenáře

Nyní postavíme hráče do situace, kde si může vybrat, zda strážného zastřelí, nebo se pokusí situaci vyřešit jinak.



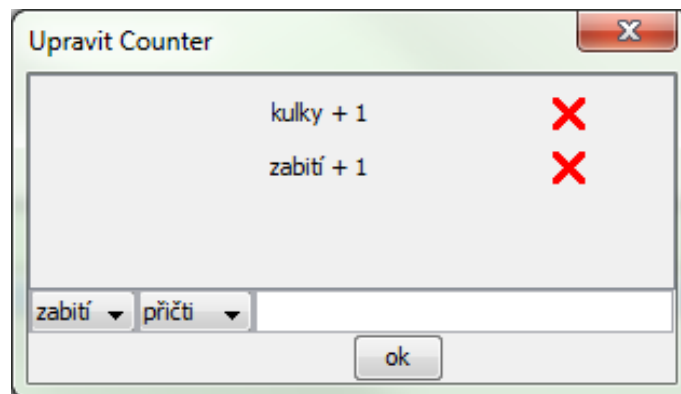
Obrázek 7.4: Situace, která upravuje počítadla

Z kontextové nabídky konkrétního stavu zvolíme možnost upravit počítadlo.



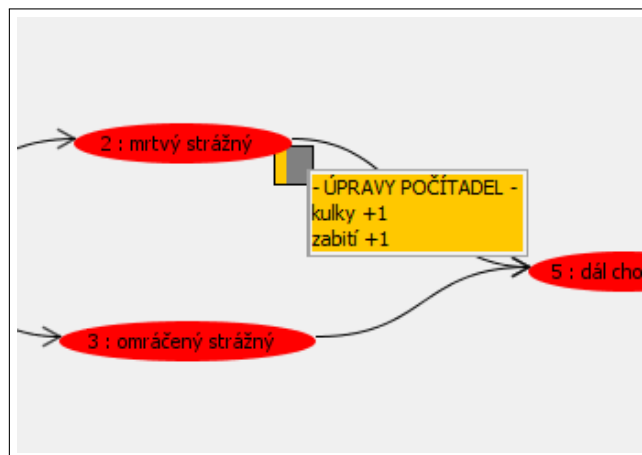
Obrázek 7.5: Kontextová nabídka stavu - práce s počítadly

V nově zobrazeném okně navolíme úpravy, které chceme s počítadlem provést. V našem případě k oběma počítadlům přičteme jedničku.



Obrázek 7.6: Okno úpravy Counteru : nastavené úpravy

Tyto úpravy se zobrazí v infoboxu stavu. Na první pohled je tedy vidět, že ve stavu jsou nějaká počítadla upravována. Po rozkliknutí infoboxu se zobrazí podrobnosti.

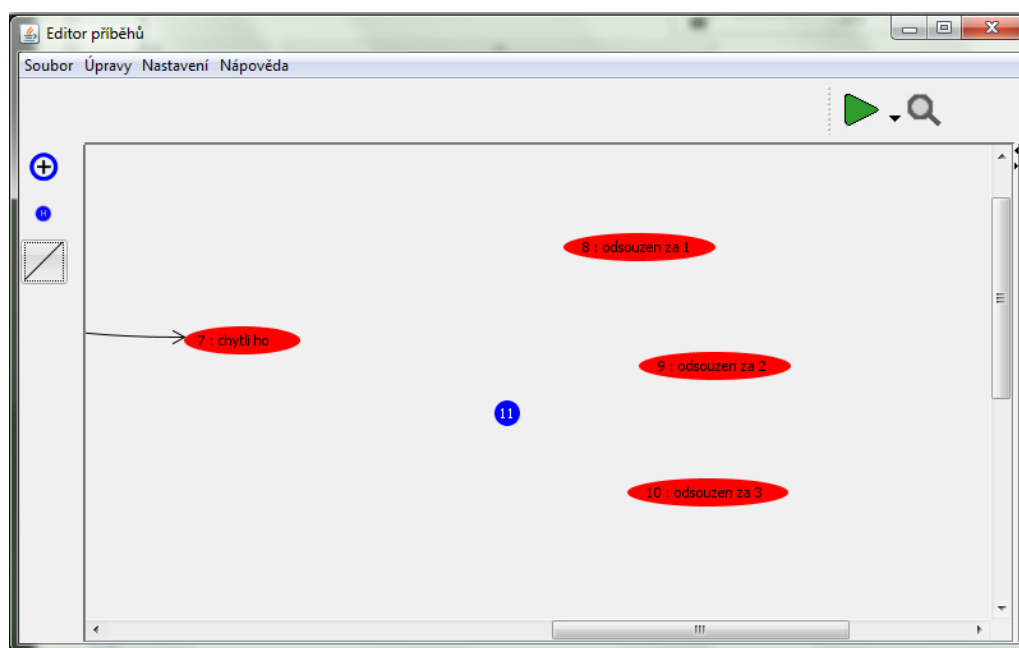


Obrázek 7.7: Infobox stavu : souhrn upravovaných počítadel

Takové úpravy počítadel umístíme do našeho příběhu pokaždé, když vězeň Solo použije zbraň. Nemusí ji použít pokaždé proti člověku. Někdy se může hodit například k odstranění zámků a tak podobně.

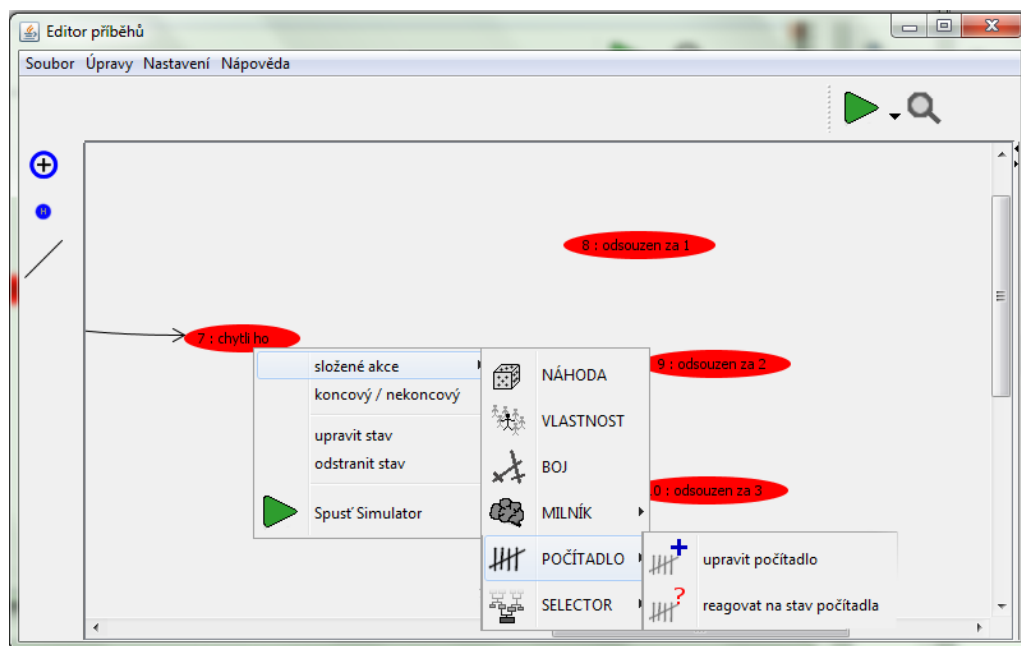
Pokud ke konci příběhu, nebo kdykoli v průběhu, budeme chtít reagovat na stav počítadla, uděláme to z kontextové nabídky stavu - volba reakce na počítadlo.

V tomto konkrétním případě Solo chytily, a ten stane před soudem. V souvislosti s jeho předchozími činy, ho chceme odsoudit. Připravíme si stavy, ve kterých je odsouzen za vraždu jednoho, dvou i tří strážných. Také budeme potřebovat jeden Holder, abychom mohli vyhodnotit složenou podmínku.



Obrázek 7.8: Situace, ve které vyhodnocujeme stav počítadla

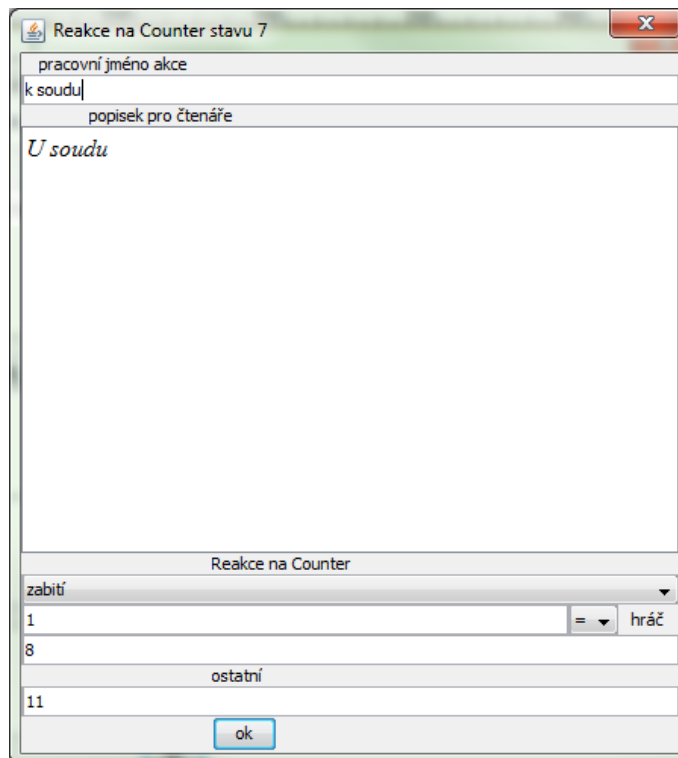
Vybereme možnost reakce na stav počítadla u stavu, ze kterého k soudu přistupuje.



Obrázek 7.9: Kontextová nabídka stavu : práce s počítadly

Nejdříve vytvoříme první část podmínky. Když Solo předstoupí před soud, napřed oddělíme případ, kdy zabil při útěku pouze jednoho člověka.

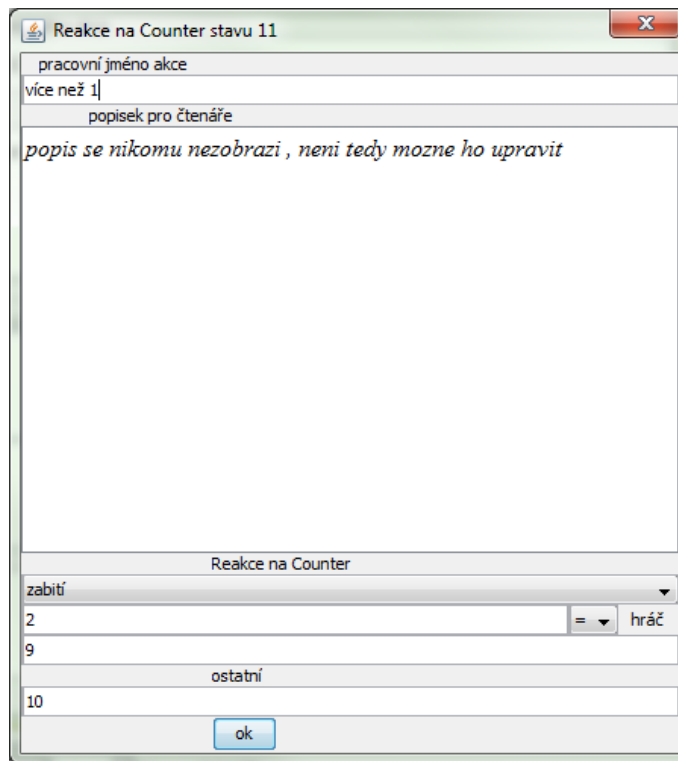
Pokud tomu tak je, pošleme hráče do stavu č. 8, ve kterém si vyslechne rozsudek za jedno zabití. Ostatní pošleme do Holderu č. 11.



Obrázek 7.10: Okno akce Counteru - první část

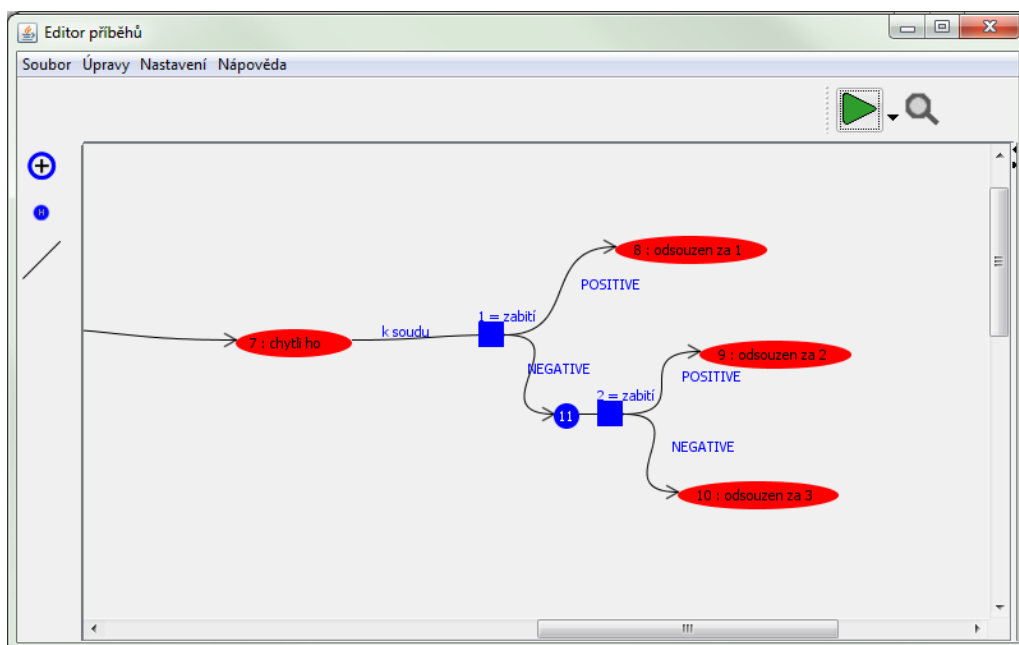
Následně vytvoříme druhou část vyhodnocení a na Holder navážeme akci, ve které pošleme hráče do stavu č. 9, pokud během svého útěku zabil 2 lidi. Ostatní (v tomto případě mohl zabit maximálně 3 lidi) pošleme do stavu č. 10. Kdybychom chtěli zapojit více výsledků soudu, postupovali bychom analogicky a použili více Holderů, přičemž bychom v každém oddělili jeden případ.





Obrázek 7.11: Okno akce Counteru - druhá část (za holderem)

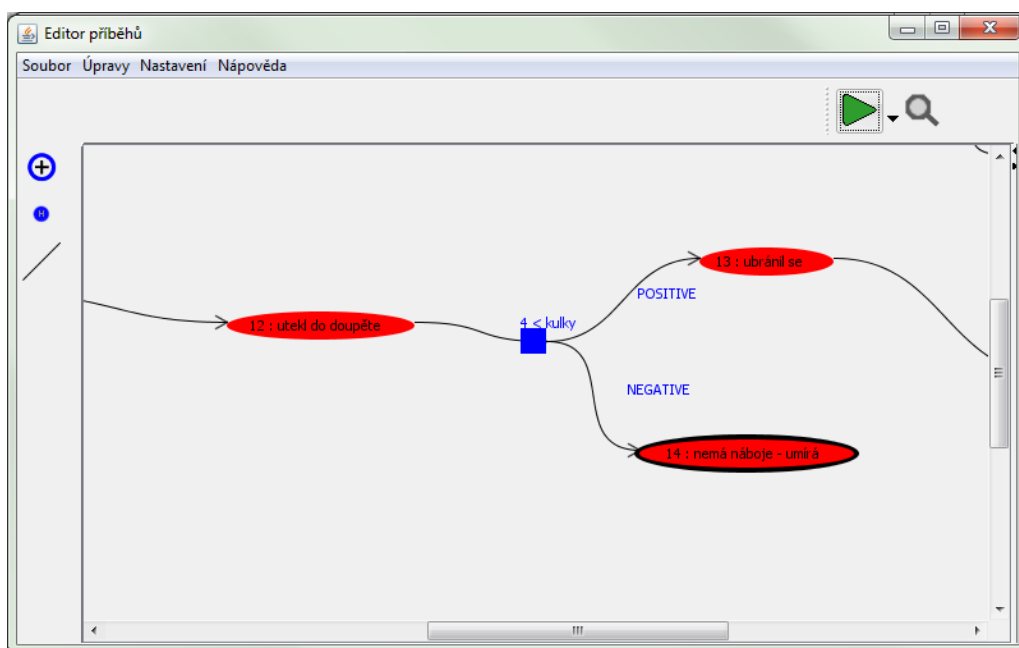
Takto vypadá celý model situace soudu reagující na předchozí počínání hráče.



Obrázek 7.12: Příběh se zakomponovanou akcí závislou na počítadlu

Analogicky můžeme do příběhu zakomponovat podobný mechanismus, ve kterém, pokud se podaří utéct z vězení a dostat se až do doupěte, kde na Sola čekají někdejší komplicové, bude záležet na tom, kolik ran v revolveru zůstalo. Budou z něj totiž

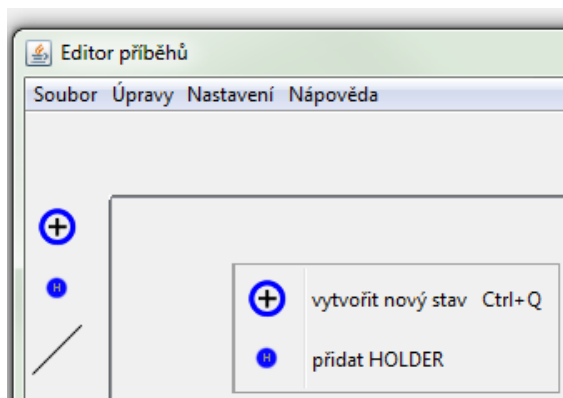
chtít dostat, kam schoval peníze po jejich společné loupeži, za kterou si šel Solo sednout.



Obrázek 7.13: Příběh se zakomponovanou reakcí na počítadlo kulek

## 8 Holder

Holder byl vytvořen pro rozšíření možnosti vyhodnocování akcí bez interakce se čtenářem. Ač se to nezdá, je to mocný nástroj, který přináší spoustu možností. Lze jej přidat pravým tlačítkem -> přidat HOLDER



Obrázek 8.1: Kontextová nabídka vyvolaná na plátně

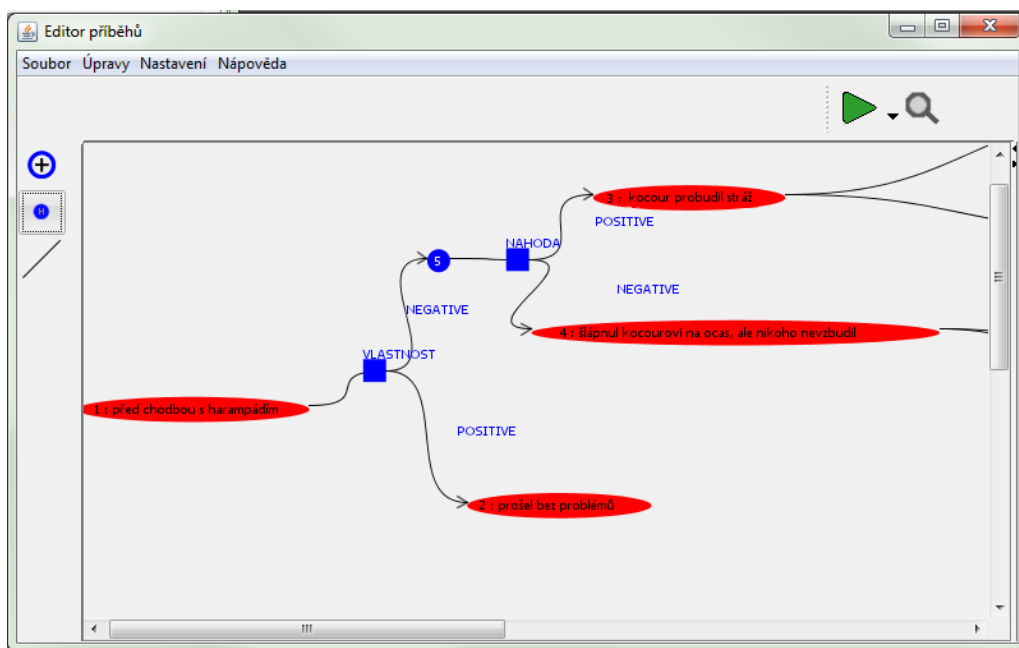
Existují některá pravidla, která musíme dodržet, pokud holder chceme použít:

- z holderu smí vycházet jen jedna akce
- v akci, která vychází z holderu, nelze upravovat popis pro čtenáře (nikde se nezobrazí )

Mohlo by se zdát, že nás holdery jen omezují, ale jejich přínos je velký.

### 8.1 Holder - přesměrování jedné akce na druhou

Pokud bychom například chtěli udělat zajímavější vyhodnocení nějakého sledu událostí, mohli bychom to udělat například takto. Postava prochází chodbou plnou harampádí. Její obratnost rozhodne, jestli šlápne spícímu kocourovi na ocas nebo ne. Pokud ano, kocour se s leknutím probudí, zachrčí, výskne, ale jestli probudí strážného, je jen dílem náhody. Může spát tvrdě, nebo jen klimbat.



Obrázek 8.2: Příběh se zakomponovaným přesměrováním výstupu jedné akce na druhou

Místo holderu by tu mohl být samozřejmě běžný stav, ale pokud by tomu tak bylo, nejdříve bychom čtenáře informovali, že kocoura probudil a museli ho nechat vybírat jen z jedné akce. Pravděpodobně by to bylo něco jako "pokračovat". Pokud se tomuto chcete vyhnout, použijte právě holder. Ten čtenář nezaznamená. Holder totiž ihned po vstupu přesměruje hrdinu na akci, která je na něm navázaná. Z tohoto důvodu také může na holder navazovat jen jedna akce a nelze k ní přidat popis pro čtenáře, protože se nikde nezobrazí.

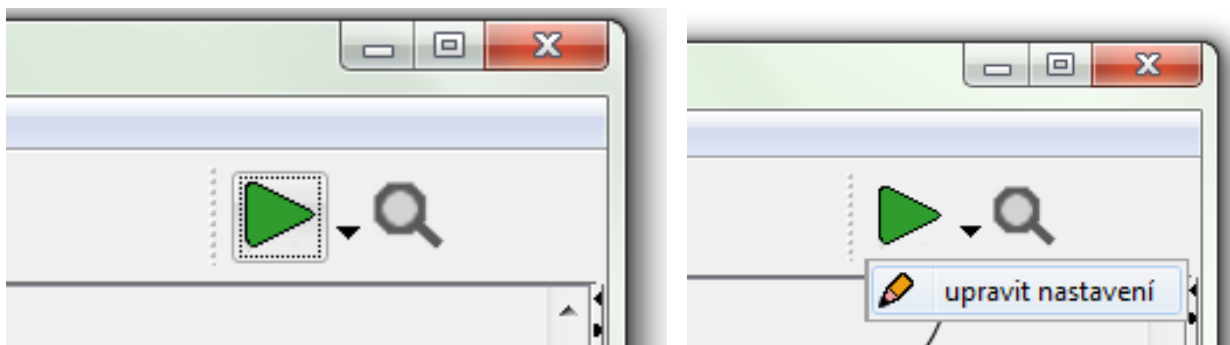
## 8.2 Holder - vytvoření oddělených startů

Holder může být pomůckou pro vytvoření oddělených startů jednotlivých gangů, ras, národů atp. Tato problematika je podrobně popsána v kapitole 6 str. 74.

## 9 Simulátor

Aby bylo možné si vytvořený příběh projít z pohledu čtenáře a také ověřit namodelovanou situaci, je v Editoru zakomponován Simulátor. Pro jeho spuštění stačí kliknout na symbol zelené šipky v pravém horním rohu (viz obr. 9.1a str. 87).

Pokud je nastaven startovní stav, simulátor se spustí na začátku příběhu. Jestliže startovní stav nastaven není, budete požádáni o jeho doplnění.



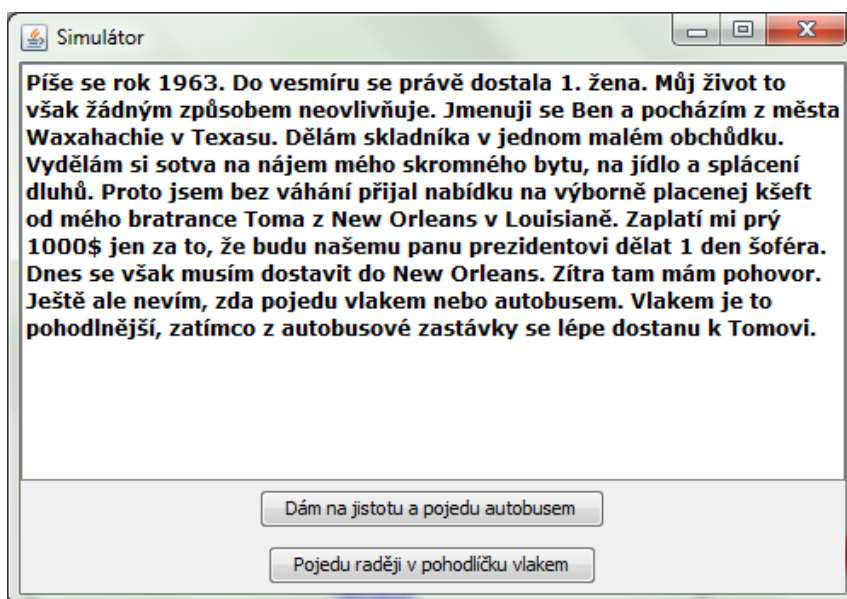
(a) Pro spuštění Simulátoru : zelená šipka

(b) Rozšířené spuštění Simulátoru

Obrázek 9.1

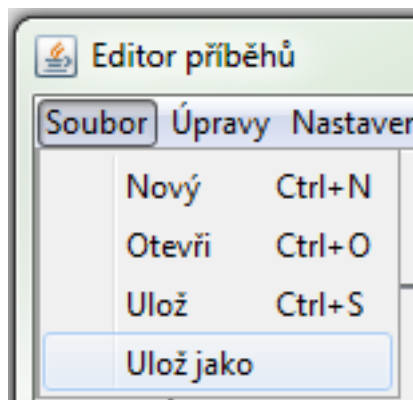
Pokud jsou ve Vašem příběhu vytvořeny nějaké Selectory při prvním spuštění se zobrazí okno, ve kterém je nastavíte. Pokud budete chtít tuto volbu později změnit, stačí rozbalit rozšířenou nabídku Simulátoru a upravit nastavení (viz obr. 9.1b str. 87).

V horní části je zobrazen text stavu, ve spodní části možné akce. Pro výběr akce stačí kliknout na některou z nich. Simulátor se přepne do dalšího navazujícího stavu (viz obr. 9.2 str. 87).



Obrázek 9.2: Okno Simulátoru

## 10 Ukládání a distribuce Gamebooku



Obrázek 10.1: Menu Soubor :  
Uložit jako

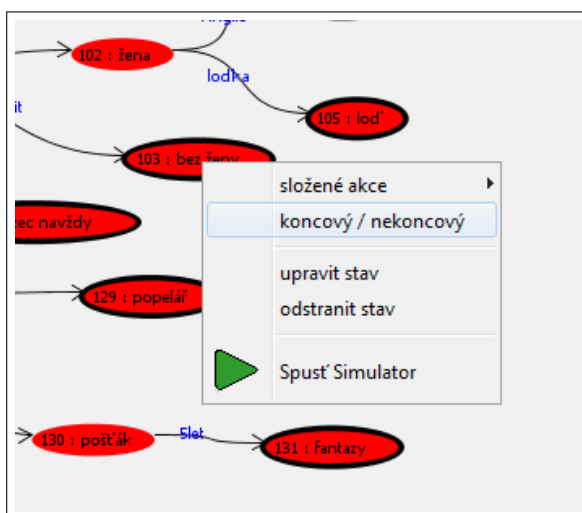
Pokud chceme rozpracovaný příběh uložit, zvolíme v menu Soubor volbu Uložit jako.

Pokud jste tak ještě neučinili, můžete být požádáni o vyplnění potřebných údajů. Mezi povinné údaje patří titul knihy a počáteční stav, tedy stav, ve kterém začíná dobrodružství.

Po uložení se vytvoří soubor. Tento soubor je možné později v editoru otevřít pro editaci, popřípadě po jeho dokončení jej nějakým způsobem distribuovat ke čtení.

### 10.1 Nezbytné úpravy před distribucí

Před distribucí je nutné všechny stavy, ve kterých příběh končí, označit. V kontextové nabídce stavu vybereme možnost koncový/nekoncový. Stav se ohraničí černě.



Obrázek 10.2: Označení koncových stavů