

LANGUAGES AND COMPILERS IN COMPUTER GAMES

Jan Hranáč

Master's Degree Programme, FIT BUT

E-mail: xhrana01@stud.fit.vutbr.cz

Supervised by: Zemčík Pavel

E-mail: zemcik@fit.vutbr.cz

ABSTRACT

This project deals with usage of programming languages and compilers in computer games. Purpose of this project is to give young people motivation for programming, which isn't always done by usual work in Pascal at high school. This idea was applied on a developed computer game with has a few integrated languages which can be used for creation of the game's content (adventures, campaigns and objects in them). This computer game was based on a RPG desk game *Dragon's Lair* (a Czech equivalent of D&D) from ALTARTM publishing company.

1 ÚVOD

Tento projekt se zabývá využitím programovacích jazyků v počítačových hrách. Tím je myšleno použití daných jazyků k vytváření herního obsahu (herního světa a objektů a událostí v něm). Důraz je kladen na snadnost užití těchto jazyků a jednoduchost prostředí, do kterého jsou zasazeny - každý člověk (se zájmem o počítače) by měl být schopen pochopit vývojový systém hry a posléze vytvořit vlastní kampaň.

V rámci tohoto projektu byla vytvořena počítačová hra *Simulátor Dračího Doupěte* obsahující skriptovací jazyky, které jsou užívány k definici zdrojů a objektů hry a jejího chování (události, scénáře, AI atd.).

Cílem tohoto projektu však není pouze vytvořit hru s obsahem definovatelným pomocí programovacích jazyků (což by nebylo nic nového), ale také vytvořit něco, co by poskytlo mladým lidem motivaci k programování. Je pravda, že studenti středních škol a gymnázií se tak jako jak učí programovat, ale obvykle pracují v jazycích jako je Pascal, nebo C. Problém je v tom, že tyto jazyky nejsou pro začátečníky příliš uspokojující, protože programátor se okamžitě nedočká velkých výsledků a není tudíž motivován do dalšího experimentování. Tím si ale uzavírají cestu k dalšími vývoji v této oblasti, neboť programovat se člověk naučí pouze programováním.

2 ROZBOR

Tato část obsahuje stručný popis projektu *DrdSim*. Je zde popisována i samotná hra, ale hlavní pozornost je věnována jazykům použitým ve hře, systému vestavěného run-time prostředí a principům celého prostředí hry pro vývoj jejího obsahu.

Vytvořená hra je tahové RPG založené na stolní hře *Dračí Doupě*TM.

2.1 VÝVOJ HERNÍHO OBSAHU

Způsob vytváření herního obsahu hry *DrdSim* byl vzdáleně inspirován PC hrou *Operation FlashpointTM* [1], která také byla hlavní motivací tohoto projektu. Tvorba misí do OFP byla založena na třech hlavních pilířích: mapa¹, objekty figurující ve hře a skripty ovládající děj hry. Třídy objektů a skripty byly psané v radikálně odlišných jazycích (což bylo logické, neboť tyto jazyky sloužily radikálně rozdílným účelům). Celý systém byl však poněkud kostrbatý a použité jazyky nebyly syntakticky pěkné.

Základním pojmem pro vývoj herního obsahu v *DrdSim* je **dobrodružství**. To je reprezentováno složkou s konvenčním prefixem. Uvnitř složky jsou podobně jako v OFP definovány tři² skupiny zdrojů:

- Mapy vytvořené ve speciální grafické aplikaci a uložené v XML.
- Třídy objektů herního světa popsané v deklarativní části jazyka DSL.
- Skripty napsané v procedurální části jazyka DSL.

Jazyk DSL je hlavním programovacím jazykem prostředí *DrdSim*, vše ostatní jsou jen jednodušší záležitosti pomocného charakteru. Původně místo něj byly dva oddělené jazyky, ale na radu vedoucího byly spojeny do jednoho. I v současné době je ale DSL stále dělen do dvou částí (viz výše).

Proč ale nemít jeden obecný jazyk, který by zvládl vše? Existují systémy, které to takto řeší. Příkladem může být např. *Unreal engine* [2]. Výsledný jazyk je však příliš komplikovaný (z pohledu laika). Systémy, které použily přístup „jeden úkol, jeden jazyk“, se naproti tomu empiricky projeví jako snáze pochopitelné (viz [1]). Proto bylo rozhodnuto použít více nástrojů, ale maximálně jednoduchých. Navíc oba jazyky se dají kombinovat a mají stejné syntaktické ladění.

Procedurální část DSL je založena na jazyce C. Mnoho věcí, které byly v C uznány za nepěkné či složité³, bylo v DSL odstraněno. Další rozdíly odrážejí fakt, že DSL je frontendem interpretového prostředí. Následkem toho zmizely např. explicitní typy proměnných. Je využívána pozdní vazba a s řetězci a polí se pracuje jednodušeji. Není možno definovat statické struktury, místo toho se pracuje s dynamickými záznamy.

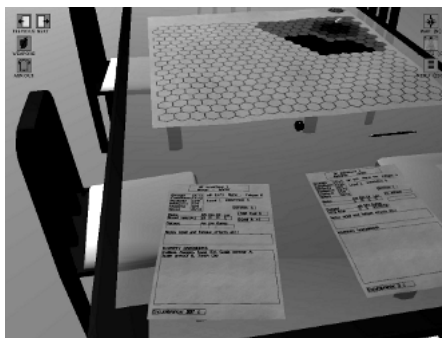
Deklarativní část DSL se vzdáleně podobá definici struktur (s přidanou dědičností). Uvnitř tříd jsou definovány atributy popisující vlastnosti objektu (např. u třídy *široký meč* bude zapotřebí definovat např. jeho cenu). Třída dědí všechny atributy nadefinované v mateřské třídě. Hodnotou atributu může být buď číslo, řetězec, nebo obslužný kód pro událost. Poslední uvedené je pouze syntaktický luxus, který nemá význam metod a který nenarušuje deklarativitu jazyka. Příklad:

```
Trigger: MapObject {
    handler = {StdPrintLn(„Base trigger triggered.“);}
}
```

¹S mapami to je v OFP poněkud složitější - jsou mapy ostrovů a k tomu ještě mapy rozmístění jednotek v okamžiku startu mise. To je však pro projekt *DrdSim* irelevantní.

²Ve skutečnosti je tam toho ještě více, ale to není důležité.

³Například ekvivalence zápisů `array[index]` a `index[array]`.



Obrázek 1: Hlavní scéna *DrdSim*.

2.2 ENGINE HRY

Engine hry obsahuje 3D grafiku, zvukový systém, framework pro uživatelské rozhraní, virtualizaci souborového systému hry a samozřejmě interprety všech použitých jazyků a formátů spojených do jednoho velkého run-time prostředí napojeného na interní systém hry.

Grafika hry byla pojata kontroverzním způsobem. *Dračí Doupě* je stolní hra, a proto i ve hře *DrdSim* vidí hráč stůl a na něm rozložené osobní deníky členů družiny a mapu současné jeskyně/města/kraje (viz obr. 1).

Grafika hry je založena na Open Inventoru a SDL. Interakce hráče s 3D scénou i s 2D GUI byla vyřešena pomocí vlastního frameworku pro tvorbu interaktivních scén.

3 ZÁVĚR

Cílem této práce bylo vytvořit hru s prostředím vývoje herního obsahu použitelného i laiky. Tento cíl byl v dostatečné míře splněn (alespoň při posouzení toho, co již bylo na projektu dokončeno). Prostedí je jednoduché a logické. Hlavní skriptovací jazyk je sice založen na populárním jazyce, ale vyžaduje menší programátorské schopnosti.

V současné době je hra ve fázi, kdy už funguje její engine, překladače a run-time prostředí pro vestavěné jazyky hry. Stála však zbývá veliké množství práce.

Jako osobní postřeh bych rád uvedl, že největší množství práce a problémů nespočívalo v oblasti jazyků a překladačů, ale v drobných věcech technického rázu (např. v *Coin2*).

PODĚKOVÁNÍ

Projekt využívá pravidla *Dračího Doupěte* verze 1.5 od nakladatelství ALTAR™. Tato práce vznikla částečně za podpory grantu VUT FIT, FIT-S-10-2 a specifického výzkumu MSM0021630528.

REFERENCE

- [1] Bohemia Interactive: OFP Wiki. http://community.bistudio.com/wiki/Main_Page, 4.12.2009.
- [2] Sweeney, T.: Reference pro UnrealScript. <http://unreal.epicgames.com/UnrealScript.htm>.