**DnD Telegram Bot**

**Dokumentace**

Filip Vopálenský

Obsah

[Anotace 3](#_Toc126183707)

[Přesné zadání 3](#_Toc126183708)

[Mapa 3](#_Toc126183709)

[Ulice 3](#_Toc126183710)

[Místa 3](#_Toc126183711)

[Postavy 3](#_Toc126183712)

[Poslíček 3](#_Toc126183713)

[Hlídka 3](#_Toc126183714)

[Kočka 3](#_Toc126183715)

[Frakce 3](#_Toc126183716)

[Úkoly 4](#_Toc126183717)

[Fáze úkolů 4](#_Toc126183718)

[Dones předmět 4](#_Toc126183719)

[Najdi 4](#_Toc126183720)

[Uteč 4](#_Toc126183721)

[Linie úkolů 4](#_Toc126183722)

[Modifikátory 4](#_Toc126183723)

[Save file 5](#_Toc126183724)

[Stav hráče 6](#_Toc126183725)

[Stavy úkolových linií 6](#_Toc126183726)

[Stavy postav 6](#_Toc126183727)

# Anotace

Problémem, který řeší můj program, je nuda. Jedná se totiž o hru, a to konkrétně o hru výpravnou inspirovanou textovými adventurami a Dračím doupětem. Uživatel, tudíž hráč, se ocitne ve městě zvaném Kritraven a musí se mu podařit schovat či uprchnout dříve, než ho dopadne hlídka. Hra je rozdělena na kola a v každém z nich musí hráč pečlivě volit svá rozhodnutí a uvážit své kroky. Hráč samozřejmě není ve městě sám, ba naopak může potkat několik NPC, se kterými může interagovat, a tak ovlivnit výsledek hry.

# Přesné zadání

# Mapa

Mapa Kritravenu, viz obrázek, sestává ze 33 ulic a N navštívitelných míst. Jedná se tedy o graf, jehož vrcholy jsou ulice a hrana mezi ulicemi (či ulicí a místem) existuje právě tehdy, když se obě ulice protínají či sbíhají.

## Ulice

Seznam ulic lze nalézt v souboru streets.csv. Každá z nich je přesně definovaná svým ID (od 0 - 33), svým jménem v češtině a seznamem ID dalších připojeným míst a ulic.

## Místa

Podobné jako ulice, avšak jsou uloženy v places.csv. Jejich zvláštní vlastností je, že se jedná o listy grafu, tudíž do nich vede právě jedna hrana.

# Postavy

Seznam všech postav a údaje o nich nalezneme v souboru characters.csv. Každá z nich má svoje jedinečné ID, jméno v češtině, seznam ID úkolů, které musí splnit, ID ulice ve které začíná a ID ulice či místa kam půjde, pokud vše splní. Pokud bude v *end\_street\_ID* hodnota -1, znamená to, že postava se bude dále volně a náhodně pohybovat po městě.

## Poslíček

Jedná se o archetyp postavy, která má pouze jeden úkol, a to Dones předmět (ID 0), a poté se bude volně pohybovat po mapě až do konce hry.

## Hlídka

## Kočka

# Frakce

Další nezanedbatelnou charakteristikou každé postavy je frakce, pro kterou pracuje. Výpis všech lze nalézt v souboru fractions.csv. Mimo ID a jména v češtině zde můžeme nalézt i vztahy mezi frakcemi a ID místa jejich základny.

Ohodnocení vztahů: 3 = nasadíme za vás svůj život, 2 = nevadíme si, 1 = klidně tě udám, 0 = zabiju tě tady a teď, -1 = nedefinováno (např. vztah sám se sebou nebo nemá smysl určovat)

Výjimky: nikam nepatřící postavy nemají residenci (ID = -1)

# Úkoly

Seznam všech úkolů a jejich všemožné podrobnosti jsou uloženy v souboru quests.csv. Řekl bych, že se jedná o nejkomplexnější vstupní data ze všech. Lze zde totiž určit mnoho modifikátorů, které přesně určí, jak mají jaké úkoly (a jejich fáze) probíhat a jak se mají postavy tento úkol plnící chovat.

## Fáze úkolů

Každý, jakkoliv komplikovaně dosažitelný cíl se dá rozložit na posloupnost několika fází. A když postupně tyto fáze splníme, dosáhneme určitého vyššího cíle.

### Dones předmět

Příklad velice základního úkolu. Postava, které byl úkol zadán, se musí dostavit na určené místo, kde převezme daný předmět. Poté se musí dostat do zadané ulice, ale cestou se musí vyhýbat určeným ulicím či postavám. Proč? Představme si, že postava něco pašuje, určitě by nechtěla upoutat pozornost městské hlídky v nějaké uličce a už vůbec by nechtěla procházet ulicí, kde sídlí.

### Najdi

### Uteč

## Linie úkolů

Když dáme dohromady více fází vzejde nám schéma nějakého většího plánu. Avšak každá fáze může mít výsledek buď pozitivní či negativní. Řekněme, že poslíček dostane předmět, aby ho doručil na určené místo. Pokud tak udělá, dostane výměnou za to jiný předmět, který má donést zpět. Pokud by mu však něco zabránilo v doručení tohoto balíčku, tak rozhodně nedostane nový balíček, ale nejspíš i někdo začne řešit, kam se balíček či poslíček poděl.

Linie úkolů jsou tedy implementovány jako stromy, kdy každý prvek má dva syny, z nichž jeden zastupuje další krok v případě úspěšného provedení fáze a druhý zastupuje následující krok při neúspěchu jeho rodiče.

Mějme tedy strom, jehož každý vrchol sestává z fáze úkolu, ke které je přidruženo několik modifikátorů, které specifikují například kdo, co a jak má provést danou fázi.

### Modifikátory

Všechny modifikátory jsou reprezentovány řetězcem znaků.

Kdo – Zde můžou nastat 2 varianty:

Fáze je určena pro určitou postavu

* Modifikátor := “char“ + ID všech postav oddělené středníky

Fáze je pro kohokoliv z určité frakce

* Modifikátor := “frac“ + ID určené frakce

Momentálně nedotažená funkce. Tento způsob sice funguje, ale když mise probíhá a má ji zadaná určitá náhodná postava, tak může být v dalším kole přidělena jiné postavě, musí se tedy implementovat jakési sledování, že někdo danou fázi plní. Podobně se zadáním více určitých postav.

Odkud – Pokud je určené místo, kde musí fáze započít: Modifikátor := ID místa, pokud určeno není, tak: Modifikátor := “-1“

Předmět – Pro určité fáze může být potřeba, aby u sebe postava, která jej vykonává, měla nějaký předmět. Ten jí bude do inventáře přidělen až se dostane na místo, kde jeho fáze začíná a když fázi splní, tak bude zase odebrán.  
Pokud by mělo nastat, že stejný předmět bude potřeba pro více fází za sebou, tak stejně bude po každé fázi odebrán a poté přidán zpět.  
Pokud by měla postava o přidělený předmět během mise přijít, tak je to bráno jako neúspěch.  
Modifikátor := ID předmětu  
Modifikátor := “x“ (pokud není určen předmět)

Kam – Pokud je přesně definováno místo, pak: Modifikátor := ID cílového místa. Může však nastat situace, že výsledné místo není přesně známo anebo se průběžně mění. V tom případě bude Modifikátor := “?“ a výsledné místo se určí dle typu fáze jiným modifikátorem.  
Pokud je však cílové místo definováno zde, tak příchod postavy do zadaného místa je ekvivalentní

Jdi za – Pokud není pevně určeno výsledné ID místa, je možné, že je zapotřebí, aby postava došla za nějakou jinou. V tom případě: Modifikátor := ID hledané postavy + “;“ + parametr akce.   
Pokud bude výsledné místo určeno jinak:   
Modifikátor:= “none“  
  
Když postava zastihne jinou, tak existuje několik možností, co bude dělat dál.

Mluv – parametr akce := „talk“, často používané jako spouštěč další fáze

Zab – parametr akce := “kill“ (o porovnávání schopností více později)

Omrač – parametr akce := “stun“

Okraď – parametr akce := “rob“

Nic – parametr akce := “none“

Přiveď – parametr akce := “bring“  
  
Momentálně nedotažená mechanika. Jak postava ví, kde se nachází někdo jiný? Pokud hledaná postava někam půjde, tak pro ostatní bude velmi těžké ji zastihnout.

# Save file

Nějakej cool úvod lol

Jelikož hru zprostředkovává bot na telegramu, tak by mohla nastat situace, kdy by hru chtělo hrát více uživatelů najednou. Pro každého z nich musí tedy existovat právě jedno uložení postupu. Který postup patří jakému uživateli se určí dle konkrétního ID chatu, které lze získat přes bota.  
Data k jedné hře tedy budou vždy uloženy ve dvou řádcích. Na prvním bude ID chatu a na druhém řetězec znaků obsahující veškeré potřebné informace, viz dále, jehož segmenty od sebe budou odděleny znakem \_ (podtržítko).

## Stav hráče

Stav hráče je definován řetězcem znaků složených z více částí, jež jsou odděleny “,“ (čárkou).

Místo, kde se hráč nachází := “place:“ + ID místa

Počet peněz, jenž má := “coins:“ + počet peněz

Předměty := “items:“ + ID předmětů oddělené středníky (např.: “items:1;3“)

Síla := “str:“ + číselná hodnota síly hráče

Rychlost := “speed:“ + hodnota rychlosti

Vztah ke frakcím := “relations:“ + hodnoty(definovány stejně jako ve Frakcích) pro každou frakci oddělené ; (středníky). Hodnoty jsou ve stejném pořadí jako indexy frakcí.

Při spuštění nové hry se nastaví výchozí pozice na index 0 (U opilého poníka), počet peněz na 25, hráč je bez předmětů, jeho síla i rychlost je 2 a vztah se všemi frakcemi je 2.

## Stavy úkolových linií

Stavy jednotlivých linií v řetězci jsou opět rozděleny “,“ (čárkami). Stavy linií jsou uloženy postupně dle rostoucích indexů.

Jelikož je každý linie úkolů implementována jako binární strom, kdy jeden syn značí úspěch a druhý neúspěch, tak můžeme cestu do konkrétního uzlu stromu definovat jako posloupnost písmen F a S, kdy F znamená, že v daném bodě nastal neúspěch a máme se posunout na takového syna. Naopak S pak znamená úspěšné splnění mise.   
Pokud je řetězec prázdný, znamená to, že je aktivní stále první fáze, tedy kořen stromu.   
Naopak pokud linie dospěje svého konce, tak se řetězec přepíše na znak “E“, aby se již příště nemusel zbytečně procházet.

## Stavy postav

Stav každé z postav je v řetězci oddělen “+“ (plus) a jednotlivé části každého stavu “,“ (čárkou). Postavy jsou též řazeny dle indexu.

Podobně jako u stavu hráče bude pro každou postavu definováno místo, počet peněz, předměty, které má momentálně u sebe, síla a rychlost. Vztahy pro postavy určeny nejsou, neboť se definují dle jejich příslušnosti k frakcím.

Další důležitá vlastnost každé postavy je, jaký úkol zrovna plní, jež bude uloženo následovně:

Linie := “line:“ + ID úkolové linie, kterou plní  
Fáze := “phase:“ + určená modifikovaná fáze, kterou právě plní, uložená ve stejném formátu, jako bylo definováno v Liniích úkolů pro fáze definované modifikátory.  
K tomu se váži možné stavy, ve kterých se tyto fáze nachází.   
Stav fáze := “stage:“ + konkrétní stav (viz níže)  
Rozlišujeme 3 různé stavy:  
Fáze ještě nezačala = postava teprve musí dojít do výchozího místa (“tostart“)  
Fáze právě probíhá = postava se musí dostavit na určené místo (“inprogress“)  
Fáze právě skončila = postava se dostavila na určené místo a provede zadanou akci (“ended“)  
Postava selhala = nějak se stalo, že postava nesplnila úkol (“fail“)  
Postava uspěla = postava splnila fázi bez problémů (“succes“)

Pokud postava žádný úkol neplní, tak bude za všemi klíčovými slovy a dvojtečkou -1 (úkolová část postavy pak bude vypadat následovně: “line:-1,phase:-1,stage:-1“)

Další podstatnou vlastní každé postavy je, jaký je jeho zdravotní stav: “state:“ + konkrétní stav  
Postava je plně při vědomí = “alive“  
Postava je omráčená, ale žije = “stun“  
Postava je mrtvá = “dead“

Rotace a pořadí tahů

1. Načtení z game save souboru
2. Zjištění hráčova stavu + výpis na GUI
3. Zadání akcí uživatelem
4. Reakce na dané akce
5. Zjištění dat o postavách
6. Kontrola, zda někdo z nich splnil úkol
7. Aktualizace stavu průběhu úkolů na základě dat od postav
8. Průchod nových dat úkolů
9. Přiřazení fází postavám
10. Pohyb postav
11. Akce postav
12. Uložení stavu hráče a postav