

**ZÁPADOČESKÁ UNIVERZITA V PLZNI**  
**FAKULTA STROJNÍ**

**Studijní program:** N2301 Strojní inženýrství  
**Studijní obor:** Průmyslové inženýrství a management

**DIPLOMOVÁ PRÁCE**

**Tvorba vážné hry v konceptu digitální továrny**

**Autor:** Bc. Miroslav Suchý

**Vedoucí práce:** Ing. Petr HOŘEJŠÍ, Ph.D.

**Akademický rok 2012/2013**

## **Prohlášení o autorství**

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě Strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů, uvedených v seznamu, který je součástí této diplomové práce.

V Plzni dne: .....

.....  
podpis autora

# ANOTAČNÍ LIST DIPLOMOVÉ PRÁCE

<b>AUTOR</b>	<b>Příjmení</b> Suchý	<b>Jméno</b> Miroslav	
<b>STUDIJNÍ OBOR</b>	N2301- Průmyslové inženýrství a management		
<b>VEDOUCÍ PRÁCE</b>	<b>Příjmení (včetně titulů)</b> Ing. Hořejší, Ph.D.	<b>Jméno</b> Petr	
<b>PRACOVIŠTĚ</b>	ZČU - FST - KPV		
<b>DRUH PRÁCE</b>	<b>DIPLOMOVÁ</b>	<del><b>BAKALÁŘSKÁ</b></del>	<b>Nehodící se škrtněte</b>
<b>NÁZEV PRÁCE</b>	Tvorba vážné hry v konceptu digitální továrny		

<b>FAKULTA</b>	strojní	<b>KATEDRA</b>	KPV	<b>ROK ODEVZD.</b>	2013
----------------	---------	----------------	-----	--------------------	------

## POČET STRAN (A4 a ekvivalentů A4)

<b>CELKEM</b>	91	<b>TEXTOVÁ ČÁST</b>	69	<b>GRAFICKÁ ČÁST</b>	22
---------------	----	---------------------	----	----------------------	----

<p style="text-align: center;"><b>STRUČNÝ POPIS (MAX 10 ŘÁDEK)</b></p> <p><b>ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY</b></p>	<p>Tématem diplomové práce, je Tvorba vážné hry v konceptu digitální továrny. Jde o vytvoření výukového programu, založeném na počítačové hře. Tvorba hry proběhla ve Wintermute engine, nástroje pro tvorbu her. Program by měl formou hry studentům vysvětlit některé metody, používané v managementu a logistice.</p>
<p style="text-align: center;"><b>KLÍČOVÁ SLOVA</b></p> <p style="text-align: center;"><b>ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE</b></p>	<p>vážná hra, edutainment, wintermute engine</p>

## SUMMARY OF DIPLOMA SHEET

<b>AUTHOR</b>	<b>Surname</b> Suchý	<b>Name</b> Miroslav	
<b>FIELD OF STUDY</b>	N2301 - Industrial Engineering and Management		
<b>SUPERVISOR</b>	<b>Surname (Inclusive of Degrees)</b> Ing. Hořejší, Ph.D.	<b>Name</b> Petr	
<b>INSTITUTION</b>	ZČU - FST - KPV		
<b>TYPE OF WORK</b>	<b>DIPLOMA</b>	<b>BACHELOR</b>	<b>Delete when not applicable</b>
<b>TITLE OF THE WORK</b>	Creation of the serious game within the digital factory concept		

<b>FACULTY</b>	Mechanical Engineering	<b>DEPARTMENT</b>	Industrial Engineering and Management	<b>SUBMITTED IN</b>	2013
----------------	------------------------	-------------------	---------------------------------------	---------------------	------

### NUMBER OF PAGES (A4 and eq. A4)

<b>TOTALLY</b>	91	<b>TEXT PART</b>	69	<b>GRAPHICAL PART</b>	22
----------------	----	------------------	----	-----------------------	----

<b>BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS</b>	The main topic of the thesis deals with serious game development. The main concept of the game corresponds with the digital factory concept. It means creation of an educational program, based on a computer game. The game development will be provided by the wintermute Engine, which is a tool for creation of computer games. The program could explain to students some methods, used in management and logistic.
<b>KEY WORDS</b>	serious game, edutainment, wintermute engine

## **Poděkování**

Na tomto místě bych rád poděkoval doc. Ing. Petru Hořejšímu, Ph.D., za cenné připomínky a odborné rady, kterými přispěl k vypracování této diplomové práce. Dále děkuji studentce Ústavu Umění a Designu na Západočeské Univerzitě, Pavlíně Kunešové za vytvoření všech ilustrací pro tento projekt.

Velké poděkování patří Ing. Michalu Wendlovi, učiteli na Gymnáziu Blovice za odbornou pomoc při tvorbě programu a při práci s editorem.

<b>Obsah.....</b>	<b>1</b>
Seznam zkratk a vysvětlení pojmů.....	3
1 Úvod .....	4
2 Vážné hry .....	5
2.1 Edutainment.....	5
2.2 Vážná hra.....	5
2.2.1 Definice a oblast působnosti .....	5
2.2.2 Historický vývoj.....	6
2.2.3 Současnost.....	9
3 Analýza současného stavu.....	10
3.1 Simulace řízení .....	10
3.2 Simulace strategií .....	11
3.3 Jiné simulace.....	11
3.4 Advergates .....	12
4 Výběr vhodného nástroje.....	14
4.1 Volba enginu hry .....	14
4.1.1 Co je herní engine .....	14
4.1.2 Účel .....	14
4.1.3 Source engine .....	15
4.1.4 Aurora engine .....	16
4.1.5 Wintermute Engine .....	18
4.2 Výběr vhodného žánru.....	19
4.3 Výběr vhodného enginu.....	20
5 Wintermute engine .....	21
5.1 Vlastnosti .....	21
5.2 Prvky WME.....	23
5.3 Ukázka hry, vytvořené ve Wintermute engine .....	23
6 Proces tvorby výukového programu .....	24
6.1 Účel vzdělávacího programu .....	24
6.2 Návrh interface a ovládání.....	24
6.3 Ilustrace .....	24
6.4 Scénář .....	25
6.5 Tvorba struktury info-tabletu.....	26
6.6 Tvorba scén.....	28
7 Tvorba samotného programu .....	29
7.1 Základní pojmy.....	29
7.2 Struktura WME.....	29
7.3 Rozlišení .....	31
7.4 Tvorba scén.....	31
7.5 Tvorba hlavního menu.....	35
7.6 Tvorba submenu .....	38
7.7 Tvorba vlastní metody Talk.....	39
7.8 Tvorba inventáře.....	41
7.9 Tvorba hlavní postavy .....	43
7.10 Tvorba informačního tabletu .....	46
7.11 Sebrání předmětu ze scény .....	51
7.12 Změna entity při akci .....	53
7.13 Tvorba vedlejší postavy.....	56
7.14 Dialogy s vedlejšími postavami.....	57
7.15 Tvorba proměnných.....	60
7.16 Kritéria tvorby programu.....	61

8	Závěr .....	66
9	Zdroje .....	67
	PŘÍLOHA č. 1 - Scénář výukového programu .....	A-1
	PŘÍLOHA č. 2 - Vývojový diagram průchodu scénami .....	B-1
	PŘÍLOHA č. 3 - Ilustrace .....	C-1
	PŘÍLOHA č. 4 – Zajímavosti z tvorby programu .....	D-1

### **Seznam zkratek a vysvětlení pojmů**

Advergames – reklamní hry. Zvláštní případ vážných her, které dělají reklamu určitému produktu

Adventura – počítačová hra, kde hlavním motivem je řešení logických hádanek

AMG – divize firmy Mercedes

Back to basic – učitelské hnutí v 70. letech 20. století v USA. Cílem bylo vrátit se k tradičním formám výuky jako je čtení, psaní a aritmetika

Decision-maker – hráč nebo více hráčů, hrající hry. Doslova: ten, co činí rozhodnutí

DPVR – Digitální podnik a virtuální realita. Předmět, vyučovaný na Západočeské Univerzitě

e-learning – elektronická forma vzdělávacích dokumentů, například skripta v e-learningovém programu

Entita – obecně objekt. V informatice může obsahovat libovolnou množinu dat

First person shooting game – druh počítačové hry. Hráč hledí na scénu pohledem z první osoby, hry jsou nazývány slangově „střílečky“

Edutainment – spojení slov education (vzdělání) a Entertainment (zábava)

Engine, Game Engine – jádro všech počítačových her. Obsahuje balíky grafiky, zvuků a vykreslovací nástroje. V engine se tvoří samotné hry

Game-learning – označení kategorie her pro vzdělávání se

Games for health – označení her pro zdraví. Hry pro aplikaci ve zdravotnictví, původně z roku 2004

Games for change – hry, zaměřené na sociální otázky, původně z roku 2004

Middleware – flexibilní a opakovatelně použitá herní platforma, jiný název pro herní engine

MMORPG - Massively multiplayer online role-playing game. U těchto her je pohled buď z první, nebo ze třetí osoby, hrají se na serverech online a hlavním prvkem této hry je plnění úkolů a vylepšování atributů postavy

PC – personal computer. Domácí stolní počítač

RPG - Role-playing game. U těchto her je pohled buď z první, nebo ze třetí osoby, hlavním prvkem této hry je plnění úkolů a vylepšování atributů postavy

Playstation – herní konzole, vyvinutá firmou Sony

Real-time strategická hra – strategická hra, odehrávající se v reálném čase (opak tahové hry)

Serious game – vážná hra. Hra určená ne primárně k zábavě, ale i ke vzdělávacím účelům

Sprite – malý, dvourozměrný obrázek, který je integrován do scény

WME – Wintermute Engine. Engine pro tvorbu 2D a 3D videoher



## 1 Úvod

Cílem diplomové práce je vytvořit tzv. serious game v konceptu digitální továrny. Výukový program je rozdělen do kapitol, každá kapitola je rozdělena podle vyučované metody. Metody užití v programu jsou například systém Kanban, Teorie omezení, či motivace zaměstnanců.

Vytvářená vážná hra (Serious game) nebo výukový program je primárně zaměřen na výuku vybraných metod průmyslového inženýrství zábavnou formou. Implementované metody byly vybrány dle subjektivní vhodnosti výkladu vhodnou formou pomocí vážné hry.

Pro vážné hry je možné využít různé formy. Je možné vytvářet například plně interaktivní 3D scény. Ty jsou však poměrně značně náročné na hardware, mnozí uživatelé, kteří nejsou zvyklí hrát moderní počítačové hry, mají často potíže s ovládáním. Naprosto univerzálně se jeví možnost tvorby v rámci žánru 2D adventura, kde je ovládání snadné a na každou akci mnoho času.

Takový výukový program je možné naprogramovat od začátku – from scratch. V této práci je ukázána možnost využití již hotového jádra – engine. Byl vybrán nástroj Wintermute Engine.

Výukový program by ve výsledku měl studentům tohoto oboru na Univerzitě pomoci lépe porozumět problematice těchto manažerských metod a nenásilnou, naopak zábavnou formou studentům pomoci si tyto znalosti osvojit.

V prvních kapitolách se tato práce zabývá pojmem serious game, jeho historií, vývojem až po aplikace v dnešní době. V dalších kapitolách je popsán proces výběru vhodného nástroje pro tvorbu tohoto výukového programu. Kapitole 5 se zabývá nástrojem, který byl pro tvorbu vybrán, tedy Wintermute Editorem, v kapitolách 6 a 7 je popsána samotná tvorba programu. V přílohách jsou vloženy ilustrace scén, scénář výukového programu, vývojový diagram postupu scénou a jako poslední zajímavosti z tvorby výukového programu.

## 2 Vážné hry

Vážná hra je každá hra, která kromě zábavného charakteru má za úkol ještě něco naučit. S jistou rezervou je možné třeba i závodní hry považovat za výukové (výuka fyzikálních vlastností vozidla). Vážné hry jsou cíleně vytvářeny často jako simulátory dílčích činností za účelem osvěty principu těchto činností. V následující kapitole si vysvětlíme některé pojmy.

### 2.1 Edutainment

Educational Entertainment, neboli Edutainment, je jakákoliv forma zábavy, vytvořená za účelem vzdělat se, nebo se něco naučit. Hry, s vysokým obsahem jak zábavy, tak vzdělávací hodnoty, můžeme označit jako Edutainment. Existuje zde obsah, který má primárně vzdělávací hodnotu, ale k tomu přidává jakousi hodnotu zábavnou. Celé to může fungovat i obráceně. Totiž primárně zábavný obsah, který má jistou vzdělávací hodnotu. [3]

Pojem Edutainment byl poprvé využit v roce 1948 Waltem Disneyem v seriálu True life adventures. Název se poprvé začal používat v roce 1973 Robertem Heymanem při produkci dokumentů pro National Geographic. Tento pojem byl rovněž použit Dr. Chrisem Danielssem v jeho díle Millennium Project, který se později stal známý spíše jako The Project World Elysian. Odnož tohoto slova, slovo Edutainer byl používán Craig Sim Webbem již před přelomem tisíciletí a popisoval jednotlivce, který nabízí výukové představení. [3]

### 2.2 Vážná hra

Vážná hra (z angl. Serious game) je taková hra, která není určena primárně pro zábavu, ale obsahuje i edukativní část obsahu. Slovo „Vážná“ zde znamená zaměření například na produkty používané v odvětvích jako je obrana, vzdělávání, vědecký výzkum, zdravotní péče, chování se v krizových situacích, plánování, inženýrství, urbanistika, náboženství nebo politika. [6]

#### 2.2.1 Definice a oblast působnosti

Vážné hry jsou navrženy za účelem řešení problému. Přestože vážné hry mohou být zábavné, jejich hlavním cílem je orientovat se na daný problém a řešit ho. Přitom se hry více či méně blíží realitě. Někdy je ve hře záměrně obětována část zábavy, aby bylo dosaženo požadovaného pokroku ze strany hráče. Vzhledem k tomu, že videohry jsou klasifikovány podle žánru (od závodních po strategické nebo bojové), u vážných her tato klasifikace neplatí a jsou řazeny podle účelu. Tyto kategorie zahrnují vzdělávací hry, reklamní hry, politické a náboženské hry. Kategorie vážných her pro vzdělávání je také někdy známa jako „game-learning“. Vážné hry jsou tedy počítačové hry a videohry, které mají za úkol nejen uživatele bavit, ale jsou v nich zahrnuty také prvky vzdělávání či odborné přípravy. Hlavním cílem vážné hry tedy není primárně publikum bavit, i když právě zábava je důležitým aspektem při výběru hry, protože právě kvůli zábavě při hraní hry hráč tuto hru hraje dál a tím se vzdělává. [6]

## 2.2.2 Historický vývoj

Pojem vážná hra byl skutečně použit dávno před zavedením počítačů a elektronických zařízení do zábavního průmyslu. V roce 1970, Clark Abt zavedl tento pojem a diskutoval o něm ve své knize *Serious Games*, publikované nakladatelstvím Viking Press. Jeho kniha uvažovala spíše o stolních či karetních hrách. Nicméně zavedl použitelnou definici, které je dnes (ve světě počítačů) stále velmi dobře aplikovatelná:

*„Hra je aktivita mezi dvěma nebo více nezávislými hráči, kteří činí rozhodnutí (decision-makers), snažícími se o dosažení jejich cílů v nějakém limitujícím kontextu.“* Více konvenční definice by řekla, že hra je kontext s pravidly mezi protivníky, kteří se snaží vyhrát nebo první dosáhnout cíle hry. Zabýváme-li se vážnou hrou, tato hra má explicitní a pečlivě promyšlený vzdělávací účel nebo účely a nejsou určeny primárně pro zábavu. [6]

Prvopočátky vážných her je třeba hledat dříve. Důstojníci Pruské armády měli k dispozici deskovou hru, pomocí které mohli trénovat své taktické a strategické schopnosti v boji. Hra pochází z roku 1812, původní název byl: „Instrukce pro reprezentaci taktických manévru.“ Hra je známá pod jejím pozdějším názvem *Kriegsspiel* (je německý název pro válečnou hru). Hra byla velmi složitá, byla zde snaha o co nejvěrnější zobrazení boje. Proti sobě stály 2 strany a hra byla určena pro až 10 hráčů a jednoho hráče, který byl správce mapy. Fungovaly zde prvky z boje jako hierarchie velení, ale i například jednotky, které na sebe neviděly, se nesměly koordinovat. Každá jednotka měla jinou rychlost pohybu a dostřel, to vše se měřilo speciálními měřidly. Časové omezení bylo 2 minuty na tah, hrálo se na mapě s měřítkem 1:8000. [1] Původní hru *Kriegsspiel* můžeme vidět, viz Obr. 2.1



Obr. 2.1: Kriegsspiel [7]

Mike Zyda poskytl aktualizaci a logický přístup k termínu vážná hra v jeho článku z roku 2005 v *IEEE Computer* s názvem: „From Visual Simulation to Visual Reality to Game“, kde pojem definuje takto: „Hra – fyzický a mentální obsah, hraje se dle pravidel, jejím cílem je pobavení. Videohra - mentální obsah, za účelem zábavy, rekreace či odměny. Vážná hra -

mentální obsah, hraje se na počítači, dle pravidel. Využití zábavy k tréningu, vzdělávání, lepšímu zdraví atp.“ [1]

Dlouho předtím, než pojem vážná hra vstoupil do širokého používání v Serious Game Initiative v roce 2002, předtím byly hry tvořeny pro nezábavní účely. Protože hry byly zaměřeny čistě na edukativní obsah, hry zpočátku nebyly ziskové, dále s rozvojem moderních technologií, byli výrobci nuceni přehodnotit celý tento koncept vážných her koncem roku 1990. Během této doby se zkoumalo využití a užitečnost her pro jiné účely. Tímto se vážné hry začaly stávat oblíbenějšími. V roce 2002 tedy ústav Woodrow Wilson International Center for Scholars ve Washingtonu zahájil Serious Game Initiative na podporu rozvoje her, které se začaly zaměřovat na politiku a manažerská témata. Poté se začaly hry více zaměřovat na konkrétní podskupiny a v roce 2004 se začaly objevovat tzv. Games For Change, které se zaměřovaly na sociální otázky a tzv. Games For Health, které byly tvořeny pro aplikaci ve zdravotnictví. [6]

Hraní her začalo být využíváno pro edukativní účely už od počátku 20. století. Využívání papírových vzdělávacích her bylo populární v 60. a 70. letech 20. století, ale muselo ustoupit pod tlakem učitelského hnutí Back to Basic. Hnutí Back to Basic byla změna ve stylu učení, která začala v 70. letech, kdy studenti ve standardizovaných textech obstáli velmi chabě. Cílem tohoto hnutí bylo zaměřit studenty zpět na čtení, psaní a matematiku a zařadit tyto předměty do osnov více, než předtím. S rozšířením počítačů v 80. letech 20. století, se stalo hraní vzdělávacích her velmi populární, jak mezi studenty, tak mezi učiteli. Mezi populární tituly patřila například hra Oregon Trail: viz Obr. 2.2 [6]



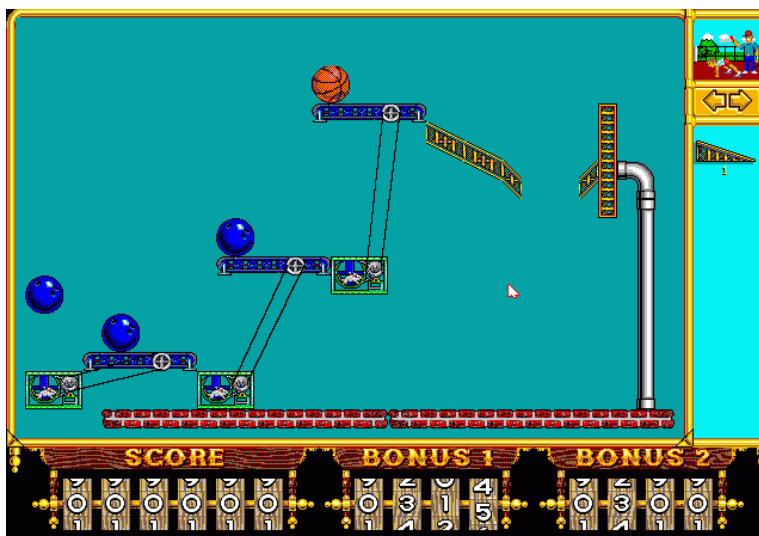
Obr. 2.2: Hra Oregon Trail pro Apple II [8]

V této hře byl student vsazen do role osadníka, který putoval z Missouri do Oregonu se svojí družinou osadníků. Hra se odehrávala v roce 1848 a měla studenty učit o realitě 19. století a tehdejších průkopnících, kteří se vydali budovat nové osady do Oregonu. [9]

Dalšími populárními hrami byly zejména matematické hry Math Blaster a Number Munchers. Ty byly původně určeny pro výuku matematiky, postupem času byly však rozšířeny i na jiné předměty, jako výuka jazyků (čtení) a věda. I když tyto hry byly nesmírně populární jak mezi učiteli, tak mezi studenty, byly také často kritizovány kvůli tomu, že studentům nepřinesou

nové formy učení, místo toho se tyto hry dají velmi dobře obejít tím, že si student prostě najde jeden způsob jak hru rychle a „bezbolestně“ vyřešit. [6]

V 90. letech 20. století přišla velká změna s nástupem nových her jako například: TIM, The Incredible Machine. [6] Ve hře máte v každém kole jiný úkol, od úkolu nechat míč proskočit basketbalovým košem, až po složité úkoly typu: Vystřel z děla, dělová koule musí spadnout na nůžky, které přestříhnou provázek, na kterém visí balonek. Variací zde bylo opravdu nesčetně, krom zábavy představuje tato hra základní fyzikální vlastnosti v praxi. Pokročilejší kola, kde se i například mění gravitace, byla velmi obtížná. První kolo hry TIM vidíme na Obr. 2.3:



Obr. 2.3: The Incredible Machine [10]

V roce 1990 byl také zaváděn internet do škol, který dostal přednost před hraním her. [6]

Na počátku 21. století došlo k nárůstu různých typů vzdělávacích her, zejména těch pro mladší studenty. Mnohé z těchto her nebyly na bázi počítače, ale na bázi konzole. V roce 1999, firma LeapFrog Enterprises představila tzv. LeapPad, jednoduchý tablet pro děti. Na Obr. 2.4. Děti si mohly na dotykovém displeji kreslit, psát, či řešit různé úkoly, které jim zařízení zadalo. Na základě v té době populárních ručních herních systémů typu Game Boy byl vytvořen nový výrobek, zvaný Leapster, pomocí kterého se děti učí abecedu a jednoduchou matematiku. Ten spatřil světlo světa v roce 2003. [6]



Obr. 2.4: LeapPad [18]

Další přelom nastal, když hlavní spotřebitel a zároveň výrobce vážných her, Americká armáda, potřebovala připravit své vojáky na situace, prostředí a kulturu v cizích zemích. Potřebovali naučit vojáky komunikovat, pochopit okolí a rychle jednat. Jako první vážná hra pro armádu je často považovaná hra Army Battlezone, projekt od firmy Atari, přerušený v roce 1980, navržený pro využití 2D tankového simulátoru pro výcvik. Dvě nejznámější plnohodnotné hry pro Americkou armádu jsou America's Army z roku 2002, viz Obr. 2.5 a Full Spectrum Warrior z roku 2004. [6]



Obr. 2.5: ukázka ze hry America's Army [11]

Mimo vládní použití je zájem především o vážné hry, zaměřené formální vzdělání, politiku, reklamu, zdraví a sociální témata. [6]

### 2.2.3 Současnost

V posledních letech zadavatelé vzdělávacích simulací (např. Americká armáda), apelovali na herní vývojáře na tvorbu nízkonákladových simulací, které by byly zároveň přesné ale i poutavé. Zkušenosti herních vývojářů s hraním a tvorbou her, tyto vývojáře činí jako nejlepší kandidáty pro tvorbu těchto typů simulací, které oproti původním armádním projektům stojí o miliony dolarů méně a zároveň nepotřebují speciální zařízení pro jejich hraní. Výhody tohoto použití jsou hlavně v té věci, že herní vývojáři dokážou hru vyrobit mnohem levněji a hlavně rychleji. Hry zahrnují funkční entity jako radar a bojová vozidla a pomocí stávající infrastruktury tak vývojáři vytvářejí hry, které simulují bitvy, procesy a události za zlomek ceny vládních dodavatelů. Tradiční simulátory stojí obvykle miliony dolarů nejen při rozvoji, ale i při nasazení a obecně vyžadují nákup specializovaného vybavení. Hra vytvořená herními vývojáři oproti tomu je spuštěna pomocí obyčejného domácího PC a vejde se na volně dostupné médium. V dnešní době lze takové hry dokonce stáhnout z internetu, odpadají zde problémy s distribucí. Další nespornou výhodou je zpětná vazba, kterou vývojář dostává od uživatele a během životního cyklu hry může samotnou hru pomocí updatů a servisních balíčků dopravovat. [6]

### 3 Analýza současného stavu

V této kapitole se podrobně seznámíme s různými typy vážných her. Prakticky každá hra, ať už vážná nebo ne, hráče/uživatele něco naučí. Šachy dokáží naučit taktice a předvídání, poker zase odhadnutí soupeře, vzhledem k tomu jaké karty má v ruce. Akční hry jsou velmi dobrý trénink na reflexy, hráči těchto her dokáží reagovat mnohem rychleji, než lidé, kteří se hraním her nezabývají. Adventury jsou logické hry, kde na základě rozhovorů a situací, jste nuceni řešit logické hádanky nebo úkoly. My se nyní seznámíme s rozdělením vážných her do žánrů, jejich účelem a výsledky, co můžou hráči přinést.

#### Rozdělení

Vážných her existuje v dnešní době nepřehledné množství druhů. Dále budou uvedeny pouze ty neznámější kategorie her. V rámci rešeršní části této práce bylo množství dále popisovaných aplikací testováno.

#### 3.1 Simulace řízení

Simulace patří mezi nejrozšířenější skupinu vážných her. Mezi jednu z nejdokonalejších simulačních vážných her patří např. MS Flight Simulator. Tato komerčně prodávaná simulace slouží mimo jiné i pro výcvik skutečných pilotů. Zde sedíte v kokpitu letadla, to má oproti reálnému stroji stejné ovládání, stejně se i chová při stejných podmínkách. Ve hře se samozřejmě například i mění počasí a let podle toho musíte přizpůsobit.

Další takovou hrou, která byla postupem let dovedena k dokonalosti, je Gran Turismo na konzoli Playstation. (Na Obr. 3.1) Jedná se o simulátor řízení vozu. Hra je graficky vymodelovaná do posledního detailu, zvuky vozů jsou opravdu nahrány na reálných vozech. Telemetrie vozů byla také nahrána na skutečných vozech. Takže pokud si ke konzoli uživatel pořídí velmi kvalitní volant, bude mít přesně ty samé podmínky jako v reálném voze. Důkazem je každoroční soutěž, kdy tvůrci hry vyhlásí jeden automobil, jeden závodní okruh a uživatelé po celém světě spolu soupeří. Vítěz je pak posazen do opravdového vozu na opravdovou trať a po krátkém tréninku jede ostrý závod. Většinou se jezdí vytrvalostní závody v tříčlenných týmech. Jen v jednom konkrétním týmu jsou pouze dva závodníci a jeden hráč této hry. Hra je až tak realistická, že poprvé co byl hráč posazen do závodního vozu, který víceméně nikdy neviděl, jeho tým se umístil na stupních vítězů. Tato hra vás naučí řídit a stačí vám k tomu pouze konzole, televize a volant.

Dále existují simulátory řízení dálkových tras kamionů, průmyslových a zemědělských vozidel.



Obr. 3.1: Gran Turismo 5 [14]

### 3.2 Simulace strategií

Do kategorie simulací lze rovněž zařadit strategické hry. Ve hře Simcity uživatel buduje město a na základě poptávky po budovách stavíte rezidenční, komerční a industriální budovy. Celé město musí připojit k elektrickým rozvodům a rozvodům vody, zavést hromadnou dopravu, policii, školy, hasiče, nemocnice a ochránit město před katastrofami od požárů po demonstrace.

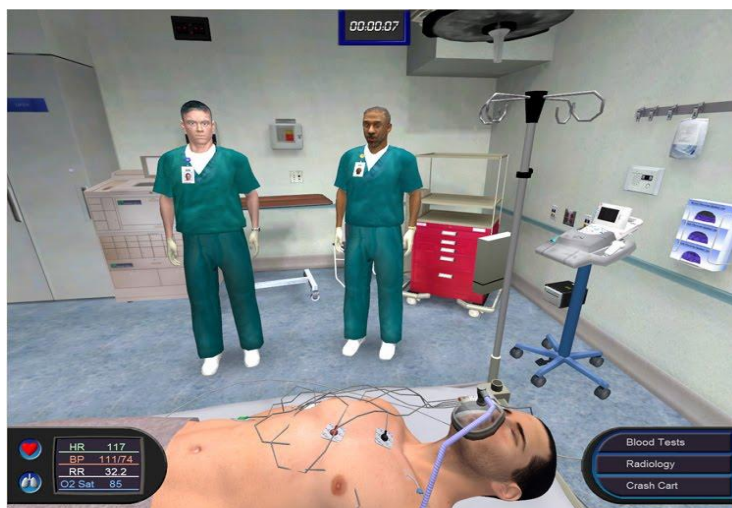
Další takovou hrou byla strategie Transport Tycoon (viz Obr. 3.2), kde jako majitel společnosti na dopravu a přepravu se snažíte opět na základě nabídky a poptávky provádět transport zboží a pasažérů z místa na místo, společnost dále rozšiřujete a bojujete s konkurencí. I když hra vyšla již v roce 1994, je svým způsobem nadčasová a velké množství hráčů se k ní opět vrací.



Obr. 3.2: Transport Tycoon [15]

### 3.3 Jiné simulace

Simulátory nemusí být pouze sportovního, či strategického zaměření. Můžeme najít spousty her se zaměřením například zdravotnickým. Na Obr. 3.3 vidíme screenshot ze hry Pulse. Jedná se o simulační hru pro profesionály, kde si mohou nacvičit lékařské zákroky bez nebezpečí na zdraví pacientů. Hráč pracuje ve virtuální nemocnici, kde musí reagovat na naléhavé situace, určit diagnózu pacienta, jako v reálném světě. Tato hra byla vyvinuta na Texaské univerzitě. [24]



Obr. 3.3: Lékařský simulátor Pulse [1]



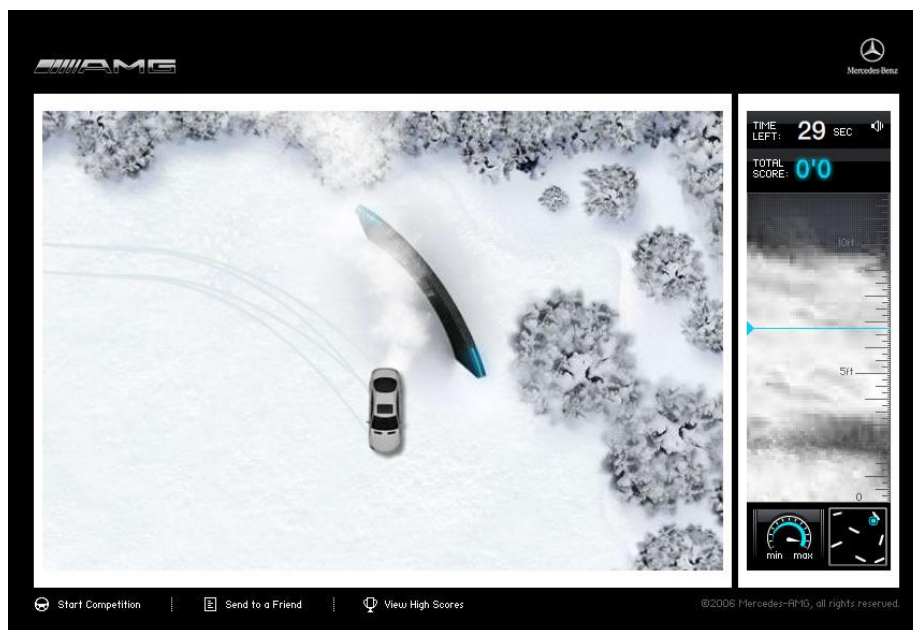
Vážné hry existují i ve strojním oboru. Na Obr. 3.4: Megacenius je vidět vážná hra z tohoto prostředí, s názvem Megacenius. Jedná se o výukovou, multiplayerovou online hru, zaměřenou na studenty strojírenských oborů. Cílem je seznámit studenty s objevováním virtuální obráběcí dílny, získáním nových dovedností na NC obráběcích strojích a s řízením virtuálních výrobních projektů. Původ této hry je ve Francii, na Champollion university. Od roku 2009 je tento projekt podporován francouzským ministerstvem ekonomiky a organizací World competitiveness cluster v letectví a kosmonautice. Multiplayer této hry byl zveřejněn v roce 2012. Herní svět je vsazen do poměrně vzdálené budoucnosti, kdy lidská rasa opouští planetu Zemi a chystá se žít jinde ve vesmíru. Hráč převezme roli učně-inženýra, který je se svojí lodí vysazen na neznámé planetě. Hráčovým cílem je opravit vznášedlo, které mu umožní cestu kolem planety. Hráč je konfrontován se řadou miniher, kde získá části vybavení, které mu pak pomáhají splnit hlavní úkoly hry. [27]



Obr. 3.4: Megacenius [27]

### 3.4 Advergames

Zvláštní skupinu her tvoří tzv. advergames, neboli reklamní hry. Většinou se jedná přímo o hru, vzniklou na zakázku. Zadavatelem je výrobce produktu nebo poskytovatel služeb, pro které má tato hra dělat reklamu. Hry bývají kvůli dostupnosti a především kvůli návštěvnosti umístěny na serverech s online hrami zdarma. [6] Například český server www.superhry.cz. Tyto hry se dají dále dělit např. podle toho, zda je hra zaměřena primárně na daný produkt, či zda se ve hře jen vyskytují reklamy na produkt (Product placement – například reklama na virtuálním ledě ve hře NHL). Zvláštní kategorii tvoří reklamní hry, pro jejichž „hraní“ je třeba hledat indicie na různých stránkách na internetu. Jedna z jednodušších her má název Mercedes Drift, viz Obr. 3.5. Jedná se o 2D hru, pohled na autíčko je shora a zde v krátkém čase musíte udělat co nejlepší přetáčivý smyk. Hru doplňuje zvuková kulisa reálného vozu. Krom poslechu zvuku AMG vidlicového osmiválce však hra nenabízí zhora nic.



Obr. 3.5: Mercedes drift game [17]

Jako další příklad uveďme velmi zajímavou hru s názvem McDonald's game. Na Obr. 3.6. Ve hře vedete pobočku restaurace jmenované značky, nezůstáváte však pouze u prodávání hamburgerů. Kontrolujete zde fáze od nakupování pozemků, kde pěstujete obiloviny a dobytek, porážku dobytka, vlastní restauraci a management. Když management oznámí nízké zisky, spustíte reklamní kampaň na podporu prodeje. Reklamní slogany od: Děti chtějí McDonald až po přesvědčení lidí, že naše jídlo je to nejlepší. Tržby se rychle zvednou, ale s tím i spotřeba masa a za chvíli vám na dveře zaklepu obezitologové, že populace kvůli vaší reklamní kampani tloustne. Takže je podplatíte. Kvůli zvyšující se poptávce ale musíte vykácet deštný prales, postavit zde farmy a už máte na krku pro změnu ekologické aktivisty. Hra je trochu naturalistická, ale rozhodně zajímavá a představuje výzvu.



Obr. 3.6: McDonald's videogame [12]

## 4 Výběr vhodného nástroje

Autor se bakalářské práci zabýval tvorbou e-learningového kurzu pro potřeby předmětu podnikové informační systémy. Jeho přínosem je, že je volně stažitelný nebo dostupný online a některá problematika se zde dá vysvětlit například pomocí flashové animace nebo grafiky. Přínosem dále byl kvíz po každé z kapitol, kde student měl na výběr ze tří možností odpovědi a program následně vyhodnotil úspěšnost studenta.

tato diplomová práce jde mnohem dále. Tato práce se (jak bylo řečeno) bude zabývat tvorbou vážné hry, která bude obsahovat ve scénáři informace, uživateli užitečné ke studiu. Aby měla hra smysl a nebyla to pouze edukativní hra bez formy jakéhokoliv zábavy, je nutné do hry přidat prvky, které tento koncept poněkud rozvedou dále a student, u hry vydrží sedět a bude schopen tyto informace vstřebat a zapamatovat si je.

I když nebylo jasné, jak bude hra vypadat, první intuitivní myšlenky směřovaly k řešení hlavní postavy, za kterou bude hráč hrát (bude ji tedy ovládat), která bude mít roli některé z nižších vedoucích pozic v nějaké fiktivní firmě. Ve firmě budou kanceláře, kde bude ředitel nebo jeho přímý nadřízený dávat postavě (hráči) úkoly a ten je pak následně bude plnit. Důležitou součástí hry bude informační kancelář, kam hráč může pokaždé zajít a zeptat se na problém, kterému nerozumí a zde mu poradí. Než ale postoupíme až do této fáze tvorby hry, bylo nejprve nutné najít vhodný nástroj pro tvorbu her.

### 4.1 Volba enginu hry

Pro tvorbu počítačové hry lze v podstatě použít dva způsoby tvorby. První je začít tzv. na zelené louce, kdy se celá hra, včetně grafiky tvoří od základu na novo. Druhým způsobem, který je mnohem jednodušší a využívaný všemi vývojářskými firmami na světě, je použít engine, který již někdo vytvořil. V této kapitole je popsáno, co je to herní engine, jaké obsahuje komponenty a způsoby jeho využití.

#### 4.1.1 Co je herní engine

Herní engine je systém, určený pro vytvoření a rozvoj videoher. Hlavní herní engine poskytuje softwarové rozhraní, které tvůrce hry používá k vytváření her jak už pro herní konzole, tak pro osobní počítače PC. Jádro hry, prezentované enginem obsahuje renderovací engine („renderer“) pro 2D a pro 3D grafiku a fyzikální engine nebo detekci kolizí objektů, zvuk, skripty, animace, umělou inteligenci, ukládání a načítání, scény atp. Proces tvorby her z ekonomických důvodů používá zaběhnuté enginey, nebo se na ně adaptuje. Vytvořit totiž celý engine hry je ekonomicky a časově obrovsky náročné. [16]

#### 4.1.2 Účel

Herní engine poskytuje vizuální vývojové nástroje v kombinaci se softwarovými komponentami. Tyto nástroje jsou obvykle poskytovány v integrovaném vývojovém prostředí, které umožní zjednodušený a rychlý vývoj hry. Vývojáři herního enginu musí být krok napřed před herními vývojáři a vytvořit robustní softwarové rozhraní, které by mohl herní vývojář použít pro vývoj samotné hry. Souprava herního enginu (game engine suite) poskytuje zázemí, které usnadňuje vývoj komponent, jako jsou grafika, zvuk, herní fyzika a funkce umělé inteligence. Tyto enginey jsou také někdy nazývány middleware, protože poskytují flexibilní a opakovaně použitelnou softwarovou platformu, která poskytuje všechny

základní potřebné funkce „hned po vybalení z krabice“ pro vytvoření hry při značném snížení nákladů jak finančních, tak časových a hra může jít do prodeje mnohem dříve. To jsou tři kritické faktory, kterých se musí herní vývojáři držet. Mezi nejznámější middleware programy patří například Gamebryo, JMonkey Engine a RenderWare. [16]

Jako jiná middleware řešení, i herní enginy obvykle poskytují určitou abstraktní platformu, což znamená, že hra může být vytvořena jak pro stolní počítače, tak pro herní konzole. Dále se vyrábějí v komponentovou technologii, některé komponenty tak lze zaměnit za jiné, více specializované na danou problematiku. Některé herní enginy, například RenderWare, jsou vytvořeny jako řada volně propojitelných middleware částí, které lze volně kombinovat a vytvořit tím engine „na míru“. Nicméně, rozšiřitelnost je hlavním požadavkem při tvorbě herních engineů. [16]

Některé herní enginy ale poskytují jen real-time 3D referování namísto široké škály funkcí potřebných pro hry. Tyto enginy se spoléhají na herního vývojáře, který vytvoří tyto ostatní funkce, nebo hru sestaví za pomoci jiných middleware. Tyto enginy jsou obecně označovány jako „graphic engine“, „rendering engine“ nebo „3D engine“. Jako příklad grafického engineu můžeme považovat například: Crystal Space, Genesis3D, Irrlicht, OGRE, RealmForge, Truevision3D a Vision Engine. Moderní hry nebo grafické enginy také poskytují tzv. scene graph, což je objektově orientovaná reprezentace 3D herního světa, která většinou zjednodušuje design hry a může být použit pro efektivnější vykreslování rozsáhlých virtuálních světů. [16]

#### 4.1.3 Source engine

První z herních engineů, který připadal pro tvorbu vážné hry v úvahu, byl Source engine. V tomto engineu jsou prováděny virtuální simulace v rámci předmětu KPV/DPVR. Viz Obr. 4.1:



Obr. 4.1: Ukázka světelných a stínových efektů Source engineu [20]

Source je 3D herní engine, vyvinutý společností Valve Corporation. Byl vydán v roce 2004 s hrou Counter Strike: Source a krátce na to v dlouho očekávané bombě, Half life 2. Engine je aktivně používán dodnes. Source byl vytvořen pro tvorbu her, tzv. first person shooting game, což je střílečka, kdy hráč se dívá na scénu z první osoby, dále pro MMORPG, to v překladu znamená Massively multiplayer online role-playing game. U těchto her je pohled buď z první, nebo ze třetí osoby, hrají se na serverech online a hlavním prvkem této hry je plnění úkolů a vylepšování atributů postavy. Dále je engine možné použít pro tvorbu real-time strategických her. [19]

Pro účely této hry ale není třeba engine tak propracovaný, spíše nějaký engine pro tvorbu rpg her nebo adventur. Source je navíc pro tento případ velmi složitý, jiných herních platform je ale velké množství.

**Klady:**

- Velké množství textur a výběru prostředí
- Engine obsahuje výběr z několika druhů postav
- Grafické zpracování

**Zápory:**

- Nejedná se o freeware
- Nutnost tvorby vlastních entit
- 3D prostředí – složitost tvorby entit

Až příliš velká složitost

#### **4.1.4 Aurora engine**

Engine Aurora (viz Obr. 4.2) je 3D nástupce dřívějšího 2D Infinity engine od firmy Bioware. Engine umožňuje real-time osvětlení a stíny a dále prostorový zvuk. Tím se liší od svého předchůdce, který byl použit na legendární hru Baldur's Gate. První použití Aurora engine bylo ve hře Neverwinter Nights, která také obsahovala Aurora toolset pro uživatele, kteří si chtěli vytvořit svoje lokace a příběhy. Další díl, Neverwinter Nights 2, už nevyvíjela firma Bioware, ale firma Obsidian Entertainment. Na druhý díl této série byl použit vylepšený engine, který se nazýval Electron Engine. Hry jako Star Wars: Knights of the Old Republic 1 a 2 zase využívaly ukatovanou verzi Aurora engine, zvanou Odyssey Engine. Aurora Engine byl také použit ve hře zaklánač, i když renderovací engine byl zde psán od nuly. [21]



Obr. 4.2: Aurora Engine, zde Neverwinter Nights [1]

Původním záměrem bylo použít editor hry Neverwinter Nights. Výhodou tohoto řešení je nesporně ta věc, že veškeré komponenty, grafika, zvuky apod. jsou v programu implementované a jde tedy pouze o to si vytvořit postavy a příběh a hra je víceméně hotova. Postavy mají také svůj inventář, což je v tomto případě potřeba. Věc, která zde naopak potřeba nebyla, je celková rozsáhlost herního systému. Postavy mají přiřazeny vlastnosti, které se jim postupem hry vylepšují, používají kouzla, zbraně, brnění a podobné věci. Pro účel našeho výukového programu je tohle poněkud zbytečné.

#### **Klady:**

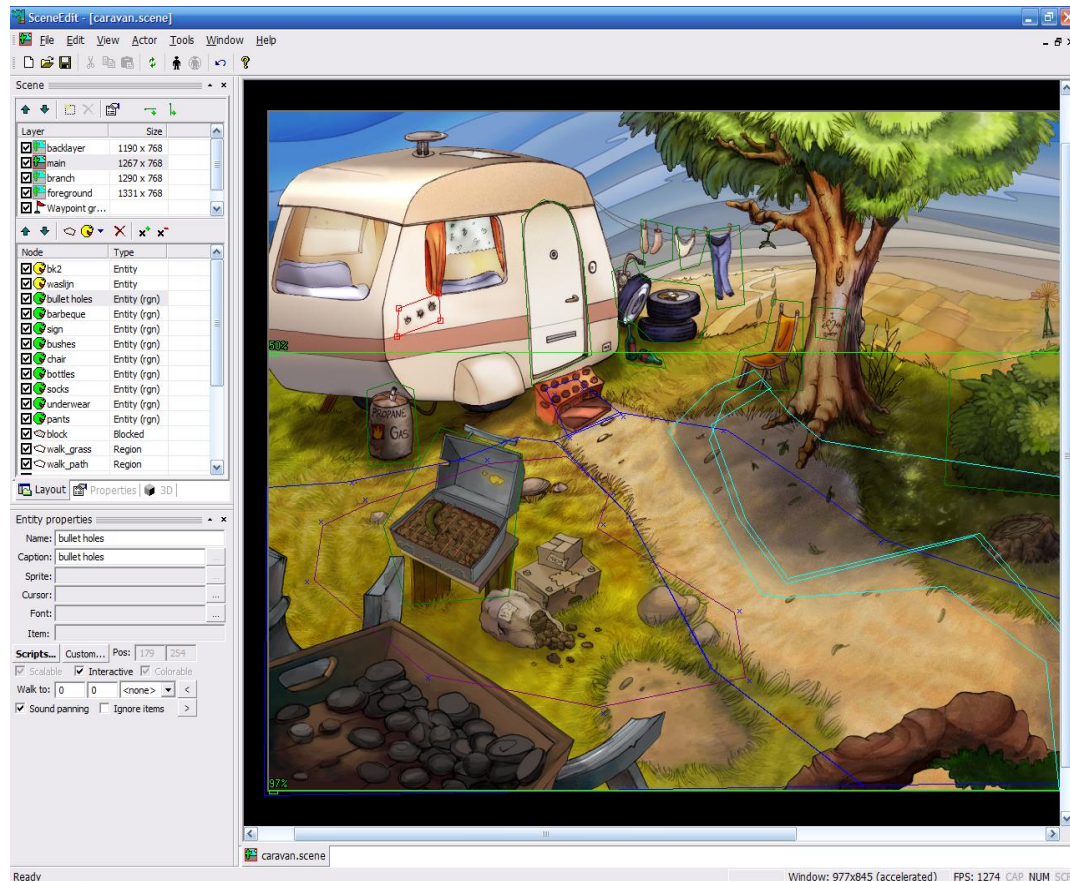
- Velké množství postav bez nutnosti jejich úpravy
- Možnost volby hlavní postavy
- Engine je určen pro tvorbu her, bohatých na dialogy
- Možnost vylepšování postavy
- Velké množství předmětů, které lze použít

#### **Zápory:**

- Nejedná se o freeware
- Nemožnost importovat vlastní entity (tato možnost je zcela zásadní)
- Nemožnost nalezení některých informací k tvorbě
- Spíše pro tvorbu misí, než pro tvorbu nového konceptu hry

### 4.1.5 Wintermute Engine

Wintermute Engine Development Kit je sada nástrojů pro tvorbu a spuštění grafických “point&click” adventure her a to jak tradičních 2D, tak moderních 2.5D her (3D postavy na 2D pozadích). Sada obsahuje interpreter her (Wintermute Engine, nebo WME) a grafické nástroje pro vytváření a správu vlastního obsahu hry (WME nástroje). [22]



Obr. 4.3: Wintermute engine - editor scén [1]

#### Klady:

- Freeware
- Početná komunita, která se zabývá tvorbou v tomto engineu
- Český program, česká komunita
- Jednoduché skriptování
- Instruktažní videa na [www.youtube.com](http://www.youtube.com)
- Entity jsou vytvářeny jako obrázek + skript, lze tedy neimportovat cokoli
- Prakticky neomezené možnosti tvorby
- Komponentní programování, kdykoliv tedy lze přidávat, či ubírat prvky
- Pravidelná komunikace s tvůrci her v tomto engineu

#### Zápory:

- Tvorba programu od nejjednoduššího základu
- Nutnost tvorby ilustrací
- Nízké rozlišení
- Nutnost tvorby hlavní postavy i vedlejších postav

## 4.2 Výběr vhodného žánru

Celý výukový program je postaven na základu počítačové hry, známé jako adventura. V adventuře vystupuje hlavní hrdina (nemusí být vždy jeden), který prochází vlastní hrou a řeší zde rozličné, většinou logické, úkoly (tzv. questy – z angl. slova posláni). Hráč postupuje hlavním příběhem ke vzdálenému cíli, přičemž je kladen důraz na řešení těchto hádanek [26]

Adventura je žánr, který je příběhově velmi spletitý a zajímavý, v tomto žánru hádanky přímo podmiňují zábavu. Strukturně je vytyčen hlavní úkol, hlavní dějová linie, po které hráč postupuje. Hlavní cíl nebo úkol klade před hráče směsici různorodých hádanek, přes které se hráč propracovává využitím různých logických myšlenek a své fantazie a tím naplňuje hlavní cíl hry. [26]

Například ve hře Horké Léto je hlavním úkolem najít svojí rodinu, kterou unesou kanibalové při dovolené v tropech. Rodina je však na jiném ostrově, hráč tak musí postavit vor, zlikvidovat žraloka a dostat se k nim.

V jiných vzdělávacích hrách zase hráč stojí před úkolem řídit společnost a zvyšovat zisk použitím nových technologií i strategie.

V našem výukovém programu se postupuje obdobným způsobem. V programu vystupuje hlavní postava, mladý začínající manažer, který bude postaven hned před několik úkolů. Protože je hra zaměřena primárně pro výuku převážně logistických metod, vyučovaných na Katedře Průmyslového Inženýrství a Managementu, týká se řešená problematika právě těchto věcí. Naprosto nejdůležitější věcí v tomto výukovém programu je propojení dvou prvků. Prvku výukového, aby celý program měl patřičný výukový a informační charakter, tak i prvku zábavného. Ten je velmi důležitý, uživatel musí mít celou dobu dojem, že hraje hru a nikoliv, že studuje skripta. Bylo nutné tedy vyřešit, jak do sebe zakomponovat oba prvky. Program se tedy nechová způsobem, kdy by uživatel pouze nastudoval informace a pak je někde použil formou testu. Do scénáře bylo třeba zakomponovat pokud možno co nejvíce interakce s prostředím. Uživatel zde tedy hovoří s postavami, mluví s nimi o jejich pracovních problémech a snaží se tyto problémy řešit. Řešení probíhá jak formou dialogů, tak i formou úkolů, kde má uživatel možnost si tyto metody „osahat“ v praxi. Jedná se o používání předmětů, interakce s nimi, jejich kombinace a spěšné použití. Program také musí být koncipován tak, aby uživatel sám od sebe věděl, co má dělat a nedocházelo k jeho „zaseknutí“ v určité lokaci, kde by dlouho hledal co s čím zkombinovat.

V tomto výukovém programu se postupuje trochu jiným způsobem, než u klasické adventury. Jelikož metody, které se zde používají, jsou jasně dané, nebyla zde velká možnost přizpůsobení a využití již výše zmíněné fantazie apod. Řešením je oddělení těchto věcí do dvou scénářových rovin. V první uživatel za pomoci informací, jak už formou skript, internetu, či informací v samotném programu, řeší otázky ohledně těchto metod, zodpovídá je ostatním postavám ve hře a názorně ukazuje, jak a kde se tyto metody používají. Toto je blíže popsáno v kapitole 6. V druhé rovině je zakomponovaný více zábavný příběh, kdy se například hledá po podniku zapomenutá karta, důležitá pro další chod příběhu. Zde bude nutné použít logické uvažování a fantazii. V programu jsou tyto dva aspekty propojeny. Pokud je vše vytvořeno dobře, uživatel by si tyto věci neměl ani uvědomit, což je jeden z hlavních cílů tohoto programu. Tzn. Učení se zábavnou formou, přičemž uživatel bude sám aktivně chtít informace získávat.



### 4.3 Výběr vhodného enginu

Každý z nabízených herních enginů nabízí širokou škálu možností, výhod a nevýhod. Protože autor neměl zkušenosti s tvorbou programů tohoto typu, byla pro něj prioritní kromě dostupnosti programu, také uživatelská podpora. Jelikož jediný Wintermute Engine disponuje širokou uživatelskou základnou, vše postupně směřovalo k použití právě tohoto editoru. Zejména po nalezení instruktážních videí na internetu, která vytvořil jeden z dlouhodobých uživatelů tohoto softwaru. Dalším faktorem je jakákoliv možnost přizpůsobení prvků programu, tvorba vlastních entit (zde nutná) a jeho hardwarová nenáročnost. Proces výběru vhodného nástroje, viz Tabulka 1.

	Source Engine	Aurora Engine	Wintermute Engine
Freeware	ne	ne	ano
Předpřipravené prostředí	ano	ano	nutnost vlastních ilustrací
Možnost importu nových entit	ano	ne	ano
Tvorba rozvětvených dialogů	ne	ano	ano
Informace o tvorbě v enginu	ne	ne	ano
Komunita uživatelů enginu	ne	ne	ano, i česká komunita
Tvorba skriptů	ne	ne	ano
Tvorba řídicích proměnných	ne	ne	ano
Dostupnost editoru	placená	placená	volně ke stažení

Tabulka 1: Srovnání enginů pro tvorbu programu

## 5 Wintermute engine

Po odzkoušení několika dalších enginů byl vybrán a Wintermute Engine Development Kit, zkráceně WME. WME je softwarový nástroj, který je produktem českého programátora Jana Nedomy.

Wintermute Engine je volně stažitelná platforma, dostupná na internetových stránkách: <http://dead-code.org/> a skládá se z několika navzájem propojených editorů, například editoru postavy, editoru scény a aktivních prvků. Toto je blíže popsáno v kapitole 7.

Wintermute Engine (WME) byl vytvořen českým programátorem Janem Nedomou, který je znám na internetových fórech pod jménem Mnemonic. První veřejná beta verze programu byla zveřejněna v lednu 2003. Engine je aktivní, každý rok je zveřejněno množství updatů. Tvorba scén v enginu Wintermute je zobrazena na Obr. 4.3 [22]

### 5.1 Vlastnosti

Wintermute engine je plný funkcí, které umožňují tvůrcům hry co největší flexibilitu a volnost při tvorbě her. [23]

#### **Rozlišení a barvy**

Engine podporuje v podstatě jakékoliv rozlišení. Hry mohou být tvořeny v zastaralém rozlišení 320x200 až po modernější 1920x768. Podporovaná barevná hloubka je 16 i 32bitů. Hráč si může sám vybrat barevnou hloubku, odpovídající výkonu jeho počítače, engine provede konverzi barev automaticky.

#### **Renderovací subsystém**

Jelikož jsou dnes běžné 3D akcelerátory, i WME dokáže využít 3D akcelerace pro rychlé vykreslování 2D grafiky ve vysokém rozlišení a to i s grafickými prvky jako jsou průsvitnost, či vyhlazování hran (antialiasing). Na starších počítačích lze WME spustit pod tzv. režimem kompatibility, který 3D akcelerátor nepožaduje, ale pokročilé grafické prvky jsou vypnuty.

#### **Vývojářské nástroje**

pro zrychlení vývoje je ve WME přítomna řada grafických nástrojů pro návrh jednotlivých scén hry, animací a pro správu souborů hry. Nástroje interně využívají pro vykreslování přímo herní engine. Momentálně je ve vývoji nová generace podpůrných nástrojů.

#### **Podporované druhy souborů**

Engine umožňuje používat různé druhy souborů pro grafiku a zvuky. Kromě toho používá vlastní formáty souborů pro definování herních objektů. Podporované grafické formáty: BMP, TGA, PNG a JPG. WME podporuje PNG a TGA soubory s alfa kanálem. Podporované zvukové formáty jsou: Ogg Vorbis (OGG) a WAV. Ačkoliv WME nepodporuje formát MP3 kvůli licenčním komplikacím, používá jako náhradu formát Ogg Vorbis. Tyto zvukové formáty lze použít jak pro zvukové efekty, tak pro hudbu. Velké zvukové soubory jsou během hraní "streamovány" z disku a nezabírají zbytečně paměť. Kromě toho umí engine přehrávat video ve formátu Ogg Theora a AVI, včetně automatického zobrazování titulků (formát SUB). [23]

### **Skriptovací jazyk**

WME nabízí flexibilní, objektově orientovaný skriptovací jazyk, pomocí kterého je vývojář schopen do hry začlenit téměř jakoukoliv vlastnost, hádanku, či minihru. Všechny herní objekty nabízejí sadu metod a atributů, které umožňují snadno ovládat herní engine. Také je možné vytvářet vlastní objekty a upravovat vestavěné metody. Skriptovací jazyk používá běžnou syntaxi, podobnou jazyku C.

### **Balíky**

Než je hotová hra distribuována, může být zabalena do jednoho nebo více "balíků". Balík obsahuje všechny soubory hry ve zkomprimovaném tvaru. Vývojář může zvolit, jestli rozdělí hru do více balíků, například pokud chce zvlášť nabízet vlastní hru a zvlášť zvuky, nebo vydává-li hru po jednotlivých kapitolách. Balíky mohou mít různou prioritu; toho lze využít třeba při vydávání opravných patchů. Balík - patch bude obsahovat pouze změněné soubory a bude mít vyšší prioritu, než původní balíky hry. Engine pak použije přednostně změněné soubory. Během vývoje není třeba balíky kompilovat. Engine dokáže pracovat přímo nad zdrojovými soubory, což ulehčuje vývoj a ladění hry.

### **Podpora pro 3D postavy**

WME obsahuje podporu pro real-time 3D postavy. To umožňuje WME vývojářům vytvářet moderní adventure hry, které kombinují 2D prostředí s 3D postavami (tzv. 2.5D hry), jako jsou například Syberia či The Longest Journey.

### **Vrstva pro tvorbu uživatelského rozhraní**

Pomocí několika základních stavebních prvků (jako jsou okna, tlačítka atd.) lze sestavit komplexní uživatelské rozhraní pro danou hru. Například okno pro nahrání/uložení hry, okno nastavení, inventář a podobně. Všechny ovládací prvky jsou plně "skinovatelné", lze tedy změnit jejich vzhled tak, aby odpovídal stylu hry.

### **Podpora lokalizace**

WME také umožňuje přeložit hru do jiných jazyků. Lokalizace není omezena pouze na texty, lze vytvořit lokalizační balíky pro jednotlivé jazyky, které budou obsahovat přeloženou tabulku textů, písma, grafiku nebo dabing v jiném jazyce.

### **Podpora usnadnění**

WME nabízí několik možností usnadnění pro zrakově postižené hráče. Jednak je možno zapnout přesměrování psaného textu do hlasového syntezátoru, takže hráč nemusí číst titulky, které jsou často příliš malé, nebo mizí příliš rychle. Dále je možné pomocí klávesové zkratky zvýraznit aktivní místa na obrazovce, jejichž nalezení může být pro zrakově postiženého hráče leckdy obtížné. Poslední možností je možnost kdykoliv pozastavit hru pomocí klávesové kombinace Ctrl+Mezerník, takže hráč může v klidu dočíst titulek či prozkoumat scénu, aniž by něco přehlédl.

### **Stálý vývoj**

WME je neustále vyvíjeno. Updaty s novými funkcemi a opravenými chybami jsou zveřejňovány prostřednictvím WME fóra. Program Project Manager navíc umí automaticky zjišťovat nové verze na internetu, takže si můžete stáhnout update okamžitě po jeho zveřejnění. [23]

## 5.2 Prvky WME

WME je soustava několika navzájem spolupracujících komponent. Základem je tzv. „Project manager“, ve kterém je tvořen celý výsledný program. V Project manageru jsou již samotné nástroje pro tvorbu konkrétních postupů. Je to zejména Scene editor, ve kterém se provádí tvorba vlastních scén, neboli lokací. Dalším editorem je Window editor, ve kterém je prováděna tvorba oken. Okna jsou prvky, které lze otevírat v rámci scény. Pro naprogramování dílčích entit, jako je například informační tablet, je zbytečné a nepraktické používat novou scénu. Je to tedy provedeno pomocí okna. Následuje tzv. sprite editor, kde jsou vytvářeny všechny předměty, aktivní oblasti programu a jsou zde vytvářeny i všechny postavy. Posledním prvkem programu je okno pro tvorbu skriptů. To má podobu klasického programu MS Notepad. Práce s jednotlivými editory je popsána v kapitole 7.

## 5.3 Ukázka hry, vytvořené ve Wintermute engine

Jedna ze známějších her, tvořená ve Wintermute engine je hra s názvem Alpha Polaris. Odehrává se uprostřed sněžných plání Grónska, kde leží Americká ropná výzkumná laboratoř. Vysoko nad vším se zrovna formuje rozsáhlá iontová bouře, která s sebou přináší zvláštní kombinaci reality a nočních můr. [23] Obrázek ze hry: Obr. 5.1.



Obr. 5.1: Alpha Polaris [25]

## 6 Proces tvorby výukového programu

Tato kapitola se blíže zabývá teoretickou částí vývoje programu. Je zde popsáno k čemu má program sloužit, co je jeho cílem kroky, které předcházely samotné tvorbě programu v editoru.

### 6.1 Účel vzdělávacího programu

Jak již bylo uvedeno, práce se zabývá tvorbou vážné hry, která obsahuje ve scénáři informace, studentovi užitečné ke studiu. Aby měla hra smysl a nebyla to pouze edukativní hra bez formy jakéhokoliv zábavy, je nutné do hry přidat prvky, které tento koncept rozvedou dále a student, u hry vydrží sedět a bude schopen tyto informace vstřebat a zapamatovat si je.

První zásadní myšlenkou je hlavní postava, kterou bude uživatel ovládat. Ta bude mít roli některé z nižších vedoucích pozic v nějaké fiktivní firmě. Ve firmě jsou kanceláře, kde ředitel nebo jeho přímý nadřízený dává postavě (uživateli) úkoly a ten je pak následně plní. Důležitou součástí programu bude informační kancelář, kam uživatel může pokaždé zajít a zeptat se na problém, kterému nerozumí a zde mu poradí.

### 6.2 Návrh interface a ovládání

WME je postaven pro jednu hlavní postavu, kterou ovládá uživatel, s podporou vedlejších postav. I ty jsou zde využity. Nejzásadnější věcí zde jsou scény, neboli lokace. V těch se odehrává děj a dochází zde k interakci s prostředím. Scény jsou tvořeny ilustracemi, entity na scénách jsou buď součástí ilustrace, nebo jsou dodávány zvlášť. Další součástí interface je inventář, který je již v základním nastavení editoru. Sem se ukládají předměty, sebrané na scéně a odtud jsou i pak následně použity.

Nový prvek interface, který není zahrnut v základním nastavení editoru, je informační tablet. Ten je vytvořen jako sekvence několika oken, do nichž je vložen učební text a uživatel má kdykoliv možnost stiskem klávesy F1 tablet otevřít a přečíst si potřebné informace.

Ovládání programu vychází ze základního nastavení WME, tedy klasického ovládání myší. Právým tlačítkem dochází k prozkoumání daného předmětu, po kliku levým tlačítkem postava zpravidla k předmětu dojde a použije ho. Použitím se rozumí sebrání předmětu ze scény, popřípadě dialog s vedlejšími postavami. Dalším prvkem ovládání je již zmíněný informační tablet, který je otevřen klávesou F1. Zavírán je pak kliknutím na logo dole uprostřed.

### 6.3 Ilustrace

Wintermute Engine podporuje široké rozmezí formátů ilustrací. Ty jsou zde třeba na veškerou grafickou stránku programu. Nejprve bylo nutné vytvořit lokace. Lokacemi se rozumí obrázek místnosti ve formátu, ve kterém bude program běžet, zde tedy 1024x768 obrazových bodů. Těchto ilustrací je použito celkem 6 a z nich jsou vytvořeny scény. Dále je zde množství menších obrázků, ze kterých je vytvořena hlavní postava, vedlejší postavy a předměty, se kterými uživatel pracuje. Ilustrace jsou tvorbou studentky 1. ročníku Ústavu umění a designu na Západočeské Univerzitě v Plzni, Pavlína Kunešové.

**Ilustrace: Viz příloha 3**

## 6.4 Scénář

Při tvorbě celého programu je scénář první a zcela zásadní věcí při jeho tvorbě. Podle scénáře se vytvářejí lokace, skripty, vedlejší postavy a proměnné.

Hlavní část programu se odehrává v průběhu dvou dnů. První den začíná tím, že uživatel musí s hlavní postavou dojít k řediteli do kanceláře pro instrukce. V kanceláři ředitele se uživatel dozví, že dělník v dílně má problém s metodou Kanban a že celý Kanban nefunguje. Je tedy nutné mu vše vysvětlit. V rozhovoru se ředitel zmíní o Kanbanu a následuje odpověď s možnostmi a,b,c. Při odpovědi a nebo b uživatel řekne řediteli, že o Kanbanu moc neví a ředitel nás následně pošle do kanceláře sekretářky, která nám má vše vysvětlit. Pokud uživatel zvolí možnost c, je seznámen s Kanbanem nebo alespoň se základy a může pokračovat rovnou do dílny. Do kanceláře sekretářky tedy může, ale nemusí.

V kanceláři sekretářky dojde k jednomu rozhovoru o metodě Kanban. sekretářka poskytne uživateli informace o metodě v tabletu, viz obr. 4. Tablet se aktivuje stiskem tlačítka F1 a vypíná se kliknutím na logo hrušky dole (logo zčervená po najetí kurzorem na něj).

Po tomto rozhovoru se uživatel přesune do dílny, kde následuje rozhovor s dělníkem. V tomto rozhovoru uživatel dělníkovi vysvětluje, co je metoda Kanban, čím se zabývá a jaké to bude mít pro něj přínosy. Rozhovor opět obsahuje otázky a několik možných odpovědí, z nichž je vždy jen jedna správná. Pokud uživatel zvolí špatnou odpověď, dělník na to zareaguje a vše se vrátí k poslednímu dialogu.

Zde je rozcestník. Pokud uživatel volí odpovědi správně, proběhne rozhovor a dělník se zeptá, zda by mu uživatel mohl ukázat, jak celý Kanban funguje. Uživatel tedy z červené bedny sebere kartu a vloží jí do kanban tabule, kde se karta vzápětí objeví. Také přibude jedna červená bedna s polotovary. Celá činnost je provázena komentářem.

Pokud uživatel bude volit špatné odpovědi, dělník pozná, že o tom moc neví a řekne, že na tohle nemá čas. Místo toho uživatele pošle do skladu, pátrat po nezvěstném skladníkovi. Uživatel se tedy přesune do skladu, kde skladník není k nalezení. Jde se tedy zeptat do kanceláře sekretářky, kdy skladník naposledy dorazil do práce. Ta mu oznámí, že skladník v práci je, takže je nutné se vrátit zpět do skladu.

Po vstupu do skladu se zde již skladník nachází a dojde k rozhovoru s ním. Ukáže se, že skladník rozumí Kanbanu, ale že od dělníků nedostává kanbanové karty, proto dělníci nedostávají materiál. Poté se uživatel jde opět poradit se sekretářkou.

U sekretářky v kanceláři se uživatel dozví o metodě 5WHYS s tím, že se má skladníka zeptat na správné otázky.

Po návratu do skladu dojde ke zbytku rozhovoru, kde se uživatel dozví potřebné informace. Pokud uživatel při rozhovoru s dělníkem odpovídal správně, vysvětlí skladníkovi, že už všechno půjde v pořádku. Pokud odpovídal špatně, od skladníka se dozví, že dělníci Kanbanu nerozumí a poté se uživatel přesune opět do dílny

V dílně následuje dokončení rozhovoru s dělníkem a vysvětlení metody Kanban. Uživatel se přesune do kanceláře ředitele.

V kanceláři ředitele dojde k vyhodnocení a pochvale za dobrou práci. Konec prvního dne.

Druhý den je nutné opět zajít k řediteli do kanceláře pro instrukce. Zde se uživatel dozví, že se bude v podniku zavádět metoda Just in Time, a že ve skladu je stále problém s kartami, skladník je stále neobdržel od dělníka. Odchod z kanceláře na chodbu.

Na chodbě se uživatel dozví, že sekretářka není v práci a informace o metodě JIT si tedy uživatel musí zjistit sám, nebo se podívat znovu do tabletu, kde je vše vysvětleno. Poté následuje krátký kvíz mezi hlavní postavou a uživatelem o metodě JIT. Poté se uživatel s hlavní postavou přesune do skladu.

Ve skladu se uživatel nedozví nic nového, pouze, že skladník od dělníka nedostal kanban karty a tím pádem nemůže objednávat další dodávky materiálu. Uživatel se přesune do dílny.

V dílně se uživatel dozví, že dělník všechny karty odevzdal. Nicméně si není jistý, tak si vezme kopie, které mu dá dělník. Přesun do skladu.

Ve skladu si uživatel vezme od skladníka originály a porovná je s kopiemi od dělníka. Návrat zpět do dílny.

V dílně dojde k porovnání originálů s kopiemi a zjistí se, že jeden originál chybí. Pak si hlavní postava všimne, že na zemi je lepidlo a dělník odpoví, že do něj šlápl několikrát. Karta se mu tedy mohla nalepít na botu a upadnout někde jinde. Odchod na chodbu.

Na chodbě uživatel najde ztracenou kartu, zjistí, že jde o ztracený originál a přesune se do skladu.

Ve skladu dojde k dokončení rozhovoru se skladníkem a po zodpovězení několika otázek o metodě JIT se uživatel vrátí do kanceláře k řediteli pro pochvalu. Konec

**Scénář: Viz příloha 1**

**Vývojový diagram: viz příloha 2**

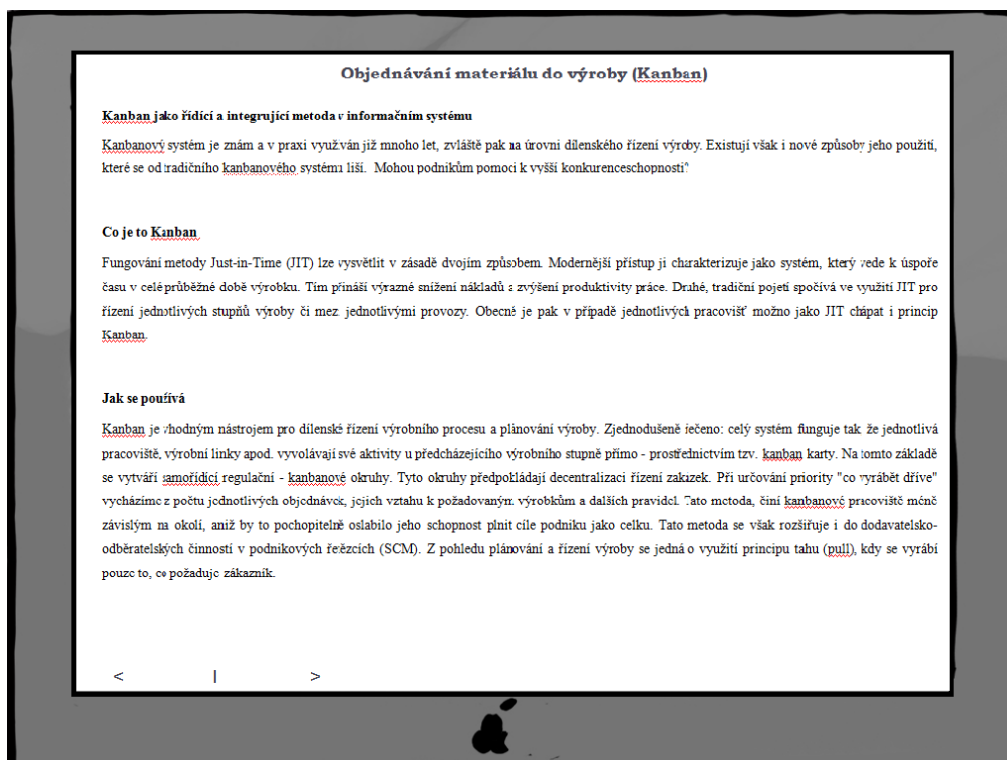
## 6.5 Tvorba struktury info-tabletu

Informační tablet je otevírán klávesou F1. Po otevření se objeví titulní obrazovka, kde si uživatel vybere, o jaké informace má zájem. Titulní strana, viz Obr. 6.1.



Obr. 6.1: Titulní okno tabletu

Poté si uživatel může vybrat z nabízených logistických metod. Jsou to metody: Kanban, Just in time a metoda 5 proč. Všechny tyto metody jsou v programu použity. Kliknutím na jednu z metod se otevře vlastní okno tabletu s textem. Okno Kanban 1, viz Obr. 6.2.



Obr. 6.2: Okno Kanban 1

Šipkami vedle čísla stránky se kliknutím myši listuje mezi okny. Kliknutím na logo hrušky dole uprostřed dojde k zavření všech oken tabletu a k návratu do původní scény, ze které byl tablet otevřen.



## 6.6 Tvorba scén

Scéna, nebo li lokace je místo v programu, kde se pohybuje postava a provádí veškeré činnosti s tím spojené. Programy podobného typu se liší svojí rozsáhlostí, zejména v počtu lokací. Vytvářený program obsahuje větší množství lokací. V první řadě musí uživatel provést nějakou činnost, nějaký cíl, kvůli kterému celý program využívá. V tomto případě se jedná o získání informací o některých logistických metodách. To se odehrává v dílně, kde uživatel komunikuje s dělníkem, dále ve skladu, kde komunikuje se skladníkem. Jelikož je zde nutné, aby někdo uživateli úkoly zadával, je pro tento případ vytvořena kancelář ředitele s ředitelem jako vedlejší postavou. Dále protože je nutné uživateli poskytnout informace o vyučovaných metodách, k tomu slouží druhá kancelář, kde se může dát do řeči se sekretářkou. Ta mu v některých věcech poradí. Posledním prvkem v programu je tablet. Po stisknutí tlačítka F1 se objeví tablet s potřebnými informacemi. To lze provést kdykoliv.

## 7 Tvorba samotného programu

V této kapitole je podrobně popsáno, jakým způsobem byl celý výukový program vytvářen. Tato kapitola bude blíže popisovat tvorbu scénáře celého programu, tvorbu jednotlivých scén, dále tvorbu hlavní postavy a tvorbu vedlejších postav celého příběhu. Jsou zde popsány nejdůležitější skripty, které mají za úkol celý program správně ovládat. Toho je docíleno soustavou proměnných, z nichž každá ovládá hned několik činností programu.

Celý program má hned několik částí. Hlavní část a několik menších částí. Hlavní část obsahuje celý scénář a uživatel se zde postupně seznámí se všemi metodami. Tři menší části jsou věnovány konkrétním metodám, Kanban, Just in Time a metodě 5WHYS. Zde je použita pouze část scénáře, věnující se konkrétní metodě a odehrává se pouze v jedné lokaci.

### 7.1 Základní pojmy

**Entita** – Množina objektů stejného typu, charakterizovaných názvem typu a popsáných pomocí jejich vlastností – atributů. Jednotlivé entity nazýváme také výskyty nebo instancemi objektů entitního typu. [28]

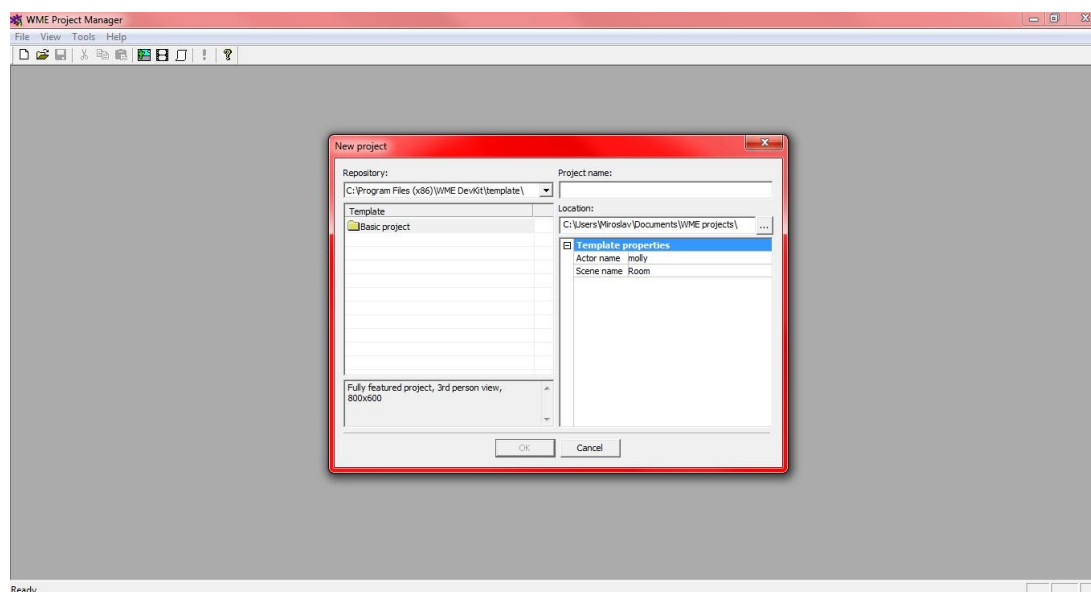
**Script** – Posloupnost příkazů, které ovlivňují chování entit

**Sprite** – Jedná se o dvourozměrný obrázek, který je integrován do větší scény [29]

**Aktivní oblast** – oblast, vytvořená kolem jednotlivých spritů. Na tuto oblast lze klikat a provádět s ní interakci pomocí hlavní postavy

### 7.2 Struktura WME

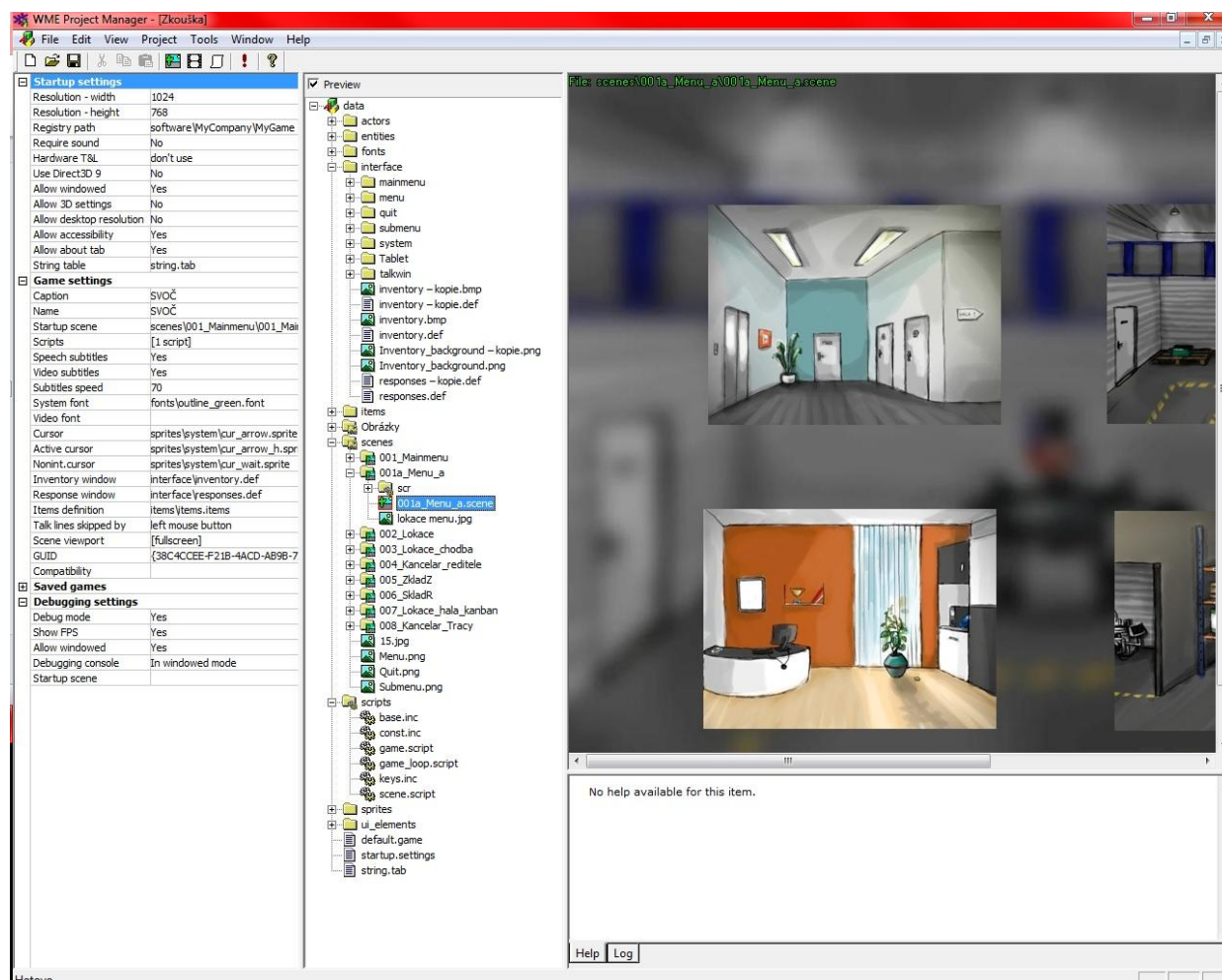
V této kapitole se seznámíme se základními kroky používání tohoto editoru. Po stažení a nainstalování editoru, dvojklikem na ikonu WME Project manager, celý editor otevřeme. Po otevření project manageru, kliknutím na File/new otevřeme okno pro tvorbu nového projektu, viz Obr. 7.1.



Obr. 7.1: Tvorba nového projektu

V tomto okně je nutné projekt pojmenovat a uložit. Project manager vytvoří základ programu a jeho adresářovou strukturu s několika předdefinovanými prvky, skripty, základními scénami bez obrázků a 3D postavu, Molly. Na Obr. 7.2 Je vidět adresářová struktura programu.

V levém sloupci s názvem Startup settings je prováděno základní nastavení výsledného programu. Všechny prvky jsou předdefinovány, je tedy nutné pouze změnit rozlišení na 1024x768 obrazových bodů. Dále v nižší části tohoto sloupce pod názvem Game settings je nutné nastavit počáteční scénu, která bude spuštěna se spuštěním výsledného programu. Nastavení počáteční scény je nutné provést samozřejmě až po jejím vytvoření.



Obr. 7.2: Adresářová struktura WME

V prostřední části se nachází adresářová struktura. Složky jsou seřazeny abecedně. Jako první je zde tedy složka actors. Tato složka je od začátku nadefinována a nachází se zde hlavní postava Molly, která je v průběhu tvorby programu nastavena postavou Greg. Složka entities obsahuje vedlejší postavy. Ve složce fonts jsou nahrány fonty pro jakýkoliv text v průběhu tvorby a užívání výsledného programu. Základní font outline je zde nadefinován vícekrát, pokaždé s jinou barvou. Jiná barva písma je následně využita pro ostatní postavy, popřípadě při kliknutí do textu, text změní barvu. Složka interface obsahuje strukturu. Obsahuje totiž všechna menu a submenu, dále inventář, informační tablet a vlastní metodu Talk. Ve složce items jsou uloženy všechny předměty, se kterými přijde postava do kontaktu ve výsledném programu. Složka obrázky je jediná složka, která není nadefinována předem. Složku lze nadefinovat buď zde přidáním nové složky pravým klikem, nebo v průzkumníku Windows.

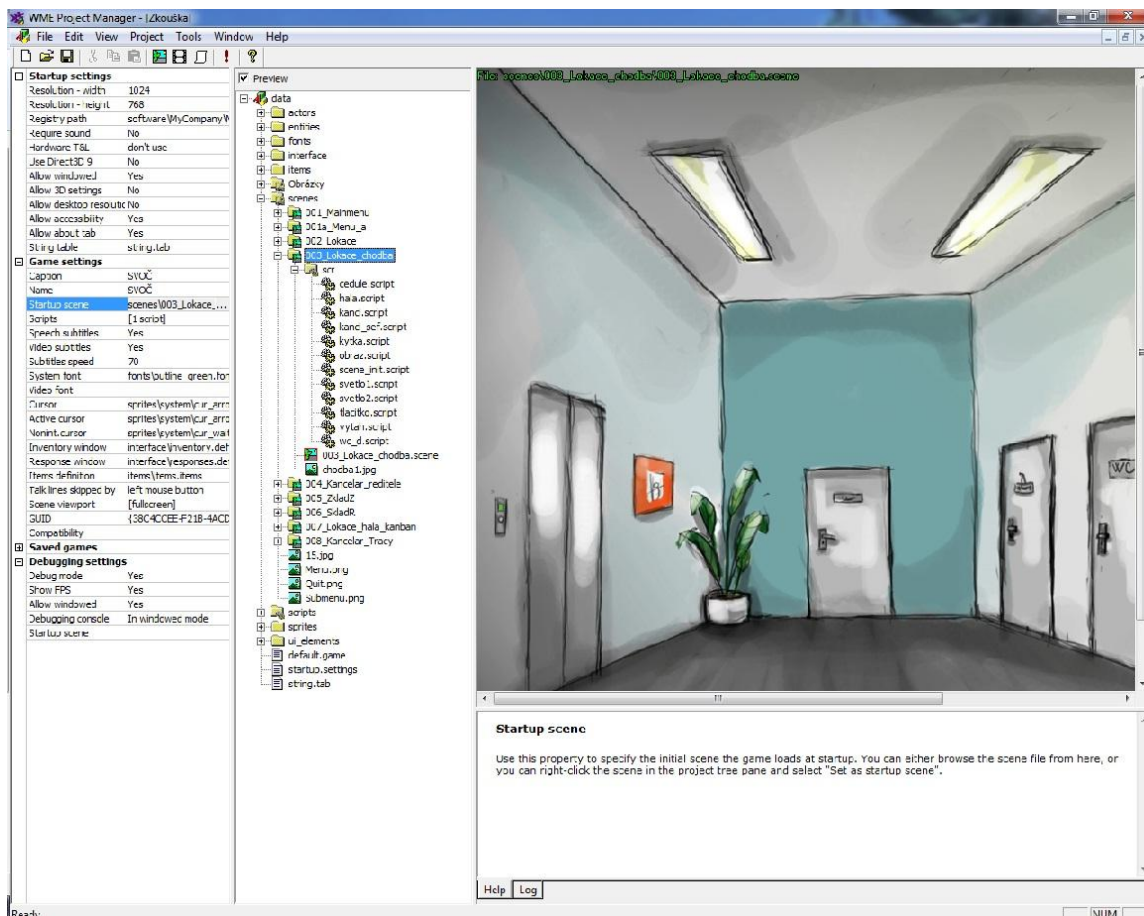
Poté je třeba kliknout pravým tlačítkem na strukturu nadřazenou (v tomto případě data) a zvolit možnost refresh. Ve složce scenes jsou uloženy veškeré scény, ve složce scripts pak hlavní skripty, nutné pro chod programu. Pomocné skripty nebo skripty k jednotlivým oknům, scénám a spritům, jsou uloženy ve stejné složce jako prvky, které ovládají. Ve složce sprites jsou uloženy základní sprity jako například kurzor. Ostatní sprity jsou rovněž uloženy ve svých složkách. Poslední složkou je složka ui\_elements, která obsahuje další pomocné sprity, jako například ikonky a části předdefinovaných oken, které v tomto programu nejsou použity.

### 7.3 Rozlišení

Celý program je tvořen v rozlišení 1024x768 obrazových bodů. Toto rozlišení bylo zvoleno kvůli nižší náročnosti, větší rozlišení totiž klade mnohem větší důraz na celkové propracování detailů a pro účely tohoto programu není tolik třeba. Další nevýhodou je, že Wintermute Engine umí zobrazovat běžící program na celou obrazovku právě pouze v tomto rozlišení. Od původního záměru, tedy vytvořit celý program v rozlišení Full HD, tedy 1920x1080 obrazových bodů, bylo nakonec upuštěno.

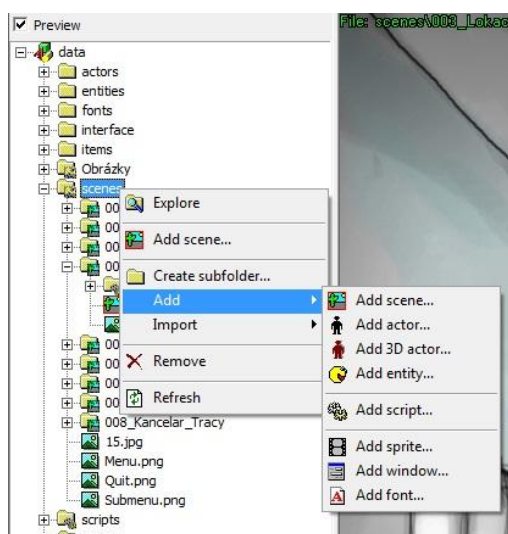
### 7.4 Tvorba scén

V této kapitole je podrobně popsána tvorba nové scény, pomocí scene editoru. Základní menu WME editoru a složka scenes je na Obr. 7.3:



Obr. 7.3: Hlavní menu editoru

Tvorba scén probíhá ve Scene editoru. Nejprve je třeba vytvořit novou scénu. V adresáři scenes, se pravým otevře následující okno, viz Obr. 7.4. Příkazem Add scene dojde k vytvoření nové scény.



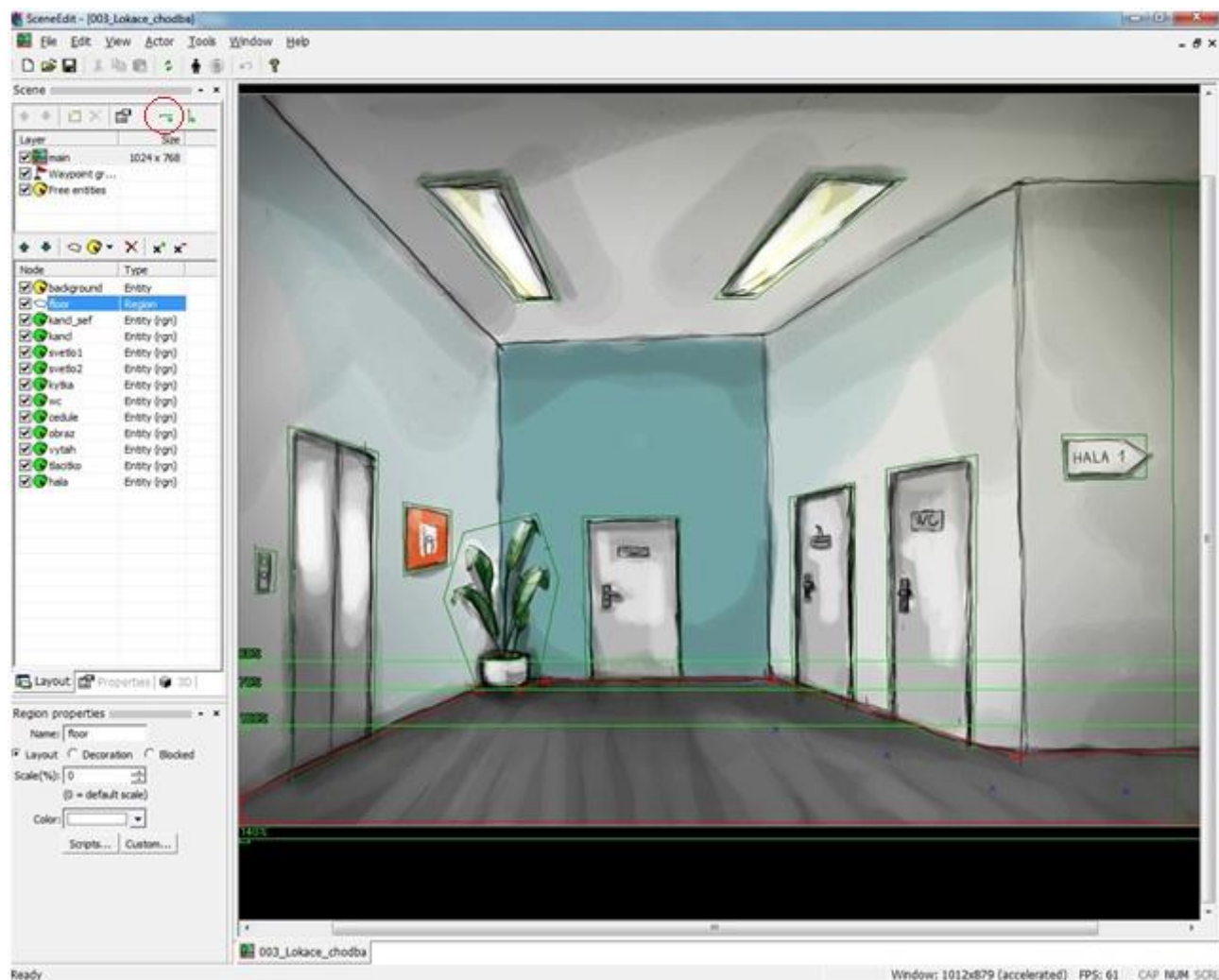
Obr. 7.4: Založení nové scény

Otevře se okno pro tvorbu nové scény. Zde je nutné scénu pojmenovat. Kvůli lepší přehlednosti jsou zde jednotlivé scény očíslované. Po otevření adresáře s novou scénou je nutné vymazat základní pozadí scény a nahrát sem ilustraci pro danou scénu. Dále následuje otevření Scene editoru, viz Obr. 7.5.

V levé části je umístěn seznam entit, použitých v této scéně. Jako první je vytvořena entita Background. Zde se žlutou ikonou, což značí, že jde o entitu typu sprite. Po kliknutí na entitu se ve spodní části editoru zadají souřadnice. U pozadí vždy  $x=0$ ,  $y=0$ . Pozadí je pouze obrázek ve formátu .jpg nebo .png, který je neaktivní a má rozměry, ve kterých je program tvořen. Zde tedy 1024x728 obrazových bodů. Aktivní oblasti se do scény přidávají pak.

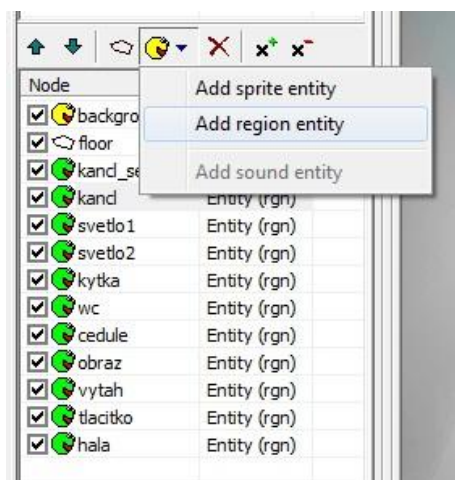
Jako první aktivní oblast na scéně je vytvořena podlaha. Na Obr. 7.5 je v levé části zakliknuta, na scéně je tedy vybarvena červeně. Podlahu ze začátku tvoří pouze obdélník. Kliknutím na tlačítko Add point, umístěné nad seznamem entit (černé x s horním indexem zelené +) se přidají další uzlové body, poté se podlaha vytvaruje do potřebného tvaru. Postava, chodící po scéně, se může pohybovat pouze po podlaze.

Aby byly entity viditelné v programu, je zde zaškrtnuta každá z nich hned vedle barevné ikonky. Pokud je nutné složit několik entit přes sebe, šipkami nad seznamem entit je nutné entity seřadit. Entita, která je níže v seznamu, je v programu na vrchu. Ve scéně 001a\_Menu\_a je podlaha v seznamu nad pozadím. Tedy je zde vidět pouze pozadí, hlavní postava je skryta.



Obr. 7.5: Scene editor

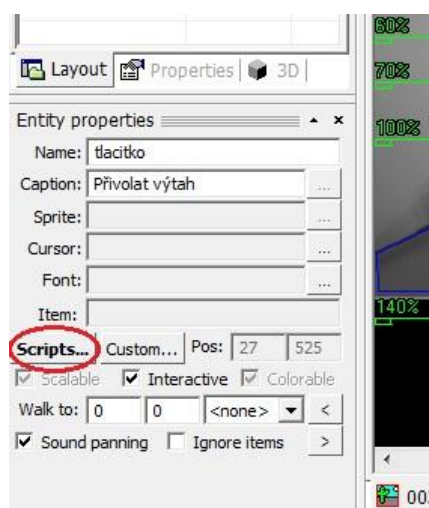
Dalším krokem je vytvoření ostatních aktivních oblastí. To se provede přidáním entit typu region (zelená ikonka). Pro každou aktivní oblast je nutná právě jedna entita. Přidání nové entity, viz Obr. 7.6:



Obr. 7.6: Přidání nové entity

Po kliknutí na novou entitu se otevře v dolní části okno Entity properties. Zde je nutné entitu pojmenovat do kolonky name. Zde bez háčků, čárek a mezer. Do pole caption je možné psát s interpunkcí. Caption je nápis, který se v programu objeví po najetí kurzorem na entitu. Poté se entita umístí do scény, upraví její velikost, popřípadě se přidají další uzlové body. Stejný postup jako u entity podlaha (floor).

Další častou činností je přiřazení skriptu k aktivní oblasti. Background a floor svoje skripty nepotřebují, tedy zde bude počet skriptů roven počtu sprite entit, plus jeden skript pro celou scénu. Skript je vytvořen následovně. Po kliknutí na entitu se v dolní části otevře okno Entity properties. Pod textovými poli name, caption a ostatními je tlačítko Scripts... Po kliknutí se objeví okno: Scripts for the "název objektu" object. Kliknutím na ikonu Edit script se otevře okno skriptu. Viz Obr. 7.7. Důležité upozornění. Okno Scripts for the "název objektu" object je nutno po editaci skriptu ukončit tlačítkem OK. Pokud dojde k ukončení jinak, skript se neuloží. Pokud je skript uložen a je aktivní, po kliknutí na entitu ve scene editoru je v dolní části tlačítko Scripts... zvýrazněné tučně, viz Obr. 7.7. červeně zakroužkované tlačítko. V tomto případě je zvýrazněné tučně, skript se tedy uložil. Pokud ne, skript se neuložil.

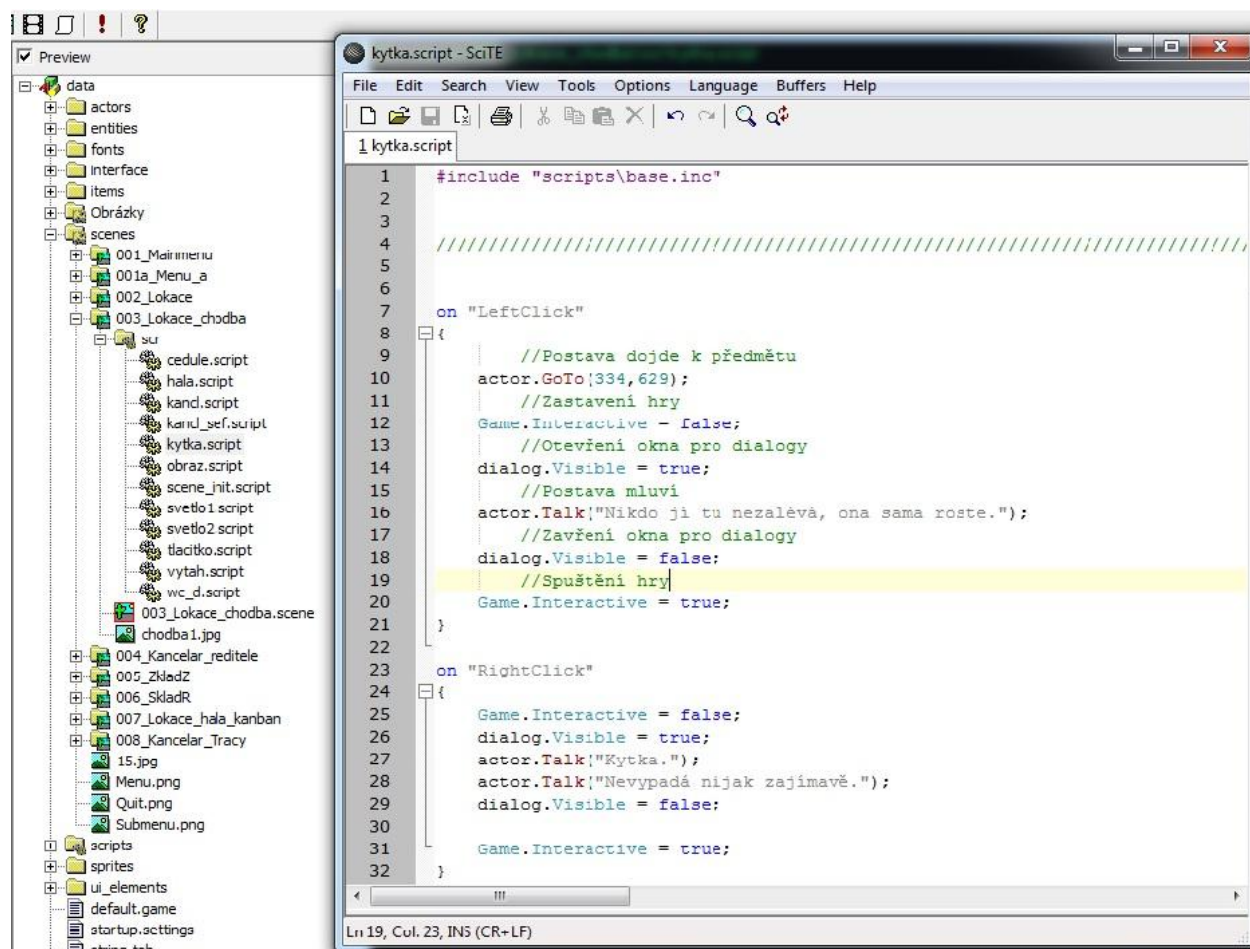


Obr. 7.7: Uložení skriptu k aktivní oblasti

Na Obr. 7.8 je vidět ukázkou skriptu pro entitu „kytka“. Po pravém kliknutí dojde k prozkoumání předmětu, postava řekne komentář. Po levém kliknutí postava dojde k předmětu a prozkoumá ho blíže. Pokud se jedná o jinou postavu, dojde k dialogu, u některých entit může dojít k jiné akci, například sebrání předmětu ze scény.

Tvorba skriptu je velmi jednoduchá. Procedura pro levé kliknutí začíná na řádce 7, na Obr. 7.8. Procedura pro levé kliknutí, příkaz on "LeftClick". Zelený text jsou komentáře. Ty se do skriptu vkládají po vložení dvou lomítek: //Komentář. Řádek 10: Postava dojde na souřadnice x=334, y=629. Příkaz pro otočení postavy do určitého směru zde není použit. Příkaz se zadává následovně: actor.TurnTo(DI\_UP); Up je směr, kdy je postava k uživateli zády. Program rozeznává celkem 8 směrů. RIGHT, LEFT, UP, DOWN a jejich kombinace. Při psaní skriptů je nutné dodržovat velká a malá písmena u příkazů. Například již zmíněný příkaz pro otočení: actor.TurnTo nebo pro chůzi: actor.GoTo. Příkazy Game.Interactive na řádcích 12 a 20 slouží k pozastavení hry během dialogů. Tento příkaz je použit pro jistotu, tím je zamezeno zablokování programu například při nechtěném kliknutí uživatelem. Po skončení dialogu je nutné interaktivitu programu opět zapnout. Příkazy dialog.Visible na řádcích 14 a 18 ovládají viditelnost dialogového okna. Bez něj by se text objevoval přímo nad hlavou

postavy, zde v tomto programu je použito řešení s dialogovým oknem ve spodní části obrazovky. Tvorba tohoto dialogového okna bude popsána v další kapitole.



Obr. 7.8: Tvorba skriptu

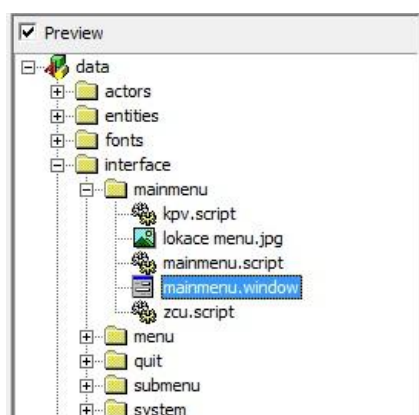
Posledním prvkem z tvorby scény je perspektiva. Scény jako například chodba na Obr. 7.5 je nakreslena do hloubky. Postava se tedy musí při pohybu dozadu (UP) zmenšovat a obráceně. Toho je docíleno zelenými vodorovnými čarami scale level s nadefinovanou procentuální velikostí postavy. Na scéně chodba jsou použity celkem čtyři. 60%, 70%, 100% a 120%. Když postava překročí tuto čáru, zmenší se. Nutné jsou minimálně dva, které jsou mimo podlahu, tedy oblast, kam postava může vstoupit. Tím nedochází ke skokovému zvětšení či zmenšení postavy při překročení čáry. Prostřední dvě jsou k upřesnění perspektivy, zde byla potřeba přizpůsobování velikosti postavy například dveřím na pravé straně chodby z pohledu uživatele. Linie scale level jsou k dispozici jak horizontální, tak i vertikální. Vložení těchto čar se provádí kliknutím na ikonu, která je červeně zakroužkována na Obr. 7.5.

## 7.5 Tvorba hlavního menu

Hlavní menu je důležitým prvkem celého programu. Obsahuje titulní stránku s ovládacími tlačítky Nová hra a Ukončit hru, dále odkazy na stránky univerzity.

Menu se nevytváří jako nová scéna, v programu je předdefinované, je tedy třeba jej pozměnit do jiného vzhledu. Hlavní menu se nachází ve složce interface/mainmenu/mainmenu.window, viz Obr. 7.9.

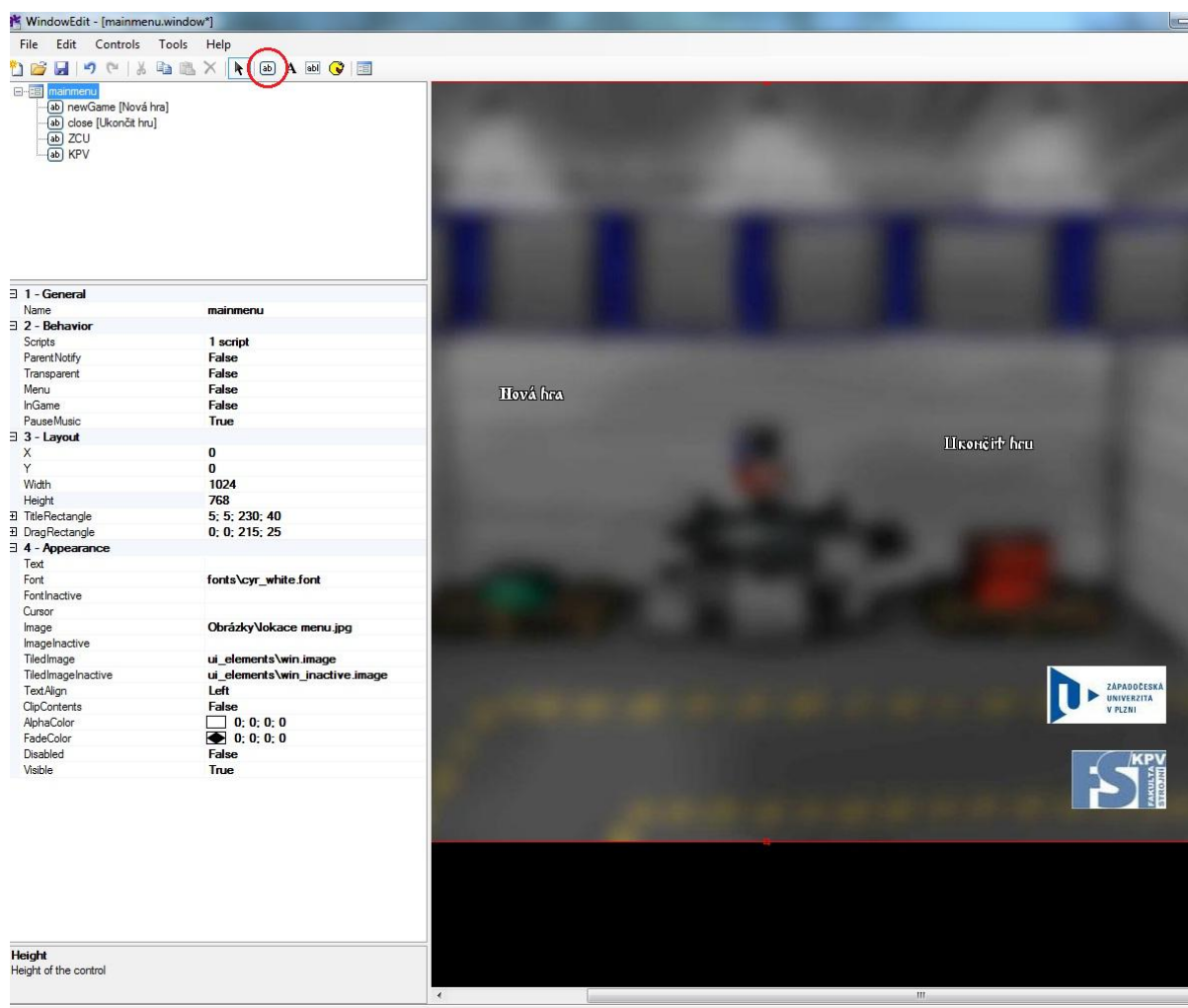




Obr. 7.9: Umístění hlavního menu

Kliknutím na okno se otevře editor oken (Window editor), kde se celé menu vytvoří. Dalším prvkem jsou skripty, které je nutno poupravit.

První věc je nahrání pozadí pro hlavní menu. Jako pozadí slouží ilustrace dílna, která je rozmazána v programu Adobe Photoshop. Po kliknutí na okno mainmenu, viz Obr. 7.10, mainmenu je označeno, dojde k rozbalení dialogové části vlevo uprostřed. U bloku 3-Layout se zvolí umístění pozadí, zde  $x=0$ ,  $y=0$ . Šířka a výška je nastavena přes celou obrazovku, tedy 1024x768 obrazových bodů.



Obr. 7.10: Editor okna hlavního menu

Poté lze přidávat tlačítka. Přidání tlačítka (button) se provede ikonou, která je na Obr. 7.10 zakroužkována červeně. Poté se tlačítko pojmenuje v textovém poli name a caption. V tomto okně jsou pouze dvě tlačítka. newGame, (caption: Nová hra) a tlačítko close, (caption: Ukončit hru). Tlačítka jsou bez obrázku, tvořena pouze textem. Font je zde zvolen cyr\_white.font a font hover, což je funkce, která mění vzhled při najetí kurzorem je zvolen cyr\_grey.font. Všechny fonty jsou uloženy ve složce fonts.

Do pozadí jsou přidána dvě loga. Jedno je logo Západočeské Univerzity a druhé je logo Katedry Průmyslového Inženýrství a Managementu. Kolem log se také vytvoří dvě tlačítka, která jsou však bez textu. Tlačítka jsou pojmenována ZCU a KPV a nemají caption. Nyní už zbývá jen přiřadit k celému menu skripty.

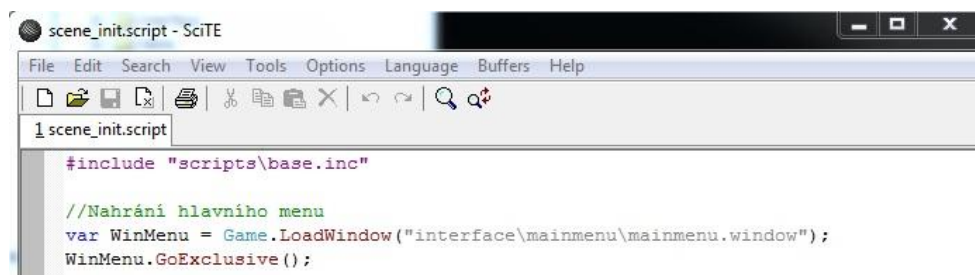
Ve skriptu mainmenu.script se přiřazují skripty k tlačítkům. První skript ovládá tlačítko Ukončit hru. Skript je na řádcích 8 až 19, Obr. 7.11. Na řádce 10 se nadeklaruje proměnná WinQuit a přiřadí se k ní akce: nahrát okno quit.window. Na řádce 11 se okno vycentruje a na řádce 12 se posune do popředí. Na řádce 14 je procedura, která při zavření programu vymaže okno z paměti. Tlačítko Nová hra je na řádcích 32 až 36. Na řádce 33 provede změnu scény na scénu Menu\_a a na řádce 34 se opět toto okno odstraní z paměti.

```
1  #include "scripts\base.inc"
2  #include "scripts\keys.inc"
3
4  this.xResult = false;
5
6
7  ///////////////////////////////////////////////////
8  on "close"
9  {
10     var WinQuit = Game.LoadWindow("interface\system\quit.window");
11     WinQuit.Center();
12     WinQuit.GoSystemExclusive();
13
14     if(WinQuit.xResult)
15     {
16         Game.QuitGame();
17     }
18     else Game.UnloadObject(WinQuit);
19 }
20
21 ///////////////////////////////////////////////////
22 on "yes"
23 {
24     this.xResult = true;
25     this.Close();
26 }
27
28
29 ///////////////////////////////////////////////////
30
31 ///////////////////////////////////////////////////
32 on "newGame"
33 { Game.ChangeScene("scenes\001a_Menu_a\001a_Menu_a.scene");
34   Game.UnloadObject(this);
35 }
36 }
```

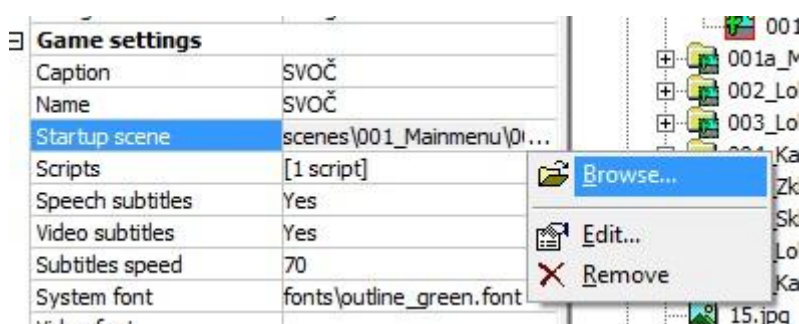
Obr. 7.11: mainmenu.script

Pro odkazy na webové stránky jsou použity dva samostatné skripty zcu.script a kpv.script. Každý obsahuje jeden řádek kódu `Game.OpenDocument("adresa URL");` Tím dojde k otevření těchto adres po kliknutí na tlačítko.

Protože se hlavní menu otevírá jako scéna, je třeba založit novou scénu s názvem `001_Mainmenu.scene`. Ta zůstává prázdná. Ve skriptu `scene_init.script` ve složce k této scéně se nadeklaruje proměnná `Winmenu`, do ní se nahraje okno s hlavním menu a opět se posune do popředí. Skript viz Obr. 7.12. Poslední věcí, která je třeba udělat, je zvolení hlavního menu jako výchozí scény. To se provede v hlavním editoru programu v levé části: `Game settings`, `Startup scene`. Zde se tedy zvolí scéna `001_Mainmenu.scene`, viz Obr. 7.13.



Obr. 7.12: Skript ke scéně hlavního menu

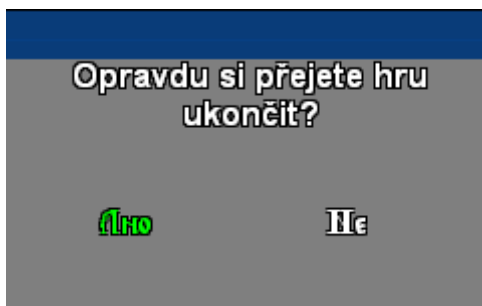


Obr. 7.13: Volba výchozí scény

## 7.6 Tvorba submenu

V této kapitole je popsáno, jak vytvořit submenu, totiž menu, které se aktivuje kdykoliv uživatel ve hře zmačká tlačítko `Esc`. Dále Okna `Quit`, `Load` a `Safe`. Všechna tato pomocná menu jsou uložena ve složce `interface/system`. Menu jsou v editoru již od začátku nového projektu, pro účely tohoto programu jsou pouze poupravena, co se týče vzhledu.

V okně `mainmenu` jsou celkem čtyři tlačítka. `Close` (Caption: Pokračovat ve hře) vrací uživatele zpět do hlavního programu, `Load` (Caption: Nahrát hru) otevírá okno pro nahrání uložené pozice. `Save` (Caption: Uložit hru) otevírá okno pro uložení pozice. `Quit` (Caption: Ukončit hru) otevírá okno pro ukončení hry. Stejně okno pro ukončení hry jako z hlavního menu. Postup vytvoření tlačítek se nijak neliší od tvorby tlačítek v hlavním menu, není třeba jej popisovat podrobněji. U ostatních oken se neměnilo nic, kromě vzhledu. Změna vzhledu okna je rovněž popsána v předchozí kapitole. Tyto pomocná okna mají všechna stejný jednoduchý vzhled, liší se pouze rozměry. Na Obr. 7.14 je okno `Quit`, které se otevře po stisknutí tlačítka `Ukončit hru`. Tlačítko `Ano` ukončí program, tlačítko `Ne` vrátí zpět. Při najetí kurzorem na tlačítko `Ano` se tlačítko vybarví zeleně. Tlačítko `Ne` červeně.

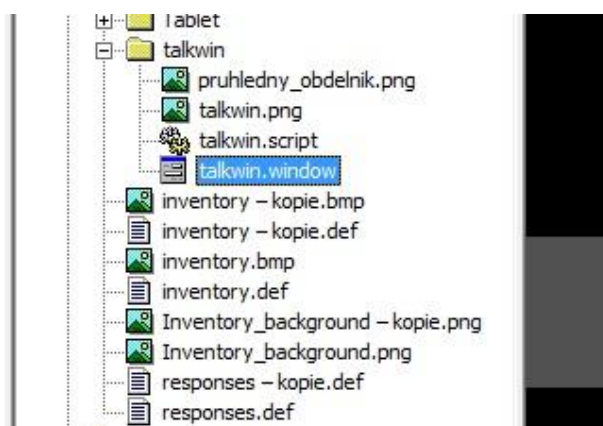


Obr. 7.14: Okno quit

## 7.7 Tvorba vlastní metody Talk

V této kapitole je popsán postup k vytvoření metody talk. Tato metoda složí k zobrazování textu při dialogu v dolní části obrazovky na poli k tomu určeném, na místo textu nad hlavou postavy. Toto původní řešení je nepraktické. Text je velmi malý a i po změně fontu je nepříjemné hledání textu pokaždé někde jinde. Po zavedení metody talk je text mnohem lépe čitelný. Pro hlavní postavu je zvolen bílý font, pro vedlejší postavy font oranžový.

Jako první je nutno vytvořit okno, kde se budou titulky zobrazovat. Ve složce interface je tedy nová složka talkwin, ve které je vytvořeno okno talkwin.window. Viz Obr. 7.15.



Obr. 7.15: Umístění metody talk

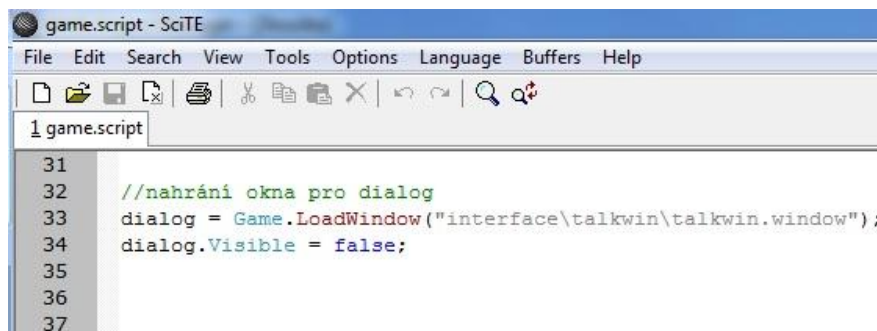
Do složky se nahraje obrázek ve formátu .png, pruhledny\_obdelnik.png. Formát .png je zvolen záměrně, podporuje totiž poloprůhlednost. Zde je zvolena průhlednost (transparency) 50%. Tím celé pole nepůsobí rušivým dojmem. Celý průhledný obdélník má rozměry 1024x80 obrazových bodů a je umístěn při dolním okraji obrazovky. Umístění je tedy  $x=0$  a  $y=768$  (výška obrazovky)  $- 80$  (výška okna)  $= 688$ . Z tvorby okna je toto vše, nyní zbývá jen přiřadit skript.

Ve skriptu base.inc, kde se deklarují proměnné je nutno nadeklarovat proměnnou dialog. Deklarace u této proměnné je provedena v tomto skriptu, protože ten je pro všechny scény a celý program, nelze tedy proměnnou deklarovat ve skriptu, patřícímu například jedné scéně. Pak by celá metoda nefungovala pokaždé. Deklarace proměnné se provede následovně:

`global dialog;`

Proměnná typu global je zde, protože se vztahuje na celý program. Dále je třeba tuto metodu aktivovat v hlavním skriptu hry Game.script. Viz Obr. 7.16. Na řádce 33 se okno nahraje do

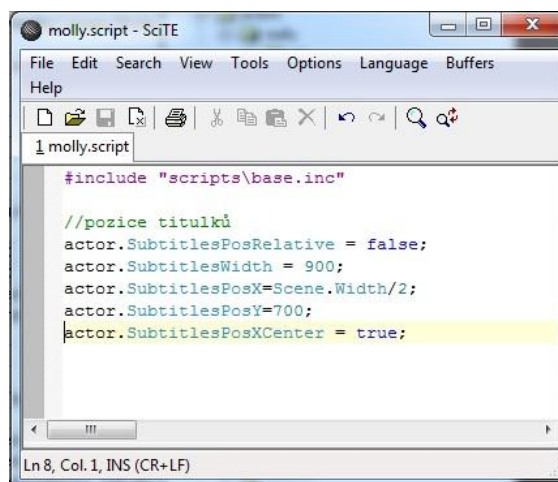
programu a na řádce 34 se odstraní jeho viditelnost. To, aby okno standardně vidět nebylo, aktivuje se pouze při dialogu. To je však nutné naskriptovat ke každému dialogu.



```
game.script - SciTE
File Edit Search View Tools Options Language Buffers Help
1 game.script
31
32 //nahrání okna pro dialog
33 dialog = Game.LoadWindow("interface\talkwin\talkwin.window");
34 dialog.Visible = false;
35
36
37
```

Obr. 7.16: Script dialogového okna

Do skriptu každé postavy je potřeba upravit pozici titulků, aby se titulky objevovaly na správném místě. Na Obr. 7.17 je skript hlavní postavy, kde je tato pozice upravena. Jejich šířka je 900, pozice na X-ové souřadnici je šíře scény/2, titulky tedy budou vždy uprostřed. Ypsilonová souřadnice titulků je 700.

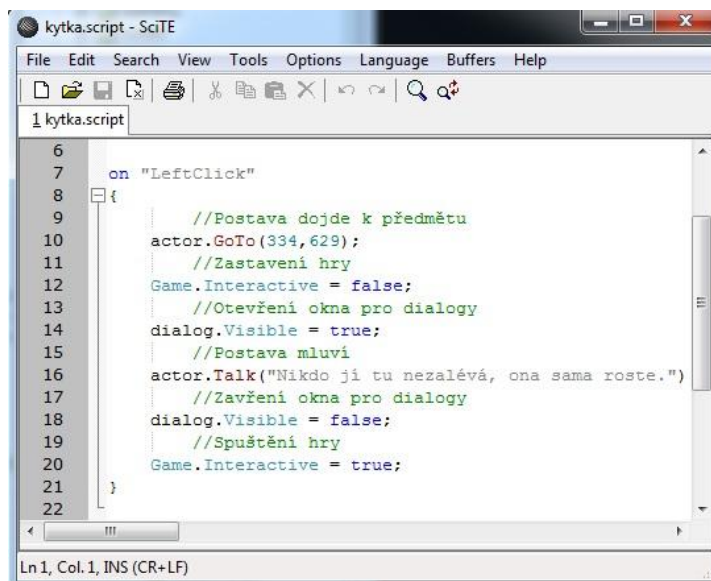


```
molly.script - SciTE
File Edit Search View Tools Options Language Buffers Help
1 molly.script
#include "scripts\base.inc"

//pozice titulků
actor.SubtitlesPosRelative = false;
actor.SubtitlesWidth = 900;
actor.SubtitlesPosX=Scene.Width/2;
actor.SubtitlesPosY=700;
actor.SubtitlesPosXCenter = true;
```

Obr. 7.17: Pozice titulků

U každého dialogu je potřeba na začátku okno zviditelnit a na konci dialogu opět viditelnost odstranit. Viz Obr. 7.18. Na řádce 14 dojde ke zviditelnění okna, na řádce 18 naopak.

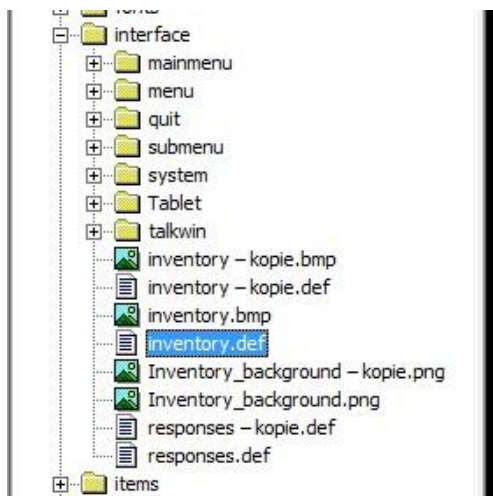


```
6
7 on "LeftClick"
8 {
9     //Postava dojde k předmětu
10    actor.GoTo(334,629);
11    //Zastavení hry
12    Game.Interactive = false;
13    //Otevření okna pro dialogy
14    dialog.Visible = true;
15    //Postava mluví
16    actor.Talk("Nikdo jí tu nezalévá, ona sama roste.")
17    //Zavření okna pro dialogy
18    dialog.Visible = false;
19    //Spuštění hry
20    Game.Interactive = true;
21 }
22
```

Obr. 7.18: Použití metody talk v rozhovoru

## 7.8 Tvorba inventáře

Inventář slouží k přenášení předmětů, prozkoumávání, jejich kombinaci a používání. Má formu okna v horním okraji obrazovky, které se vysune při najetí kurzorem do dané oblasti. Inventář je v editoru připraven již od začátku. Pro účely tohoto programu byl pouze pozměněn, co se týká grafické stránky. Umístění souboru pro tvorbu inventáře, inventory.def, viz Obr. 7.19.



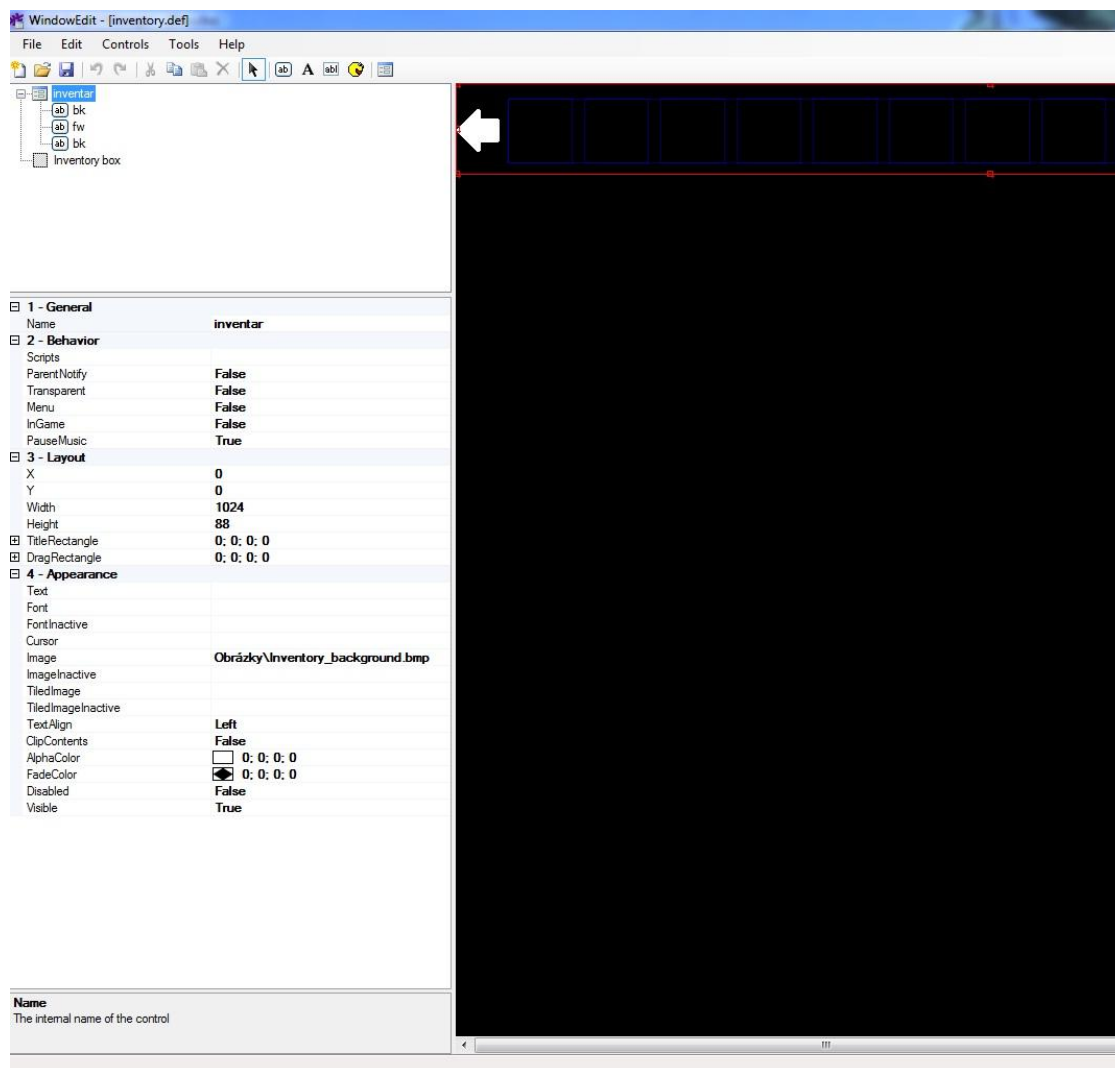
Obr. 7.19: Umístění okna inventáře

Původní inventář byl nevyhovující po grafické stránce. Stačí tedy pouze nahrát místo původního obrázku pro inventář obrázek nový. Obrázek inventory\_background.bmp se nachází ve složce obrázky. Protože okno inventáře má být průhledné, lze jej vytvořit dvěma způsoby. Prvním způsobem je vytvoření obrázku ve formátu .png s plnou průhledností. Druhý způsob, který je zde použit, využívá předdefinovanou vlastnost editoru, průhlednost určitých barev. Editor má od základu nadefinovanou jednu barvu, která je po spuštění editoru neviditelná, tedy průhledná. Jedná se o růžovo-fialovou barvu, rgb kód 255, 0, 255. Obrázek je tedy v této barvě, viz Obr. 7.20. Rozměry obrázku jsou 1024x88 obrazových bodů.



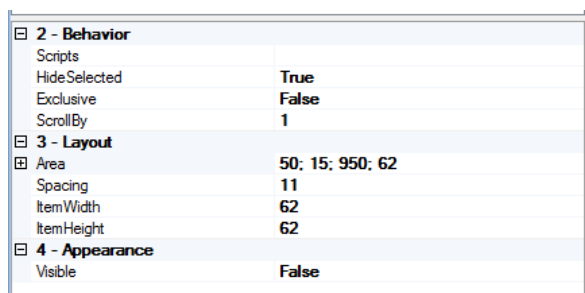
Obr. 7.20: Okno inventáře

V okně pro editaci inventáře, inventory.def, viz Obr. 7.21, se ve složce 3-Layout zvolí umístění tohoto okna  $x=0$ ,  $y=0$ . Okno je tedy v horní části obrazovky. Velikost okna se zvolí podle velikosti obrázku, tedy 1024x768 obrazových bodů. Dále jsou zde dvě tlačítka fw a bk, která slouží ke scrollování v inventáři zleva doprava a naopak. Tato tlačítka jsou již předem nafedinována, je tedy třeba je jen posunout a upravit, aby byla na bílých šipkách.



Obr. 7.21: Editace inventáře

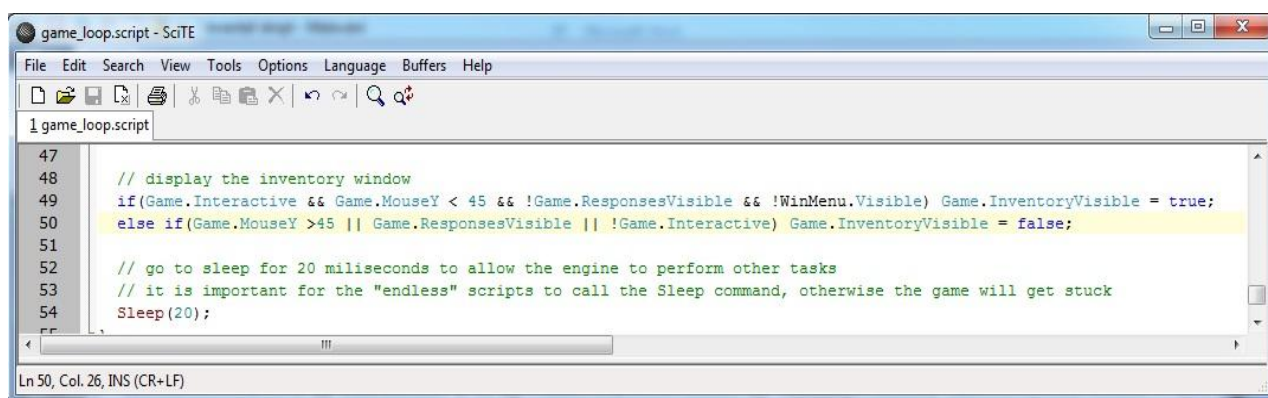
Dále je třeba upravit rozměry samotných políček, kam se budou ukládat předměty. Předměty mají všechny velikost 62x62 obrazových bodů, i okna pro ně mají stejné rozměry. Nastavení rozměrů oken se provede kliknutím na inventory box v levé horní části editoru okna, viz Obr. 7.21. Nastavení okna, viz Obr. 7.22.



<b>2 - Behavior</b>	
Scripts	
HideSelected	True
Exclusive	False
ScrollBy	1
<b>3 - Layout</b>	
Area	50; 15; 950; 62
Spacing	11
ItemWidth	62
ItemHeight	62
<b>4 - Appearance</b>	
Visible	False

Obr. 7.22: Nastavení inventory boxu

Vzdálenost boxu od levého okraje je 50, od horního okraje 15. Šířka je 950 a výška 62 obrazových bodů. Prostor mezi okny je 11 a šířka i výška oken je 62 obrazových bodů. Poslední věcí, kterou je třeba nadefinovat, je kdy se má inventář objevit na scéně. To je nadefinováno ve skriptu `game_loop.script` ve složce `scripts`. Skript, viz Obr. 7.23.



```
47
48 // display the inventory window
49 if(Game.Interactive && Game.MouseY < 45 && !Game.ResponsesVisible && !WinMenu.Visible) Game.InventoryVisible = true;
50 else if(Game.MouseY > 45 || Game.ResponsesVisible || !Game.Interactive) Game.InventoryVisible = false;
51
52 // go to sleep for 20 miliseconds to allow the engine to perform other tasks
53 // it is important for the "endless" scripts to call the Sleep command, otherwise the game will get stuck
54 Sleep(20);
```

Obr. 7.23: Script pro ovládání inventáře

Na řádce 49 je příkaz, když je kurzor na souřadnicích  $y < 45$ , okno inventáře je viditelné. Na řádce 50 je příkaz opačný, který okno schovává zpět. Tímto je nadefinován kompletně celý inventář.

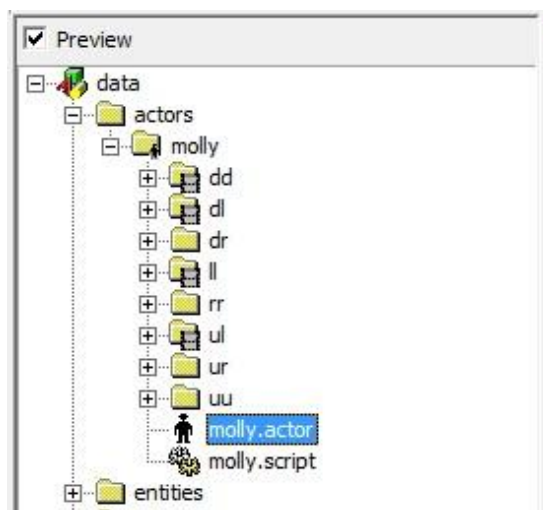
## 7.9 Tvorba hlavní postavy

Hlavní postava, kterou ovládá uživatel, je naprosto zásadní. S hlavní postavou uživatel prochází lokacemi a mluví s ostatními postavami a plní různé úkoly. Hlavní postava, v editoru nazvána jako Actor, je již předem nadefinována, včetně všech skriptů. Bylo nutné tedy vytvořit postavu novou a předdefinovanou postavu nahradit.

Editor podporuje formát 2D, nebo 2,5D, což je 3D postava na 2D pozadí. Původní postava z editoru, Molly, je vytvořena ve 3D. K tomuto však nebyly prostředky, byly k dispozici pouze 2D ilustrace a i samotný editor mívá problémy s postavami 3D, jedná se tedy o postavu 2D. Postavou měl být mladý muž, začínající manažer jménem Greg. Toto jméno vzniklo úplnou náhodou, více v příloze 4: Zajímavosti.

Zpět k původní postavě Molly. I když se jedná o 3D model postavy, samotný editor s 3D objekty pracovat neumí. Proto je tedy každý pohyb postavy rozdělen na sérii obrázků ve formátu `.bmp`, kde okraje, které nemají být vidět, jsou opět v růžovo-fialové barvě. Pro tu má editor nastavenou plnou průhlednost. Postava Molly je vytvořena pro celkem 8 směrů. Umístění hlavní postavy je v adresáři `actors`, viz Obr. 7.24.



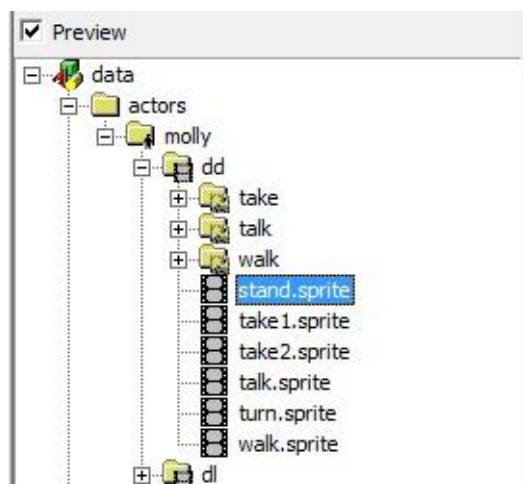


Obr. 7.24: Adresář s hlavní postavou

Postava Molly je vytvořena pro celkem 8 směrů, z nichž každý je ve svém adresáři, viz Obr. 7.24. Směry jsou označeny následovně: DD: down down, postava je čelem k uživateli. DL: Down left, postava je otočena mezi dolní a levou stranou. Obdobně pro ostatní směry. Pro každý směr jsou zde vytvořeny činnosti, pro které má hlavní postava přiřazené úkony. Jedná se o činnost také, kdy postava sbírá předmět, činnost talk, kdy postava hýbe ústy a činnost walk, kdy postava chodí. Jelikož se jedná o sekvenci obrázků, poskládaných dohromady, obrázků zde musí být velké množství. Například pro chůzi je třeba 9 snímků pro každý směr, celkem tedy 72. U postavy Greg nebylo použito takové množství snímků, bylo nutné upravit časování animací.

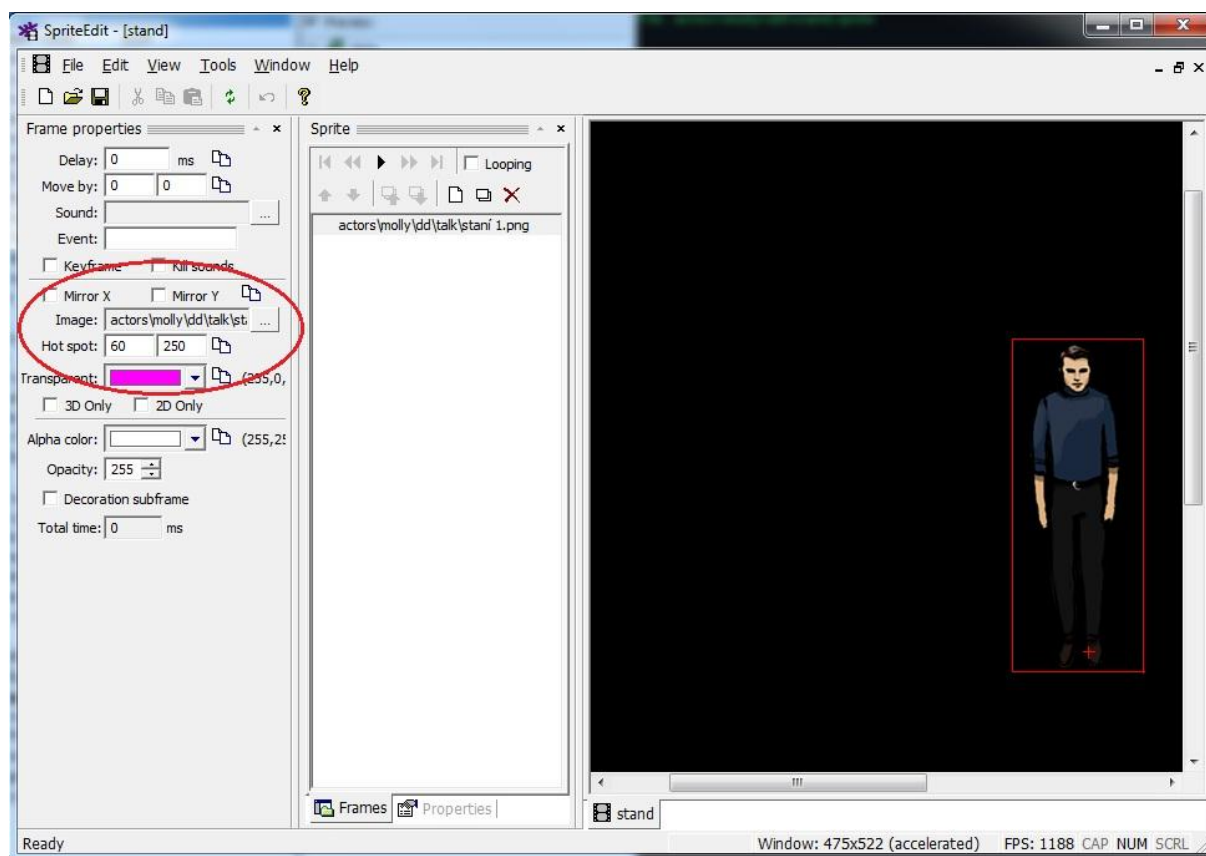
Postava Greg je vytvořena ve formátu .png, nejsou zde třeba růžové okraje, png podporuje průhlednost obrázků. Ukázalo se, že editor s tím nemá nejmenší problém. Dále je postava Greg vytvořena pouze pro 4 základní směry. Dopředu, dozadu, nahoru, dolů. Směry UR a DR (up right, down right) nejsou použity, místo nich se tedy použil pouze směr doprava. Obdobně se postupovalo v případě levé strany. Pro chůzi je použito 7 obrázků pro každý směr, 4 obrázky pro činnost talk a jeden pro činnost stand. Činnost take zde použita není. Celá postava Greg je tedy vytvořena ze 48 obrázků.

Po rozkliknutí konkrétního směru, například směru dd, viz Obr. 7.25, se objeví sprite entity, určené pro dané činnosti a adresáře s obrázky.



Obr. 7.25: Sprite entity, ovládající postavu

Prvním spritem, který se upravuje, je sprite stand, čili jednoduché stání postavy. Po dvojkliku na stand.sprite se otevře okno pro jeho tvorbu, viz Obr. 7.26.

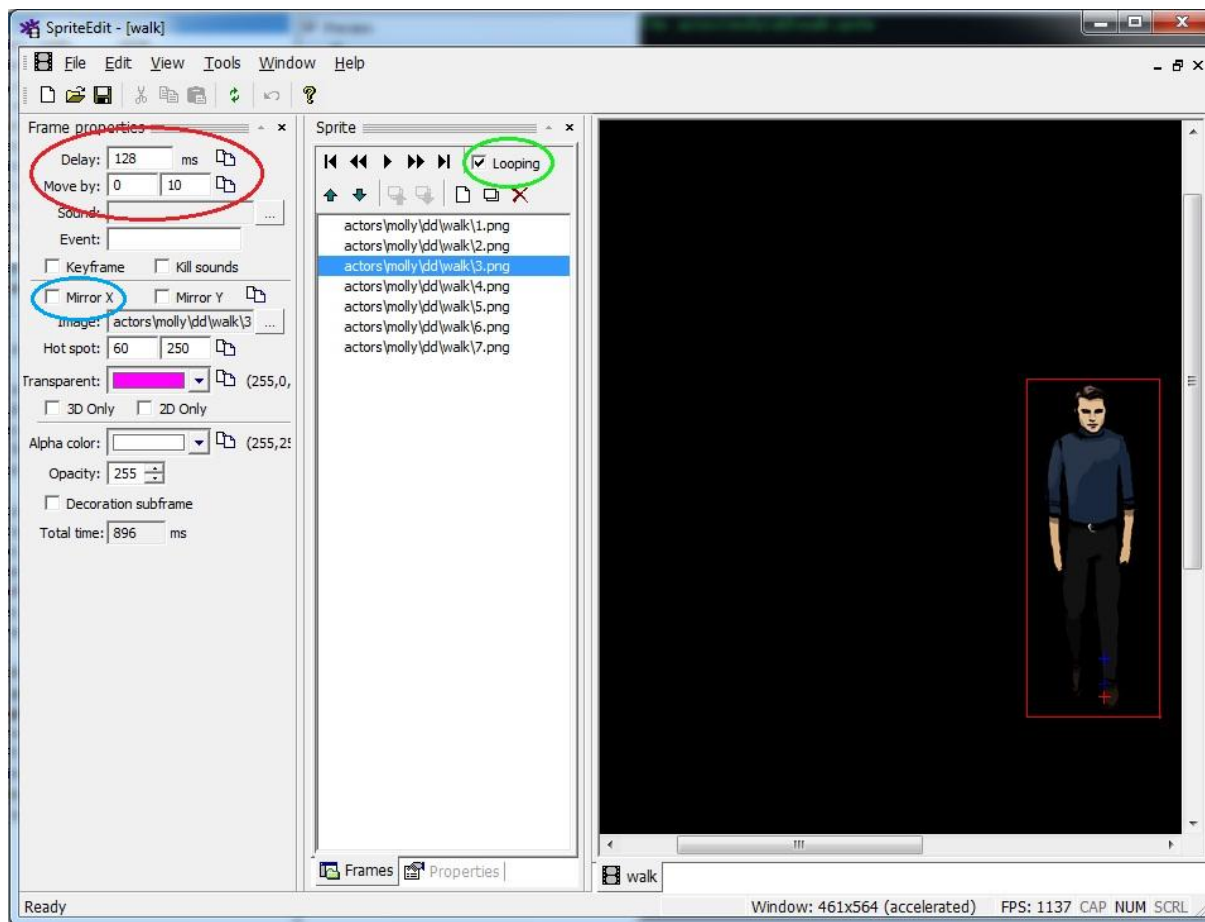


Obr. 7.26: Tvorba sprite stand

V červeně zakroužkovaném poli se nachází položka Image. Do ní se nahraje obrázek z adresáře. Obrázky je dobré vkládat do adresářů přímo tomu určených, ne z jednoho velkého. Dochází pak ke zmatkům a zbytečně to zdržuje v práci. Po nahrání obrázku je nutné upravit souřadnice bodu Hot spot. To je bod, určující polohu postavy, ať je kdekoliv na scéně. Pokud se na některé scéně zadávají souřadnice pohybu postavy, jedná se vždy o tento bod. Zde jsou nastaveny souřadnice  $x=60$ ,  $y=250$ . Pro tento případ není nutné další nastavení. Jen je třeba vymazat z tohoto okna obrázky postavy Molly a uložit. Sprite stand je kompletní. Pro jeden směr.

Složitější situace nastala při tvorbě postavy v pohybu, tedy činností talk a walk. Ilustrovat postavu v pohybu je velmi náročné. Proto například v drtivé většině komiksů má superhrdina plášť, který velkou část těla zakrývá. Chůze je sprite, který je víceméně animací několika obrázků řazených sekvenčně za sebe. Editace sprite walk, viz Obr. 7.27. Import obrázků je proveden stejně jako v předchozím případě, včetně souřadnice bodu Hot Spot. Protože zde se však jedná o animaci, je nutné nastavit čas, jak dlouho budou jednotlivé obrázky vidět. To se provede v kolonce Delay, na obrázku červeně zakroužkováno, v tomto případě tedy 128ms. Aby se vše nemuselo zadávat zvlášť, vpravo od konkrétní číselné hodnoty je ikonka dvou listů, kde po kliknutí je daná vlastnost nastavena pro všechny obrázky v sekvenci. Příkaz „move by“ je číselná hodnota pohybu postavy mezi jednotlivými obrázky. Pro pohyb dd, tedy čelem k uživateli se postava pohybuje pouze po ypsilonové dráze a o 10 na každém snímku. Toto je rovněž nastaveno pro všechny snímky v dané sekvenci. Pokud se „move by“ nastaví moc velký, postava bude bruslit po scéně. Pokud malý, postava bude bruslit na místě.

Nastavení posunu bylo provedeno metodou pokus omyl. Posledním prvkem, který je třeba nastavit je Looping, na obrázku zakroužkováno zeleně. Když je looping aktivní, sekvence se bude promítat pořád dokola. Pro směry do stran, tedy levá a pravá a směry mezi, byly použity pouze obrázky pro jednu stranu. Když se tvořila druhá strana, stačilo pouze v okně zaškrtnout Mirror X a změnit souřadnice posunu na mínusové. Na obrázku zakroužkováno modře.



Obr. 7.27: Tvorba sprite walk

Metoda talk je vytvořena stejným způsobem. Použity jsou celkem 4 obrázky, které jsou naskládány do sekvence v pořadí: 1,2,3,4,3,2. Delay u jednotlivých obrázků je 140ms a posun Move by se zde nenastavuje.

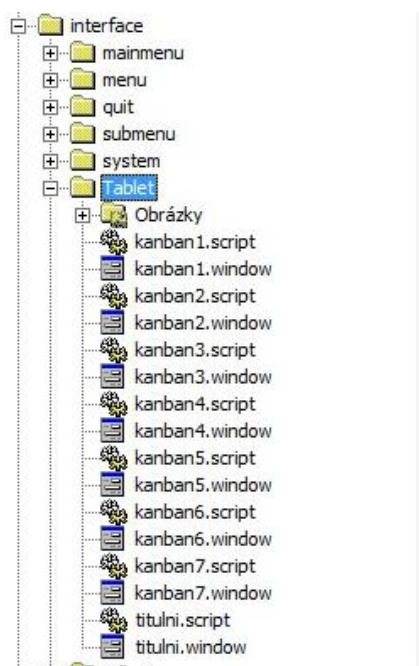
Po vytvoření těchto spritů do ostatních směrů je postava vytvořena kompletně. Největší „výzvou“ zde byly tedy samotné ilustrace. Rozpočítání postavy je po seznámení se se systémem dílem několika desítek minut.

## 7.10 Tvorba informačního tabletu

Zdaleka největší překážkou při tvorbě celého tohoto projektu, byla tvorba informačního tabletu. Jedná se totiž o prvek, který v samotném editoru není, a tedy nebylo na čem stavět. Původním záměrem bylo, aby se po stisku určité klávesy otevřelo okno s tabletem, ve kterém by uživatel mohl scrollovat v jakýchsi interaktivních skriptech. Po stisknutí klávesy by se mělo vše vrátit do původní scény. Problémů zde vystavalo hned několik.

Prvním zásadním problémem byl formát okna. V ideálním případě se mělo jednat o PowerPoint prezentaci, kam lze pohodlně vložit text a kdykoliv jej upravit. Tento formát však do editoru vložit nelze, vše se tedy muselo vytvořit mnohem komplikovanějším způsobem. Celé okno tabletu je vytvořeno jako ilustrace. Při zkoumání možností, se jako nejschůdnější varianta jevila, vytvořit tablet jako novou scénu, kde by byla nahrána jednotlivá okna tabletu. Tento způsob však nešel využít, protože do jiné scény se lze dostat pouze příkazem Game.ChangeScene. Pokud by tedy tablet byl jako samostatná scéna, nešlo by scénu zavřít, pouze se přepnout do scény jiné. Tablet je však nutné otevřít z jakékoliv scény a po jeho zavření, se do této scény opět vrátit. Zde bylo jasné, že tablet musí být vytvořen jako okno.

Okno lze kdykoliv zavřít příkazem close a tím dojde k návratu do původní scény. Nyní bylo potřeba vyřešit samotný problém ve formátu textu. Po vyzkoušení několika variant, byla zvolena alternativa nejméně elegantní. V MS PowerPoint byla vytvořena prezentace a po jednotlivých listech byla vložena do oken. Vše bylo provedeno v programu Malování. Konkrétní list prezentace se přenesl přes PrintScreen do programu Malování, tam se upravil na velikost okna tabletu a uložil pod číslem strany. Tedy Vznikla strana titulní, kde uživatel vybere logistickou metodu a ke každé z nich pak několik stran textu i s obrázky. Nyní už zbývalo vše vložit do editoru a celé oživit.



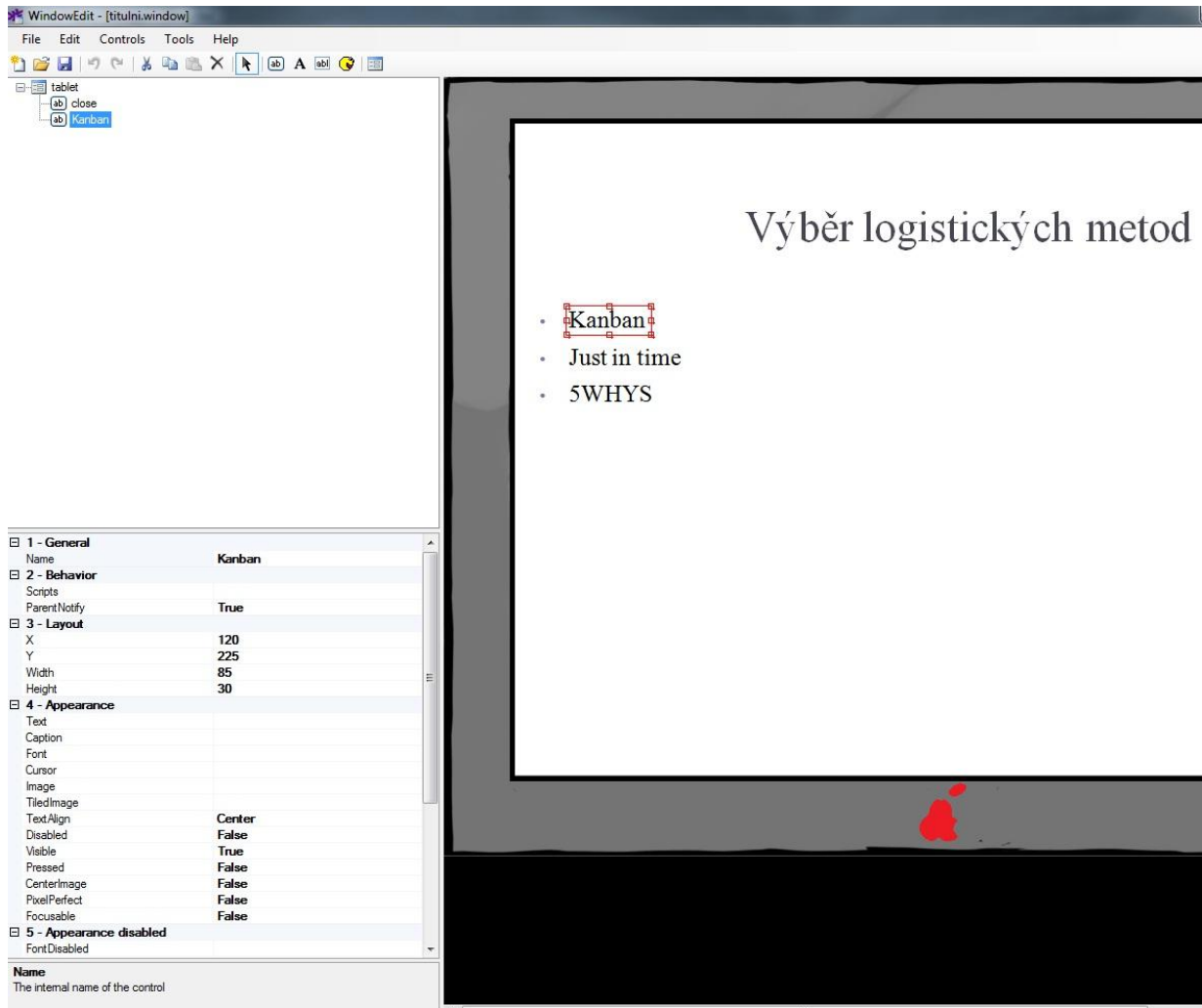
Obr. 7.28: Adresář okna Tablet

Ve složce interface byla nejprve vytvořena složka Tablet. Ve složce tablet pak další složka Obrázky, do které byly nahrány všechny obrázky. Poté bylo vytvořeno okno pro každou stranu a k němu přiřazen skript.

Na Obr. 7.29 je samotný editor oken. V něm je otevřeno okno tabletu s názvem Titulní. Toto okno se otevře, když uživatel stiskne klávesu F1 a zde si také vybírá logistické metody, o kterých se chce dozvědět potřebné informace.

Do okna byl nahrán obrázek okna titulní a jsou zde použita tlačítka (button). Jedno tlačítko je na nakousnuté hrušce, která funguje jako tlačítko pro zavření okna. Zavření okna opět klávesou F1 bylo mnohem náročnější. Musí se totiž vytvořit skript, který zjišťuje, zda je okno otevřeno. Pokud otevřeno je, příkaz F1 znamená zavřít a pokud otevřeno není, příkaz F1

znamená otevřít. Tato procedura je však velmi náročná, proto je zde použito pouze zavírání tlačítkem na nakousnuté hrušce. Ta po najetí kurzorem na ni zčervená. K oběma tlačítkům je přiřazen skript.



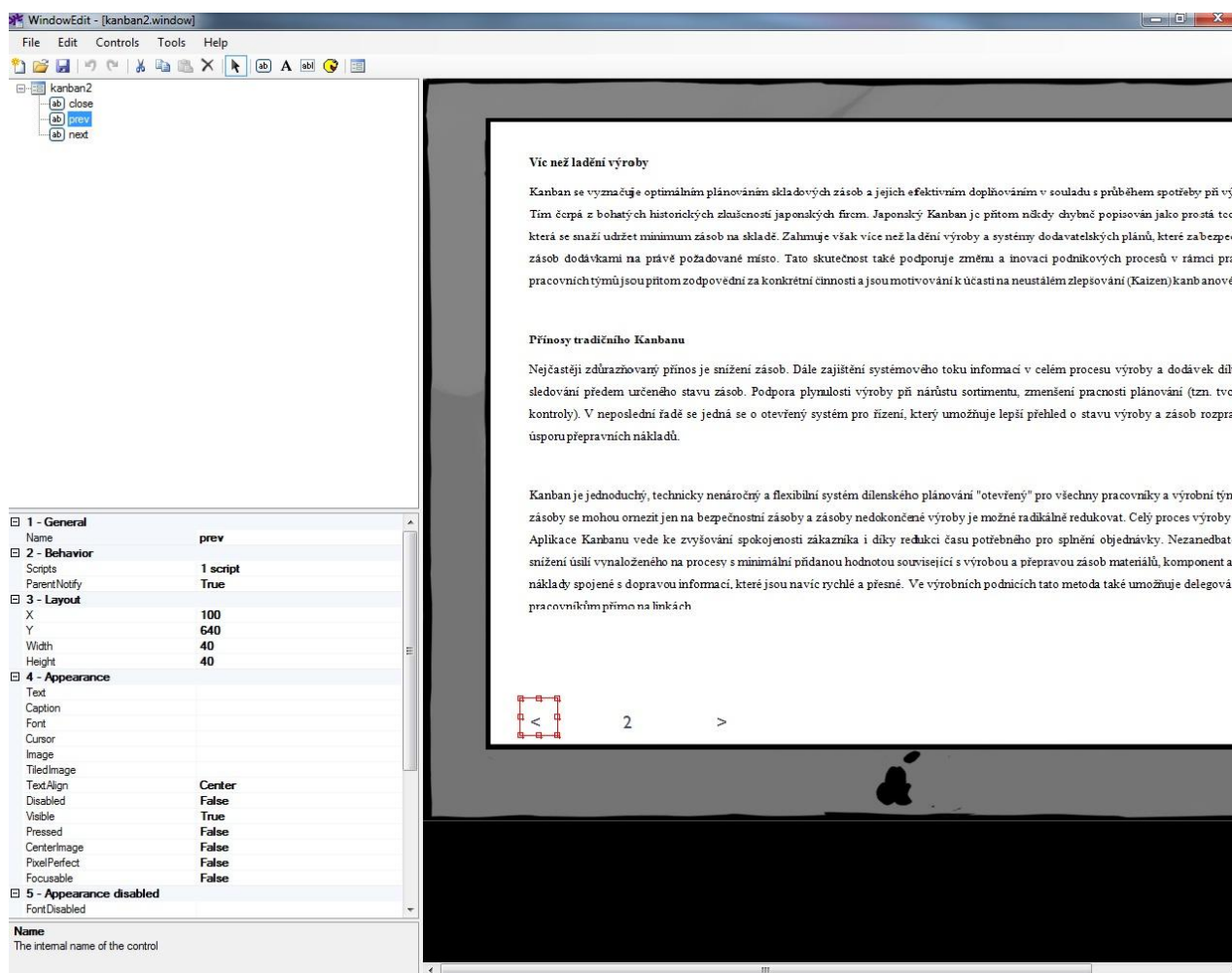
Obr. 7.29: Editace titulního okna tabletu

```
1 titulni.script *
1  #include "scripts\base.inc"
2  #include "scripts\keys.inc"
3
4
5  //////////////////////////////////////
6  on "close"
7  {
8    this.Close();
9  }
10
11 on "Kanban"
12 {
13   // Přepnutí okna na okno Kanban 1
14
15   var k1 = Game.LoadWindow("interface\Tablet\kanban1.window");
16   k1.Center();
17   k1.GoSystemExclusive();
18   this.Close();
19   Game.UnloadObject(k1);
20
21 }
```

Obr. 7.30: Skript pro titulní okno

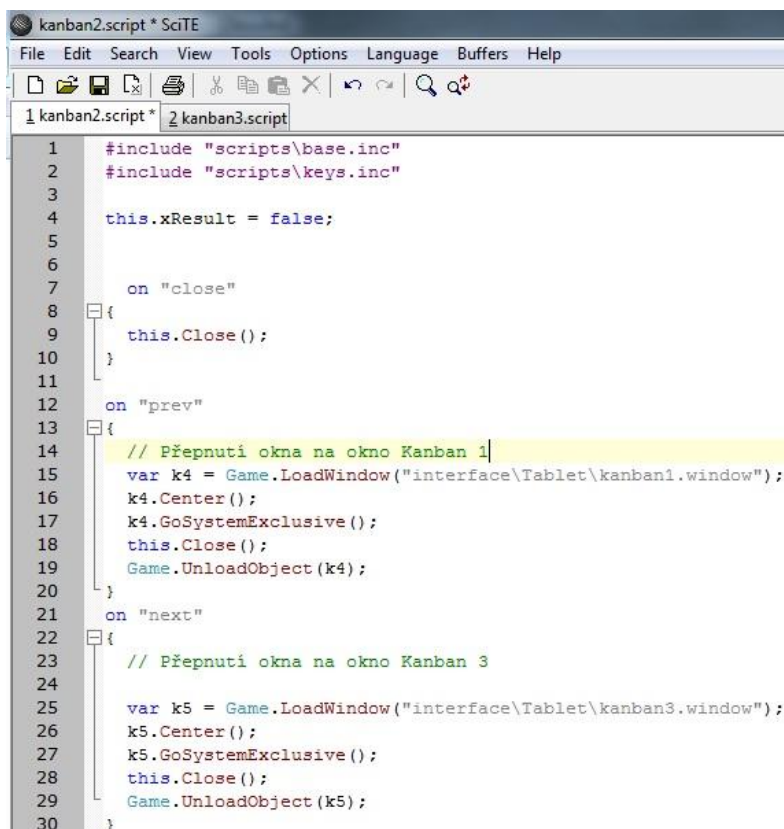
Skript pro titulní okno obsahuje příkazy pro ovládání okna close a příkaz pro přepnutí do konkrétní logistické metody. Skript, viz Obr. 7.30. Na řádcích 6 až 9 je příkaz, který po kliknutí na nakousnutou hrušku zavře okno s tabletem. Na řádcích 11 až 20 je skript pro přepnutí na okno Kanban 1. Na řádce 15 je nutné nejprve vytvořit proměnnou, pod kterou se bude nové okno otevírat. Proměnná je různá pro každé okno tabletu, je tedy potřeba pokaždé vytvořit novou proměnnou s novým názvem. Na řádcích 16 a 17 dojde k vycentrování okna a přesunutí okna do popředí. Na řádce 18 je příkaz pro zavření okna titulní. Vyřešení tohoto problému byl snad největší problém při tvorbě celého programu. Samotné přepnutí okna na jiné není problém, děje se příkazem `Game.LoadWindow`. Problém nastal při pokusu zavřít tablet kliknutím na tlačítko close, tedy na nakousnutou hrušku. Okno se sice zavřelo, ale místo něj se otevřelo okno předchozí. Při pokusu zavřít další okno se otevřelo opět okno předchozí. Pokud byla otevřena jen 2 okna, problém to není. Problém nastal, pokud uživatel mezi okny přepíná, tak by musel zavřít po sobě třeba 20 oken a to je nemyslitelné. Po vyzkoušení mnoha možností, se nakonec našlo řešení, a to v podobě vymazání předchozího okna z paměti. To se děje příkazem na řádce 19, příkazem `Game.UnloadObject`. Tento příkaz po zavření okna vymaže okno z paměti a při zavření se již toto okno neotevře. Celý proces nefunguje na 100%, při zavření je vidět, jak se postupně zavírají všechna okna, svůj účel to ale splnilo. Přepínání a zavírání oken je tedy tímto vyřešeno.

Po kliknutí na metodu se otevře okno k dané metodě, kde lze šipkami přepínat mezi jednotlivými okny. Jedno z oken s informacemi o metodě Kanban, viz Obr. 7.31. Okno je tedy stejné jako okno titulní, liší se pouze textem uvnitř a aktivními tlačítky. Tlačítko Close zde zůstává, je na všech oknech. Na pravé i levé straně od čísla stránky jsou šipky, na kterých jsou vytvořena tlačítka prev a next. Ta přepínají uživatele mezi okny doprava a doleva. Pouze u posledního snímku z řady není použita šipka next.



Obr. 7.31. Okno Kanban 2

Skript pro ovládání okna Kanban 2, viz Obr. 7.32. Příkaz close na řádcích 7 až 10 je totožný s předchozím případem. Příkaz prev na řádcích 12 až 20 přepíná na předchozí okno. Opět je zde zavedena nová proměnná K4, do které se nahraje nové okno a to se pak zavře a vymaže z paměti. Identický postup je pro tlačítko next, pouze s jinou proměnnou a s přepínáním na jiné okno. Postup se opakuje i pro ostatní okna. Tím je vytvořen informační tablet pro uživatele.



```
1 #include "scripts\base.inc"
2 #include "scripts\keys.inc"
3
4 this.xResult = false;
5
6
7 on "close"
8 {
9   this.Close();
10 }
11
12 on "prev"
13 {
14   // Přepnutí okna na okno Kanban 1
15   var k4 = Game.LoadWindow("interface\Tablet\kanban1.window");
16   k4.Center();
17   k4.GoSystemExclusive();
18   this.Close();
19   Game.UnloadObject(k4);
20 }
21 on "next"
22 {
23   // Přepnutí okna na okno Kanban 3
24
25   var k5 = Game.LoadWindow("interface\Tablet\kanban3.window");
26   k5.Center();
27   k5.GoSystemExclusive();
28   this.Close();
29   Game.UnloadObject(k5);
30 }
```

Obr. 7.32: Skript k oknu Kanban 2

## 7.11 Sebrání předmětu ze scény

Protože je celý program postaven na bázi hry, známé jako adventura, a protože kromě edukativního obsahu je nutné zahrnout i obsah zábavný, jsou v programu zahrnuty interakce s předměty a jejich používání. Nejprve je třeba sebrat určitý předmět ze scény. To je vysvětleno v této kapitole na příkladu, kdy hlavní postava sebere z přepravky kanbanovou kartu po rozhovoru s dělníkem. Karta je následně uložena do inventáře.

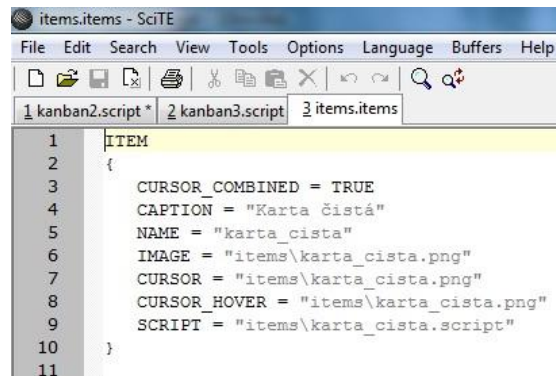
K tomu, aby šlo předmět použít v programu, je třeba mít obrázek tohoto předmětu. V tomto případě jde o ilustraci s rozměry 62x62 obrazových bodů, tedy stejný rozměr jako okno inventáře. Obrázek i se skriptem je uložen ve složce items, viz Obr. 7.33. Obrázek je uložen opět v rozměru, podporovaném programem. Tedy .jpg, .bmp, .png. Okraje, které se nemají zobrazovat jsou opět růžové, nebo v případě formátu .png průhledné.

Další položkou, kterou je třeba upravit, je položka items.items, opět ve složce items, viz Obr. 7.34. Zde je třeba nadefinovat parametry předmětu. Nejdříve CAPTION, tak se bude předmět nazývat v programu po najetí kurzorem. Dále NAME, bez háčků a čárek, tak se bude předmět nazývat při skriptování. Za IMAGE se přiřadí obrázek předmětu, CURSOR a CURSOR\_HOVER se nechává stejný jako původní obrázek a jako poslední je nutno přiřadit skript.





Obr. 7.33: Složka items



Obr. 7.34: Definování entity karta

Samotný předmět se nachází na jedné ze scén. Může se sebrat při rozhovoru, kdy jej uživateli někdo předá, může volně ležet na scéně a může se také objevit například po prozkoumání některého předmětu. To je přesně tento případ. Předmět karta čistá se bude nacházet v přepravce na scéně Dílna. Po dokončení rozhovoru s dělníkem a prozkoumání přepravky dojde k sebrání karty a uložení do inventáře.

Celý skript, ovládající tuto akci je skript paleta\_2.script. Z palety 2 hlavní postava sebere kartu. Skript, viz Obr. 7.35.

```
1 paleta_2.script
1  #include "scripts\base.inc"
2
3
4  on "LeftClick"
5
6  if (var_004==true)
7  {
8      //Postava dojde k předmětu
9      actor.GoTo(758,570);
10     actor.TurnTo(DI_UP);
11
12     Game.Interactive=false;
13     dialog.Visible=true;
14     actor.Talk("Vezmeme kanbanovou kartu z přepravky a dáme jí na tabuli.",null,3000);
15     var kartal=Scene.GetNode("karta_cista");
16     kartal.Visible=false;
17     kartal.Active=false;
18     dialog.Visible=false;
19
20     //sebrání předmětu
21     Game.TakeItem("karta_cista");
22
23     Game.Interactive=true;
24
25     //spusit hru
26     Game.Interactive=true;
27 }
28 else
29 {
30     //Postava mluví
31     Game.Interactive = false;
32     dialog.Visible = true;
33     actor.Talk("Tady není nic, co by se mi hodilo.");
34     Game.Interactive = true;
35     dialog.Visible = false;
36 }
37
```

Obr. 7.35: Skript paleta\_2

Na řádce 6 je vytvořena podmínka, která hlídá, zda uživatel už mluvil se skladníkem. Pokud ano, hodnota proměnné je true, lze sebrat z přepravky kanbanovou kartu. Pokud k rozhovoru zatím nedošlo, Greg pouze pronese větu, že v přepravce není nic, co by se hodilo. Pokud je podmínka splněna, po levém kliknutí na přepravku, postava dojde k přepravce a otočí se směrem nahoru. Viz řádky 9 a 10. Poté dojde k zastavení interaktivity programu, objeví se dialogové okno, viz řádky 12 a 13. Postava dále pronese větu, viz řádek 14. Na řádce 15 je samotný skript, ovládající sebrání předmětu ze scény. Poté na řádcích 16 a 17 dojde k zneviditelnění karty a ke zrušení její aktivity na scéně. Na řádce 21 je procedura, která kartu vytvoří v inventáři.

Po levém kliknutí tedy karta zmizí ze scény a objeví se v inventáři. Pokud by karta volně ležela na scéně, je třeba celou tuto proceduru vytvořit ve skriptu ke kartě. Tímto je popsáno, jak vytvořit proceduru k sebrání předmětu ze scény.

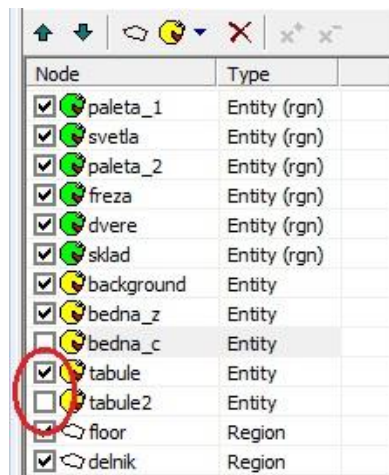
## 7.12 Změna entity při akci

Pokud je potřeba různě kombinovat předměty, musí dojít ke změně entity při nějaké akci. V tomto programu bude tato změna entity vysvětlena na příkladu kanbanové tabule. Tabule se nachází v lokaci dílna. Tabule je přidána do lokace zvlášť, jako sprite entita. Do ní je pak nahrán obrázek tabule a vytvořen skript, který ovládá chování této entity. Po sebrání karty

z palety a přemístění karty do inventáře, má uživatel za úkol umístit kartu na kanbanovou tabuli. Po umístění karty na tabuli, tabule změní obrázek (objeví se na ní karta) a změní se i přepravky. Na pravé straně jedna přibude a na levé straně jedna zmizí.

Nejprve je nutné na scénu přidat entity tabule a tabule2, která se objeví po přidání karty na tabuli. Přidání entity na scénu bylo popsáno v kapitole 7.4

Tvorba scén. Zaškrtnutí aktivní a neaktivní entity, viz Obr. 7.36. Nebude tedy zaškrtnuta (políčko vedle ikonky entity bude prázdné). Aktivní prvky naopak musí být zaškrtnuty



Obr. 7.36: Výběr aktivní a neaktivní entity

Obě tabule musí být na stejném místě, aby se po změně entity nová tabule neobjevila na druhé straně obrazovky. V seznamu entit na Obr. 7.36 dále musí být entita tabule\_2 pod entitou tabule, protože entita, která je v seznamu níže, je po spuštění programu na obrazovce na vrchu. Obě tabule, viz Obr. 7.37. Nyní už zbývá jen vytvořit skript pro tuto situaci.



Obr. 7.37: Kanbanová tabule před a po změně

```
1 tabule.script | 2 tabule2.script
14 }
15
16 on "RightClick"
17 {
18     Game.Interactive = false;
19     dialog.Visible = true;
20     actor.Talk("kanban tabule.");
21     actor.Talk("Sem se dávají kanban karty.");
22     dialog.Visible = false;
23     Game.Interactive = true;
24 }
25
26 on "karta_cista"
27 {
28     //Postava dojde k tabuli
29     actor.GoTo(362,589);
30     actor.TurnTo(DI_UP);
31     //odstraníme předmět vybraný
32     Game.SelectedItem = null;
33
34     //postava hovoří
35     Game.Interactive=false;
36     dialog.Visible=true;
37     actor.Talk("Tak. Karta patří sem na nástěnku.");
38     dialog.Visible=false;
39     Game.Interactive=true;
40
41     //výmaz předmětů z inventáře
42     Game.DropItem("karta_cista");
43     Game.DeleteItem("karta_cista");
44
45     //Přepnutí obrázku na tabuli2
46     Game.Interactive=false;
47     var tabule2=Scene.GetNode("tabule2");
48     tabule2.Active=true;
49     tabule2.Visible=true;
50     dialog.Visible=true;
51     actor.Talk("Když je kanbanová karta na svém místě, skladník nám přinese další přepravku.",null,3000);
52     dialog.Visible=false;
53
54     //Objevení se další bedny
55     Game.Interactive=false;
56     var bedna=Scene.GetNode("bedna_c");
57     bedna.Active=true;
58     bedna.Visible=true;
59     dialog.Visible=true;
60     actor.Talk("Tak a přepravka je tu.",null,3000);
61     actor.Talk("Takže takhle funguje Kanban mezi dílnou a skladem.",null,4000);
62     dialog.Visible=false;
63     Game.Interactive=true;
64 }
65
```

Obr. 7.38: Skript kanbanové tabule

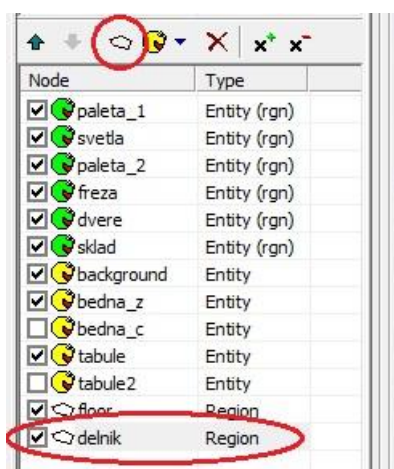
Na Obr. 7.38 je zobrazen skript, který mění entitu tabule na entitu tabule 2. Skript začíná na řádce 26, kdy se kartou klikne na tabuli. Postava nejprve dojde k tabuli, viz řádky 29 a 30, poté se karta odstraní z kurzoru, viz řádek 32. Dále postava řekne jednu větu, viz řádky 35 až 39. Skript na řádcích 42 a 43 pak kartu vymaže z inventáře. Na řádce 46 dojde k zastavení interaktivity programu a následně dojde ke zviditelnění a zaktivnění tabule 2, viz na řádcích 48 až 50. Postava poté pronese jednu větu a obnoví se interaktivita programu, viz řádky 51 a 52. Obdobně je postupováno od řádku 55 dále, kdy se po objevení tabule 2, objeví další přepravka na scéně.

Poslední věcí, kterou je třeba upravit, je přiřazení skriptu k entitě tabule 2. V předchozích procesech se totiž tabule překryla tabulí 2 a ta je nyní na vrchu. Je tedy aktivní a zakrývá původní tabuli. Je tedy vytvořen skript pro tabuli 2, takže na ni lze opět kliknout a nechat postavu, říct o předmětu několik vět.

### 7.13 Tvorba vedlejší postavy

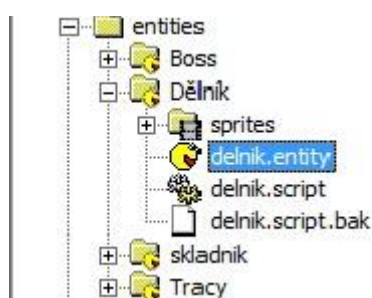
V programu se kromě hlavní postavy objevuje i několik vedlejších postav. Jako první, s kým přijde uživatel do styku, je ředitel. Ten uživateli zadává úkoly. Další je sekretářka Tracy, která s hlavní postavou sdílí jednu kancelář. Na dílně pracuje dělník, kterému bude uživatel vysvětlovat, k čemu jsou různé logistické metody a co to pro něj bude mít za přínosy a ve skladu skladník, kterého bude uživatel muset chvíli hledat. Vedlejší postavy jsou každá pouze v jedné lokaci, nepřemísťují se a pouze mluví, není u nich naskriptován pohyb.

Postava se vkládá do lokace opět jako entita typu region. Jedná se však o region typu podlaha, vložení entity, viz Obr. 7.39. Vloží se pouze prázdná entita, u které se vyplní položka NAME. To je pro zatím celé, nyní stačí jen uložit celou scénu a odejít z editoru scény.



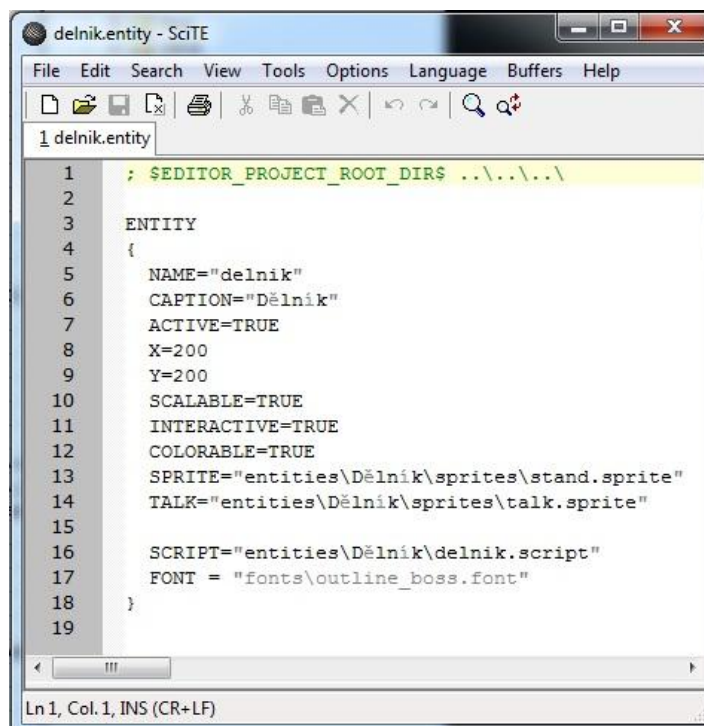
Obr. 7.39: Vkládání vedlejší postavy do scény

V adresáři entities, viz Obr. 7.40 se pravým kliknutím přidá entita a vytvoří složka sprites. Ve složce sprites se vytvoří opět sprite stand a talk, stejným způsobem jako při tvorbě hlavní postavy. Sprite stand je tvořen opět jedním obrázkem, sprite talk sekvencí několika po sobě jdoucích obrázků.



Obr. 7.40: Adresář s vedlejšími postavami

Kliknutím na položku delnik.entity se otevře okno, viz Obr. 7.41. Zde se upraví jméno a caption, dále se přiřadí sprite pro metodu stand a talk, skript a na posledním řádku font. Font boss\_font je pojmenován podle ředitele, u kterého se začal používat jako první a poté bylo rozhodnuto, že tento font bude použit pro všechny vedlejší postavy.



```
delnik.entity - SciTE
File Edit Search View Tools Options Language Buffers Help
1 ; $EDITOR_PROJECT_ROOT_DIRS ..\..\..\
2
3 ENTITY
4 {
5     NAME="delnik"
6     CAPTION="Dělník"
7     ACTIVE=TRUE
8     X=200
9     Y=200
10    SCALABLE=TRUE
11    INTERACTIVE=TRUE
12    COLORABLE=TRUE
13    SPRITE="entities\Dělník\sprites\stand.sprite"
14    TALK="entities\Dělník\sprites\talk.sprite"
15
16    SCRIPT="entities\Dělník\delnik.script"
17    FONT = "fonts\outline_boss.font"
18 }
19
Ln1, Col.1, INS (CR+LF)
```

Obr. 7.41: Úprava entity dělník

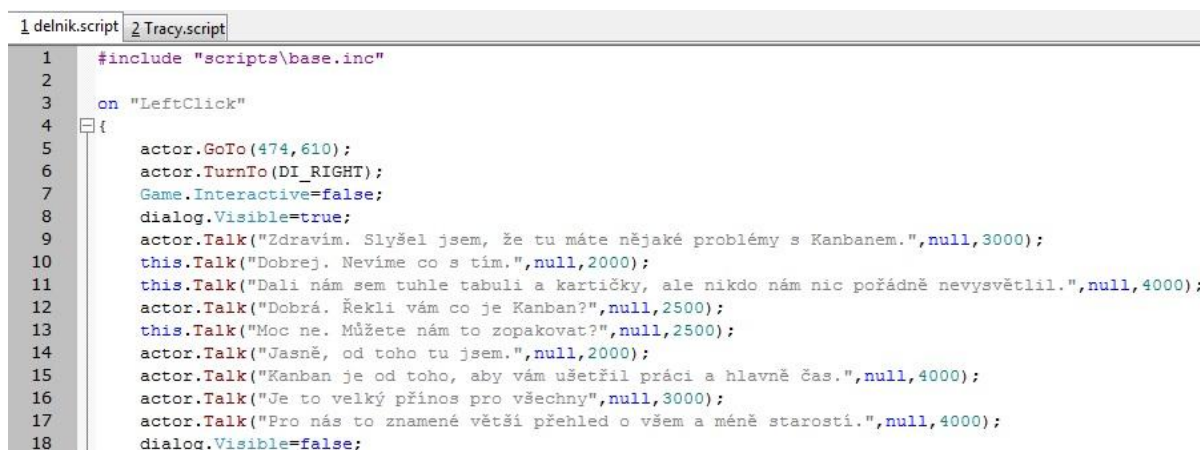
Posledním krokem k oživení vedlejší postavy je přiřazení pozice titulků do skriptu. Skript je popsán v kapitole 7.7 Tvorba vlastní metody Talk.

Tímto je dokončena tvorba vedlejší postavy. Sprite pro mluvení vedlejší postavy je připraven, v další kapitole je podrobně popsáno, jak tvořit dialogy.

## 7.14 Dialogy s vedlejšími postavami

Wintermute editor umožňuje použití velkého množství variant dialogů. To zahrnuje rozhovory dvou postav, dále rozhovory podmíněné nějakou podmínkou a rozhovory s několika možnostmi odpovědi. Všechny tyto možnosti jsou v programu využity.

Prvním případem je obyčejný dialog dvou postav. Ten je vytvořen v souboru skriptu vedlejší postavy. Na Obr. 7.42 je vidět skript pro tento jednoduchý dialog.



```
1 delnik.script 2 Tracy.script
1 #include "scripts\base.inc"
2
3 on "LeftClick"
4 {
5     actor.GoTo(474, 610);
6     actor.TurnTo(DI_RIGHT);
7     Game.Interactive=false;
8     dialog.Visible=true;
9     actor.Talk("Zdravím. Slyšel jsem, že tu máte nějaké problémy s Kanbanem.",null,3000);
10    this.Talk("Dobřej. Nevíme co s tím.",null,2000);
11    this.Talk("Dali nám sem tuhle tabuli a kartičky, ale nikdo nám nic pořádně nevysvětlil.",null,4000);
12    actor.Talk("Dobrá. Řekli vám co je Kanban?",null,2500);
13    this.Talk("Moc ne. Můžete nám to zopakovat?",null,2500);
14    actor.Talk("Jasně, od toho tu jsem.",null,2000);
15    actor.Talk("Kanban je od toho, aby vám ušetřil práci a hlavně čas.",null,4000);
16    actor.Talk("Je to velký přínos pro všechny",null,3000);
17    actor.Talk("Pro nás to znamená větší přehled o všem a méně starostí.",null,4000);
18    dialog.Visible=false;
```

Obr. 7.42: Část skriptu rozhovoru s dělníkem

Po levém kliknutí postava dojde k dělníkovi a otočí se na něj, viz řádky 5 a 6. Poté dojde k zastavení interaktivity hry a zobrazení dialogového okna. Na řádku 9 začíná rozhovor hlavní postava (příkaz: actor.Talk). v závorce rozhovoru je dále položka null a za ní číslice. Ta udává čas v milisekundách, po který je tato konkrétní věta zobrazena. Zejména u delších vět je toto nutné upravit, popřípadě zobrazit další větu rozhovoru po kliknutí. Na řádku 10 se do rozhovoru zapojuje i dělník. Jelikož se jedná o skript pro dělníka, je zde použito příkazu: this.talk, tedy: tato entita bude vykonávat činnost talk. Pro činnost talk je již zaveden sprite mluvicí postavy, viz předchozí kapitola. Po skončení tohoto rozhovoru je nutné opět vypnout dialogové okno, viz řádek 18.

Složitější situace nastává, když je nutné provést rozhovor s několika možnostmi odpovědí. V tomto programu je těchto rozhovorů několik, většinou se jedná o 3 možnosti odpovědí, z nichž jen jedna je správná. Nejprve je nutné si všechny možnosti, otázky a odpovědi připravit ve scénáři. I tak je to však časově náročná činnost.

Na Obr. 7.43 je vidět další část rozhovoru s dělníkem. Jedná se o část, kde se rozhovor větví a jsou tedy na výběr 4 možnosti. Dělník se ptá Grega, co je to vlastně Kanban. Na uživateli je, aby vybral správnou odpověď. Po vybrání správné odpovědi postavy řeknou několik vět a rozhovor pokračuje dále. Při špatné odpovědi řeknou postavy také několik vět, ale vše se vrací zpátky k těmto možnostem.

Na obrázku je tedy dialog, který je ve scénáři označen jako dialog 1. Na řádku 21, je nadeklarována proměnná options. Do této proměnné se nahrávají možnosti, viz řádky 22 až 25. Možnosti jsou číslovány od nuly dále. Tyto věty jsou vidět na obrazovce a na jednu z těchto vět uživatel klikne a tím ji vybere. Na řádku 26 je nadeklarována druhá proměnná selected. Do této proměnné se nahrává odpověď. Nyní k samotným podmínkám rozhovoru. Na řádku 27 je podmínka while. Ta udává, že pokud uživatel nezvolí první možnost, zde tedy možnost 0, rozhovor se bude dále opakovat. Na řádcích 29 až 32 je procedura, která možnosti nahrává na obrazovku. Příkaz Game.AddResponse je příkazem pro kliknutí na odpověď, přičemž pokud uživatel zvolí špatnou odpověď, tato odpověď se v rozhovoru objeví znovu. Pokud by zde byla podmínka: Game.AddResponseOnceGame, tak je tato odpověď zobrazena pouze jednou. Tedy pokud uživatel zvolí ze čtyř odpovědí jednu špatnou, podruhé má na výběr o tuto možnost méně, tedy jen 3. Příkaz na řádku 34 složí k nahrání odpovědi do proměnné select. Pro tyto možnosti je vypnuto dialogové okno. Pokud se totiž jedná o 3 a více možností, již se nad sebe do dialogového okna nevejdou. Na řádku 37 dochází ke spuštění dialogového okna, od této chvíle již probíhá normální rozhovor. Na řádku 38 je příkaz, na který hlavní postava řekne vybranou větu. Poté dochází k přepnutí na tuto možnost, viz řádek 39.

Od řádku 40 dochází k použití jedné z možností. Na řádku 41 začíná možnost 0, tedy v tomto případě se jedná o správnou odpověď. Po dialogu následuje opět zavření dialogového okna, viz řádek 46 a poté příkaz break na řádku 47. Ten zastavuje další postup rozvětveného dialogu. Pokud by příkaz break nebyl použit, dojde k odříkání všech 4 možností. Příkaz break tedy zastaví další dialog, který je určen pro jiné možnosti. Stejný postup je proveden i pro ostatní možnosti a ostatní dialogy.

```
1 delnik.script | 2 Tracy.script
19 {
20 //Scénář, dialog 1
21 var options;
22 options[0] = "Je to metoda k řízení skladů, ale i například administrativy.";
23 options[1] = "Je to metoda pro odstranění úzkých míst ve výrobě.";
24 options[2] = "Je to metoda, podle které zjistíme, která pracoviště jsou vytižena a která ne.";
25 options[3] = "Je to metoda k odstranění zmetků ve výrobě.";
26 var selected;
27 while (selected !=0) //Rozhovor pokračuje do doby, dokud uživatel neklikne na první možnost (možnost 0)
28 {
29     Game.AddResponse(0,options[0]);
30     Game.AddResponse(1,options [1]);
31     Game.AddResponse(2,options[2]);
32     Game.AddResponse(3,options[3]);
33     Game.Interactive=true;
34     selected=Game.GetResponse();
35     Game.Interactive=false;
36
37     dialog.Visible = true;
38     actor.Talk(options[selected]);
39     switch(selected)
40     {
41     case 0:
42         this.Talk("Jo, to už jsme slyšeli. A můžete nám to říct česky, abychom to my pochopili?",null,3000);
43         actor.Talk("Jak jsem říkal, od toho tu jsem.",null,2000);
44         actor.Talk("Jenom vám to vysvětlím a bude to všechno v pořádku. ",null,3000);
45         actor.Talk("Kartičky jsou tam kvůli tomu, aby skladník poznal, kdy vám má dodat další materiál, a tudíž
46         dialog.Visible=false;
47         break;
48     case 1:
49         this.Talk("A k čemu ty kartičky?",null,3000);
50         actor.Talk("Asi jsem řekl pěknou hloupost, co?",null,3000);
51         dialog.Visible=false;
52         break;
53     case 2:
54         this.Talk("A na ty kartičky si budeme psát co?",null,3000);
55         actor.Talk("Tohle asi nebude správně. Gregu, měl bys to zkusit znovu.",null,4000);
56         dialog.Visible=false;
57         break;
58     case 3:
59         this.Talk("A k tomu nám pomůžou kartičky? Zmetky do nich schováme, aby nebyly vidět?",null,3000);
60         actor.Talk("Máte pravdu, tohle asi nebude správná odpověď.",null,2000);
61         dialog.Visible=false;
62         break;
63     }
64 }
65 }
66 dialog.Visible=true;
67 this.Talk("A co s těmi kartami? ",null,2000);
68 actor.Talk("Správná otázka.",null,2000);
69 dialog.Visible=false;
70 {
```

Obr. 7.43: Druhá část rozhovoru s dělníkem

Posledním případem dialogu, je dialog podmíněný. V programu jsou dialogy, které se dají absolvovat až po provedení určitých činností. Například ve skriptu pro dělníka jsou rozhovory pro oba pracovní dny. To se odděluje proměnnou. Pokud je proměnná hodnoty například false, proběhne rozhovor pro první den. Na konci prvního dne se proměnná přepne na true a při dalším rozhovoru s dělníkem se jedná už o dialog pro den druhý.

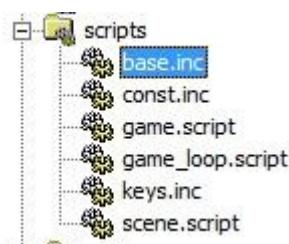
Dialogy jsou velmi obsáhlé, co se týče počtu řádků. Procedura sama o sobě patří mezi složitější a zejména při tvorbě dialogů poprvé se vždy vyskytne mnoho chyb, které se musí postupně odstraňovat. Jakmile je toto provedeno, vše funguje výborně. V další kapitole je popsána tvorba proměnných, které mohou řídit například dialogy.



## 7.15 Tvorba proměnných

Program bylo nutno napsat za použití tzv. řídicích proměnných, o kterých bylo zmíněno v předchozích kapitolách. Tvorba a použití těchto proměnných je popsáno níže.

Celý program je proveden tak, aby uživatele sám vedl. Je zde tedy použita proměnná global var\_001, která je nadeklarována ve skriptu base.inc, který je umístěn ve složce scripts, viz Obr. 7.44.



Obr. 7.44: Umístění skriptu base.inc

Tato proměnná nemá zatím přiřazenou hodnotu. Ve skriptu kancl\_sef.script je příkaz: Pokud má proměnná var\_001 hodnotu null, změnit hodnotu na false. Proměnná má tedy pro zatím hodnotu false. V lokaci chodba je tato proměnná využita hned několikrát. Ve skriptu hala.script, který uživatele přesouvá z chodby do dílny, je příkaz: Pokud má proměnná var\_001 hodnotu false, postava pronese větu: Musíme za šéfem pro instrukce. Stejný příkaz je u skriptu kancl.script, který uživatele přesouvá do kanceláře sekretářky. Tedy pokud má proměnná var\_001 hodnotu null, je v lokaci chodba změněna na false. Pokud má proměnná hodnotu false, program uživatele nepustí do kanceláře sekretářky a do dílny. Po rozhovoru s ředitelem se na konci rozhovoru hodnota této proměnné změní na true, program uživatele pustí do dílny a do kanceláře sekretářky, do kanceláře ředitele už ale ne. Postava pronese větu: S ředitelem jsme už mluvili. Deklarace řídicích proměnných, viz Obr. 7.45.

```
1 #include "scripts\const.inc"
2
3 global Scene;
4 global Keyboard;
5 global actor;
6 global dialog;
7
8 global var_001; //podminka, zda jsme už mluvili se šéfem
9 global var_002; //Při rozhovoru se šéfem jsme nevěděli, co je Kanban. Po rozhovoru s Tracy můžeme jít do dílny
10 global var_003; //Mluvili jsme s Tracy. Podruhé postava řekne, že už ví všechno
11 global var_004; //Mluvili jsme s dělníkem
12 global var_005; //Karta se objevila na nástěnce. Změna obrázku nástěnky
```

Obr. 7.45: Deklarace řídicích proměnných

Další proměnnou, využitou zde, je proměnná var\_002. Ta je použita při rozhovoru se šéfem. Při rozhovoru se šéf zeptá Grega na Kanban. Pokud uživatel odpoví správně, tedy něco o Kanbanu ví, proměnná var\_002 použita není. Pokud však odpoví špatně, tedy o Kanbanu není dostatečně informován, proměnná změní hodnotu na true. Viz část skriptu rozhovoru s ředitelem, na Obr. 7.46. Zde v tomto případě je správná možnost 2, čili možnost poslední. V možnosti 0 a 1 je na řádcích 37 u možnosti 0 a 45 u možnosti 1 změna proměnné var\_002 na true. Pokud uživatel odpoví špatně, bude poslán do kanceláře sekretářky, která mu s metodikou poradí. Skripty jsou tedy nastaveny tak, že po odchodu z kanceláře ředitele na chodbu proměnná pustí uživatele pouze do kanceláře sekretářky. Tam se po rozhovoru objeví proměnná var\_003, která již uživatele pustí do dílny.

```
25     selected=Game.GetResponse();
26     Game.Interactive=false;
27
28     actor.Talk(options[selected]);
29     dialog.Visible = true;
30     switch(selected)
31     {
32     case 0:
33         this.Talk("Ty nevíš, co je Kanban?");
34         this.Talk("Skoč si za Tracy, ta ti k tomu dá nějaké podklady.");
35         this.Talk("Nastuduj si je a skoč za chlapama na dílnu jim to vysvětlit.");
36         actor.Talk("Dobrá. Zatím nashledanou.");
37         var_002= false;
38         dialog.Visible=false;
39         break;
40     case 1:
41         this.Talk("Ty nevíš co je Kanban?");
42         this.Talk("Skoč si za Tracy, ta ti k tomu dá nějaké podklady.");
43         this.Talk("Nastuduj si je a skoč za chlapama na dílnu jim to vysvětlit.");
44         actor.Talk("Dobrá. Zatím nashledanou.");
45         var_002= false;
46         dialog.Visible=false;
47         break;
48     case 2:
49         this.Talk("Přesně tak. Máme tu ale problém.");
50         this.Talk("Budeš tam muset zajít a trochu jim vysvětlit o co jde.");
51         this.Talk("Vysvětlí jim to normálně, každý se někdy něco učí.");
52         actor.Talk("Jasně. Jenom se nesmí učit stejnou věc celý život.");
53         this.Talk("Jo, to máš pravdu. Tak za nimi zajdi. Ahoj.");
54         actor.Talk("Zatím");
55         this.Talk("Ještě něco. Pokud si nejsi jistý, co přesně jim vysvětlovat, zaj");
56         actor.Talk("Dobrá. Díky a zatím nashledanou.");
57         dialog.Visible=false;
58         break;
```

Obr. 7.46: Použití proměnné var\_002 v rozhovoru s ředitelem

Ostatní proměnné se deklarují a používají stejným způsobem. Je dobré si psát vedle na papír poznámky, co která proměnná dělá, když je null, false, nebo true. Zamezí se tím nepříjemnostem při tvorbě programu. Někdy je také nutné si proměnnou vyrušit ručně v kódu skriptu, například kvůli vyzkoušení některých dialogů. Rovněž je dobré si vše psát na papír, aby programátor věděl, co kdy a kde pozměnil.

K metodice tvorby samotného programu je toto vše. V postupu se již nevyskytují další nové postupy, tedy i úplný začátečník si s tímto návodem vystačí. Pokud ne, je zde možnost navštívit internetové fórum o WME editoru ani stránkách [http:// www.dead-code.org](http://www.dead-code.org).

## 7.16 Kritéria tvorby programu

Pro účel tohoto výukového programu bylo nutné krom výběru vhodného nástroje, jeho zprovoznění a grafické stránky a scénáře skoubit dohromady několik prvků, díky kterým bude celý program fungovat tak, jak je požadováno. Hlavní kritéria jsou:

1. Jednoduché ovládání programu, pokud možno bez nutnosti nápověd
2. Uživatel celou dobu musí vědět, co je jeho cílem
3. Poskytnutí informací k vyučovaným metodám

## 1) Jednoduché ovládání programu

Zde je využito klasické ovládání myši. Uživatel pohybuje po obrazovce kurzorem. Kurzor je rovněž vytvořen interaktivně, při najetí kurzorem na určitou, použitelnou entitu, dochází ke změně barvy kurzoru. Je také možné upravit ikonku kurzoru, jak je to zvykem v jiných takto koncipovaných programech. Totiž aby například při najetí kurzorem nad ostatní postavy se kurzor změnil na ikonku nějak připomínající mluvení (ústa, či několik písmen). Zde je využita pouze změna barvy kurzoru.

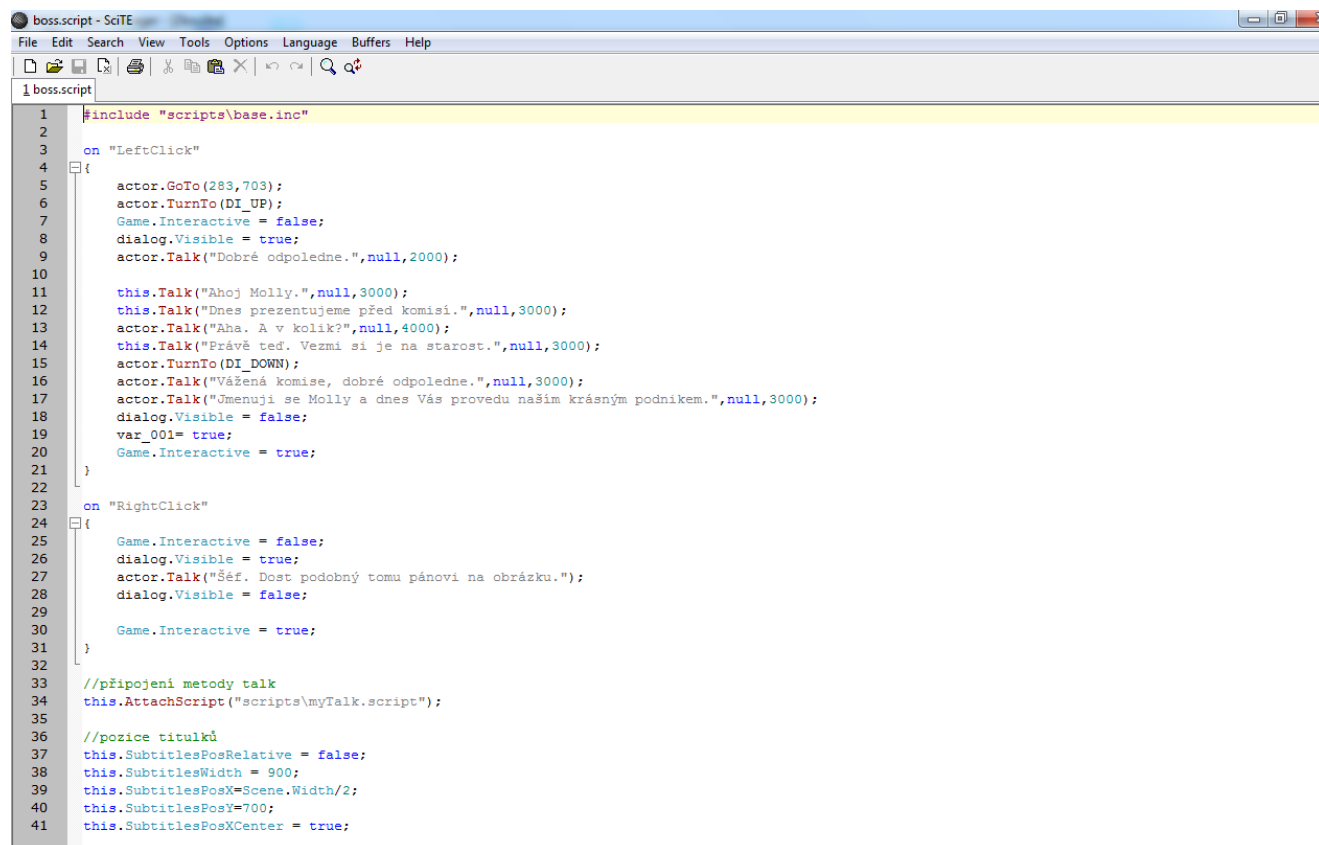
Další věcí je interaktivita samotného prostředí. V podobných programech tohoto žánru velmi často docházelo hned k několika menším problémům.

Prvním z nich byla značná nepřehlednost. Stává se totiž, že v inventáři má uživatel uloženo velké množství předmětů a neví na co je použít. Dalším problémem je obvykle to, že na danou činnost by šlo použít předmětů z inventáře hned několik, ale tvůrce hry hru napsal pro ten jeden konkrétní. A pokud platí Murphyho zákon, tak tím předmětem bude zrovna ten, který v inventáři nemáte. V tomto případě je tento problém vyřešen už samotným konceptem programu. Předměty v inventáři a jejich kombinace zde samozřejmě je, ale všechno je zjednodušené tak, aby uživatel nemusel přemýšlet nad jejich kombinací a mohl se plně věnovat pohodlnému užívání programu. Ten je rozdělen na několik menších tematických celků a v každém tomto celku se nachází jen minimum předmětů a dále je zde od začátku do konce jasné, co s tímto předmětem dělat. Směřují k tomu rozhovory i úkoly samotného programu.

Další z těchto problémů přichází ve chvíli, když vstoupíte do nové lokace a v ní se nachází velké množství použitelných entit. Jedná se například o dveře, okna, předměty na scéně, které nelze přímo sebrat. Uživatel je obvykle nucen všechny tyto předměty prozkoumat a pak dále přemýšlet, není-li třeba na tyto entity použít něco z inventáře, či se na tu věc zeptat jiné postavy.. K prozkoumání předmětu dojde po najetí kurzorem na daný předmět nebo entitu a kliknutím pravým tlačítkem. Postava pak k předmětu přijde (pokud může) a řekne o něm jednu, či dvě věty. Uživatel tak okamžitě pozná, jedná-li se o předmět použitelný dále ve hře, či nikoliv. Pokud na předmět postava nedosáhne, nebo nelze z nějakého důvodu použít, postava zůstane stát na místě, nepříjde tedy blíže k předmětu, řekne k němu komentář a uživatel tak jasně pozná, že tento předmět již dále ve hře využívat nebude a nemusí se jím tedy dále zabývat. Toto řešení byla navrhována pro snadnější použití programu, z angl. user friendly. Je nutné totiž brát v potaz, že tento program nebudou využívat pouze hráči her, tudíž lidé, s hrami a herními mechanismy seznámení.

## 2) Úkoly

Ve výukovém programu je nutné zajistit, aby uživatel byl celou dobu seznámen s cíli samotného programu. Musí tedy celou dobu hraní vědět, co je jeho úkolem. Příběhová část je vzazena do několika obyčejných dnů mladého manažera, kdy jako první věc musíme zajít za ředitelem a ten nám dá hlavní úkol. Úkoly se vzájemně neprolínají, po dokončení jednoho úkolu následuje úkol další. Nicméně bylo nutno vše naprogramovat tak, aby zde nebyla nutnost jakýchkoliv náповěd a uživatel sám věděl, co je jeho cílem, či kde se má pohybovat. Toho je docíleno soustavou proměnných. Viz Obr. 7.47:



```
1 #include "scripts\base.inc"
2
3 on "LeftClick"
4 {
5     actor.GoTo(283,703);
6     actor.TurnTo(DI_UP);
7     Game.Interactive = false;
8     dialog.Visible = true;
9     actor.Talk("Dobré odpoledne.",null,2000);
10
11     this.Talk("Ahoj Molly.",null,3000);
12     this.Talk("Dnes prezentujeme před komisí.",null,3000);
13     actor.Talk("Aha. A v kolik?",null,4000);
14     this.Talk("Právě teď. Vezmi si je na starost.",null,3000);
15     actor.TurnTo(DI_DOWN);
16     actor.Talk("Vážená komise, dobré odpoledne.",null,3000);
17     actor.Talk("Umenuji se Molly a dnes Vás provedu naším krásným podnikem.",null,3000);
18     dialog.Visible = false;
19     var_001= true;
20     Game.Interactive = true;
21 }
22
23 on "RightClick"
24 {
25     Game.Interactive = false;
26     dialog.Visible = true;
27     actor.Talk("Šéf. Dost podobný tomu pánovi na obrázku.");
28     dialog.Visible = false;
29
30     Game.Interactive = true;
31 }
32
33 //připojení metody talk
34 this.AttachScript("scripts\myTalk.script");
35
36 //pozice titulků
37 this.SubtitlesPosRelative = false;
38 this.SubtitlesWidth = 900;
39 this.SubtitlesPosX=Scene.Width/2;
40 this.SubtitlesPosY=700;
41 this.SubtitlesPosXCenter = true;
```

Obr. 7.47: Skript rozhovoru hlavní postavy s ředitelem

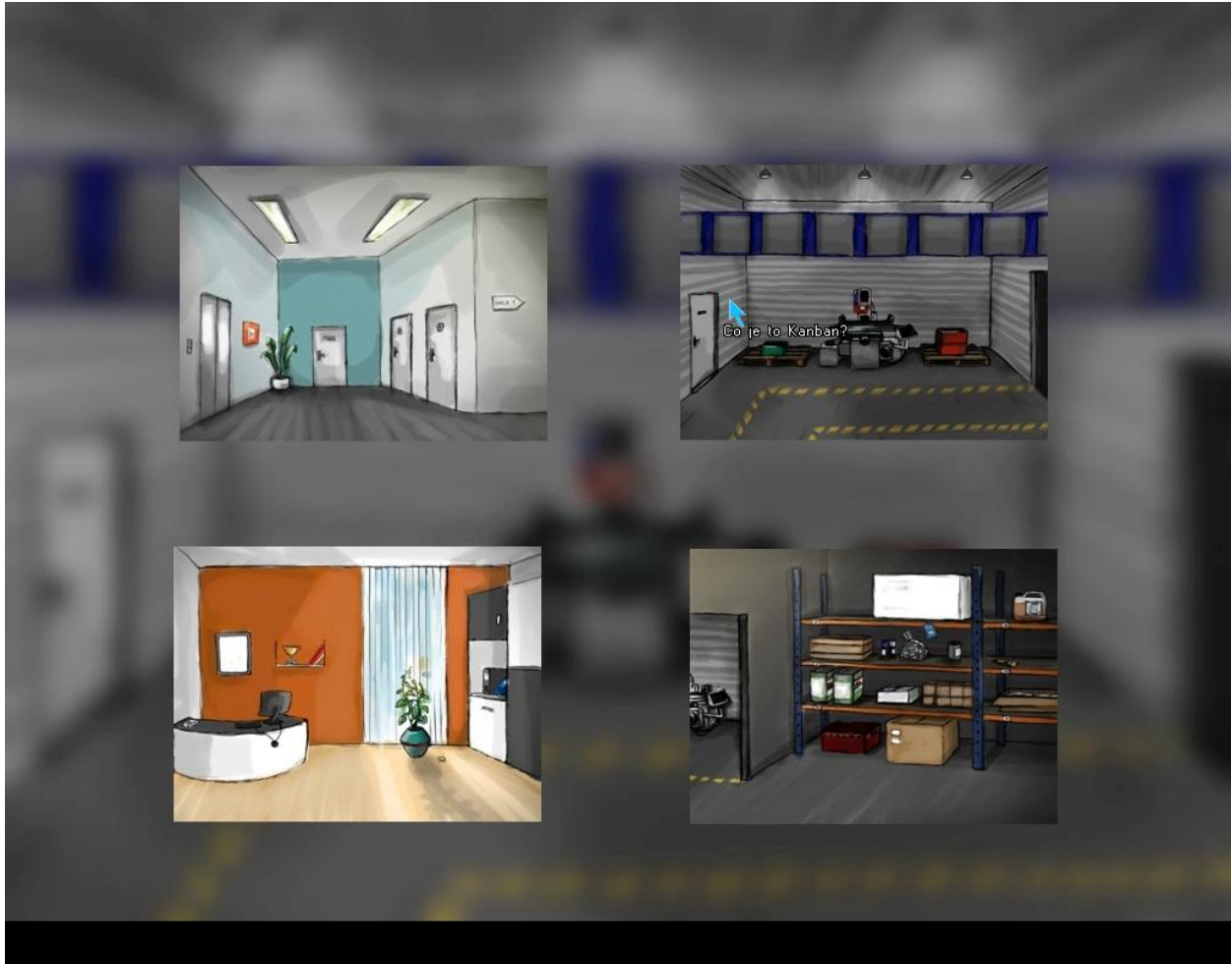
Na Obr. 7.47 je vidět část skriptu, který ovládá rozhovor s ředitelem. Na řádce 19 je důležitá proměnná `var_001`. Tato proměnná je nadeklarována v základním skriptu hry a má hodnotu `false`. Ta ovládá činnost v průběhu celé hry. Na začátku se postava nachází ve svojí kanceláři, a jak je popsáno výše, je nutno zajít k řediteli pro instrukce, aniž by bylo nutné použít jakoukoliv náповědu. Z kanceláře tedy s postavou jdeme na chodbu, ze které se dá dostat ke kanceláři ředitele. Pokud bychom však chtěli jít do dílny, nebo někam jinam, je nastaven skript pro proměnnou `var_001`. Pokud má hodnotu `false`, hra nás nepustí nikam jinam, než k řediteli do kanceláře. Po vstupu do kanceláře má proměnná stále hodnotu `false` a skript pro kancelář ředitele určuje, že z kanceláře nelze odejít. (Postava okomentuje slovy: Musím nejdříve promluvit se šéfem). Po rozhovoru proměnná změní hodnotu na `true`, na obr. 1, řádek 19 a poté již lze odejít z kanceláře. Hodnota proměnné zůstává stále na hodnotě `true`, nyní tedy je možné pokračovat do dílny a zároveň program nedovolí vstoupit zpět do kanceláře ředitele (Komentář: U ředitele jsme už byli, nebudeme ho rušit).

Podobných proměnných je v programu celá řada, starají se o to, aby uživatel hru procházel, tak jak byla hra navržena a nebyla zde možnost odbočit, či se někde zaseknout.

### 3) Poskytnutí informací k vyučovaným metodám

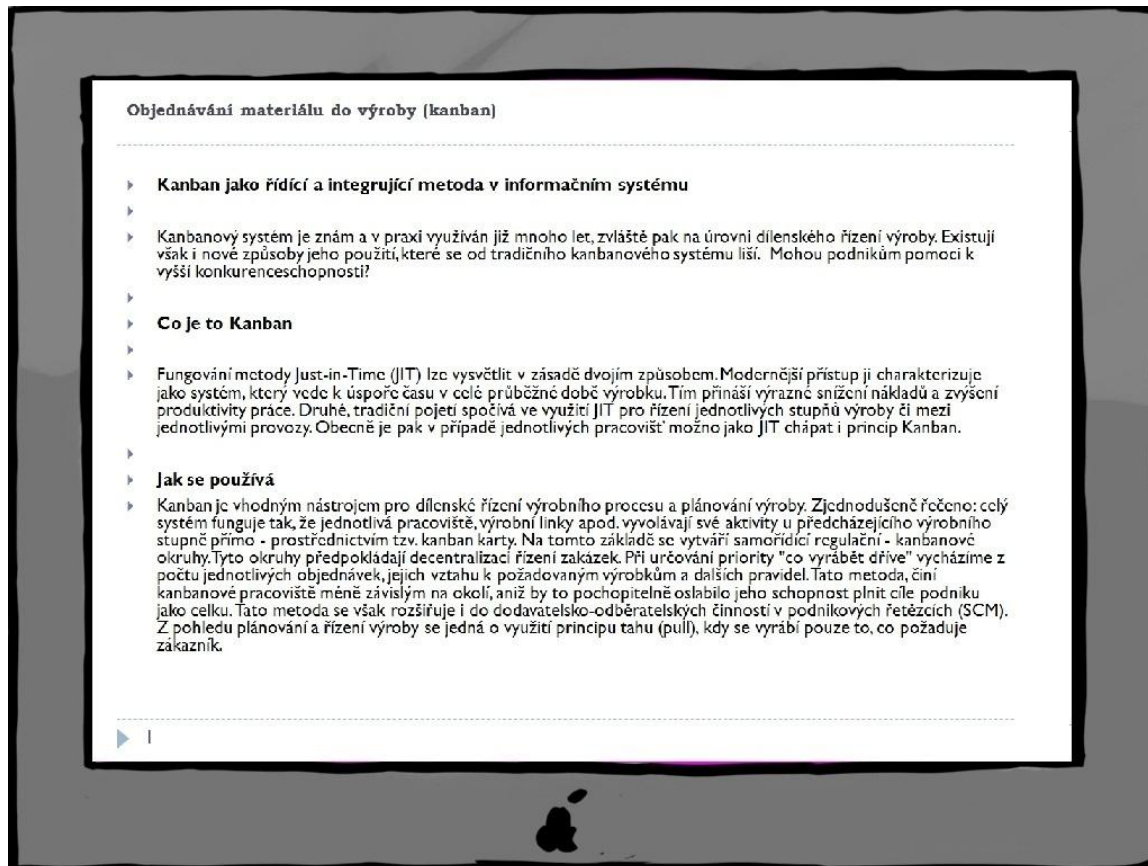
Protože účel programu je výukový, je zde samozřejmě možnost získat informace o dané problematice. Po briefingu u ředitele následuje rozhodnutí. Pokud je uživatel s problematikou seznámen, může pokračovat dále, zpravidla do dílny, kde po rozhovoru s dělníkem dojde k „otestování“ v podobě kvízu pro danou logistickou metodu. Odpovědi jsou zpravidla: a, b, c. Dochází zde také ke sčítání špatných odpovědí s tím, že pokud jich je určitý počet, dělníkovi se to přestane líbit a vykáže uživatele pryč. Ten buď může absolvovat rozhovor znovu, či v jednom případě jde do další lokace, tentokrát hovořit se skladníkem a k dělníkovi se zpět vrátí pak. Toto je opět určeno proměnnou.

Pokud uživatel není seznámen s problematikou, nebo se o tématu chce něco dozvědět, má několik možností. První z možností, na kterou narazí, je odkaz na podprogram s danou tématikou. Uživatel uloží pozici ve hře, přepne do hlavního menu a tam je na výběr krom celé hry i několik mini her. V tomto programu se postava nachází pouze v jedné lokaci a postava zde vysvětluje, jak a k čemu se daná metoda používá. Po skončení se uživatel opět přepne do hlavního programu, kde tyto informace využije. Volba kapitoly programů je na Obr. 7.48:



Obr. 7.48: Volba kapitoly programu

Další možnost je přímo v hlavním programu. Jedná se o příručku v tabletu. Tablet se nachází v kanceláři a po sebrání ze stolu se objeví v inventáři. Po kliknutí na tablet se na jeho displeji objeví informace o dané problematice. Zde však jen teoreticky, v podprogramu má uživatel možnost si vše vyzkoušet. Čtení informací z tabletu, Obr. 7.49:



Obr. 7.49: Informace o systému Kanban uvnitř programu

Poslední případ se nachází v druhém dni scénáře, kdy je nutné vysvětlit skladníkovi, co to znamená metoda Just in Time. Postava zde sama řekne uživateli, aby si o této metodě našel informace sám. Ať už ze skript, či na webu, směřuje uživatele k větší samostatnosti. Pokud si uživatel najde informace sám od sebe, snadněji si o všem udělá svůj názor. Pokud si informace najde na více zdrojích, přínos je o to větší.

Všechny tyto prvky jsou do programu zahrnuty proto, aby uživatel co nejlépe prošel celým scénářem programu a nedocházelo u něj ke zdržení z důvodu nedostatečné informovanosti jak o použitých metodách, tak v tom, co je zrovna jeho cílem.

## 8 Závěr

Cíl diplomové práce, tedy vytvořit interaktivní výukový program formou počítačové hry, se ukázal jako v praxi proveditelný. Po provedení analýzy trhu, výběru vhodného nástroje a žánru, byl vybrán Wintermute Editor, nástroj pro tvorbu počítačových her. Po zvážení všech možností a aspektů byl nakonec vybrán výukový program, typu 2D adventura. Toto řešení bylo zvoleno kvůli přílišné složitosti 3D animací.

Tento program je určen pro studenty ZČU/KPV, kterým zábavnou formou nabídne informace o logistických metodách Kanban, Just in time a 5WHYS.

Využití tohoto programu pro studijní účely ukáže až použití časem. Nicméně možnosti editoru jsou téměř neomezené, do programu lze vsadit prakticky jakoukoliv problematiku. Seznamování se s editorem je časově náročné, ale pokud vývojář toto zvládne, není pak pro něj nejmenší problém přidávat další entity, lokace, nebo scénář. Největší výhodou editoru, je cílový program ve formě komponent. Program není vytvořen jako monolit, je sestaven z mnoha částí, které spolupracují. Není tedy problém kdykoliv něco přidat, či ubrat. Stačí pouze upravit entity a k tomu jejich skripty.

Vize do budoucna je využití tohoto programu, nebo jeho modifikací, jako training programu v podnicích, již mimo univerzitu. Metody, které se zde zmiňují, není problém vhodným způsobem rozšířit přímo pro danou společnost a využít je při školení zaměstnanců. Takovýto nástroj může být obrovským přínosem jak pro školitele, tak pro zaměstnance.

Slabá stránka tohoto projektu je vždy vývojář. Tam jsou totiž omezení, přes která se nedá dostat. Editor sám o sobě je volně šířitelný program, který se dá vylepšovat. Při práci s ním nebyl nalezen problém, který by byl neřešitelný.

V této diplomové práci byly použity výsledky z projektu OP VK č.CZ.1.07/2.3.00/09.0163.

## 9 Zdroje

- [1] Görner, T., Hořejší, P., Kurkin, O., VYZTYMDP : Virtuální realita : úvodní úroveň, e-book, SmartMotion 2012, Plzeň, ISBN 978-80-87539-07
- [2] Hořejší, P., Görner, T., Kurkin, O., VYZTYMDP : Virtuální realita : základní úroveň, e-book, SmartMotion 2012, Plzeň, ISBN 978-80-87539-07
- [3] Z <http://www.wikipedia.org> [online] [cit. 2012-12-4] dostupné z WWW: <[http://en.wikipedia.org/wiki/Educational\\_entertainment](http://en.wikipedia.org/wiki/Educational_entertainment)>
- [4] Z <http://www.wikipedia.org> [online] [cit. 2012-12-4] dostupné z WWW: <[http://en.wikipedia.org/wiki/Educational\\_toy](http://en.wikipedia.org/wiki/Educational_toy)>
- [5] Z <http://www.wikipedia.org> [online] [cit. 2012-12-4] dostupné z WWW: <[http://en.wikipedia.org/wiki/Entertainment\\_in\\_education](http://en.wikipedia.org/wiki/Entertainment_in_education)>
- [6] Z <http://www.wikipedia.org> [online] [cit. 2012-12-4] dostupné z WWW: <[http://en.wikipedia.org/wiki/Serious\\_game](http://en.wikipedia.org/wiki/Serious_game)>
- [7] Z <http://www.kriegsspiel.homestead.com> [online] dostupné z WWW: <[http://kriegsspiel.homestead.com/files/Accessories/prussian\\_equipment\\_4.jpg](http://kriegsspiel.homestead.com/files/Accessories/prussian_equipment_4.jpg)>
- [8] Z <http://www.upload.wikimedia.org> [online] dostupné z WWW: <<http://upload.wikimedia.org/wikipedia/en/f/fd/OregonTrailScreenshot.png>>
- [9] Z <http://www.en.wikipedia.org> [online] [cit. 2012-12-5] dostupné z WWW: <[http://en.wikipedia.org/wiki/The\\_Oregon\\_Trail\\_\(video\\_game\)](http://en.wikipedia.org/wiki/The_Oregon_Trail_(video_game))>
- [10] Z <http://www.gamesforyou.ic.cz> [online] dostupné z WWW: <<http://www.gamesforyou.ic.cz/hry/images/incredible3.png>>
- [11] Z <http://www.digitalislam.eu> [online] dostupné z WWW: <[http://www.digitalislam.eu/dwn/1003/2076C520x390\\_rep3.jpg](http://www.digitalislam.eu/dwn/1003/2076C520x390_rep3.jpg)>
- [12] Z <http://www.mcvideogame.com> [online] dostupné z WWW: <[http://www.mcvideogame.com/images/home\\_imm.png](http://www.mcvideogame.com/images/home_imm.png)>
- [13] Z <http://www.bpa.atech.edu> [online] dostupné z WWW: <<http://bpa.atech.edu/team3/pictures/kinect-sports-.jpg>>
- [14] Z <http://www.recenzeher.eu> [online] dostupné z WWW: <<http://www.recenzeher.eu/img/novinky/2745/g1.jpg>>
- [15] Z <http://www.starepchry.cz> [online] dostupné z WWW: <<http://starepchry.cz/wp-content/uploads/2012/03/OpenTTD.png>>
- [16] Z <http://www.en.wikipedia.org> [online] [cit. 2012-12-6] dostupné z WWW: <[http://en.wikipedia.org/wiki/Game\\_engine](http://en.wikipedia.org/wiki/Game_engine)>



- [17] Z [http:// www.benzworld.org](http://www.benzworld.org) [online] dostupné z WWW:  
<[http://www.benzworld.org/forums/attachments/general-amg/108658d1166138640-amg-winter-drifting-game-13\\_mercedes\\_amg\\_drift\\_competition\\_s.jpg](http://www.benzworld.org/forums/attachments/general-amg/108658d1166138640-amg-winter-drifting-game-13_mercedes_amg_drift_competition_s.jpg)>
- [18] Z [http:// www. cdn3.pcadvisor.co.uk](http://www.cdn3.pcadvisor.co.uk) [online] dostupné z WWW:  
<[http://cdn3.pcadvisor.co.uk/cmsdata/reviews/3374150/LeapPad2\\_LR.jpg](http://cdn3.pcadvisor.co.uk/cmsdata/reviews/3374150/LeapPad2_LR.jpg)>
- [19] Z [http:// www en.wikipedia.org](http://www.en.wikipedia.org) [online] [cit. 2012-12-6] dostupné z WWW:  
<[http://en.wikipedia.org/wiki/Source\\_\(game\\_engine\)](http://en.wikipedia.org/wiki/Source_(game_engine))>
- [20] Z [http:// www.en.wikipedia.org](http://www.en.wikipedia.org) [online] dostupné z WWW:  
<[http://en.wikipedia.org/wiki/File:Half-Life\\_2\\_City\\_17\\_Train\\_Station\\_Square.png](http://en.wikipedia.org/wiki/File:Half-Life_2_City_17_Train_Station_Square.png)>
- [21] Z [http:// www.en.wikipedia.org](http://www.en.wikipedia.org) [online] [cit. 2012-12-6] dostupné z WWW:  
[http://en.wikipedia.org/wiki/Aurora\\_engine#Aurora\\_Engine](http://en.wikipedia.org/wiki/Aurora_engine#Aurora_Engine)
- [22] Z [http:// www.dead-code.org](http://www.dead-code.org) [online] [cit. 2012-12-6] dostupné z WWW:  
<<http://dead-code.org/home/>>
- [23] Z [http:// www. http://en.wikipedia.org](http://www.en.wikipedia.org) [online] [cit. 2012-12-10] dostupné z WWW:  
< [http://en.wikipedia.org/wiki/Wintermute\\_Engine](http://en.wikipedia.org/wiki/Wintermute_Engine)>
- [24] Z [http:// www.healthgamesresearch.org](http://www.healthgamesresearch.org) [online] [cit. 2012-12-12] dostupné z WWW:  
< <http://www.healthgamesresearch.org/games/pulse-the-virtual-clinical-learning-lab>>
- [25] Z [http:// www.dead-code.org](http://www.dead-code.org) [online] dostupné z WWW:  
< <http://res.dead-code.org/lib/exe/detail.php/games:ap02.jpg?id=games%3Astart&cache=cache>>
- [26] Z <http://www.wikipedia.org> [online] [cit. 2012-12-4] dostupné z WWW:  
<<http://cs.wikipedia.org/wiki/Adventura>>
- [27] Z <http://www.mecagenius.univ-jfc.fr> [online] [cit. 2013-05-22] dostupné z WWW:  
<<http://mecagenius.univ-jfc.fr/en/home>>
- [28] Z [http://cs.wikipedia.org/wiki/Entita\\_\(informatika\)](http://cs.wikipedia.org/wiki/Entita_(informatika)) [online] [cit. 2013-05-22] dostupné z WWW: < <http://cs.wikipedia.org/wiki/> >
- [29] Z [http://cs.wikipedia.org/wiki/Sprite\\_\(počítačová\\_grafika\)](http://cs.wikipedia.org/wiki/Sprite_(počítačová_grafika)) [online] [cit. 2013-05-22] dostupné z WWW: < <http://cs.wikipedia.org/wiki/> >

## PŘÍLOHA č. 1

### Scénář výukového programu

## Den 1

### V kanceláři:

Šéf: Ahoj Gregu, máme problém. Na lince jsme zavedli Kanban.

Greg: a) Kanban? Co to je?

b) Kanban? Myslíte hledání úzkých míst ve výrobě?

c) **Dali jste jim kartičky?**

Šéf: a) Ty nevíš co je Kanban? Skoč si za Tracy, ta ti k tomu dá nějaké podklady, nastuduj si je a skoč za chlapama na dílnu jim to trochu vysvětlit. Dveře do její kanceláře jsou na chodbě hned vedle dveří sem.

b) Ty nevíš co je Kanban? Skoč si za Tracy, ta ti k tomu dá nějaké podklady, nastuduj si je a skoč za chlapama na dílnu jim to trochu vysvětlit.

c) **Přesně tak. Máme tu ale problém.**

Greg: a) Konec rozhovoru, jít za Tracy

b) ...

Neví co s tím?

Šéf: **Budeš tam muset zajít a trochu jim vysvětlit o co jde. Vysvětlí jim to normálně, každý se někdy něco učí.**

Greg: **Jasně. Jenom se nesmí učit stejnou věc celý život.**

Šéf: **Jo, to máš pravdu. Tak tam za nimi zajdi. Ahoj**

Greg: **Zatím...**

Šéf: **Ještě něco. Pokud si nejsi jistý, co přesně jim vysvětlit, zajdi si za Tracy, ta ti k tomu dá podklady. Kancelář má hned vedle její kanceláře.**

Greg: **Dobrá. Díky a zatím nashle.**

U Tracy v kanceláři (možnosti a, b)

Tracy: Nazdar Gregu, tak s čím máš problém?

Greg: Kanban.

Tracy: Kan co?

Greg: Kanban. Neříkej, že o tom nic nevíš.

Tracy: Jenom tě zkusím ☺ Tady. Prostuduj si tohle, máš tam o tom info, i konkrétně o tom co jsme zavedli tady ve firmě.

**Tablet s informacemi o systému Kanban**

Greg: Tak ti děkuju. Bonboniéry mi už došly.

Tracy: Stejně si hlídám linii.

Greg: Chápu. Přinesu ti lahev vína ☺

Tracy: To sice moc na linii není, ale neodmítnu.

Greg: Dobrá. Měj se zatím.

### V dílně:

Greg přichází do dílny a dává se do řeči s dělníkem, který pracuje u obráběcího stroje.

Greg: Zdravím chlapi. Slyšel jsem, že tu máte nějaké problémy s Kanbanem.

Dělník: Dobrej. Nevíme co s tím. Dali nám sem tabulky a kartičky, ale nikdo nám nic pořádně nevysvětlil.

Greg: Dobrá. Řekli vám co je Kanban?

Dělník: Moc ne. Můžete nám to zopakovat?

1) Greg: Jasně, od toho tu jsem. Kanban je tu k tomu, aby vám ušetřil práci a hlavně čas.

Je to velký přínos pro všechny. Pro nás to znamená větší přehled o všem a pro vás mnohem méně starostí.

a) **Je to metoda k řízení skladů, ale i například administrativy a podobně.**

- b) Je to metoda pro odstranění úzkých míst ve výrobě.
- c) Je to metoda, podle které zjistíme, která pracoviště jsou vytížena a která ne.
- d) Je to metoda k odstranění zmetků ve výrobě.

Dělník: b) A k čemu ty kartičky?

Greg: Asi jsem řekl pěknou hloupost, co?

c) A na ty kartičky si budeme psát co?

Greg: Tohle asi nebude správně. Gregu, budeš to muset zkusit znovu.

d) A k tomu nám pomůžou kartičky? Zmetky do nich schováme, aby nebyly vidět?

Greg: Máte pravdu, tohle asi nebude správná odpověď.

Dělník: Jste si jistý, že víte, co vám tu povídáte?

Greg: Moc právě ne. Ale já to napravím.

*Zpět k dialogu 1)*

Dělník: a) Jo, to už jsme slyšeli. A můžete nám to říct česky, abychom to my pochopili?

Greg: Jak jsem říkal. Od toho tu jsem. Jenom vám to vysvětlím a bude to všechno v pořádku. Kartičky jsou tam kvůli tomu, aby skladník poznal, kdy vám má dodat další materiál, a tudíž vy se o dodávky chybějícího materiálu nemusíte starat a všechno necháte na nich.

Dělník: A kde jsou ty karty?

2) Greg: Správná otázka.

a) Máte je u sebe, a když potřebujete materiál, napíšete na ně požadavek a pak si ji převezme skladník.

b) Jsou v přepravce s materiálem a pokaždé, když si přepravku vezmete, kartu z ní vyndáte a dáte jí skladníkovi.

Dělník: a) To budu pokaždé vypisovat kartu? Opravdu je to tak?

Greg: To je pravda, to nezní moc logicky.

*Zpět k dialogu 2)*

Dělník: b) Aha. Přijde si pro ni, nebo mu tu kartu musím dát osobně?

3) Greg: a) Až půjde skladník kolem, tak mu tu kartu jednoduše předáte, kolem chodí pravidelně, je to součást systému.

b) Vedle sebe máte nástěnku. Tam tu kartu dáte a skladník si ji vezme, až půjde okolo. Kolem chodí pravidelně, je to součást systému.

Dělník: a) Takže se pořád musím dívat, kdy půjde kolem skladník, abych ho chytil?

Není to trochu nepraktické?

Greg: Teď když to slyším, mi to tak opravdu připadá. Dát kartičku na nástěnku, co je tady vedle, by bylo asi praktičtější.

Dělník: Taky si myslím.

b) Takže vezmu bednu, kartu z ní dám na nástěnku a dál se už o nic nemusím starat?

Greg: Přesně tak. Je to celé kvůli tomu, abyste nemuseli hledat neustále materiál, ale měli ho celou dobu po ruce.

Dělník: A co s tou druhou bednou na hotové výrobky? Tam se taky dává nějaká karta?

4) Greg: a) Ano, kartu musíte vypsát a tu pak dáte do bedny na obrobky, až bude plná.

b) Ne, žádná další karta už není třeba.

c) Karty tady máte připravené. Do bedny dáváte vždy stejný počet polotovarů a až ji naplníte, tak do ní dáte kartu a skladník všechno odveze pryč.

Dělník: a) Takže propisku s sebou?

Greg: Ano, jenom napíšete počet kusů a pošlete dál.

Dělník: c) Takže jenom dát pokaždé STEJNÉ MNOŽSTVÍ hotových výrobků do bedny a pošlu ji dál?

Greg: Ano, přesně tak.

Dělník: A když mám nějaké malé díly a může jich být hodně?

Greg: Pak přichází v úvahu i možnost a) , kde na konci napíšete na kartu počet dílů, který se dozvíte buď z počítačadla, nebo z váhy.

Dělník: b) A kdy tedy skladník pozná, že si může bednu vyzvednout?

Greg: To je pravda, to asi nepozná.

*Zpět k dialogu 4)*

Dělník: Takže tohle je všechno, co musím udělat, aby to všechno fungovalo?

Greg: Ano, víc toho není.

Prostě když si budete brát bednu s obrobky, vezmete z ní kartu a tu pak dáte sem na nástěnku.

Skladník, který pojede kolem, tu kartu vezme a místo ní vám přiveze novou bednu.

To samé uděláte, když budete mít bednu s hotovými výrobky, dáte do ní kartu a bednu dáte na místo odvozu a skladník ji odveze dál.

Tím budete mít pořád dostatek materiálu a nemusíte se o nic dále starat.

Dělník: To zní logicky. Můžete mi to ukázat v praxi?

Greg: Samozřejmě.

Greg: Vzít kartu z červené bedny do inventáře.

Kartu přesunout z inventáře na nástěnku.

Odstoupit

Skladník: Přijet, vzít kartu z nástěnky a nechat na místě novou červenou bednu. (místo prázdné bedny se objeví bedna plná)

Greg: Vzít kartu z frézky do inventáře.

Přesunout kartu z inventáře do zelené bedny.

Odstoupit.

Skladník: Přijet, vzít zelenou bednu a odjet.

Dělník: Díky moc, zas to není tak složité, jak jsme se obávali.

Greg: Není problém, od toho tu jsem. Až zase něco nebudete vědět, stačí říct a já se pokusím vám poradit.

### **Při počtu špatných odpovědí 0 až 1:**

Dělník: Mohl byste nám ještě s něčím pomoci?

Greg: Povídejte. Snad to zvládnou.

Dělník: Ztratil se nám skladník.

Obyčejně mu oběd trvá jenom 2 hodiny, ale už jsem ho neviděl asi dva dny.

Greg: Jenom dva dny? Nechcete mu dát ještě týden?

Dělník: No klidně, ale nám už dlouho nikdo nepřivezl věci ze skladu.

Greg: A to si pro ně nemůžete dojít?

Dělník: No, za první to není moje práce a za druhý to není moje práce.

Greg: No dobrá, jdu se po něm podívat.

Greg: Odchází do skladu.

### **Pokračování ve skladu 1**

### **Při počtu špatných odpovědí 2 a více:**

Dělník: Vy o tom opravdu moc nevíte.

Greg: Možná i jo. Ale slibuju, že se vám to vysvětlím

Dělník: Hele, ztrácíme tu čas. Musíme makat a akorát nás tu zdržujete.

Dělník: Ale s něčím bychom potřebovali pomoc.

Greg: Povídejte. Snad to zvládnou.

Dělník: Ztratil se nám skladník.

Dělník: Obyčejně mu oběd trvá jenom 2 hodiny, ale už jsem ho neviděl asi dva dny.

Greg: Jenom dva dny? Nechcete mu dát ještě týden třeba?

Dělník: No klidně, ale nám už dlouho nikdo nepřivezl věci ze skladu.

Greg: A to si pro ně nemůžete dojít?

Dělník: No, za prvý to není moje práce a za druhý to není moje práce.

Greg: No dobrá, jdu se po něm podívat.

Greg: Odchází do skladu.

### **Pokračování ve skladu 1**

#### **Sklad 1: tma**

Greg: Haló. Je tu někdo?

Haló

Nikde nic.

Nikde nic a zejména a hlavně světlo.

Musím najít světlo, bojím se tmy.

ííííííáááíííííí

Au, teď jsem do něčeho šláp.

Jednou jsem si sedl na dvě kočky a znělo to přesně takhle.

Ještě šlápnu na hrábě a jdu domů. Tady bych čekal opravdu všechno.

Tak, teď už by to asi chtělo najít to světlo.

Hledání vypínače (pixel hunting)

nalezení vypínače, přepnutí scény na sklad 2

#### **Sklad 2. Rozsvíceno, prázdné regály**

Greg: Tady to vypadá jak po vloupačce.

Prázdnou.

Jak v Moskvě v supermarketu před revolucí.

Teď už jenom najít toho skladníka.

Po chvíli:

Jdu se zeptat, jestli dneska vůbec přišel do práce.

#### **U Tracy v kanceláři**

Tracy: Ahoj Gregu, jak to jde?

Greg: Nic moc, už dva dny se prý pohřešuje skladník.

Tracy: Zajímavé, že si toho někdo všimnul až teď.

Greg: Jo, všimli si toho až potom co se vyprázdnily sklady a nikdo jim nedodal materiál.

Tracy: Aha. A co potřebuješ?

Greg: Můžeš se podívat do evidence, jestli dneska přišel do práce? Ať alespoň víme, že tu je.

Tracy: Jasně, minutku.

Měl by tu být, zrovna byl asi na obědě nebo někde.

Greg: Díky, jdu se po něm podívat.

#### **Sklad 1: tma**

Greg: Ale ne, už zase.

Hledání vypínače (pixel hunting)

nalezení vypínače, přepnutí scény na sklad 2

### Sklad 2. Rozsvíceno, prázdné regály

(Skladník je ve skladu, ale vypadá, že spí)

Greg: Vstávat pánové, zaplatíme a jdeme domů.

Skladník: Ale já jsem v práci

Greg: No já vím, vstávat a makat.

Skladník: A vy jste kdo?

Greg: Já jsem tvoje noční můra.

Vstávej.

Jakto, že dělníci nemají materiál a nemakají?

Skladník: To není moje vina.

Greg: Moje taky ne.

Tak co se děje?

Skladník: Od té doby, co se tu zavedl Kanban, tak mi tu nefunguje nic. Nikdo se se mnou nebaví, ale taky zároveň nedostávám Kanban karty.

Greg: Vy chápete, jak funguje Kanban?

Skladník: Jo, kluk studuje, tak mi to doma všechno vysvětlil.

Greg: To je dobře, alespoň někdo ví, jak to funguje.

Dobrá. Jsem tu od toho, abych vám pomohl. Zjistím, co a jak abych vám mohl pomoci.

Odchod do kanceláře

### U Tracy v kanceláři

Tracy: Tak co, našel jsi ho?

Greg: Jo, našel.

Tracy: A?

Greg: Horší, než jsem myslel. Ve skladu to vypadá jako v mojí peněženke před výplatou.

Tracy: A nikdo nic neví. Ani ty ne.

Greg: Přesně.

Můžeš mi nějak poradit?

Tracy: Ale to víš, že jo.

Znáš metodu 5whys?

Greg: Jo, metoda pěti proč.

Netuším, o co jde.

Tracy: Tady o tom něco mám, přečti si to.

Tablet s informacemi o metodě 5whys

Greg: Jo, to nebylo vůbec zlý. Díky moc.

Tracy: Měj se Gregu.

Odchod do skladu

### Při počtu špatných odpovědí 2:

Sklad 2

Skladník: Tak co pro mě máte?

Greg: No, víceméně jenom otázky.

Skladník: Tak se ptejte.

Greg: Jakto, že jsou tu prázdné regály?

Skladník: Protože mi nikdo nepřivezl materiál.

5) Greg: a) Dobrá, jdu zjistit co s tím.

b) A proč vám nikdo nepřivezl materiál?

a)

Skladník: To bylo celé?

Greg: Co jsem potřeboval zjistit, jsem zjistil.

Skladník: Nejmenuje se ta metoda 5 proč?

Greg: Jo, to máte pravdu. Tak ještě jednou.

Zpět k dialogu 5)

b)

Skladník: Protože tu nemám kanban karty.

Greg: A proč tu nemáte kanban karty?

Skladník: Protože mi je z dílny nevrátili.

Greg: A proč vám je nikdo z dílny nevrátil?

Skladník: Protože tomu chlapi nerozumí.

Greg: Jakto, že tomu oni nerozumí a vy jo?

Skladník: Já jsem si o tom něco přečetl, jinak nám tu nikdo nic nevysvětloval.

Greg. Aha, už všechno chápu. Jdu s tím něco udělat.

Odchod do dílny.

Znovu proběhne rozhovor s dělníkem o Kanbanu, teď už ale bez počítadla špatných odpovědí.

Po vysvětlení závěr v kanceláři a pochvala

### **Při počtu špatných odpovědí 0 až 1:**

#### **Sklad 2**

Skladník: Tak co pro mě máte?

Greg: No, víceméně jenom otázky.

Skladník: Tak se ptejte.

Greg: Jakto, že jsou tu prázdné regály?

Skladník: Protože mi nikdo nepřivezl materiál.

5) Greg: a) Dobrá, jdu zjistit co s tím.

b) **A proč vám nikdo nepřivezl materiál?**

a)

Skladník: To bylo celé?

Greg: Co jsem potřeboval zjistit, jsem zjistil.

Skladník: Nejmenuje se ta metoda 5 proč?

Greg: Jo, to máte pravdu. Tak ještě jednou.

Zpět k dialogu 5)

b)

Skladník: Protože tu nemám kanban karty.

Greg: A proč tu nemáte kanban karty?

Skladník: Protože mi je z dílny nevrátili.

Greg: A proč vám je nikdo z dílny nevrátil?

Skladník: Protože tomu chlapi nerozumí.

Greg: To bude v pořádku, už jsem jim to vysvětloval dneska ráno.

Skladník: takže zítra by mělo všechno být v pořádku?

Greg: Ano, slibuju.

Skladník: Dobrá, děkuju vám. Moc jste nám tu pomohl.

Odchod do kanceláře

#### **V kanceláři**

Šéf: Ahoj Gregu. Jak to šlo?

Greg: Nakonec jsme se domluvili, chlapi nejsou hloupí, jenom jim to nikdo nevysvětlil.



Ukázal jsem jim, co a jak a myslím, že bude všechno v pořádku.

Šéf: Dobrá práce, hned máme o starost míň. Už vím, na co tě budu příště „využívat“ ☺

Greg: Pokud za to dostanu Vánoční prémie, proč ne ☺

Šéf: Tak mi o nich ale neříkej o Velikonocích, ano?

## Den 2

### Kancelář

Greg: Dobré ráno. Sháněl jste se po mně?

Šéf: Ahoj Gregu. Potřebuju, abys mi vyřešil drobný problém.

Greg: O co jde?

Šéf: Včera jsi jim to vysvětlil, ale volal mi sem ráno skladník, že pořád čeká na dodávku.

Greg: Říkal ještě něco dalšího?

Šéf: Zeptej se ho sám. Pak se tu zastav mi říct, jak jsi to vyřešil.

Greg: Dobrá. Ještě něco?

Šéf: Ty jsi dobrej v tom vysvětlování.

Budeme tu zavádět Just in time. Vysvětli ve skladu, o co jde.

Greg: Dobrá, jdu na to. Nashle

šéf. Ahoj Gregu.

### Chodba

Greg: Zajdu za Tracy, ať se dozvím, co je JIT.

Vzít za dveře.

Greg: Hm, není tu.

Bude lepší se podívat do skript nebo na web, co vy na to? ☺

Najděte si, co je Just in Time ať můžeme jít do skladu.

Hotovo?

1) Greg: Dobrá, takže co je metoda Just in time?

a) Metoda monitorování skřítků na evropských kulturních festivalech v závislosti na vypitém absintu, právě hned po dopití.

b) Metoda, která značně urychluje výrobu, zavedením nových postupů

c) **Metoda, umožňující vyrábět požadované výrobky v určeném množství a určeném čase.**

Greg: a): Hráč zřejmě absinth pil. Nicméně se nejedná o úplně správnou odpověď.

b): Nene, to není ono. Zkuste to ještě jednou.

Zpět k dialogu 1)

c) Výborně.

Název se do češtiny nepřekládá, ale dá se vyložit jako „Právě včas.“

Principem je tedy dodávat to, co zákazník požaduje, přesně když to požaduje.

2) Greg: Co je tedy hlavním cílem této metody?

a) **Aby odběratel nemusel držet zboží skladem.**

b) Aby odběrateli klesly náklady, spojené s dopravou.

c) Zásobování afrických států gumovými medvídky. Protože jak víme, po vypuknutí krize je jich tam obrovský nedostatek.

Greg: b) Ale notak. ekonomická krize se přeci produkce gumových medvídků vůbec nedotkla.

c): U zkoušky by to možná i prošlo, nicméně se nejedná o tu úplně správnou odpověď.

Zpět k dialogu 2)

a) Přesně tak. Zavedení systému včasných dodávek nemusí mít odběratel tolik zboží na skladě. Ideálně žádné.

3) Greg: Na co se dále klade velký důraz?

- a) Vyrábět jen to, co je potřebné a tak efektivně, jak je to jen možné.  
b) Mít na skladě co nejvíce výrobků tak, abychom mohli odběrateli vždy dodat, co potřebuje.  
c) Na trpaslíky. Trpaslíky do každé zahrady.
- Greg: b): To by bylo akorát drahé a nic víc. To nebude ono.  
c): Nebuďte jako můj soused, prosím.
- a) Výborně. Klade se zde důraz na komunikaci a důvěru mezi dvěma obchodními partnery. Dodávat potřebné zboží v pravý čas a v předepsané kvalitě.
- 4) Greg: Jak je tedy dosaženo tohoto systému včasného dodávání?  
a) Velmi časté a včasné dodávky, někdy i několikrát v průběhu dne.  
b) Pečlivé rozplánování výroby a vytvoření jedné velké dodávky v přesný čas, aby nebyly moc vysoké náklady za dopravu.
- Greg: b) Nene. Nikdo nedokáže plánovat všechny proměnné ve výrobě. Až se vám to povede, rádi vás tu zaměstnáme.  
a) Přesně tak. Odběratel sice platí vyšší náklady za dopravu, nemusí však držet zboží skladem a zboží dostává přesně v tu dobu, kdy jej potřebuje. Tím dojde ke značnému snížení nákladů celkově.
- 5) Greg: Co je další výhodou této metody krom výhod, které jsme zmiňovali předtím?  
a) Zkrácení cyklu plánování a zvýšení pružnosti výroby.  
b) Menší časová náročnost spedičního nebo i celního odbavení.
- Greg: b) Nene, je to právě naopak.  
a) Ano, je to tak. Pravidelnými dodávkami menšího počtu výrobků nebo zboží můžeme výrazně zkrátit cyklus plánování a můžeme pružněji reagovat na poptávku zákazníka.
- 6) Greg: Má tato metoda nějaké nevýhody?  
a) Víceméně ne. Dojde ke snížení nákladů a zvýšení pružnosti.  
b) Nárůst rozsahu přepravy stále menších zásilek stále větším počtem nákladních automobilů.  
c) „Pomalé výrobky“ se musejí vyrábět jen na zakázku a ne na sklad.
- Greg: a) Nic na světě není bezchybné. Teda jenom komunizmus byl, ale ten byl tak bezchybný až nakonec padnul  
c) To není zas taková nevýhoda. Vyrábění na sklad je věc, kterou je potřeba eliminovat, než podporovat.  
b) Ano, toto může být problém hlavně pro dodavatele. Například zácpa nebo nehoda na dálnici může způsobit v nejhorším případě i zastavení výroby na straně odběratele.
- Greg: Dobrá, metodu JIT jsme probrali a můžeme se jít podívat do skladu, co se tam vlastně děje.

## Sklad 2 – rozsvíceno

Greg: Dobré ráno, tak jak to jde?

Skladník: No, až na dva menší problémy to jde docela dobře.

Greg: Asi začínám tušit.

Co jsou ty dva problémy?

Skladník: První je metoda Just in time, se kterou mi tu teď všichni straší.

Greg: A ten druhý?

Skladník: Pořád to tu vypadá jak po zásahu středně velkým tornádem.

Greg: A nebylo by dobrý tu trochu uklidit?

Skladník: Ně, já myslím náš menší problém s dodávkami zboží.

Greg: Dělníkům jsem vysvětlil co a jak.

Skladník: No jo, ale nic se nezměnilo.  
Greg: Skočím do dílny a poptám se.  
Skladník: Díky moc, já tu aspoň zametu.

### Dílna

Greg: Zdravím šéfe. Jak to jde?  
Dělník: Super, od včerejška už nemám výpadky  
Greg: Je tu ale problém ve skladu.  
Dělník: A proč se na to ptáte mě?  
Greg: Nedostal od vás kanban karty a když je neposlal dál, nikdo mu nedodal zboží.  
Dělník: Já všechny karty odevzdal.  
Greg: A ty kopie, co tu zůstávají, tady jsou?  
Dělník: Jo, měly by tu někde být.  
Tady v koši. Ale je jich tu moc.  
Greg: Skočím do skladu pro originály.  
Odchod do skladu.

### Sklad 2 – rozsvíceno

Skladník: Tak co jste zjistil?  
Greg: Je možné, že nějakou kartu ztratil.  
Jenomže těch kopií tam má víc.  
Skladník: Takže potřebujete ty originály, co se mi mají vracet?  
Greg: Přesně tak.  
Skladník: Není problém. Tady jsou.  
Sebrat kanban karty (originály)  
Greg: Tak já to jdu porovnat.  
Odchod do dílny

### Dílna

Greg: Tak to tady mám.  
Dělník: Tady jsou ty kopie karet.  
Greg: Tak se na to podíváme.  
Porovnat originály s kopiemi. Jeden originál chybí.  
Greg: Tahle karta tu chybí.  
Dělník: Ale kopie tu je, takže mi to rukama prošlo.  
Greg: Ze kdy to je?  
Ze dneška.  
Takže tu někde musí být.  
Dělník: To by asi měla, ale nikde jí tu nevidím.  
Greg: Co je to tu na zemi?  
Dělník: To mi ukáplo lepidlo.  
Greg: A šlápnul jste do něj?  
Dělník: Jo, několikrát.  
Proč se ptáte? Já to uklidím.  
Greg: Byl jste v jídelně nebo na záchodě?  
Dělník: Jo, byl.  
Je to důležitý?  
Greg: Mohla se vám nalepit na podrážku.  
Dělník: A to by mě nenapadlo.  
Greg: Jdu se podívat.

Odchod na chodbu.

### Chodba

Greg: A hele, tady jí máme.

Sebrání karty do inventáře.

Porovnání karet.

Karta je ta chybějící.

Odchod do skladu

### Sklad

Greg: Našel jsem tu chybějící kartu.

Skladník: Ano, to je ona.

Díky moc, teď už by mělo být všechno v pořádku.

Greg: To jsem moc rád.

Teď už snad začneme i něco vyrábět.

Skladník: Děkuju moc.

Měl bych ještě jednu otázku.

Co je metoda Just in time?

Greg: To vám vysvětlím hned.

Protože už jsme si o metodě JIT povídali, tak tento rozhovor urychlíme ☺

...

A to je všechno.

Skladník: Takže je to metoda tahu?

1) Greg: a) Nene, je to systém tlaku.

b) Tah i tlak.

c) Ano, je to metoda tahu.

Greg: a) No, osobně nevím, co bychom kam chtěli tlačit.

b) Nene, to nebude ono.

c) Přesně tak. Odběratel si od dodavatele „vytahuje“ požadavky.

Skladník: Tak dobrá, to bude asi všechno.

Děkuju vám

Greg: Za málo. Nashle

Skladník: Nashle a ještě jednou díky

Odchod do kanceláře ředitele

### Kancelář

Šéf: Ahoj Gregu. Tak jak to šlo?

Greg: Měli jsme tu malou detektivku, ale všechno jsme s chlapama vyřešili.

Šéf: Mám se ptát na podrobnosti?

Greg: Radši ani ne, myslím že pro hráče bylo rozhovorů dost ☺

Šéf: Dobrá.

Chlapi tě chválili. Jen tak dál.

Greg: Díky za pochvalu.

Tak já zase jdu.

Nashle

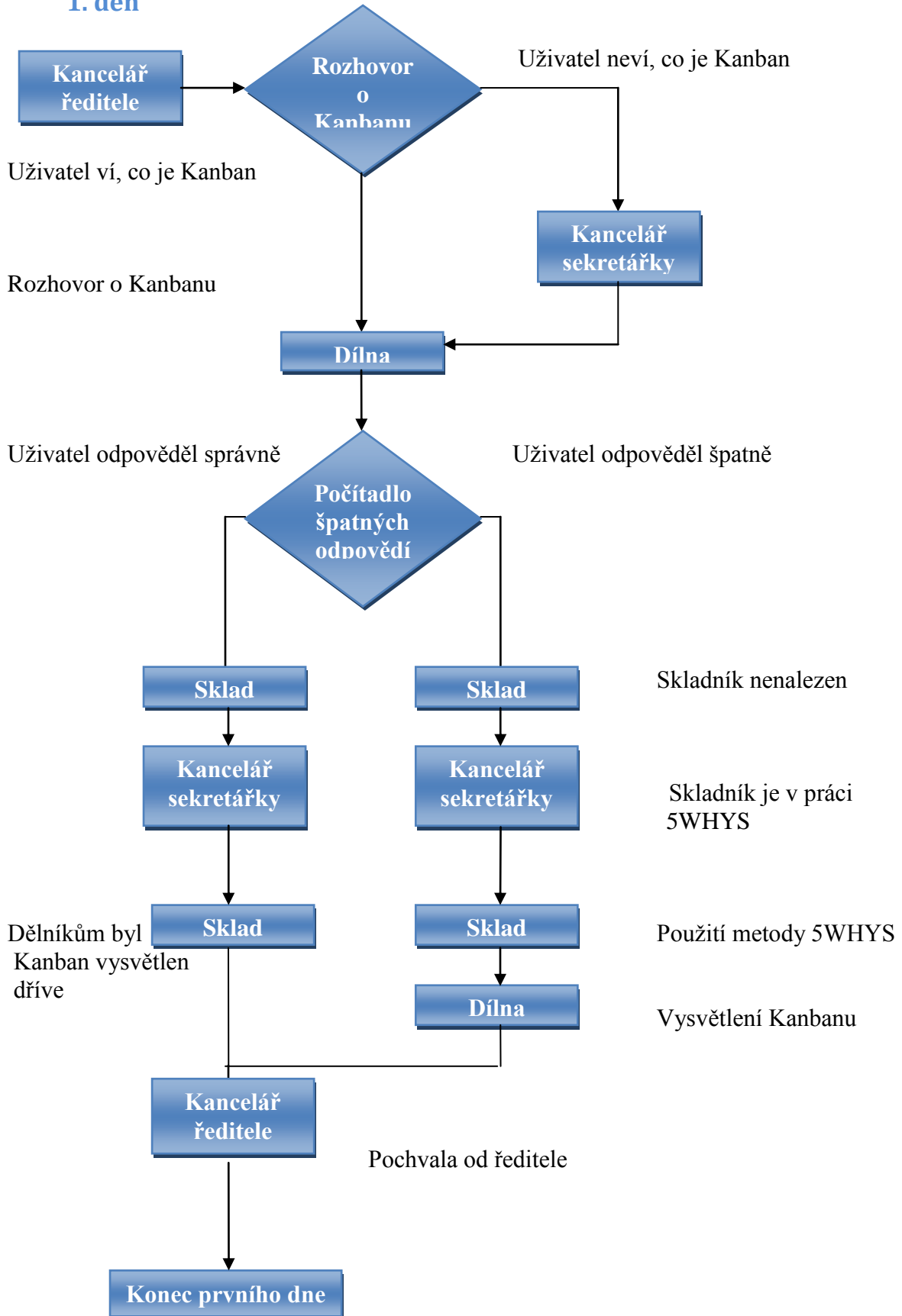
Šéf: Ahoj Gregu.

## PŘÍLOHA č. 2

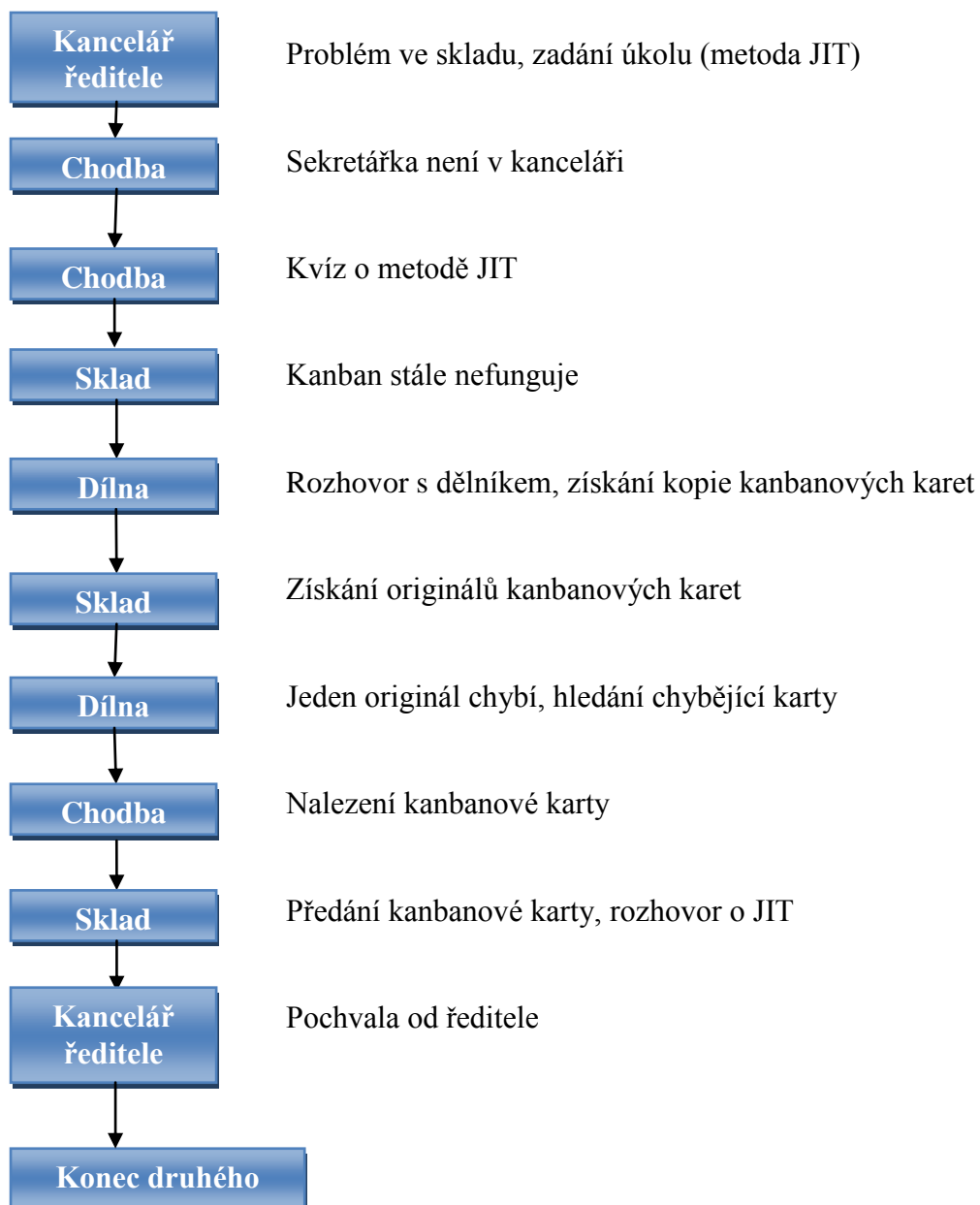
### Vývojový diagram průchodu scénami

## 10 Vývojový diagram scénáře

1. den



## 2. den



## PŘÍLOHA č. 3

### Ilustrace





Ilustrace 1: Lokace chodba



Ilustrace 2: Lokace výrobní hala



**Ilustrace 3: Lokace sklad - zhasnuto**



Ilustrace 4: Prázdný sklad – rozsvíceno



**Ilustrace 5: Lokace kancelář ředitele (entity ředitel a stůl jsou přidávány až v programu)**



Ilustrace 6: Kancelář Grega a Tracy

## PŘÍLOHA č. 4

Zajímavosti z tvorby programu

## Zajímavosti z tvorby programu

V počítačových hrách napříč všemi žánry, se vyskytují různé zajímavosti, překvapení pro hráče a různé vtipné momenty. Některé z nich odkazují na známé filmy, jiné zase na známé osobnosti či události. I v tomto výukovém programu je jich celá řada a zde je jejich výčet:

Při tvorbě scénáře bylo nutné vymyslet jméno hlavní postavy. Bylo zvoleno neutrální jméno Greg. Při vymýšlení jména sekretářce bylo zvoleno jméno Tracy. Greg Tracy je motocyklový závodník a kaskadér. To bylo zjištěno až mnohem později.

Při vstupu do kanceláře ředitele je zde odkaz na první díl známého seriálu IT Crowd, kde sedí ředitel za stolem a za ním je obrázek jeho, sedícího za stolem. Stejně je to zde.

Dalším odkazem na seriál IT Crowd, je kancelář Grega a Tracy. V seriálu měli hlavní hrdinové kancelář s obrovským množstvím nepořádku, zde je na skříni rozbitý počítačový monitor. Při pravém kliknutí na monitor se od Grega dozvíme, že s Tracy soutěžili, kdo dohodí monitorem dál. A že Tracy vyhrála.

Tracy chodí do práce v růžovém. Růžový kabát však není její součástí oděvu. Jedná se o starý, vysloužilý vařák, o který se Tracy s Gregem střídali, když jim v zimě netopilo auto.

Plakát na skříni, vedle které sedí greg v kanceláři, je vytržená stránka z kalendáře. Konkrétně duben 2012.

V podniku nikdo nezalévá květiny, i přesto stále žijí.

Po nakreslení ilustrace pro lokaci chodba jsme zjistili, že tam kde měly být dveře, jsou dveře výtahu a tím pádem chybí dveře k Tracy a Gregovi do kanceláře. Muselo tedy dojít ke sloučení toalet na společné toalety a zbylé dveře použít, jako dveře do kanceláře.

Greg se bojí tmy. Po vstoupení do zhasnutého skladu, musí uživatel nejdříve najít vypínač, jinak Greg odmítá jít dále.

Výtah v lokaci chodba je podle Grega ještě pomalejší, než na ZČU.

V kanceláři Grega a Tracy se Greg omylem přizná, že si v kufříku schovávají Playstation.

Není známo, jak se jmenuje podnik, ve kterém se děj odehrává, ani v jakém oboru podniká.