

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

Průvodce hlavními funkcemi programu Stencyl
BAKALÁŘSKÁ PRÁCE

Martin Červený
Informatika se zaměřením na vzdělávání

Vedoucí práce: Mgr. Tomáš Přibáň, Ph.D.

Plzeň, 2016

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni, 1. dubna 2016

.....
vlastnoruční podpis

Poděkování

Tímto bych chtěl poděkovat vedoucímu bakalářské práce, Mgr. Tomášovi Přibáňovi, Ph.D, za pomoc a veškerý čas, který mi věnoval při konzultacích a za cenné rady, které mi pomohly tuto práci vytvořit.

ZDE SE NACHÁZÍ ORIGINÁL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

OBSAH

Úvod	2
1 ZAŘAZENÍ PROGRAMU STENCIL.....	3
1.1 CO STENCIL PŘEDSTAVUJE	3
1.2 VYUŽITÍ PROGRAMU STENCIL	4
1.3 ALTERNATIVY	4
1.4 VÝVOJOVÉ PROSTŘEDÍ HAXE	5
2 HISTORIE PROGRAMU STENCIL	6
2.1 UVEDENÍ	6
2.2 VERZE 3.0	6
3 POPIS PROSTŘEDÍ PROGRAMU STENCIL	7
3.1 UVEDENÍ DO PROGRAMU	7
3.2 ZÁKLADNÍ NABÍDKA	8
3.3 ZÁKLADNÍ POJMY	9
4 POSTUP PRÁCE ZNÁZORNĚNÝ POMOCÍ ANIMACÍ	11
4.1 ACTOR TYPES	11
4.2 BACKGROUNDS	15
4.3 TILESETS	16
4.4 FONTS	16
4.5 SOUNDS	17
4.6 SCENES	17
5 PRAKTICKÉ PŘÍKLADY	19
5.1 JEDNODUŠŠÍ HRA	19
5.1.1 Cíl	19
5.1.2 Popis hry	19
5.1.3 Řešení	19
5.2 SLOŽITĚJŠÍ HRA - HOROLEZEC	27
5.2.1 Cíl	27
5.2.2 Popis hry	27
5.2.3 Řešení	28
ZÁVĚR.....	32
RESUMÉ	34
SEZNAM LITERATURY	36
SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ	38
PŘÍLOHY	I

Úvod

Bakalářská práce se zabývá programem Stencyl, jeho základními funkcemi a použitím. Toto téma jsem si vybral hlavně z toho důvodu, že v dnešní době se rozvíjí informační technologie velice rychlým tempem. Elektronika nás obklopuje více a více. Pokud chceme, aby fungovala tak, jak má, neobejdeme se bez programování. Proto je důležité, aby se základy algoritmizace učily děti v co nejútlejším věku. Pokud se ale před školáky někdo zmíní o programování, většinou si představí pod tímto pojmem psaní dlouhých kódů. Nelze se divit, že se o takovéto věci zajímá pouze malá část studentů. Se správným přístupem ale můžeme zjistit, že ne každé programování znamená psaní kódu a ladění syntaxe. Na základních školách se již vyučují základy programování například v programu Scratch. Tento program se stal oblíbeným pro ty, kteří s programováním nemají zkušenosti, je jednodušší na pochopení a má uživatelsky příjemnější prostředí. Proto začaly vznikat další programovací jazyky jemu podobné, právě jako například Stencyl.

Smartphone je pro dnešní mládež již téměř samozřejmostí, nedovedli by si život bez chytrého telefonu představit. Pokud si chce někdo vytvořit vlastní aplikaci, a že takovýchto lidí přibývá, je program Stencyl ideální volbou. Může se jednat o aplikaci s výukovým charakterem, na procvičení aktivity mozku či jednoduše o hru pro zábavu. Stejně tak, pokud chcete začít s programováním, je tento program velice šikovný a intuitivní. Internetové stránky programu nabízí několik kurzů, kterými si může člověk projít a naučí ho základy práce s ním. Pokud má již uživatel zkušenosti s programováním, práce v tomto programu by pro něj neměla být příliš složitá. Nevýhodou by pro někoho mohlo být, že Stencyl není přístupný české jazykové verzi.

1 ZAŘAZENÍ PROGRAMU STENCIL

1.1 CO STENCIL PŘEDSTAVUJE

Program Stencil je vývojové prostředí, které umožňuje vytvářet aplikace nejen pro počítače a web, ale i pro telefony či tablety. Výhodou je, že je možné si zvolit rozlišení pro každou nově vytvářenou aplikaci a tím se lze přizpůsobit jednotlivým zařízením. Stencil je podobně jako Scratch vizuální programovací jazyk. Nejedná se tedy o klasický programovací jazyk, ve kterém tvoříme program psaním kódu, ale nabízí sestavení programu skládáním bloků s předdefinovanými příkazy bez nutnosti psát jediný řádek. Tvorba je tedy podobná jako při programování v klasickém vývojovém prostředí, ale máme ulehčenou práci v tom, že není třeba kontrolovat syntaxi. Uživatelé, kteří někdy zkusili napsat počítačovou hru pomocí jazyku, jako je ActionScript od Adobe nebo Objective-C od Apple, dobře vědí, jak těžké je zajistit, aby každá závorka, středník nebo jiný speciální symbol byl na správném místě. Tento problém je se Stencylem minulostí. Způsob programování spočívá v přesouvání a skládání připravených bloků, které do sebe zapadají a tvoří tak tělo programu. Do vytvářené aplikace může tvůrce vkládat vlastní obrázky či zvuky. Tento program je v základní verzi k dispozici zdarma a tato základní verze nám dovoluje publikovat vytvořenou aplikaci ve formátu swf, HTML5 (v současnosti v experimentální fázi) či přímo do Chrome Web Store. Verze Pro, která je placená, umožňuje exportovat vytvořenou aplikaci buď pro desktopové (podporuje Windows, Linux a OS X) nebo mobilní využití (podporuje Android, iOS a Xcode project).

Stencil tedy není vývojová platforma jako například program Eclipse, kde se programuje pomocí zadávání kódu a psaní jednotlivých příkazů, ale ani profesionální herní engine jako například Unity, který umožňuje vytváření složitých 3D situací a herních scén. Program Stencil je něco mezi těmito dvěma. Je to software určený pro vytváření a rozvoj videoher či interaktivních aplikací. Má již předdefinované některé často používané funkce, jako fyzika – tedy chování předmětů v rámci gravitace nebo detekce kolizí. Poskytuje nám jednodušší vytvoření žádaného projektu. Je tedy uživatelsky přívětivější a umožňuje nám rychlejší vytvoření funkcí, které se často používají.

1.2 VYUŽITÍ PROGRAMU STENCYL

Jedná se o přehledný software, který je plný intuitivních nástrojů a velmi usnadňuje a urychluje samotný proces vývoje. Tvoření programu je možné pomocí skládání bloků a výsledek je brzy vidět. Při začátcích programování to ocení hlavně mladší uživatelé, poněvadž mnozí z nich si řeknou už při vyslovení slova „programování“, že to je něco, co nezvládnou. Ocení to také vyučující, kteří mohou snadno ukázat funkci podmínek a cyklů na několika málo blocích bez nutnosti zabývat se syntaxí. Díky tomu je ideální pro výukové účely a dá se využít v mnoha různých projektech včetně simulací, či pro tvorbu animovaných příběhů. Konkrétní ukázky a možnosti použití jsou nastíněny v praktické části práce. Díky možnosti vytvoření spouštěcího souboru pro různé platformy se stává jednodušší i rozšíření mezi uživatele po celém světě, používající různá zařízení. Nemusí se tedy vytvářet například stejná aplikace zvlášť pro systém Android a systém iOS, protože Stencyl je vyvíjen tak, aby zvládl vytvoření spouštěcího souboru pro oba systémy, a to za použití stejného kódu.

Vývojové prostředí Stencyl může být nainstalováno na operační systém Microsoft Windows (XP, Windows Vista, Windows 7, Windows 8 a Windows 10), Mac OS X, Linux. Na těchto systémech může být taktéž hra spuštěna. Cílových platform pro spuštění hry vytvořené programem Stencyl je ale mnohem více. Pro mobilní zařízení Apple běžící na iOS (verze 4.1 a vyšší) jsou to iPod touch, iPad, iPhone. Pro zařízení běžící na systému Android (2.2 a vyšší) jsou to veškeré telefony a tablety. Dále také fungují na mnoha prohlížečích podporujících webový standard HTML5.

1.3 ALTERNATIVY

Do stejné kategorie jako Stencyl bychom mohli zařadit také program Scratch, který byl oficiálně vydán již v roce 2007 a na kterém se dnes učí programovat i žáci základních škol. Výhodou tohoto programu je, že se nemusí instalovat, pracuje se s ním online v internetovém prohlížeči. Dalšími alternativami podobného stylu programování jsou Gamesalad či Construct 2, vyvinuté společností Scirra. Ve všech těchto programech se programuje převážně systémem přesouvání připravených bloků, či nastavováním hodnot, namísto psaní kódu na klávesnici.

1.4 VÝVOJOVÉ PROSTŘEDÍ HAXE

Program Stencil je vyvíjen v jazyce Haxe, který se poprvé objevil v roce 2005. Haxe je moderní multiplatformní programovací jazyk. Tento jazyk může vytvářet aplikace a zdrojový kód pro mnoho různých platforem z jedné kódové báze. Obsahuje sadu společných funkcí, která je podporována na všech platformách, jako například číselné datové typy, pole. Programovací jazyk Haxe používá Haxe kompilátor a kód napsaný v tomto jazyce může být kompilovaný do kódu Java, C++, C#, PHP, Python, JavaScript programů, ActionScript 3 kódu atd. To umožňuje mimořádně efektivní vývoj napříč různými platformami a v konečném důsledku šetří čas a prostředky. Používají jej tisíce vývojářů po celém světě při vývoji her, aplikací, nástrojů a frameworků. Ze známých společností můžeme uvést například Toyota, BBC, Hasbro či Coca-Cola.(1)

Kdo je již obeznámen s C++, Java, PHP nebo jinými objektově orientovanými jazyky, není pro něj těžké se pracovat s programovacím jazykem Haxe. Byl speciálně navržen tak, aby se přizpůsobil chování různých platforem. Standardní knihovna Haxe obsahuje společnou sadu vysoce testovaných API (rozhraní mezi jednotlivými programy), které umožňují správné chování všech platforem. To zahrnuje datové struktury, matematiku a datum, šifrování, serializaci, atd. Standardní knihovna Haxe obsahuje také specifické API, které umožní přístup do důležitých částí možností platformy a dá se jednoduše rozšířit.

Následující úsek kódu v Haxe demonstruje vypsání textu Hello World (viz Obrázek 1):

```
class HelloWorld {
    static public function main() {
        trace("Hello World");
    }
}
```

Obrázek 1 Úsek kódu v Haxe (2)

2 HISTORIE PROGRAMU STENCIL

2.1 UVEDENÍ

Program nemá dlouhou historii. Tento software byl poprvé představen v roce 2011, konkrétně 31. května. Byl původně nazvaný StencilWorks, ale později byl zkrácen na pouhý Stencil. Z počátku fungoval na platformách Java, ActionScript 3, Objective-C a C++.

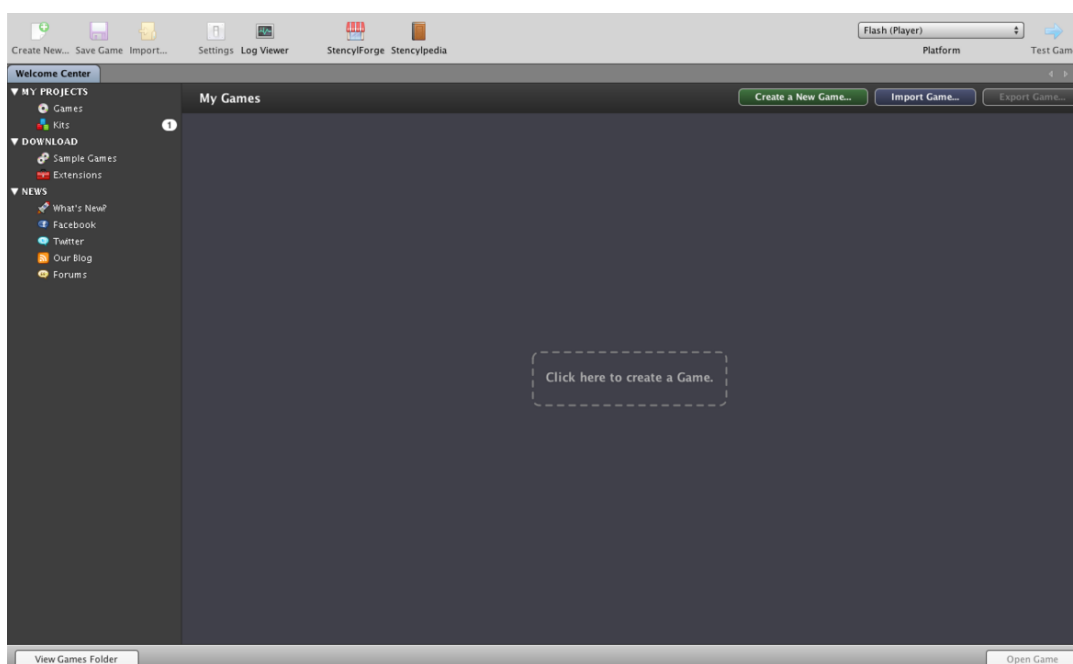
2.2 VERZE 3.0

Od verze 3.0, která vyšla 21. února roku 2014, funguje na multiplatformním programovacím jazyce Haxe. V této verzi proběhlo ještě několik malých úprav, jako například přidání tlačítka Create New Game dialog (Vytvořit nové dialogové okno), které uživatele přeměruje na stránku Sample Games (ukázkové hry), než byla nahrazena verzí 3.1. V této verzi bylo opraveno zpožděné přehrávání zvuku v některých zařízeních Android, přibyl například Bitmap Font Editor, kreslení actors v rámci vrstev a mnoho dalších úprav. K nim došlo v květnu 2014. V prosinci téhož roku došlo k vydání verze 3.2. V ní došlo opět k několika vylepšením. Bylo zkvalitněno hlášení o chybě (error), ale důležitou věcí byla kompatibilita s Javou 7/8 a také s iOS 8 a OS X Yosemite. Následovala verze 3.3, která vyšla v březnu 2015. Hlavní položkou zde byla podpora 64-bitové verze pro iOS. V Design Mode byl zvýšen výkon a přidáno mnoho oprav. Tato verze byla ještě upravena v červnu 2015, kde bylo navrženo řešení pro některé Mac a Windows systémy, pro které byl program Stencil nepoužitelný. To byla verze 3.3.1. a dále byl program upravován od července do září 2015 na verzi 3.3.2. Tato verze přidává experimentální podporu pro HTML5, která snižuje herní dobu testování až o 66%. V říjnu roku 2015 je spuštěna testovací verze 3.4.0, která přináší opět několik oprav.

3 POPIS PROSTŘEDÍ PROGRAMU STENCYL

3.1 UVEDENÍ DO PROGRAMU

Nejnovější verzi programu lze jednoduše stáhnout do počítače z domovské stránky www.stencyl.com. Po nainstalování a spuštění se nám zobrazí úvodní obrazovka (viz Obrázek 2) programovacího prostředí Stencyl, které je defaultně v angličtině.



Obrázek 2 Úvodní obrazovka

Jazyk lze změnit na dalších 13 jazyků po kliknutí na File a následně Preferences. Češtinu však program bohužel nenabízí. Pro vytváření her není nutné být registrován, bez přihlášení však není možné přispívat do fóra nebo využívat možností StencylForge, který nabízí mnoho užitečných funkcí. V základním menu existuje v levém sloupci možnost vybrat a otevřít již vytvořené projekty, pokud již nějaké máme, dále si můžeme stáhnout ukázkové hry připravené vývojáři Stencylu a také se podívat na novinky či sociální sítě, blog a fórum. V tomto případě vidíme již vytvořené projekty, které nemusí být nutně hrou, jsou však uloženy pod složkou Games, protože Stencyl je určen převážně pro tvoření her.

Pro vytvoření nové hry klikneme intuitivně na Click here to create a new game. (viz Obrázek 3)



Obrázek 3 Vytvoření nové hry

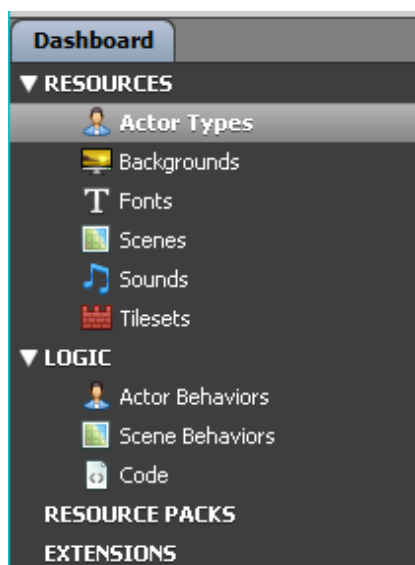
Zde se nám nabídne možnost Blank Game, tedy prázdná hra, nebo můžeme vyzkoušet některý Crash Course, což jsou hry vytvořené vývojáři Stencylu. Ty je potřeba stáhnout a nahrát do správné složky. Při vytváření nové (prázdné) hry (obr. 4) nejprve zadáme název, šířku a výšku okna v pixelech. U názvu je nutné použít pouze písmena, číslice a mezerník. Rozlišení hry lze během tvorby ještě měnit. Také nesmí být použita diakritika.

Obrázek 4 Nová hra - základní informace

3.2 ZÁKLADNÍ NABÍDKA

Po vytvoření nové hry se nám nabídne Dashboard (viz Obrázek 5), základní nabídka. Ta zahrnuje Resources – možnosti, které zahrnují Actor Types, kam vkládáme jednotlivé herce (Actors). V Resources pak dále najdeme Backgrounds, kam vkládáme pozadí pro

jednotlivé scény; Fonts, kde vytváříme různé druhy písma; Scenes, asi nejdůležitější část pro úpravu hrací plochy; Sounds, kam se vkládá zvuk pro různé události a Tilesets, pro vytvoření platformy. Další částí Dashboard je Logic (logika), ve které najdeme Actor Behaviors a Scene Behaviors, kde se nastavuje chování pro herce nebo pro scénu. Toto chování můžeme vytvořit samostatně jako událost nebo je možné použít stažené. Velké množství různých chování lze jednoduše najít v obchodu StencylForge, který je zdarma, je však nutný přístup k internetu. V obchodě se hledá pomocí nabídky v levém sloupci, dále pak podle nejnovějších, nejméně stahovaných, nejužitečnějších, nejlépe hodnocených nebo doporučených. Další způsob je pomocí vyhledávání v pravém horním rohu, kam zadáme klíčové slovo. Chování pak můžeme použít vícekrát pro různé herce či scény. Nakonec je tady možnost Code, která umožní vývojáři vložit klasicky psaný kód. Stencyl také nabízí vložit Resource Packs, neboli balíček možností, které již byly vytvořeny a Extensions, také předem vytvořené rozšíření hry například v podobě reklam.



Obrázek 5 Dashboard

3.3 ZÁKLADNÍ POJMY

Jelikož při vytváření každé hry se objevují základní termíny, je vhodné je nejdříve vysvětlit, neboť tyto termíny budou používány i dále v textu. Jsou to: Actors – herci, Players – hráči, Regions – regiony, oblasti, Tiles – dlaždice. Můžeme si ale vytvořit další sami, např. Enemies – nepřátelé. U scény máme možnost dále upravovat samozřejmě samotnou scénu, dále Behavior – chování; Events – události, které tvoříme skládáním bloků a zadávání konstant; Physics – fyzické vlastnosti, gravitace; Properties – vlastnosti

(např. velikost v pixelech). U herců se může navíc upravovat kolizní ohraničení, tedy například polygon, který bude kopírovat tvar předmětu a vytvářet tak realistickou srážku. Také v nastavení fyziky je více možností. Tyto pojmy budou vysvětleny podrobněji dále.

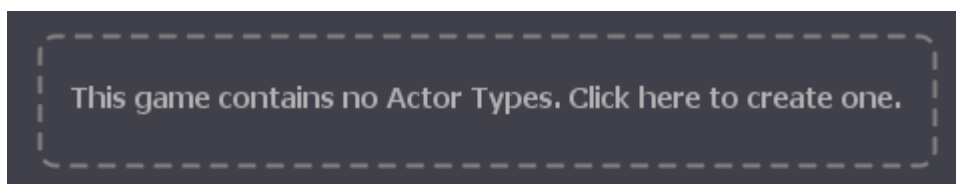
4 POSTUP PRÁCE ZNÁZORNĚNÝ POMOCÍ ANIMACÍ

V této kapitole je obecně naznačeno, jak pracovat s programem Stencyl, jak používat jednotlivé komponenty a základní funkce k vytvoření funkční spustitelné hry.

Pokud máme vytvořen název hry a zadanou šířku s výškou, můžeme přistoupit k samotnému vytváření hry. Na začátku je třeba vytvořit scénu, abychom měli kam umístit herce, dlaždice apod. Kliknutím v Dashboard na položku Scenes se zobrazí hláška: This game contains no scenes. Click here to create one. V překladu: Tato hra neobsahuje žádné scény. Pro vytvoření scény klikneme sem. To je jedna možnost jak vytvořit novou scénu, druhou je kliknutí na zelené tlačítko Create new (Vytvořit novou) v pravé horní části obrazovky. Po vybrání jedné z možností se nám ukáže nabídka, kam je stejně jako u vytváření nové hry nutné zadat název a velikost v pixelech nebo v tiles, což jsou dlaždice, které se zde používají pro vytváření terénu. Každá dlaždice má v základu nastavenou velikost na 32 x 32 pixelů, tu lze ale také libovolně měnit. Jako poslední zde nastavíme barvu pozadí, která nabízí tři možnosti: jakoukoliv barvu z nabídky HSV, HSL, RGB a CMYK; žádnou barvu nebo vertikální prolínání dvou barev, které může simulovat například oblohu. Potvrdíme tlačítkem Create. Nyní se nám otevřelo okno s nově vytvořenou scénou a možnostmi její úpravy (viz Animace č. 1 Vytvoření scény).

4.1 ACTOR TYPES

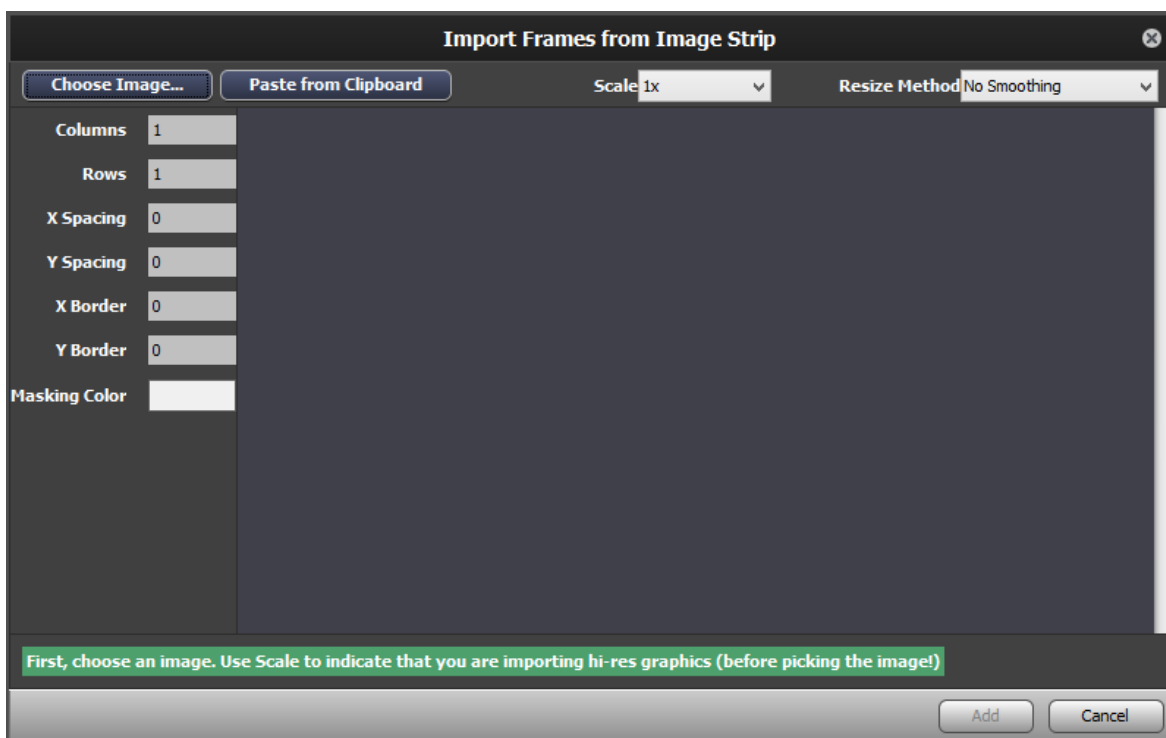
Než se budeme nadále zabývat úpravami scény, vrátíme se do Dashboard a vytvoříme si herce. Kliknutím na Actor Types se otevře složka s herci. My ale zatím žádného nemáme, a tak klikneme na políčko (viz Obrázek 6) a vytvoříme si prvního herce.



Obrázek 6 Vytvoření nového herce

Objeví se okénko Create New, kam zadáme název nového herce a potvrdíme kliknutím na tlačítko Create. Stencyl vytvoří nového herce a automaticky otevře kartu s jeho nastavením. Nový herec ale zatím neobsahuje žádný obrázek, takže musíme nějaký přidat, aby byl herec vidět. Ať chceme vytvořit animaci nebo pouze jeden obrázek, vše se vkládá do Animations. Nemáme žádnou vytvořenou, a tak klikneme na Click here to add

animation. Vytvořilo se nové pracovní prostředí, kde můžeme animaci upravovat. Animace se automaticky pojmenovala Animation 0. Pro větší pořádek si ji můžeme přejmenovat podle funkce, kterou bude zastávat. Může to být například animace chůze, skoku, letu, pádu, atd. Nyní přidáme nějaké obrázky, abychom herce oživil. Kliknutím na Click here to add a frame se otevře nové okénko s podtitulem Import Frames from Image Strip (viz Obrázek 7).



Obrázek 7 Import obrázku

Klikneme na Choose Image... a vybereme obrázek, který chceme pro vytváření herce použít. Názornou ukázkou lze vidět v Animaci č. 2 Vytvoření herce. Nyní jsme zpět v editoru animace, kde můžeme přidávat libovolný počet obrázků tvořících animaci kliknutím na Click here to add a frame. Dvojklikem na obrázek se nám otevře editor samotného obrázku, kde se nastavuje doba v milisekundách, po kterou bude tento jeden obrázek vidět, než ho nahradí další. Tímto způsobem se vytvoří animace, kterou můžeme vidět vlevo nahoře jako nekonečnou smyčku. Pokud zde máme pouze jeden obrázek, bude animace samozřejmě nehybná. Můžeme také obrázek ještě upravit v jednoduchém editoru, který nalezneme pod záložkou Edit image.... V levém dolním rohu můžeme animace buď přidávat tlačítkem + nebo mazat pomocí tlačítka -. Animace se pak tvoří stejným způsobem. Když máme u jednoho herce vytvořeno několik animací, je třeba

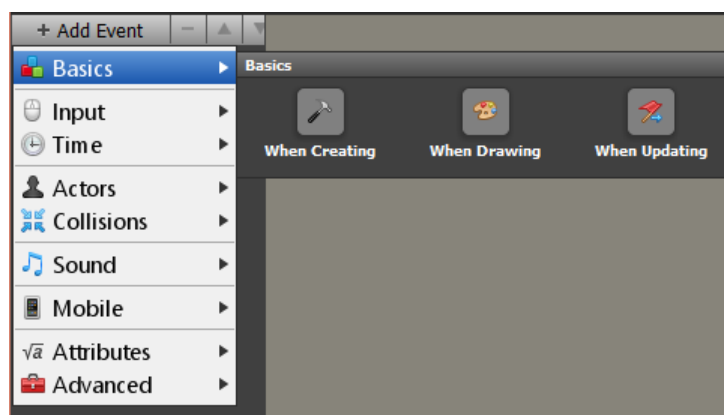
zvolit, která bude použita, až bude herec zasazen do scény. To zajistíme tím způsobem, že vybereme danou animaci a klikneme na hvězdu umístěnou vlevo dole vedle možností přidávání a mazání animací. Hned vedle je ještě tlačítko, kterým je možné jakoukoliv animaci duplikovat. Nyní se budeme věnovat dalším možnostem nastavení a naprogramování herce. Tyto úpravy obrázků a animací jsme provedli pod záložkou Appearance, kterou vidíme uprostřed v horní části obrazovky (viz Obrázek 8).



Obrázek 8 Možnosti herců

Záložka Behaviors umožňuje použít již vytvořené chování (viz Animace č. 3 Přidání staženého chování). Toto chování si můžeme sami vytvořit nebo stáhnout některé ze StencylForge a aplikovat je na více herců zároveň. Vytvoření vlastního chování lze vidět v Animaci č. 4 Tvorba vlastního chování. Můžeme samozřejmě přiřadit více chování. Ta pak můžeme stále ještě upravovat a přizpůsobit si je tak přesně pro naši potřebu, můžeme je deaktivovat či aktivovat nebo úplně smazat.

Nyní se dostáváme k důležité záložce Events, kde se dá říci, že probíhá samotné programování herce. Programovací plocha je rozdělena na tři sloupce. Vlevo je sloupec, kam se přidávají události kliknutím na tlačítko + Add Event, vpravo jsou vytvořené bloky, kde má každý svoji funkci, a jejich skládáním a spojováním v prostředním sloupci tvoříme program. Nejprve je nutné vytvořit událost (viz Obrázek 9), které pak přiřadíme nějakou funkci.



Obrázek 9 Nová událost

Hned jako první se nám nabízí událost When Creating, to znamená, že jakmile bude na scéně vytvořen tento herec, bude mu přiřazena tato část programu. Zde například (viz

Obrázek 10) nastavíme, aby byl tento herec vytvořen na pozici $x - 20$ a $y - 50$, pokud bude na scénu zavolán. Konkrétní ukázka je v Animaci č. 5 Vytvoření herce na pozici. Dále můžeme vytvářet kupříkladu reakci na kliknutí myši, na kolizi dvou hráčů, přidávat hudbu, atd.



Obrázek 10 Nastavení pozice

Tyto dva bloky, které vidíme výše, jsme vzali z pravého sloupce (viz Obrázek 11), jenž nám nabízí veškeré naprogramované bloky, které program Stencyl má. Buď můžeme nahoře použít vyhledávač, nebo zde máme šikově rozdělené funkce pod jednotlivé skupiny.



Obrázek 11 Nabídka programovacích bloků

Další záložkou je Collision, kde se herci nastavuje ohraničení, které bude symbolizovat kolizi s ostatními předměty, ať už jde o další herce, dlaždice či regiony. Máme na výběr z obdélníku, kruhu a polygonu. Do polygonu můžeme přidávat další body a vytvořit tak přesný obrys našeho herce. Můžeme jich přidat dokonce několik. Zpětně je pak mažeme pomocí Delete. Práce s kolizemi je znázorněna v Animaci č. 6 Vytvoření kolize.

Vedle Collision je další důležitá záložka a tou je Physics. V obecném nastavení General upravujeme fyzické vlastnosti, například jestli se herec může hýbat, jestli se může otáčet nebo jestli bude přitahován gravitací. Pokud budeme vytvářet nějakou lidskou nebo zvířecí postavu, určitě se nám gravitace bude hodit. Ne však pro herce typu

vesmírná loď. Dále můžeme nastavovat váhu a úhlovou váhu herce (v Heaviness) a další podrobnější nastavení, které je potřeba pro složitější aplikace.

Poslední záložkou je Properties, kde můžeme změnit název herce, napsat nějaké informace do jeho popisku, a důležitou položkou je Choose Group, tedy vybrat skupinu, do které herec patří.

Nyní máme vytvořeného herce, ale potřebujeme se podívat, s čím vším má náš herec nastavenou kolizi. To uvidíme v Settings → Groups. Herec patří do skupiny Actors. Jak vidíme, tato skupina má nastaveno Collides With: Tiles, Actors (zeleně označené). To znamená, že pokud se hráč potká s jiným hráčem, nastane kolize, stejně tak, když se hráč dotkne dlaždic. Stencil má přednastaveno několik základních skupin, které není možné smazat. Do těchto skupin jsou automaticky přidávány kolize těch komponent, které jednotlivé skupině náleží, například všichni herci jsou automaticky vytvářeni ve skupině Actors. Můžeme také vytvořit novou skupinu (kliknutím na Create New) a vytvořit si tak například skupinu nepřátel, u které pak můžeme vytvořit speciální událost pro kolizi s hráčem (s hlavním hrdinou). Podle potřeby je možné u daného herce v záložce Properties tyto skupiny měnit. Pokud by bylo třeba, je možné přiřadit herci i skupinu Tiles. Nejvíce se ale využijí tyto skupiny, když je ve scéně více herců a potřebujeme nastavit kolize jen mezi některými a s druhými se mýjeli beze srážky. Herec teď může být přidán do scény.

4.2 BACKGROUNDS

Pro vytvoření pozadí je třeba vybrat příslušnou položku v Dashboard a tou je Backgrounds. Opět nemáme žádné vytvořené, a tak klikneme na možnost vytvoření nového. Po zadání jména se nám otevře nastavení pro pozadí. Nemáme vložen žádný obrázek, a tak klikneme na Click here to add a frame a otevře se nám podokno, kde se nám v levém horním rohu zobrazí možnost Choose Image.... Po kliknutí vybereme příslušný obrázek a otevřeme jej v programu. Klikneme na Add a tím ho přidáme do vytvořeného pozadí (viz Animace č. 7 Tvorba pozadí). Pokud chceme, aby se nám toto pozadí měnilo, můžeme přidat další obrázek obdobným způsobem jako u tvoření animací herců. Stejně tak můžeme měnit dobu zobrazení v milisekundách. Pro některé scény se hodí možnost posunu pozadí. Ať už je potřeba posun doprava, doleva, nahoru či dolů, tuhle možnost nám dává Scroll Speed. Nastavuje se v pravém horním rohu obrazovky

a můžeme samozřejmě nastavit i rychlost, s jakou se bude pozadí posouvat, a dokonce i opakování (Repeat Background). Můžeme tak dát do pozadí kupříkladu obrázek jednoho stromu, díky této funkci se bude opakovat a tím máme vytvořené dlouhé, dá se říci nekonečné pozadí. V Properties v horní části uprostřed obrazovky je k dispozici možnost změnit název pozadí a přiřpat k němu popisek.

4.3 TILESETS

Důležitou část tvoří dlaždice neboli Tilesets, kterou nalezneme v Dashboard. Tyto dlaždice tvoří plošinu, po které může herec chodit, jezdit, skákat... Vytvoříme ji kliknutím na This game contains no Tilesets. Click here to create one. nebo na zelené tlačítko Create new v pravém horním rohu. Zadáme název a hned se nám nabídne podokno pro import obrázku. Kliknutím na Choose Image... se otevře průzkumník a my můžeme vybrat obrázek vhodný pro tvoření dlaždic. Můžeme vkládat buď jednotlivé dlaždice, které pak budeme do scény zasazovat, nebo vložíme obrázek a ten se nám rozdělí na určitý počet dlaždic podle zvolené výšky a šířky dlaždice (Tile Width, Tile Height). V takto vytvořeném obrázku je třeba vytvořit ohraničení kolizí pro každou dlaždici zvlášť. Pokud máme pouze rovnou plochu, stačí všude nastavit Square, pokud chceme stoupající či klesající kopec, je nutné vybrat některý z šikmých, většinou trojúhelníkových obrazců. Když je součástí obrázku i „vzduch“, do kterého není žádoucí, aby herec narážel, nastavíme mu No Collision. Detailní postup práce nalezneme v Animaci č. 8 Tvorba terénu. Kdyby nám nestačily předpřipravené obrazce, můžeme některý z nich upravit nebo přidat nový box či polygon. Polygon umožní modelaci jakéhokoliv tvaru přidáváním více bodů, nepovolí však konkávní úhly. I zde je možnost změnit název či napsat popisek záložce Properties.

4.4 FONTS

Při tvoření hry se nezdá, že používá vypisování textu na obrazovku. Pro vytvoření typu písma (fontu) vybereme v Dashboard položku Fonts. Hra nemá zatím vytvořené žádné písmo, klikneme tedy na možnost vytvoření nového (This game contains no Fonts. Click here to create one.). Po zadání názvu se otevře nastavení Font Style, kde je možné libovolně měnit velikost, styl atd. Po kliknutí na Color je k dispozici výběr barvy písma nebo vytvoření gradientu, tedy postupně se prolínající dvě barvy. V Effects je možné ještě

nastavit okraje písma a stín. Použit lze také Font, který máme stažený v počítači. V Properties můžeme stále měnit název písma a připsat k němu nějaký popisek.

4.5 SOUNDS

Pokud vytváříme plnohodnotnou hru, neobejdeme se bez zvukových efektů. To nám v Dashboard umožní položka Sounds. Klikneme na vytvoření nového zvuku (This game contains no Sounds. Click here to create one.) a zadáme název. Zde vidíme všechny možnosti na jedné stránce. Můžeme měnit název, napsat popisek, zvolit, jestli se jedná o zvukový efekt nebo o hudbu, importovat zvukovou stopu ve formátu MP3 nebo OGG. Formát MP3 je určen pouze pro hry tvořené pro Flash, pro ostatní jako je Android, iOS atd. je určen formát OGG. Je zde i možnost upravovat zvukovou stopu (Edit MP3 in External Editor nebo Edit OGG in External Editor), k tomu však potřebujeme jiný program. Editor hudby není ve Stencylu vestavěn, jako je tomu při upravování obrázků. Stencyl sám doporučuje program Audacity. S takto připravenou hudbou pak můžeme dále pracovat, přiřazovat ji k různým událostem ve scéně či u herců nebo v chování.

4.6 SCENES

Nyní se vrátíme do již připravené scény přes Dashboard → Scenes a dvojklikem na vybranou scénu ji otevřeme. V pravé dolní části obrazovky vidíme vrstvy neboli Layers, které je dobré použít pro odlišení vrstev dlaždic, herců či pozadí. Můžeme je libovolně přidávat či mazat, ale také si je například přejmenovat nebo zneviditelnit, pokud chceme lépe vidět jinou vrstvu. Podívejme se též do pravého sloupce, kde se nám nahoře v paletě (Palette) ukázaly vytvořené dlaždice (Tiles) a herci (Actors). U dlaždic můžeme označit jakýkoliv počet dílků a následně je vložit do scény, aby se nám vytvořila platforma, po které bude moci herec chodit. Pokud jich máme vytvořeno více, můžeme mezi nimi měnit v prostřední části sloupce pojmenované Tilesets. Podrobný postup je ukázán v Animaci č. 9 Umístění terénu na scénu. Herce pak uvidíme zpět v paletě, kde překlikneme na záložku Actors. Tam jen vybereme, kterého chceme použít do scény a umístíme jej. Nalevo od scény vidíme lištu s nejdůležitějšími funkcemi. Úplně nahoře je černá šipka umožňující výběr (Select) čehokoliv na scéně. Můžeme tak chytit herce či dlaždici a přesunout na nové místo, herce můžeme také libovolně transformovat. Další možnosti

je pero (Pencil), které zajišťuje vkládání herců nebo dlaždic. Další důležitou možností je přidání sloupce (Insert Column), který nám rozšíří hrací pole vlevo či vpravo o sloupec výšky scény a šířky 32 pixelů, kterou jsme zadávali při vytváření scény. Na tomto políčku lze vidět malá černá šipka, to znamená, že ho můžeme rozkliknout pravým tlačítkem u myši a zobrazí se nám další možnosti. Těmi jsou přidání řádky nahoru či dolů v šířce vytvořené scény a délce 32 pixelů a vymazání řádku či sloupce. Joints slouží ke spojování či pro konstantní posun různých objektů na scéně. Další zajímavou funkcí je přidávání regionů (Add Region), které mohou být ve tvaru obdélníku, kruhu nebo polygonu. Vytvoření regionu je předvedeno v Animaci č. 10 Tvorba regionu. Regionu pak můžeme přiřadit funkci či událost vázanou například na moment, kdy do něho vstoupí herec. Při tvoření scény regiony vidíme, při spuštění hry však viditelné už nejsou. Samozřejmostí je pak přiblížení a oddálení scény pro lepší viditelnost detailů. Dalšími funkcemi jsou například sejmutí obrazovky (Take Screenshot) nebo zobrazení mřížky (Show Grid).

Pokud se posuneme na záložku Behaviors (v horní části obrazovky), existuje možnost každé scéně přiřadit nějaké chování stejně, jako tomu bylo u herců. V záložce Events tvoříme samotný kód, který bude náležet dané scéně. Když chceme scéně nastavit gravitaci, je třeba kliknout na záložku Physics. Tam je možnost výběru horizontální (pravé či levé) a vertikální (horní či dolní) gravitace. Pro simulaci zemské gravitace nastavíme dolní gravitaci (Down) na 85. Pro změnu názvu scény, výšky a šířky či barvy pozadí je zde opět záložka Properties.

5 PRAKTICKÉ PŘÍKLADY

Tato kapitola je věnována praktickým příkladům, ve kterých bude znázorněno jak vytvořit hratelnou hru za použití jedné, následně více scén. Bude zde několik informací o obchodu StencylForge, o možnosti rozpohybovat herce, jak používat nastavení gravitace. Použije se názorná ukázka více skupin a kolize mezi nimi.

5.1 JEDNODUŠŠÍ HRA

5.1.1 CÍL

Cílem této hry je naučit se správně používat skládání bloků při programování, používat chování (Behaviors), rozpohybovat herce, ukázat si možnosti scény.

5.1.2 POPIS HRY

V této hře se objevují základní prvky pro vytvoření funkční hry. Hlavním hrdinou bude jakýsi balón, se kterým bude možné pohybovat doprava, doleva a skákat směrem nahoru pomocí šipek. Hráč se nadále bude muset vyhýbat nepřítelům pomocí skoků, snížení rychlosti apod. Hráč je také dále nucen dojet do cíle dříve, než bude předešnán nepřítelem v podobě ptáka.

5.1.3 ŘEŠENÍ

Při tvoření hry ve Stencylu nestačí jen programovat, je nutné mít připravený grafický podklad, se kterým budeme pracovat. Ten je třeba vytvořit v některém z vektorových či bitmapových editorů, jako jsou Inkscape nebo Gimp. Pro tuto hru je třeba vytvořit balón, nepřítele v podobě ptáka a kotoučové pily, mraky a stromy jako pozadí a dlaždice jako platformu, po které se balón bude moci kutálet či odrážet se z ní.

Klikneme pro vytvoření nové hry, zadáme název a zvolíme šířku 640 pixelů a výšku 480 pixelů. Při tvoření herce už víme jak přidat obrázek, vložíme tedy připravený balón, který je potřeba mít ve formátu PNG, aby se nám vytvořil kulatý herec a netvořily se bílé rohy, jako by tomu bylo například u formátu JPG. Aby se mohl balón přirozeně točit, musíme mu nastavit kolizi ve tvaru kruhu tak, že smažeme automaticky vytvořenou kolizní hranici ve tvaru čtverce/obdélníku a nahradíme ji kruhem (Add Circle). Pozici X (vlevo) a pozici Y (nahore) necháme na nule a Radius nastavíme na 13, neboť máme balón o velikosti 26 x 26 pixelů a potvrdíme tlačítkem OK. Když znovu klikneme na vytvořenou kolizi, můžeme ještě pozice x, y či radius ještě měnit. Aby se mohl míč točit a byl

přitahování gravitací, klikneme v záložce Physics → Can Rotate? na Yes a v Affected by Gravity? také na Yes. Aby se mohl balón hýbat, je nutné naprogramovat události pro stisknutí kláves pohybu (šipka vpravo, vlevo a nahoru). V záložce Events vytvoříme událost kliknutím na Add Event → Basics → When Updating. Vytvořil se nám blok always (vždy, viz Obrázek 12), kam máme možnost vložit nový blok z pravého sloupce. V kartě Flow → Conditions najdeme podmínku if, kterou si přesuneme do pracovní plochy (uprostřed), abychom s ní mohli dále pracovat. Aby se nám herec pohyboval směrem vpravo, potřebujeme nejdříve zadat, na jakou událost bude reagovat. V kartě User Input → Keyboard & Mouse najdeme zelený blok Control i down, který si přesuneme do pracovní plochy a Control nastavíme na right. Right znamená, že bude reagovat na šipku vpravo. Ve výchozím nastavení každé hry vytvořené ve Stencylu, které můžeme najít v Settings → Controls, je totiž nastaveno ovládání pro šipku vpravo, vlevo, nahoru, dolů, pro Enter a pro klávesy Z a X. Dále v tomto bloku necháme is down (je stisknuto) a vložíme jej přímo do podmínkového bloku if, kam vidíme, že nádherně zapadá i tvarem. Z karty Actor → Motion vybereme blok set x-speed to 0 for Actor a přesuneme jej do podmínky if přesně tam, kam podle tvaru zapadá. Políčko s číslem přepíšeme na 30, zbytek necháme nastavený tak, jak je. Tímto jsme vytvořili rychlost 30 pro sebe (herce, kterého máme zrovna otevřeného) po x-ové přímce. Z karty Flow ještě přidáme podmínku otherwise if a vložíme stejné bloky, jen u ovládání pravé šipky změníme is down na was released a x-speed nastavíme na 0. To nám zajistí, že když hráč stiskne a poté uvolní příslušnou klávesu, balón se zastaví. Obdobným způsobem si vytvoříme pohyb doleva a nahoru. Pohyb nahoru bude po ose Y a bude zadávána záporná hodnota.



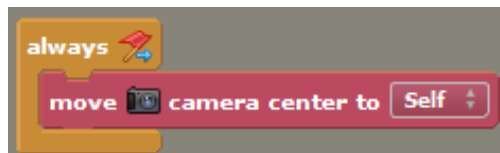
Obrázek 12 Pohyb

Pro tohoto herce si vytvoříme chování v Dashboard v sekci Logic pod záložkou Actor Behaviors. Klikneme na příslušné místo pro vytvoření nového a pojmenujeme jej Pohyb kamery. Nyní se otevřelo okno s nově vytvořeným chováním a my mu přiřadíme jednoduchou událost. V záložce + Add Event klikneme na Basics (základní) a následně na When Updating. Opět se nám vytvořil blok always, kam máme možnost vložit nový blok. Jsou dvě možnosti výběru bloku: vybrání záložky Scene a následně View (viz Obrázek 13), kde si vyhledáme příslušný blok, nebo zadáme do vyhledávače klíčové slovo v angličtině (např.: move nebo camera) a nabídne se nám úzký výběr bloků.



Obrázek 13 Scene

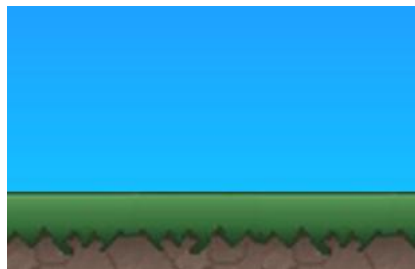
Není již těžké dohledat požadovaný move camera center to Self, kde necháme vybraný Self a vložíme jej do připraveného bloku always (viz Obrázek 14).



Obrázek 14 move camera

Kliknutím na Save Game se nastavené chování uloží. Nyní jej nalezneme v Actor Behaviors a můžeme s ním nadále pracovat. Pro přidání chování herci otevřeme již vytvořený balón, klikneme na záložku Behaviors a vlevo dole na + Add Behavior, kde se nám v podokně zobrazí chování Pohyb Kamery. Vybereme jej a potvrdíme. Takto je chování přiřazeno herci a ten jej od teď používá. Abychom si mohli vyzkoušet funkčnost, je potřeba ještě vytvořit dlaždice, po kterých se bude moci míč kutálet. V Dashboard → Tilesets vytvoříme nové dlaždice vhodně zvolenými obrázky rovné země a kopce, kterým nastavíme správné kolizní hranice. V Dashboard → Scenes přejdeme k vytvoření scény, kde si zvolíme výšku 480 pixelů, šířku 2000 pixelů a barvu pozadí jako Vertical Gradient. Pro simulaci nebe zvolíme první barvu modrou a druhou tyrkysovou. Vidíme, že scéna je široká tak, že se na

monitor nemusí vejít. Při vytváření hry jsme však nastavili šířku 640 pixelů, a proto když bychom zkusili spustit hru pomocí tlačítka Test Game vpravo nahoře, zjistili bychom, že velikost okna spuštěné hry je pořád 640 x 480 pixelů. 2000 pixelů je pouze velikost scény, se kterou můžeme pracovat. Abychom měli hru přehlednější, klikneme v pravém sloupci v kartě Layers na tlačítko + (dole) a vybereme New Tile Layer, čímž se nám vytvoří nová vrstva, kterou pojmenujeme například Zem. Do této vrstvy budeme přidávat pouze dlaždice tvořící zem, po které se balón kutálí. V Paletě máme zobrazeny dlaždice, které jednoduše vybereme chycením myši a poté kliknutím na scéně vytvoříme zemi (viz Obrázek 15).



Obrázek 15 Zem

Přidáme další vrstvu, kterou nazveme například Balon, a v paletě klikneme na záložku Actors a vybereme balón. Ten umístíme do scény tak, aby byl těsně nad zemí. V základním nastavení programu je definováno, že herci (Actors) kolidují s dlaždicemi (Tiles). Můžeme se o tom přesvědčit v Settings → Groups (Skupiny). Zeleně označená políčka znamenají, že vybraná skupina s těmito koliduje. Abychom simulovali zemskou gravitaci, nastavíme 85 v záložce Physics pro Gravity (Vertical), Down. V tomto momentě, když spustíme hru (tlačítkem Test Game), ověříme si funkčnost hned několika věcí. Po zmáčknutí klávesy vpravo se nám míč rozjede, když klávesu uvolníme, míč se zastaví. Pokud budeme klávesu držet déle, míč se bude kutálet dále napravo, a protože máme nastavený pohyb kamery podle středu herce, posune se nám kamera tak, že na herce stále uvidíme. Ověřili jsme také neměnnou velikost okna a větší šířku scény. Když však budeme pokračovat až na konec, balón se nám na kraji nezastaví a ze scény zmizí. V obchodě zvaném StencylForge (viz Obrázek 16) můžeme nalézt užitečné pomůcky pro programování. Jsou to již vytvořená chování, která můžeme zdarma stáhnout a použít. Nejlepším způsobem je zadat do vyhledávání požadovanou funkci (nebo její část) v angličtině. Pro naši situaci se

hodí Cannot Exit Screen. Ukáže se nám několik výsledků, kde každý z nich je možné stáhnout a vyzkoušet, ale většinou jsou všechny stejné či velice podobné.



Obrázek 16 StencylForge

Pokud máme nějaké chování stažené, přejdeme do chování herce (Behaviors) a vlevo dole klikneme na + Add Behavior, najdeme stáhnuté chování a vybereme jej tlačítkem Choose. Nyní je chování přiřazeno herci, a když spustíme hru a vyzkoušíme pohyb do strany, balón už obrazovku neopustí.

V Dashboard → Backgrounds si vytvoříme 2 pozadí. Klikneme pro vytvoření nového a pojmenujeme jej Stromy. Pro toto pozadí si vytvoříme obrázek s několika stromy různé velikosti stojícími vedle sebe bez dalšího pozadí. Vložíme tento obrázek ve formátu PNG do vytvářeného pozadí a v pravé části zaškrtneme Repeat Background?. Tak docílíme, že v širší scéně se bude tento obrázek opakovat stále za sebou a budeme tak mít neustále v pozadí stromy. Druhé pozadí nazveme Mraky a použijeme obrázek s několika mraky vedle sebe, nejlépe výšky 480 pixelů (stejně jako scéna), abychom věděli, v jaké výšce mají mraky přibližně být (opět ve formátu PNG). Zaklikneme Repeat Background? a ve Scroll Speed nastavíme 1.0 pro Horizontal Scroll Speed. Výchozí nastavení horizontálního posunu je vlevo (Left). Tímto nastavením zajistíme, že se budou mraky pomalu posouvat vlevo a hra bude vypadat realističtěji. Přesuneme se opět do scény, kde v pravém dolním rohu vytvoříme novou vrstvu. Tentokrát použijeme New Background Layer (vrstva pro pozadí) a vybereme pozadí Stromy. Nově vytvořenou vrstvu si pojmenujeme stejně jako pozadí, tedy Stromy. Obdobně vytvoříme novou vrstvu, vložíme pozadí a přejmenujeme vrstvu pro Mraky. Když otestujeme hru, uvidíme, že se mraky hýbou, a když pohneme hercem směrem doprava, uvidíme opakující se pozadí.

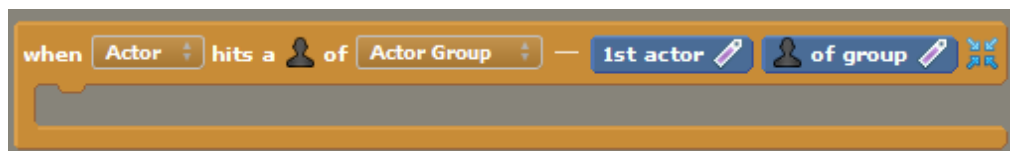
Dalším vylepšením je přidání tzv. Enemies (nepřátel). V Settings v sekci Groups klikneme na zelené tlačítko Create New a pojmenujeme Enemies. Generovala se nová skupina, které vytvoříme kolizi s herci (Collides With → Actors). V Dashboard → Actor Types vytvoříme nového herce s názvem Ozubene Kolo. Pro simulaci otočného kola

vytvoříme animaci, kde budou dva obrázky s různě natočeným kolem, a každému nastavíme délku trvání 150 ms (viz Obrázek 17).



Obrázek 17 Animace

V záložce Collision smažeme předpřipravenou kolizní hranici a nahradíme ji kruhovou. Dále v kartě Physics nastavíme, aby se herec nemohl pohybovat (What kind of Actor Type? → Cannot Move) a v Properties vybereme skupinu Enemies (Choose Group → Group → Enemies). Několik těchto herců pak vložíme do scény a následně se přesuneme (ve scéně) do záložky Events, kde nastavíme herci událost, která nastane po střetnutí s ozubeným kolem. Při vytváření nové události si vyhledáme Collisions → Any actor collides with... → Member of Group. Objeví se podmínka when (viz Obrázek 18) znázorňující kolizi určitého herce s jakýmkoliv hercem ze skupiny, kterou vybereme. Zvolíme herce Balon a pro políčko skupiny vybereme Enemies.

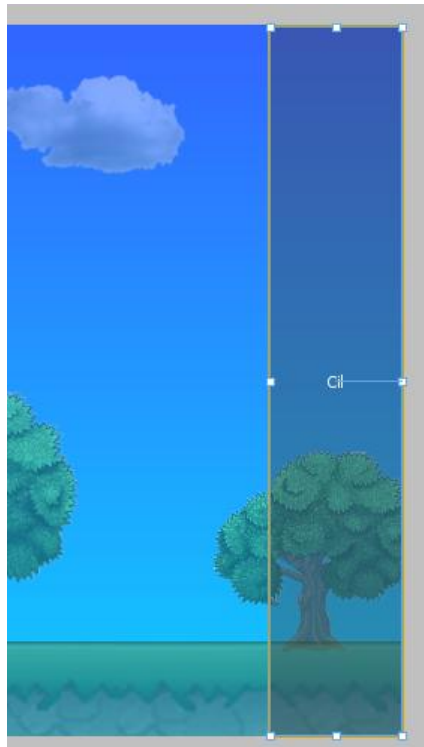


Obrázek 18 Kolize hráče se skupinou

V této chvíli je třeba vytvořit atribut v záložce Attributes → Getters. Kliknutím na Create an Attribute se nám otevře podokno, kam napíšeme název atributu. Pojmenujeme jej Zabit? a nastavíme typ Boolean. Co se týče atributů, nabízí se zejména vytvoření klasického atributu, znázorněného modrou barvou. Tyto atributy fungují v rámci jedné scény nebo jednoho herce. V rámci této hry si s těmito atributy vystačíme, pokud bychom však chtěli použít atributy v rámci více scén, s těmito bychom si nevystačili a museli bychom použít globální atributy (Game Attributes) znázorněné fialovou barvou. Defaultně jsou veškeré atributy typu boolean nastaveny na hodnotu false. V záložce Setters vybereme vytvořený blok, vložíme jej do podmínky a nastavíme hodnotu prázdného políčka na true. Jednou možností je kliknout přímo na toto prázdné políčko

a v Comparison → Booleans (Yes/No) vybrat true, druhou je vyhledat tento blok pod možností Flow. Dalším bodem v této události bude naprogramování zabití neboli zmizení herce. Vyhledáme si kill Actor, vložíme do podmínky a nastavíme herce jako Balon. Po kolizi s hráči ze skupiny Enemies nám tedy zmizí balón, a abychom znovu načetli scénu s novým míčem, vyhledáme slovo reload a vložíme blok reload and Fade Out for 2 secs using (vybereme černou barvu) and Fade In for 2 secs using (vybereme černou barvu). Takto nám po zmizení balónu scéna zčerná a objeví se znovu na začátku s novým míčem. Druhou možností je použití reload and Crossfade for 2 secs. To by znamenalo, že se scéna načte znovu během dvou vteřin, avšak bez zčernání obrazovky. Pro využití vytvořeného atributu vytvoříme novou událost Basics → When Drawing. Vložíme podmínku if, do prázdného pole vložíme atribut Zabit? a v sekci Drawing vybereme první položku draw text, kam zadáme text informující hráče o nastalé situaci a umístíme jej na pevnou pozici X a Y.

Je na každém z programátorů, jak vytvoří hru složitou a hratelně zajímavou. Připravené dlaždice mu dávají volnou ruku, stejně tak nepřátelé, které může do scény umisťovat neomezeně, až dokud nedosáhne konce. Zde je potřeba vytvořit cíl, do kterého když se herec/hráč dostane, zobrazí se hláška „Vítězství“. Tuto akci vytvoříme pomocí regionu, vloženého na samý konec scény po celé výšce a pojmenujeme jej Cil (viz Obrázek 19).

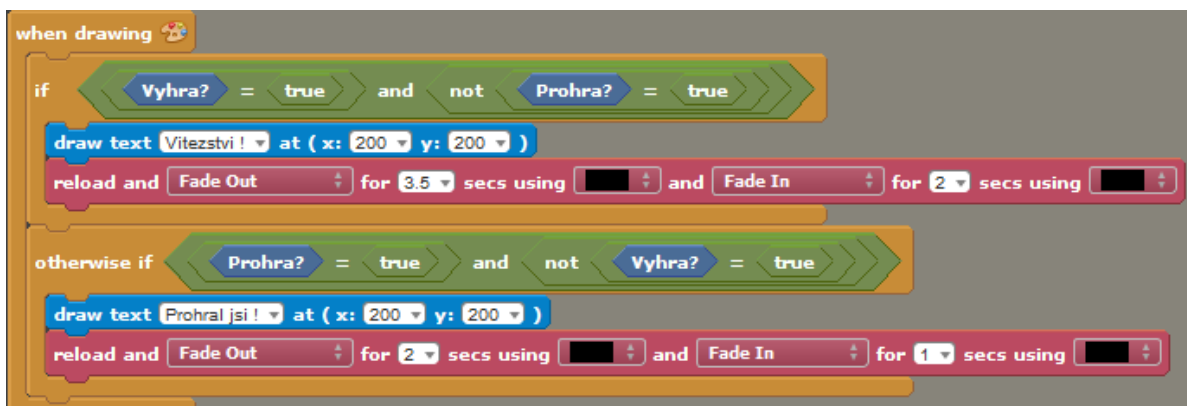


Obrázek 19 Region - Cil

V Events vytvoříme další atribut datového typu boolean, tentokrát s názvem Vyhra? a vložíme blok set Vyhra? to true do nové události Actors → Enters or Leaves a Region → Specific Actor. Vybereme herce Balon → enters → region Cil, to znamená, že když se herec Balon dostane do regionu Cil, nastaví se atribut Vyhra? na true. V nové události typu When Drawing vytvoříme podmínku if Vyhra? = true, vložíme do ní draw text s hláškou „Vitezstvi !“ a následně blok z karty Scene → Game Flow. Pro znovunačtení scény použijeme jednu z možností reload and... Pokud bychom si připravili další scénu (další level hry), nabízí se nám switch to Scene, kde bychom vybrali již zmíněnou druhou scénu. V tomto okamžiku se při dojetí do cíle zobrazí hláška „Vitezstvi !“ a scéna bude znovu načtena.

Pro ještě zajímavější a hráčsky náročnější hru si vytvoříme nového herce se jménem Ptak s příslušným obrázkem či ještě lépe animací. Vytvoříme mu stejnojmennou skupinu v Settings → Groups, kde nastavíme, aby nenastala kolize s žádnou z ostatních skupin a herci tuto skupinu v Properties přiřadíme. Na scéně mu pak vytvoříme novou vrstvu se stejným názvem a herce umístíme do levého horního rohu scény. Herci nastavíme událost Input → Universal → Any Key, kde mu nastavíme rychlost na 25, tedy o trochu méně než herci Balon (set x-speed to 25 for Self). Takto poletí v horní části

obrazovky pták a hráč bude nucen dokončit daný level v kratším čase a s menší dobou na rozmyšlení při vyhýbání se překážkám/nepřátelům. Ve scéně je zapotřebí vytvořit nový atribut typu boolean s názvem Prohra? a v nové události when Actor enters Region (tím máme na mysli Ptak enters Cil) nastavíme Prohra? na true. Do již vytvořené události Drawing vložíme další podmínku otherwise if s bloky draw text „Prohral jsi !“ a reload. Aby nenastala situace, kdy se dostanou do cíle (region Cil) Balon i Ptak současně a zobrazí se hlášky o vítězství i prohře najednou, bude program vypadat takto (viz Obrázek 20).



Obrázek 20 Vyhra/Prohra

5.2 SLOŽITĚJŠÍ HRA - HOROLEZEC

5.2.1 CÍL

V této složitější hře se naučíme především použití více scén a propojení mezi nimi. Nepostradatelnou součástí je použití globálních atributů, které umožní uložení proměnných, používaných ve více scénách.

5.2.2 POPIS HRY

Tato hra bude simulovat horolezce, který se snaží dostat na vrchol hory za použití správného vybavení a obsahu batohu, důležitého pro přežití a úspěšný výstup na vrchol.

Výstup na vrchol hory bude rozdělen do tří částí. Před každou z nich si hráč bude moci vybrat, jaké vybavení si s sebou vezme, s tím, že bude mít informaci o počasí před každou částí výstupu. Pokud vybere špatně, vypíše se mu příslušná hláška a bude moci vybírat znovu. Pokud vybere správně, bude mu umožněn postup k dalšímu záchytnému bodu. Než se tam ale dostane, bude muset projít/vylézt přes strastiplnou cestu, kde na něj budou čekat nemilé překážky.

Hráč bude mít na výběr z věcí: triko s krátkým rukávem, triko s dlouhým rukávem, bundu, rukavice, boty, mačky, malá lahev s pitím, velká lahev s pitím, termoska, jídlo, kraťasy, dlouhé kalhoty, čepice, cepíny, kšiltovka, helma, pantofle, pláštěnka. Některé z těchto věcí jsou úplně zbytečné. Po výběru věcí a potvrzení se vypíše hláška, že chybí něco důležitého, popřípadě, že obsahuje nějaké zbytečné věci pro danou část výstupu.

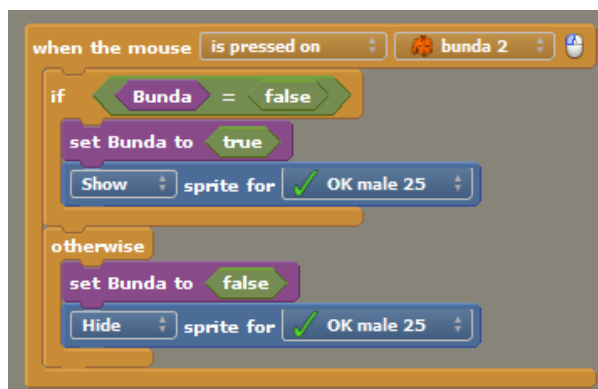
Po úspěšném výběru věcí bude moci hráč ovládat horolezce, jenž se bude vyhýbat překážkám, které mu znemožňují výstup, a bude muset uhýbat padajícím kamenům, lavinám, mít se na pozoru před medvědy apod. Když dorazí v pořádku do dalšího stanoviště, bude muset opět vybírat věci, které s sebou vezme s ohledem na danou část hory a počasí. Takto bude pokračovat až na vrchol kopce.

5.2.3 ŘEŠENÍ

Jelikož je tato hra složitější na řešení, musíme si práci předem rozvrhnout a mít dobrou představu o tom, jak má ve výsledku vypadat. Důležitou součástí takové hry je grafický podklad, který je třeba si opět předem připravit. S vytvořenými herci, dlaždicemi a pozadím pro scény pak už můžeme nadále pracovat. Je potřeba si vytvořit několik scén. Jako první z nich vytvoříme úvodní obrazovku, která hráče uvítá ve hře a sdělí mu prvotní informace o tom, co má dělat, tedy podle počasí si vybrat správné vybavení z obchodu, aby mohl vyrazit na cestu. Další scénou bude obchod, ve kterém se bude vybírat správné vybavení na následující část výstupu. Zapotřebí bude scéna s počasím a také scéna s herní částí pro samotný výstup na vrchol hory. Jelikož má hra 3 úrovně, bude potřeba mít tyto 4 scény vytvořené pro každou z nich. Celkem tedy 12 scén a jedna pro situaci, kdy hráč dokončí celou hru a bude mu nabídnuto začít hrát znovu od začátku.

Na scénu s úvodní obrazovkou umístíme 3 regiony na místa, kam má hráč kliknout, aby se dostal do obchodu s oblečením a potravinami, do místnosti s předpovědí počasí nebo aby mohl vyrazit na cestu. Pro lepší přehlednost regiony pojmenujeme tak, aby dávaly smysl (Obchod, Předpoved pocasi...) a v kartě Events vytvoříme tři události when the mouse is pressed on Region. U každé vybereme jeden z regionů a naprogramujeme switch to scene, kde zvolíme scénu příslušící regionu. V tomto momentě se z úvodní obrazovky dostaneme do jakékoliv ze tří zbylých scén. Pro scénu s předpovědí počasí stačí přidat tlačítko OK naprogramováním události pro přechod na úvodní obrazovku po kliknutí na region v místě potvrzování. Ve scéně s obchodem je nutné taktéž vytvořit

potvrzovací tlačítko, programování zde však bude o mnoho více. V jednodušších hrách s jednou scénou se vytváří klasické atributy, pokud ale tvoříme hru složitější a s více scénami, je nutné vytvářet tzv. Game Attributes, které je možné použít i v rámci více scén. Vytvoříme si tedy několik Game Attributes (tlačítko Create New Game Attribute) typu boolean s názvem Boty, Bunda, Termoska, Cepin, atd. Dáme vzniknout také potvrzovací fajfce v podobě herce, kterého umístíme do scény ke každému produktu v obchodě. Aby fajfky při vstupu do scény nebyly vidět, zvolíme událost when created a všem fajfkám nastavíme Hide sprite for (Choose Actor). Pro každý produkt v obchodě je potřeba naprogramovat událost po kliknutí na něj. Bude to when the mouse is pressed on (Choose Actor), kde zvolíme jako herce například Bunda a vložíme podmínku if a otherwise. Když bude Bunda false (což je nastaveno defaultně) nastaví se na true a ukáže se fajfka, pokud bude true, fajfka se schová (viz. Obrázek 21).



Obrázek 21 Show/Hide

Takto naprogramujeme každý produkt v obchodě. Přesuneme se zpět do scény s úvodní obrazovkou do události kliknutí na region umožňující vyrazit na cestu. Zatím by stačilo, aby hráč na region klikl, a bude puštěn dále. My ale chceme, aby mu byl další postup dovolen pouze v případě, kdy vybere v obchodě správné věci pro výstup. Proto vytvoříme podmínku if, do které vložíme programovou část s globálními atributy. Nastavíme, aby když Boty = true and Kratasy = true and Tricko = true and not Bunda = true and not Macky = true, bude umožněn postup dále. Tyto atributy si nastavíme podle toho, co se nám hodí do této úrovně a co je pro tuto část výstupu zbytečné. Vytvoříme nový atribut datového typu boolean s názvem Spravny vyber?. Do této podmínky dále vložíme set Spravny vyber? to true a switch to scene, kde vybereme scénu Level 1. Pod tuto podmínku vložíme otherwise a do něj set Spravny vyber? to false. Do nové události when drawing vložíme

podmínku `if Spravny vyber? = false`, uvnitř které se bude nacházet `draw text` s hláškou „Vyber spravne veci z obchodu.“). V této chvíli bude hráč vpuštěn do první úrovně pouze tehdy, když vybere správné věci z obchodu, jinak vyběhne hláška a bude muset vybírat znovu. Co se týče dalších úrovní, budou vypadat podobně, ve scénách se zastávkou mezi úrovněmi budou změny ve výběru potřebných věcí na další část výstupu, tedy i v podmínce, která určuje, jestli hráče pustí na další výstup nebo ne a také změníme regiony v událostech nastávajících po kliknutí na daný region. U obchodu se nám přímo nabízí možnost `Duplicate` (zkopírovat), která se nám ukáže při stisku pravého tlačítka na myši, když máme kurzor na scéně s obchodem. Zde je též třeba změnit, do jaké scény se hráč dostane po kliknutí na tlačítko `OK`. Do každé ze tří úrovní je také třeba dát na konec region, do kterého když se hráč/herce dostane, přepne se do zastávky mezi úrovněmi, pokud již půjde o poslední level, zobrazí se scéna s hláškou o vítězství a možnost začít hru znovu od počátku. Vytvoříme tedy opět region, který bude odkazovat na úvodní obrazovku. Nyní jen stačí vytvořit překážky v herních scénách dle fantazie tak, že vytvoříme skupinu `Enemies` a herce této skupiny (kameny, výčnělky skály, které horolezec nebude schopen překonat) vložíme do scény. Těmto hercům nastavíme, aby se nemohli hýbat (`Physics → Cannot Move`) a aby tato skupina kolidovala se skupinou `Actors`, ve které je herec `Horolezec`. `Horolezci` pak nastavíme, aby se nemohl otáčet (`Can Rotate? → No`) a aby na něj neměla vliv gravitace (`Affected by Gravity? → No`). V sekci `Behaviors` přidáme chování `8 Way Movement` a `Cannot Exit Scene`, což horolezci umožní pohyb v osmi směrech a nebude moci zmizet ze scény. V chování `8 Way Movement` nastavíme ovládání šipkami a rychlost pohybu `15.0`. Aby hráč neměl výstup k vrcholu hory tak jednoduchý, vytvoříme v herních scénách jednotlivých úrovní padající kameny či led. Při vytváření těchto herců naprogramujeme v události `When Updating set y-speed to 15 for Self` a `set turning speed to 500 for Self`. V momentě, kdy se na scéně objeví tento herec, bude padat dolů rychlostí `15` a otáčet se kolem svého středu rychlostí `500`. To simuluje padající kameny či led, odlamující se od skály (ledovce). Aby tyto úlomky mohly padat přes výčnělky skály, ale pokud by strefily herce, nastala kolize, je třeba vytvořit novou skupinu, která bude mít nastavenou kolizi pouze se skupinou `Actors` a v sekci `Properties` ji těmto padajícím úlomkům přiřadit. Ve scénách s horolezcem vystupujícím na vrchol je třeba vytvořit událost `when created`, kam ze záložky `Flow → Time` vložíme blok `do every _seconds` a nastavíme `4` sekundy. Uvnitř bude `shake screen for 0.5 sec with`

intensity 1% a create (padající úlomek) at (x:_ y:_). Jako X nastavíme random number between 1 and 639. Souřadnici Y nastavíme y-center of Horolezec - 400. Nyní se každé 4 vteřiny lehce zatřese obrazovka a objeví se úlomek ve výšce 400 pixelů nad horolezcem v náhodné šířce. Pro kolizi padajícího úlomku s horolezcem nastavíme v nové události Collisions → Collisions between → Type & Type znovunačtení scény (reload).

ZÁVĚR

Hlavním cílem této práce bylo seznámit se s programem Stencyl, uvést jeho historii a možnosti využití, pomocí animací ukázat primární funkce a na praktických příkladech pak předvést práci s tímto vývojovým prostředím.

V úvodní části jsem se věnoval tomu, co program Stencyl představuje, jakým způsobem se s ním pracuje a jaké je jeho využití napříč různými platformami, či kde se dá takovýto typ vývojového prostředí nejlépe využít. Zmínil jsem alternativy, tedy programy, které bychom zařadili do stejné kategorie. Uvedl jsem, v jakém jazyce je program Stencyl vyvíjen, z jakého důvodu je to výhodné a jaké přední společnosti světa jej též používají. V krátké historii tohoto programu jsem zmínil hlavní změny, které za tu dobu byly provedeny, a také jaká vylepšení nese od své nejnovější verze 3.0.

Během zpracování tohoto tématu mi byly největším pomocníkem stránky samotného Stencylu, jelikož v knižní podobě bylo o tomto programovacím prostředí vydáno velice málo, a už vůbec ne k dostání v České republice, natož s českým překladem. Pro vypracování tohoto tématu byla tedy nezbytnou podmínkou znalost anglického jazyka a samozřejmě přístup k internetu.

V další části jsem se věnoval popisu prostředí, tedy uvítací obrazovce programu, vytvoření první hry a základní nabídce. V té se objevují základní pojmy nezbytné pro další práci s programem. Postupně jsem rozebral, k čemu se jednotlivé možnosti používají a jak s nimi při vytváření hry naložit.

Jednotlivé frakce jsem poté v další kapitole rozebral podrobněji a některé postupy ukázal na přiložených souborech v podobě animací. Jednalo se hlavně o vytváření herců, pozadí, dlaždic a scén, o vytvoření prvních programovatelných událostí a bloků představujících programovou část vytvářené hry. Ukázal jsem také jak vytvořit písmo nebo zvuk.

V prvním praktickém příkladu jsem ukázal, co vše je zapotřebí k vytvoření jednoduché hry s využitím jedné scény, několika herců, jak se používají vrstvy, dlaždice, pozadí, kolize, jak využít obchod StencylForge, jak použít chování a vytváření skupin potřebných k nastavení kolizí mezi nimi. V druhém, složitějším praktickém příkladu je již vidět práce s více scénami a propojení mezi nimi pomocí regionů. Použil jsem také

složitější programování v podobě globálních atributů potřebných k uložení nastavení při přechodu mezi různými scénami. Bylo zapotřebí vytvoření více skupin pro kolize mezi různými typy herců.

RESUMÉ

The main goal of this thesis was to familiarize with a program Stencyl, present its history and options of using, show primary functions by using animations, and to demonstrate the work with this development environment on some practical examples.

In the introductory part I dealt with introducing the program Stencyl, with the way the program works, and with application this program in the different platforms or with the best application of this kind of developmental environment. I mentioned some alternatives of programs, which we would put into the same category. I stated a language of the program Stencyl, advantages of it and the leading world companies, which also use this program. In short history of this program I mentioned main changes, which were made lately, and the improvements from the newest version 3.0.

During a processing this topic I used websites of Stencyl the most, because there are not many books about this developmental environment in the Czech Republic, much less in the Czech language. Therefore, to elaborate this topic it was necessary to have some knowledge of English language and have an access to the Internet.

In the next part I dealt with a description of the developmental environment - the welcoming screen of the program, creating the first game and the basic menu. In the menu there are basic concepts necessary for next work with the program. I successively analyzed using of individual options and acting with them during creating the game.

Individual fractions are analyzed in the next capitol in detail. Some methods are demonstrated on attached files in the form of animations. It mainly concerned about a creating of Actor Types, Backgrounds, Tilesets, and Scenes, about creating of the first programmable events and blocks, which represent the program part of the game. There was also shown how to create a font or sound.

In the first practical example I outlined what is needed to create simple game with one scene and several actors, next how to use layers, tile sets, backgrounds, collisions, how to utilize a shop StencylForge, how to apply behavior and creating of groups required to set a collisions between them. In the second, more complicated practical example, we can see the work with scenes and interconnection between them by using regions. We also used more difficult programming in the form of global attributes, which are needed

to save the settings during crossing between various scenes. It was necessary to create more groups for collisions between various types of actors.

SEZNAM LITERATURY

1. Who Uses Haxe. *Haxe - The Cross-platform Toolkit*. [Online] 2016. [Citace: 25. 3 2016.] <http://haxe.org/use-cases/who-uses-haxe.html>.
2. Language Introduction. *Haxe - The Cross-platform Toolkit*. [Online] 2016. [Citace: 25. 3 2016.] <http://haxe.org/documentation/introduction/language-introduction.html>.
3. Borkwood, Innes. *Learning Stencyl 3.x Game Development*. Birmingham : Packt Publishing Ltd., 2013. ISBN 978-1-84969-596-1.
4. Stencylopedia - Table of Contents. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 15. 2 2016.] <http://www.stencyl.com/help/toc/>.
5. Crash Course: Let's Make a Game! on Stencylopedia. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 23. 2 2016.] <http://www.stencyl.com/help/viewArticle/143>.
6. Crash Course: Let's Make a Game! (Part 2) on Stencylopedia. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 23. 2 2016.] <http://www.stencyl.com/help/viewArticle/144/>.
7. Crash Course: Let's Make a Game! (Part 3) on Stencylopedia. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 23. 2 2016.] <http://www.stencyl.com/help/viewArticle/145/>.
8. Crash Course: Let's Make a Game! (Part 4) on Stencylopedia. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 23. 2 2016.] <http://www.stencyl.com/help/viewArticle/146/>.
9. Crash Course: Let's Make a Game! (Part 5) on Stencylopedia. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 23. 2 2016.] <http://www.stencyl.com/help/viewArticle/147/>.
10. Stencyl - Pricing. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 20. 2 2016.] <http://www.stencyl.com/pricing/>.
11. Use Stencyl to teach students how to code. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 20. 2 2016.] <http://www.stencyl.com/education/overview/>.
12. Stencyl Sample Games. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 21. 2 2016.] <http://www.stencyl.com/developers/samples/>.
13. Stencyl: Make iPhone, iPad, Android, Windows, Mac, Flash and HTML5 Games without code. *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 21. 2 2016.] <http://www.stencyl.com/features/>.
14. What's New in Stencyl? *Stencyl: Make iPhone, iPad, Android & Flash Games without code*. [Online] 2016. [Citace: 3. 1 2016.] <http://www.stencyl.com/features/history/>.
15. Haxe - The Cross-platform Toolkit. *Haxe - The Cross-platform Toolkit*. [Online] 2016. [Citace: 10. 3 2016.] <http://haxe.org/>.
16. Toolkit Introduction - Haxe - The Cross-platform Toolkit. *Haxe - The Cross-platform Toolkit*. [Online] 2016. [Citace: 10. 3 2016.] <http://haxe.org/documentation/introduction/toolkit-introduction.html>.

17. What is Haxe? - Haxe - The Cross-platform Toolkit. *Haxe - The Cross-platform Toolkit*. [Online] 2016. [Citace: 10. 3 2016.] <http://haxe.org/manual/introduction-what-is-haxe.html>.
18. History - Haxe - The Cross-platform Toolkit. *Haxe - The Cross-platform Toolkit*. [Online] 2016. [Citace: 10. 3 2016.] <http://haxe.org/manual/introduction-haxe-history.html>.
19. About this Document - Haxe - The Cross-platform Toolkit. *Haxe - The Cross-platform Toolkit*. [Online] 2016. [Citace: 10. 3 2016.] <http://haxe.org/manual/introduction-about-this-document.html>.

SEZNAM OBRÁZKŮ, TABULEK, GRAFŮ A DIAGRAMŮ

Obrázek 1 Úsek kódu v Haxe (2)	5
Obrázek 2 Úvodní obrazovka	7
Obrázek 3 Vytvoření nové hry	8
Obrázek 4 Nová hra - základní informace	8
Obrázek 5 Dashboard	9
Obrázek 6 Vytvoření nového herce	11
Obrázek 7 Import obrázku	12
Obrázek 8 Možnosti herců	13
Obrázek 9 Nová událost	13
Obrázek 10 Nastavení pozice	14
Obrázek 11 Nabídka programovacích bloků	14
Obrázek 12 Pohyb	20
Obrázek 13 Scene	21
Obrázek 14 move camera	21
Obrázek 15 Zem	22
Obrázek 16 StencylForge	23
Obrázek 17 Animace	24
Obrázek 18 Kolize hráče se skupinou	24
Obrázek 19 Region - Cil	26
Obrázek 20 Vyhra/Prohra	27
Obrázek 21 Show/Hide	29

PŘÍLOHY

Animace č. 1 Vytvoření scény.swf

Animace č. 1 Vytvoření scény.avi

Animace č. 2 Vytvoření herce.swf

Animace č. 2 Vytvoření herce.avi

Animace č. 3 Přidání staženého chování.swf

Animace č. 3 Přidání staženého chování.avi

Animace č. 4 Tvorba vlastního chování.swf

Animace č. 4 Tvorba vlastního chování.avi

Animace č. 5 Vytvoření herce na pozici.swf

Animace č. 5 Vytvoření herce na pozici.avi

Animace č. 6 Vytvoření kolize.swf

Animace č. 6 Vytvoření kolize.avi

Animace č. 7 Tvorba pozadí.swf

Animace č. 7 Tvorba pozadí.avi

Animace č. 8 Tvorba terénu.swf

Animace č. 8 Tvorba terénu.avi

Animace č. 9 Umístění terénu na scénu.swf

Animace č. 9 Umístění terénu na scénu.avi

Animace č. 10 Tvorba regionu.swf

Animace č. 10 Tvorba regionu.avi

Ball game.stencyl

Ball game.swf

Horolezec.stencyl

Horolezec.swf