

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Trněný Michal

Studijní program: Otevřená informatika  
Obor: Softwarové systémy

Název tématu: Generování webových šablon pro 2D skeletální animace postav

### Pokyny pro vypracování:

Analyzujte možnosti interaktivní webové prezentace 2D skeletálních animací postav. Navrhněte modulární šablonovací systém, pomocí kterého bude možné roztřídit a rozdělit interaktivní celky tak, aby je bylo možné co nejlépe kombinovat a znovupoužívat. Navrhněte a implementujte jednoduše použitelný nástroj, který z konfiguračních souborů a připravených šablon vygeneruje webovou stránku. Šablony budou sloužit k vygenerování interaktivní webové stránky, která má za účel propagaci připravované desktopové hry v Unity, používající postavy vytvořené v animačním programu Spine. Zvažte možnost využití již existujících komponent připravených v herním projektu v Unity. Porovnejte dostupné knihovny (spine runtimes) z pohledu responzivity a optimalizací pro mobilní zařízení.

### Seznam odborné literatury:

- [1] Web Animation using JavaScript: Develop & Design. Julian Shapiro. Peachpit Press, 2015.
- [2] Unity, <https://unity3d.com/>
- [3] Spine: <http://esotericsoftware.com/>

Vedoucí: Ing. David Sedláček, Ph.D.

Platnost zadání do konce letního semestru 2017/2018

prof. Dr. Michal Pěchouček, MSc.  
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 16.2.2017



České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů



Bakalářská práce

**Generování webových šablon pro 2D skeletální animace  
postav**

*Michal Trněný*

Vedoucí práce: Ing. David Sedláček, Ph.D.

Studijní program: Otevřená informatika, bakalářský

Obor: Softwarové systémy

23. května 2017





## Poděkování

Chtěl bych poděkovat Ing. Davidu Sedláčkovi, Ph.D. za podporu, vstřícnost a přínosné rady, bez kterých by tato práce byla jen stěží zrealizována.



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 15. 5. 2017

.....



# Abstract

TRNĚNÝ, Michal: Generating web templates for 2D skeletal character animation. [Bachelor thesis] - Czech Technical University in Prague. Faculty of Electrical Engineering, Department of computers. Supervisor: Ing. David Sedláček, Ph.D.

The subject of the thesis is to create a generator of interactive scenes that are based on 2D skeletal animations. The generator is made as a editor window in Unity and its output is a template for web presentation with interactive units managed by scripts in independent modules. Generator usage is tested by generating templates for the webpage of the game developed in Unity called End of Line. Skeletal animations were prepared in the Spine animation software.

The second part of the thesis deals with the selection of technology for manipulation with animations and their performance comparisons. The following chapters then describe the tool itself and the configuration files that the tool generates. Finally, the thesis deals with the script of the website and the interactive scenes themselves.

# Abstrakt

TRNĚNÝ, Michal: Generování webových šablon pro 2D skeletální animace postav. [Bakalářská práce] - České vysoké učení technické v Praze. Fakulta elektrotechnická, Katedra počítačů. Vedoucí: Ing. David Sedláček, Ph.D.

Předmětem práce je tvorba generátoru interaktivních scén, které jsou založeny na 2D skeletálních animacích. Generátor je součástí editorového okna Unity a jeho výstupem je šablona pro webovou prezentaci. Interaktivní celky jsou pak ovládány skripty v nezávislých modulech. Použitelnost generátoru je demonstrována vygenerováním šablon pro propagační stránku hry End of Line vyvíjené v Unity. Skeletální animace byly připraveny v programu Spine.

V první části se práce zabývá specifikací podkladů a animačními technikami použitelnými pro výslednou webovou stránku. Druhá část práce se zabývá výběrem technologie pro manipulace s animacemi a jejich výkonnostním porovnáním. Následující kapitoly pak charakterizují nástroj samotný a konfigurační soubory, které nástroj generuje. V závěru se práce zabývá skriptem webové stránky a samotnými interaktivními scénami.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Motivace . . . . .	1
1.2	Cíle . . . . .	2
<b>2</b>	<b>Podklady a zdrojové soubory</b>	<b>4</b>
2.1	Grafické podklady . . . . .	4
2.2	Animační podklady . . . . .	4
2.3	Komponenty herního projektu . . . . .	5
<b>3</b>	<b>Animace</b>	<b>6</b>
3.1	Animační software Spine . . . . .	6
3.1.1	O programu . . . . .	6
3.1.2	Klíčové vlastnosti . . . . .	7
3.1.3	Struktura exportovaných souborů . . . . .	8
3.2	Druhy webových animací . . . . .	9
3.2.1	Reakční animace . . . . .	9
3.2.2	Čekací animace . . . . .	10
3.2.3	Animace vyprávění příběhu . . . . .	11
3.2.4	Dekorativní animace . . . . .	11
3.2.5	Reklamní animace . . . . .	12
3.3	Animace postav . . . . .	12
3.3.1	Animační principy . . . . .	12
3.3.2	Zajištění kvality animací . . . . .	15
<b>4</b>	<b>Kritéria výběru technologií</b>	<b>17</b>
4.1	Využití funkcionality programu Spine . . . . .	17
4.2	Požadavky na webovou prezentaci . . . . .	17
<b>5</b>	<b>Přehled animačních technologií</b>	<b>19</b>
5.1	CSS animace . . . . .	19
5.2	GIF animace . . . . .	19
5.3	Video . . . . .	20
5.4	SVG animace . . . . .	20
5.5	Flash animace . . . . .	20
5.6	Unity export a WebGL . . . . .	21
5.7	JavaScriptové animace . . . . .	21

<b>6</b>	<b>Analýza technologií</b>	<b>22</b>
6.1	Spine StageXL . . . . .	22
6.2	Spine WebGL . . . . .	22
6.3	HTML5 Embeddable Widget . . . . .	23
6.4	Srovnání technologií . . . . .	23
6.4.1	Příprava výkonnostního testu . . . . .	23
6.4.2	Porovnání výkonu na desktop prohlížečích . . . . .	24
6.4.3	Porovnání výkonu na Android prohlížečích . . . . .	25
6.4.4	Verdikt . . . . .	25
<b>7</b>	<b>Nástroj pro tvorbu scén</b>	<b>28</b>
7.1	Úlohy nástroje . . . . .	28
7.2	Výběr formátu scény . . . . .	28
7.3	Výběr technologie editoru . . . . .	29
7.4	Klíčové vlastnosti editoru . . . . .	30
<b>8</b>	<b>Webová stránka</b>	<b>32</b>
8.1	Scény . . . . .	32
8.1.1	Obecný popis scén . . . . .	32
8.1.2	Souboj s monstrem . . . . .	33
8.1.3	Demonstrace zbraní . . . . .	33
8.1.4	Dialogy s postavami . . . . .	35
8.1.5	Další scény . . . . .	35
8.2	Přehled článků . . . . .	36
8.2.1	Úvodní článek . . . . .	36
8.2.2	Článek s baterkou . . . . .	36
8.2.3	Článek s vesmírnou stanicí . . . . .	36
8.3	Načítání stránky . . . . .	36
8.4	Struktura kódu . . . . .	37
<b>9</b>	<b>Závěr</b>	<b>40</b>
9.1	Shrnutí . . . . .	40
9.2	Budoucnost nástroje a stránky . . . . .	41
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>44</b>
<b>B</b>	<b>Tabulky</b>	<b>45</b>
B.1	Měření na desktopovém zařízení . . . . .	45
B.2	Měření na mobilních zařízeních . . . . .	47

# Seznam obrázků

3.1	Spine - ukázka chůze monstra v End of Line . . . . .	6
3.2	Spine - postavy se stejnou kostrou . . . . .	7
3.3	Exportované soubory z programu Spine . . . . .	9
3.4	Impossible Bureau - efekt najetí myši . . . . .	10
3.5	Úschovna - postup odesílání zásilky . . . . .	10
3.6	Apple Mac Pro - představení produktu . . . . .	11
3.7	Idea Branch - dekorativní animace . . . . .	12
3.8	Pravidlo č. 1 - Splácnutí a roztáhnutí . . . . .	13
3.9	Pravidlo č. 6 - Pomalu před akcí a pomalu po akci . . . . .	14
3.10	Pravidlo č. 10 - Přehánění . . . . .	15
3.11	Rotoskopie – postava z End of Line . . . . .	16
6.1	Ukázka stránky, na které bylo testování prováděno. . . . .	24
6.2	Graf naměřených hodnot rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Opera 42. Render 01 reprezentuje čas vykreslení první animace. . . . .	25
6.3	Graf naměřených hodnot rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Opera 42. Render 15 reprezentuje čas vykreslení poslední animace. . . . .	26
6.4	Graf zprůměrovaného měření rychlosti načítání v milisekundách na telefonu Huawei P8 Lite. Render 01 zastupuje načtení první animace, Render 15 pak té poslední. . . . .	27
7.1	Schéma popisuje parametry nastavitelné v konfiguračním souboru pro webovou stránku v UML formátu jako diagram tříd. . . . .	29
7.2	Grafické rozhraní pro úpravu skriptů je vyřešeno Unity editorovým oknem. . . . .	30
8.1	Vrstvy scény jsou zobrazeny přes sebe pomocí absolutního pozicování. Přední vrstvu obaluje třída UiManager a dvě zadní zase třída Stage-Manager. . . . .	32
8.2	Ukázka interaktivní scény, ve které uživatel může odrazit nepřítele stisknutím tlačítka. . . . .	34
8.3	Ukázka scény se zbraněmi. Animace střelby je realizována pomocí inverzní kinematiky. . . . .	34
8.4	Ukázka dialogové scény. K dispozici jsou tři možnosti dialogové interakce. . . . .	35



8.5	Zjednodušený diagram demonstruje návaznosti tříd. Červeně jsou označeny třídy definující interaktivitu scén, která je specifická pro projekt	
	End of Line. . . . .	38

# Seznam tabulek

B.1	Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Chrome 55. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	45
B.2	Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Firefox 50. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	45
B.3	Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Microsoft Edge 14. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	46
B.4	Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Opera 42. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	46
B.5	Zprůměrované měření rychlosti načítání na desktopovém zařízení v milisekundách. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	46
B.6	Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Sony Xperia E5 se systémem Android 6.0.0 v prohlížeči Chrome 58. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	47
B.7	Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Huawei P8 Lite se systémem Android 6.0.0 v prohlížeči Chrome 58. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	47
B.8	Zprůměrované měření rychlosti načítání v milisekundách na telefonu Sony Xperia E5. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	47
B.9	Zprůměrované měření rychlosti načítání v milisekundách na telefonu Huawei P8 Lite. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget. . . . .	47

# Kapitola 1

## Úvod

### 1.1 Motivace

V nezávislém herním studiu Digital Shard Interactive se podílím na vývoji 2D sci-fi příběhově orientované hry zvané End of Line. Vývoj hry je zamýšleno zakončit prodejem na známé herní distribuční platformě Steam. Vzhledem ke skutečnosti, že Digital Shard Interactive čítá pouze čtyři členy, bylo žádoucí, aby se někdo z týmu postaral o propagaci formou webové stránky. Výzvu jsem se rozhodl využít. Zmíněná hra využívá v hojné míře 2D skeletální animace postav, a proto hlavní problematika spočívala v převedení těchto komponent na webovou stránku. Cílem bylo tedy vytvořit jednoduše použitelný nástroj, který by pomohl vygenerovat interaktivní části webové stránky.

Návštěvníci by se skrze interaktivní scény měli seznámit s klíčovými rysy hry. Chtěli bychom tak prezentaci odlišit od standardních, často zmatečných webových stránek, a přiblížit se kvalitnější formě zpracování, kterou úspěšně udržuje například Blizzard. Záměrem je zároveň pokusit se o zanechání dojmu srovnatelným se samotným průchodem hry. Zážitek z průzkumu stránky by pak měl návštěvníka pobídnout ke koupi produktu.

Pro herní průmysl jsou typické velké změny v průběhu vývoje, proto by bylo žádoucí, aby vzniklé úpravy bylo možné rychle a snadno zapracovat do webové stránky. Je proto vhodné, aby nástroj zároveň umožnil snadnou výměnu nových grafických prvků a úpravy rozložení jednotlivých scén. Momentálně se spekuluje o další změně grafického vzhledu postav. Dle předchozích zkušeností se tak může stát před dokončením projektu ještě vícekrát. Nástroj by měl zajistit snadnou konverzi nového materiálu a minimalizovat nutné úpravy, aby nemusel být nijak restrukturalizován kód ovládající interaktivní scény jako takové.

### 1.2 Cíle

Jedním z cílů práce je realizovat webovou prezentaci, která bude dostatečně věrně přenášet dojmy ze samotného hraní podkladové hry. Nicméně se předpokládá, že samotná hra nebude na webu zpřístupněná. Níže jsou uvedeny klíčové vlastnosti, které by měla propagační stránka pokrýt:

- **Příběh:** Hra je postavena na silném příběhu postaveném na vztazích mezi hlavním protagonistou a hologramem mladé dívky, která mu pomáhá v souboji svými radami a schopnostmi ovlivňovat dění na vesmírné stanici, kde se odehrává hlavní děj. Mimo jiné také může manipulovat se světelnými zdroji. Protagonista se také setkává se sarkastickým robotem jménem Todd. Poznávání jejich charakterů je realizováno konverzacemi.
- **Grafický styl:** Hra se snaží udržovat realistický dojem, a i proto jsou pro velkou část prvků využívány 3D modely vykresleny do 2D a posléze editovány. Pro animace byly využity podklady z reálného světa.
- **Souboj:** Souboj se skládá z konfliktu s monstry, která jsou nepředvídatelná a číhají v temných zákoutích. Souboj probíhá jak na dálku pomocí různých zbraní, tak z blízka formou odražení.
- **Světlo a stín:** Jednou z hlavních komponent, která ovlivňuje průchod hry, je světlo, respektive jeho nedostatek. Jeho množství ovlivňuje chování umělé inteligence nepřátel a zároveň může jeho intenzitu kontrolovat výše zmíněný hologram, a hráčovi tak pomáhat, nebo v případě hádky škodit.
- **Tematické zasazení:** Celá hra je zasazena ve sci-fi prostředí na opuštěné vesmírné stanici.

Samotná webová stránka by měla zaujmout širokou veřejnost a z této podmínky plynou následující klíčové body:

- **Otevřenost pro široké publikum:** Stránka by měla být schopná přenést dojem na mobilních i stolních zařízeních. Vzhledem k motivaci by stránka měla být zobrazitelná ve všech standardních prohlížečích (konkrétní specifikace v kapitole 4).
- **Poutavost a přehlednost:** Stránka by měla využít moderních animačních a designových technik, aby poutala pozornost a zároveň nezahltěla návštěvníka přebytečným obsahem.

Pro snadnou tvorbu a editaci stránky vznikají také požadavky na existenci nástroje, který by pomohl jednotlivé interaktivní scény spravovat. Mezi klíčové vlastnosti nástroje pak patří níže uvedené body:

- **Obměnitelnost obsahu scén:** Nástroj by měl umožňovat snadné vizuální úpravy jednotlivých scén bez zásahů do logiky stránky. Jednotlivé grafické prvky by měly být nástrojem předzpracovány tak, aby jejich použití stránku nijak nezpomalovalo.
- **Parametrizovatelnost scén:** Jednotlivé scény by měly mít dostatek parametrů, aby bylo možné jejich vlastnosti upravovat, a docílit tím tak odlišnosti ve funkcionalitě anebo lepší výkonnosti.
- **Ovladatelnost:** Kromě autora samotného by měl být nástroj ovladatelný také dalšími lidmi, aby jej později mohli použít například další členové týmu Digital Shard Interactive. Pokud to bude možné, je vhodné, aby nástroj úplně odstínil úpravy logiky (kódu) a nevyžadoval nutně po uživateli znalosti webových technologií a hlubší porozumění konfiguračním souborům. Z tohoto důvodu by nástroj měl mít grafické rozhraní.

## Kapitola 2

# Podklady a zdrojové soubory

### 2.1 Grafické podklady

Grafické podklady použité pro hru přímo odpovídají těm, které by měly být k vidění na webové stránce. Grafický styl je postavený na realistickém základu, a proto většina podkladů vzniká nejprve ve 3D programech. Prostředí umístěné v pozadí za postavami je - stejně jako například zbraně - vytvořeno v programu Cinema4D. Pro udržení proporcí postav v různých úhlech byly postavy vytvořeny v intuitivním nástroji MakeHuman a posléze přeexportovány do Cinema4D pro případné úpravy. V tomto modeláři je pak nastavena izometrická kamera, s jejíž pomocí jsou pak modely vykresleny do 2D jako rastrovaný obrázek ve formátu PNG.

Vykreslené grafické podklady pak prochází editací v programu Adobe Photoshop. Jedná se obvykle o ořezávání a přebarvení. V některých případech bylo také nutné dokreslit části objektů a případně též přidat textury. Grafické podklady, týkající se prostředí v pozadí, připravil a zpracoval člen vývojářského týmu Štěpán Pelíšek.

### 2.2 Animační podklady

Grafické podklady vznikají postupem popsáním v předchozí sekci. Výstupem tohoto postupu jsou v případě postav rastrované obrázky končetin a dalších částí těla, se kterými pak manipuluje program Spine (o jeho vlastnostech více v kapitole 3.1.2). Výstupem animačního softwaru pak může být sekvence obrázků ve formátu GIF nebo video, pro účely projektu jsou však vyexportovány soubory popisující tyto animace a atlas s odkazy na všechny příslušné grafické podčásti (o struktuře těchto souborů více v 3.1.3).

Spine tak zajišťuje tvorbu jednotlivých animací, které je pak v rámci projektu nutné programově ovládat, řadit, zajišťovat jejich přechody a reagovat změnou těchto animací na uživatelský vstup.

## 2.3 Komponenty herního projektu

Výše zmíněné materiály pak figurují v herním enginu Unity. Z grafických podkladů v pozadí, u kterých lze díky editoru měnit jejich pozici ve scéně, zvětšovat je, modifikovat jejich průhlednost aj., vzniká samotný statický vizuál jednotlivých herních úrovní. Animované objekty jsou pak řízeny programově za pomoci Spine-Unity runtimes. Díky těmto knihovnám lze přímo v rozhraní editoru nastavit základní chování animovaných objektů, jako je přehrávání animace ve smyčce. Složitější interakce zahrnující animace, které musí reagovat na uživatelský vstup (jako například míření, střelba a souboj samotný), jsou řízeny pomocí skriptů v jazyce C#.

Události (například otevření dveří či spuštění dialogu) ovlivňující dění hry jsou řízeny vizuálním skriptováním, pro které projekt využívá koupenou komponentu z Unity Asset Store jménem FlowCanvas. FlowCanvas slouží i pro tvorbu dialogů mezi postavami a v kombinaci s nadstavbou pro konverzace vytvořenou na míru umožňuje velmi flexibilní propojování existujících replik postav. Skriptované konverzace tak dovolují například tvorbu dialogových cyklů, výběr z několika odpovědí nebo přiřazení zvukové stopy k dané textové zprávě.

Jedním z dalších klíčových herních systémů zmíněných v cílech (viz kapitola 1.2) jsou světla. Pro testovací účely projekt využívá základní Unity 3D světla, nicméně aktuálně dochází ke změně a pracuje se na nových 2D světlech. Umělá inteligence nepřátel je prozatím také v testovací fázi, a tak zatím nedemonstruje všechny plánované varianty chování.

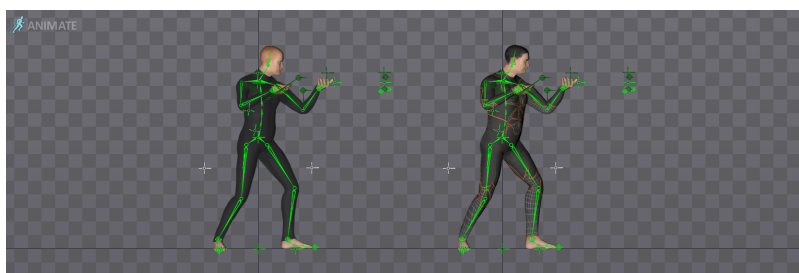




Tvorba skeletálních animací ve Spine editoru začíná v takzvaném SETUP plátně. Zde animátor připraví kostru objektu, který chce animovat. Kostra obsahuje kořen a skládá se z kostí. Každá kost kromě kořene má svého předchůdce a může mít libovolné množství potomků, dohromady tedy kostra vytváří stromovou strukturu. Na každou kost může animátor navěsit libovolné množství obrázků, které respektují pozici, natočení, zvětšení a zkosení u kosti samotné.

Zhotovená kostra je pak animována na takzvaném ANIMATE plátně. Zde animátor vytváří jednotlivé animace. Každá animace disponuje vlastní časovou osou. Animátor pak nastavuje pozice, natočení, zvětšení a zkosení jednotlivých kostí v určitém čase, a klíčuje tak pozici kostry jako takové. Protože obrázky navěšené na kostech reflektují nastavení kosti, změny vlastností kostí se promítají do těchto obrázků. Program pak automaticky dopočítá výše zmíněné vlastnosti mezi jednotlivými klíčovými pozicemi a tím vzniká animace samotná.

### 3.1.2 Klíčové vlastnosti



Obrázek 3.2: Ukázka dvou různých postav, které díky výměně skinů mohou používat stejnou kostru.

Spine disponuje mnoha vlastnostmi, které zjednodušují práci jak tvůrcům animací, tak programátorům. Navíc zlepšují dojem z animací samotných. Níže jsou uvedeny nejdůležitější vlastnosti:

- **Mesh deformace textur:** Obrázky navěšené na kosti mohou být přeměny na takzvaný mesh, který umožňuje rozčlenit texturu do polygonů. Jednotlivé vrcholy polygonu lze rovněž animovat a v praxi tím lze docílit například animace mimiky obličeje. Spine navíc umožňuje použít takzvané navázení u kostí, kdy jednotlivé vrcholy v procentuální míře přiřadí kostem (vybraným animátorem). Vlastnosti kostí se podle daného poměru přenáší na vrcholy, a tak lze pouhou změnou pozic kostí dosáhnout změny tvaru textury. Zkušením animátorů pomocí této techniky vytvoří plastický dojem z 2D animovaných obrázků.
- **Výměna skinů:** Skupinu obrázků navěšených na kosti lze vyměnit za jinou pomocí přepnutí takzvaného skinu. Tímto postupem lze využít jednu vytvořenou kostru vícekrát. V praxi lze pro všechny postavy stejného tělesného typu použít stejnou kostru. Obdobným použitím je například výměna sady oblečení dané postavy.

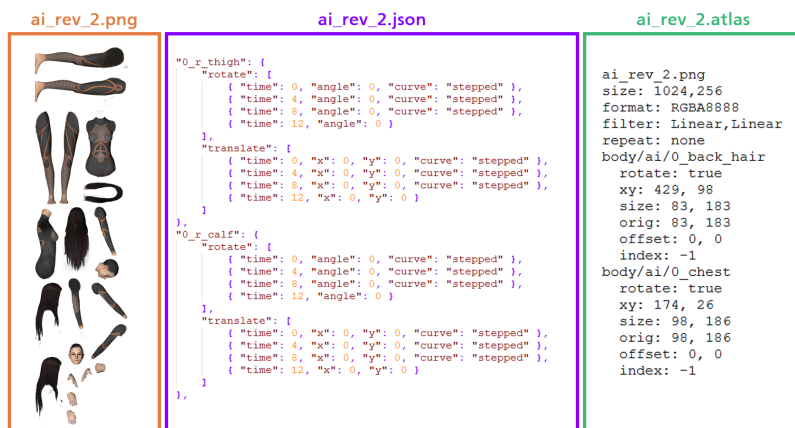
- **Události:** Do časové osy je možné také umístit událost. Událost má svůj název a ke každé události lze přiřadit jeden řetězec, celé číslo a číslo s plovoucí desetinnou čárkou. Události je pak možné odchytnout programově a přímo na ně reagovat. Toho lze využít, pokud má postava přehrát zvuk právě tehdy, když její noha došlápne na zem. Dalším příkladem je soubojová animace, při které dojde ke zranění postavy v konkrétním momentu na časové ose.
- **Mixování animací:** Animace jsou na postavě spuštěny ve vrstvách, z čehož vyplývá, že jich může být spuštěno více. Animace v nejvyšší vrstvě přepisuje zaklíčované kosti na svou hodnotu, nezaklíčované kosti neovlivňuje. Díky této premise je tak možné na jedné postavě bez problémů přehrávat dvě nezávislé animace. V případě animací, které pracují se stejnými kostmi, je tak většinou jen nutné nastavit korektní hierarchii vrstev animací.
- **Procedurální animace:** Animace mohou být díky knihovně řízeny i programově. Není složité ovlivňovat pozice kostí přímo v kódu, a tak lze například zajistit, aby postava hleděla na svého protivníka apod. V programu lze také zasahovat do časové osy a například použít jen jeden konkrétní snímek z vybrané animace. Tyto funkcionality lze dobře kombinovat s mixováním animací.
- **Blendování animací:** Jednotlivé animace do sebe mohou jednoduše přecházet. Tomuto jevu se říká blendování animací. Knihovny Spine umožňují nastavit pro dané dvě animace čas, který bude použit k tomu, aby se jedna animace prolнула do druhé. Během tohoto prolnutí dojde k postupné změně vlastností kostí. Výsledkem je plynulý přechod animací nehledě na to, v jakém bodě na časové ose k tomuto přechodu došlo.
- **Inverzní kinematika:** Standardní nastavení pozice kostí funguje tak, že se nejprve určí pozice a rotace rodičovských kostí, poté se nastaví pozice kostí potomků. Inverzní kinematika umožňuje podle nastavení pozice a rotace kosti potomka dopočítat optimální otočení a pozice kostí rodičovských. Body inverzní kinematiky na potomkovi tak mohou udržovat svou pozici na fixním místě, zatímco ostatní rodičovské kosti dopočítají svou rotaci. Tím bývá často zjednodušena práce animátora. Tato technika umožňuje snížit počet zaklíčovaných fází animace. Výsledkem jsou obvykle mnohem plynulejší animace. V kombinaci s programově řízenými procedurálními animacemi lze také ovlivňovat polohu těchto bodů inverzní kinematiky v existujících animacích, čehož se zase využívá například při animaci chůze po nerovném terénu.

Další významnou vlastností je export do formátu JSON a programově řízená manipulace s obrázky. To přináší ve srovnání se snímkanou animací kromě výkonnostních výhod (není zapotřebí načítat každý snímek animace) i další výhody popsané podrobněji v další podkapitole.[5]

### 3.1.3 Struktura exportovaných souborů

Spine poskytuje možnost vyexportovat jednotlivé animace do standardních formátů jako GIF či video formátů jako AVI a MOV. Tímto způsobem však přichází programá-

tor o výhody manipulace s animacemi jako je mixování a blendování. Pro maximální využití funkcí programu Spine je výhodné exportovat animace ve formátu JSON. V tomto textovém, lidsky čitelném, formátu je nejprve definováno rozložení kostí se všemi jejich vlastnostmi, včetně obrázků k nim připnutých. Posléze jsou zde definovány animace, které obsahují změny těchto vlastností v čase.



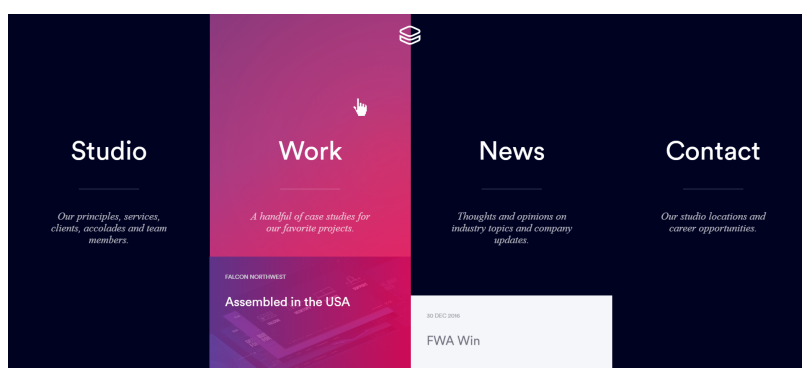
Obrázek 3.3: Exportováním do JSON vznikne kromě samotného JSON souboru, který popisuje rozložení kostí, obrázků a klíčových pozic, také soubor formátu PNG se seskupenými texturami. Pozice textur pak definuje k němu náležící soubor s koncovkou ATLAS.

JSON popisuje manipulaci s grafickým materiálem, který je uložen v souboru formátu PNG. Tento soubor obsahuje všechny použité rastrované obrázky, v případě postav jsou jimi tak části těla, oblečení a vybavení. Všechny tyto textury jsou seskupeny do jednoho souboru, a to zpravidla zlepšuje dobu nutnou k jejich vykreslení[11]. Jednotlivé subtextury jsou pak popsány v souboru s koncovkou ATLAS, který je ve standardizovaném formátu LibGDX.

## 3.2 Druhy webových animací

### 3.2.1 Reakční animace

Reakční animace podávají uživateli informaci o tom, že element zareagoval na uživatelský vstup. Běžně se s nimi setkáváme při změně vzhledu tlačítka po kliknutí či najetí myši. Tento typ animace rovněž sděluje uživateli, že úloha, kterou se snaží vyřešit (například odeslání emailu), byla úspěšně dokončena. [8]



Obrázek 3.4: Screenshot z webové stránky Impossible Bureau ukazuje efekt změny barvy prvku po najetí myši, který upozorňuje na rozkliknutelný odkaz.<sup>1</sup>

### 3.2.2 Čekací animace

Informuje uživatele o stavu aplikace a procesech, které běží na pozadí. Typickým příkladem v desktopovém světě byla změna kurzoru z šipky na přesýpací hodiny. Změněný vzhled kurzoru uživateli nepřímě oznámil, že aplikace ještě pracuje. Tím se zabránilo tomu, aby uživatel vykonal, v domněnku, že se nic neděje, vybranou akci znovu. Podobně je tak nepřímě uživateli podána informace o tom, že jeho požadavek ještě nebyl zpracován, a neměl by tak aplikaci ukončovat. Ve světě webových animací jsou to pak často například ukazatele průběhu načtení nějakého souboru. [8]



Obrázek 3.5: Screenshot z webové stránky Úschovna ukazuje animované rozhraní oznamující uživateli postup odesílání zázilky.<sup>2</sup>

<sup>1</sup>Screenshot z Impossible Bureau, pořizen 5.5.2017.

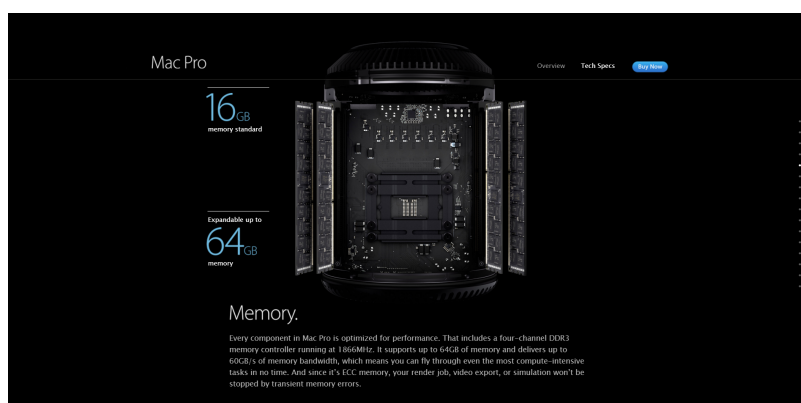
Dostupné z: <http://www.impossible-bureau.com/>

<sup>2</sup>Screenshot ze služby Úschovna, pořizen 5.5.2017.

Dostupné z: <http://www.uschovna.cz/poslat-zasilku>

### 3.2.3 Animace vyprávění příběhu

Jedná se o animace, které mají za úkol vzbudit emoce a přenést informace. V některých případech jsou postavy animovány jako v animovaných filmech, jindy se jedná o animace vysvětlující vznik nebo historii produktu či firmy. Dalším příkladem jsou webové stránky výrobků, které se před očima uživatele rozloží či složí. Těchto technik využívá například Apple nebo Sony na propagačních stránkách svých produktů.



Obrázek 3.6: Screenshot z webové stránky Apple Mac Pro poukazuje na poutavý způsob, jakým představit produkt.<sup>3</sup>

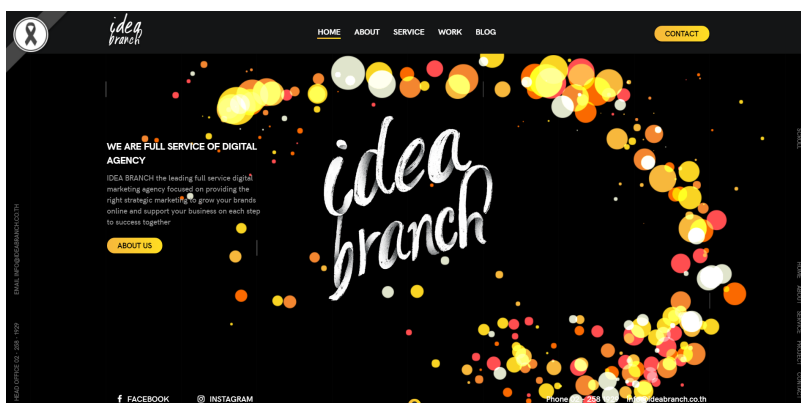
Význam těchto animací záleží na kontextu stránky, kvalitě provedení animací a nárocích na výkon. Trhané animace mohou snadno pokazit celkový zážitek. [8]

### 3.2.4 Dekorativní animace

Tyto animace nemají žádný funkční význam a slouží jen ke zkrášlení stránky. V mnoha případech však uživatele rozptylují a odvádí pozornost od podstatných částí stránky. Právě proto by měly být skryté a objevovat se jen po specifických událostech, jako je přejetí myši přes specifický element apod. Může být tedy rozumné zredukovat jejich opakování na jednorázové spuštění.

---

<sup>3</sup>Screenshot ze stránky Mac Pro, pořízen 5.5.2017. Dostupné z: <https://www.apple.com/mac-pro/>



Obrázek 3.7: Screenshot z webové stránky Idea Branch demonstruje dekorativní animaci, ke které dochází při pohybu myši. Vznikající barevné bubliny mohou být poněkud rozptýlující.<sup>4</sup>

Dalším příkladem je takzvaný „Easter-egg“, kdy se za specifických (často pro většinu uživatelů neznámých) podmínek na stránce přehraje neočekávaná animace sloužící k překvapení a pobavení uživatele. [8]

### 3.2.5 Reklamní animace

Přidáním animací do reklam se zvyšuje pozornost, kterou jim návštěvník stránek věnuje, ovšem tato pozornost bývá často nechtěná či vynucená, a proto může návštěvníky odradit. Navíc, podobně jako dekorativní animace, ruší uživatelskou soustředěnost. [8]

## 3.3 Animace postav

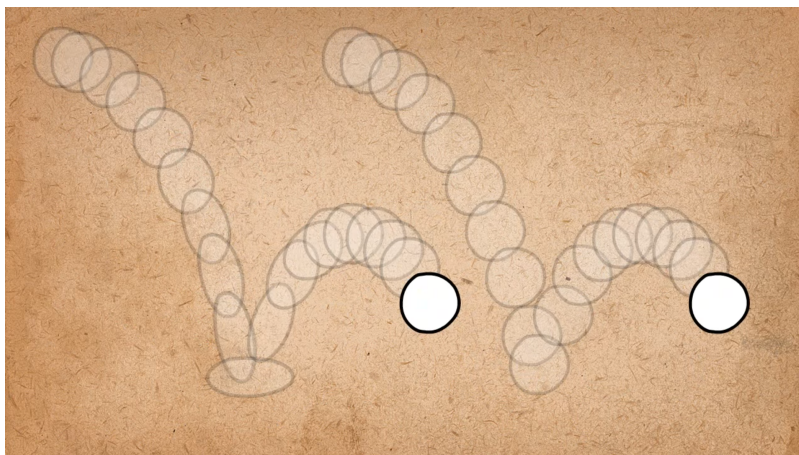
### 3.3.1 Animační principy

Jednou z obvyklých cest, jak zajistit, aby byly animace postav uvěřitelné, je držet se dvanácti níže uvedených pravidel animací, které jsou zpracovány v knize *The Illusion of Life: Disney Animation*. [18] Jedná se o:

1. **Splácnutí a roztáhnutí:** Během pohybu předmětů je často vhodné měnit jejich proporce (tedy délku nebo výšku). Animátor tím sděluje pozorovateli například to, jakou má předmět váhu a jakou letí rychlostí. Je zřejmé, že tímto způsobem lze vystihnout i materiál, ze kterého je předmět vyroben – to je vizualizováno tím, jak moc se předmět splácne při nárazu.[6]

---

<sup>4</sup>Screenshot z Idea Branch, pořízen 5.5.2017. Dostupné z <https://ideabrand.co.th/>

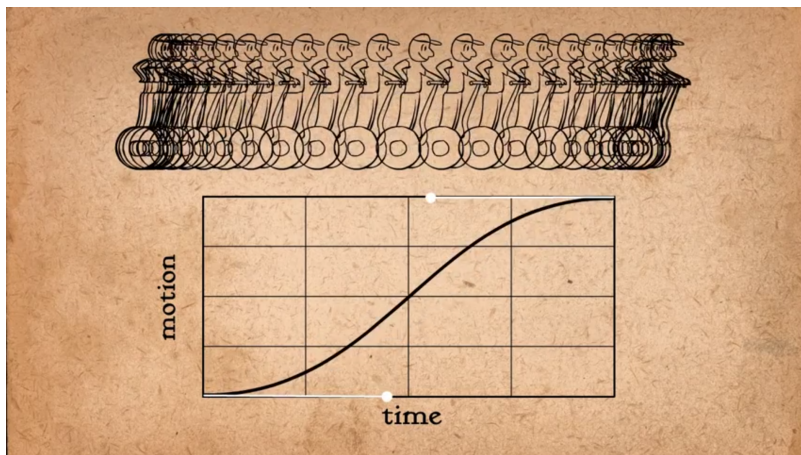


Obrázek 3.8: Obrázek popisuje pravidlo č. 1, splácnutí a roztáhnutí. Deformace během pohybu podává divákovi informace o vlastnostech pohybujícího se předmětu.<sup>5</sup>

2. **Očekávání:** Bývá vhodné diváka připravit na pohyb, který se postava chystá vykonat. To lze realizovat pomocí protipohybu. Pokud má postava například udeřit kladivem do kovádky, měla by se nejprve rozmáchnout. Samotné rozmáchnutí se přitom zpravidla bude trvat delší čas než samotný prudký pohyb.[6]
3. **Režie:** Jedná se o vhodné umísťování animovaných objektů na plátno v čase. Správného rozmístění lze docílit více způsoby. Pro více animovaných prvků například platí, že by neměly soupeřit o divákovu pozornost, nýbrž si ji postupně předávat tím, jak se ve scéně objevují. Do této kategorie spadá i správné umístění aktérů vůči kameře.[6]
4. **Animování přímo proti animování klíčových pozic:** Jedná se o postup, jakým vznikají jednotlivé snímky animace. Animování klíčových pozic znamená, že si animátor nejprve připraví klíčové pozice, kterými má postava během svého pohybu projít. Poté teprve dokreslí pozice mezi nimi. Animování přímo naopak znamená, že animátor kreslí snímky animace popořadě. Animace klíčových pozic se používá při předvídatelných pohybech postav, kdy je nutné kontrolovat průběh pohybu (např. skok nebo chůze postavy). Naopak u nepředvídatelných animací (např. hořící oheň nebo kapka vody) je vhodnější užít animování přímo.[6]
5. **Setrvačnost a překrývající se akce:** Objekty navěšené na rychle se pohybující postavu by měly podléhat jistému zpoždění. To lze nejlépe vysvětlit na konkrétním příkladu. Postava v plášti se snaží rozeběhnout, ale plášť se ještě nestihl rozvlnit. V určitý moment podléhá pohybu postava i plášť. Jakmile postava doběhne na určité místo, plášť se ještě vlní, a teprve se zpožděním se jeho pohyb uklidní. Touto technikou lze opět vykreslit charakter materiálů například délkou setrvačného pohybu těchto navěšených objektů.[6]

<sup>5</sup>Screenshot z videa 1. Squash & Stretch - 12 Principles of Animation, pořízeno 5.5.2017.  
Dostupné z: <https://www.youtube.com/watch?v=haa7n3UGyDc>

6. **Pomalů před akcí a pomalů po akci:** Téměř všechny pohyby začínají a končí pomalů. Animace s konstantní rychlostí na diváka často působí mechanicky – možná právě proto, že jsou typické pro roboty. Nastavení ovšem záleží na okolnostech, například při výstřelu ze zbraně nemá smysl vystřelené kulce přidávat počáteční zpomalení.[6]



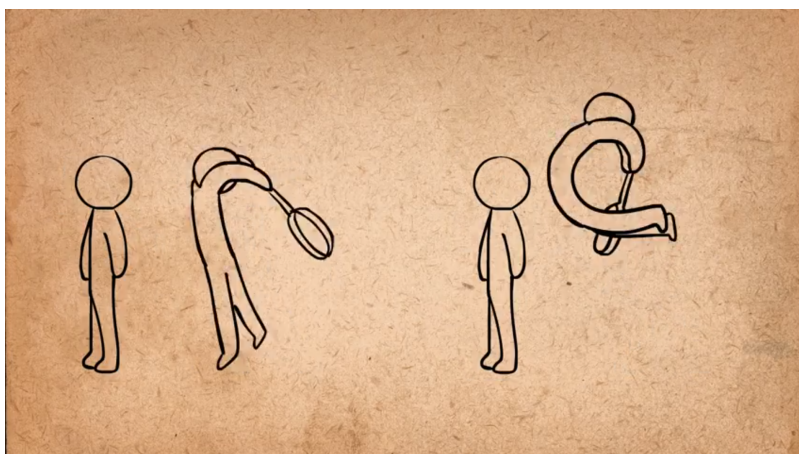
Obrázek 3.9: Upravením Bézierovy křivky definující rychlost pohybu lze docílit plynulejšího dojmu z animace.<sup>6</sup>

7. **Oblouky:** Většina žijících bytostí se pohybuje po kruhových cestách. Přidáním obloukového pohybu se tak zpravidla docílí živějšího dojmu.
8. **Druhotná akce:** Jedná se o gesta, která doplňují hlavní akci, a rozšiřují tak divákovu představu o pocitech postav nebo o situaci, ve které se nachází. V konkrétním příkladě se může jednat o chodící postavu, která při svém pohybu skřípe zuby a drží ruce zaťaté v pěst. Těmito druhotnými akcemi divák pochopí, v jakém rozpoložení se postava nachází a jaké motivy postava má.[6]
9. **Časování:** Princip časování se zabývá počtem snímků, které by animace měla mít. Pokud animátor zvolí počet snímků nízký, bude působit animace rychle. Při vysokém snímkování naopak pomalu. Přidáním snímků lze také upravit význam akce. Během akce může pak postava měnit svůj výraz a tím významně ovlivnit celkový význam animace.[6]
10. **Přehánění:** Téměř každou akci si lze představit v intenzivnějším podání a nalezení této extrémní varianty by mělo být animátorovým cílem. Extrémní provedení totiž umocňuje dopad na diváka. Z určitého úhlu pohledu se tak animace stává realističtější, neboť je tota výrazná varianta pochopitelnější, a tím pádem i věrohodnější. Doporučením na závěr je tedy najít maximální extrém akce a pak jej jen trochu zmírnit, aby nebyla akce příliš přehnaná.[6]

---

<sup>6</sup>Screenshot z videa 6. Slow In & Slow Out - 12 Principles of Animation, pořízeno 5.5.2017.  
Dostupné z: <https://www.youtube.com/watch?v=fQBFsTqbKhY>





Obrázek 3.10: Přeháněním přípravy akce dosáhneme často živějšího dojmu.<sup>7</sup>

11. **Dovednosti v rýsování:** Věrohodnosti animace často pomáhá kvalitní ztvárnění. Při kresbě postaviček je tak vhodné věnovat zvláštní pozornost perspektivě. Dekompozice animovaných objektů na jednodušší 3D tvary, které se správně zkreslují vlivem perspektivy, může výsledný dojem značně ovlivnit. Spolu s tím souvisí i přesahy částí objektu, pomocí kterých divák odvodí, v jaké hloubce se která část objektu nachází.[6]
12. **Charisma postav:** Diváka by měly animované postavy zaujmout. Toho lze dosáhnout více způsoby, jedním z nich je například použití netradičnějších základních tvarů, ze kterých se postavy skládají. Inspiraci je možné hledat také u karikaturistů, kteří vyzdvihují nejvýraznější rysy lidí, a pokud se snaží vytvořit sympatický dojem, zvětší například oči a zmenší tělo.[6]

### 3.3.2 Zajištění kvality animací

Jednou z technik zajištění kvality animací je takzvaná rotoskopie. V praxi funguje tak, že animátor kreslí přes záběry natočené kamerou (respektive snímky nafocené fotoaparátem). Samotnému natáčení by mělo předcházet kreslení takzvaných storyboardů, tedy náčrtů, které demonstrují, jak bude scéna vypadat a pod jakým úhlem bude záběr na postavu veden. V případě, že záběry nebudou použity pro překreslování, jedná se pouze o získávání živých referencí.[17]

<sup>7</sup>Screenshot z videa 10. Exaggeration - 12 Principles of Animation, pořízeno 5.5.2017.  
Dostupné z: <https://www.youtube.com/watch?v=HfFj-VQKiAM>



Obrázek 3.11: Vystřížené fotky z videa s přidanou animovanou postavou. Video sloužilo jako podklad pro animace předchozí verze hlavního protagonisty ve hře End of Line.

O tom, že rotoskopie často zajistí kvalitní animaci a že má v herním průmyslu svůj prostor odedávna, svědčí i její použití ve hře Prince of Persia z roku 1989.[12]

## Kapitola 4

# Kritéria výběru technologií

### 4.1 Využití funkcionality programu Spine

Požadavky tohoto druhu vychází z postupu, jakým vznikly grafické a animační prvky pro desktopovou hru. Zadání úlohy vyžaduje, aby vzhled animací a celkový dojem odpovídal co nejvěrněji animacím použitým ve hře. Vzniklé animace se opírají o techniky animačního softwaru Spine (popsány přesněji v kapitole 3.1.2) a tím vznikají tyto požadavky na funkcionalitu:

- Reakce na události
- Mixování animací
- Blendování animací
- Reprodukce inverzní kinematiky
- Mesh deformace textur

### 4.2 Požadavky na webovou prezentaci

Požadavky vychází z předpokladu, že by webová prezentace měla být k dispozici co nejširší veřejnosti, aby bylo možné produkt efektivně propagovat. Vyžaduje se:

- **Responzivní design:** Stránka by měla respektovat rozlišení monitoru, na kterém je zobrazena. Stejně tak by se měla přizpůsobit mobilním zařízením s menším rozlišením.
- **Funkčnost v standardních prohlížečích:** Standardními prohlížeči se rozumí aktuální verze Google Chrome, Mozilla Firefox a Microsoft Edge, případně Internet Explorer 11 a Opera. Animace postav by se měly chovat shodně ve všech těchto aplikacích, stejně tak by nemělo dojít ke komplikacím, které jsou způsobeny specifickými problémy v konkrétním prohlížeči.

- **Funkčnost na mobilních zařízeních:** Očekává se, že stránku bude možné zobrazit a ovládat na tabletech i mobilech.
- **Nízké načítací časy stránky:** Bylo by vhodné snížit co nejvíce načítací časy stránky. Zároveň je žádoucí, aby byl návštěvníkovi stránky poskytován postupně důležitý obsah a aby byl návštěvník informován o tom, že se stránka ještě načítá a není zatím kompletní.
- **Nízké časy kompletního provozu animací:** Je prioritou vybrat takový systém animací, který nebude ve srovnání s ostatními technologiemi příliš pomalý.
- **Plynulost animací** Plynulost stránky jako takové často tvoří větší část dojmu než kvalita animací samotných. Z tohoto důvodu by měla být vždy upřednostněna plynulost animací před jejich množstvím, aby bylo zajištěno vysoké snímkování.

## Kapitola 5

# Přehled animačních technologií

### 5.1 CSS animace

CSS animace slouží k animacím přechodů mezi různými CSS styly. Přechody jsou klíčovány počátečním a koncovým stavem, případně i dalšími průchozími body. Ve srovnání s JavaScriptovými animacemi jsou jednodušší na tvorbu a mívají často dobrý výkon díky technice přeskokování snímků, pomocí které udržují plynulost. Zde je nechán prostor pro prohlížeč, aby optimalizoval a případně omezil frekvenci snímkování animací v panelech (tabs), které nejsou pro uživatele aktuálně viditelné, a reagoval tak na skutečné vytížení zařízení. Tyto animace se používají především pro dvoustavové přechody, které nejsou za chodu ovlivňovány či přerušovány.[3]

Použití CSS animací pro komplexní animaci postav je nevhodné. Jednoduchost tvorby animací zároveň omezuje komplexitu, například animace změny velikosti elementu a jeho nezávislou rotaci nelze prakticky realizovat, neboť se obě tyto změny ukrývají pod vlastností transform. Dalším praktickým problémem je obdélníková struktura elementů, kterou lze jen obtížně konvertovat na trojúhelníky.

CSS animace však mají svůj prostor pro dekoraci webové stránky mimo animované postavy. Opakovanou animací na tlačítkách lze upozornit uživatele na jejich interaktivitu. Do určité míry je také možné chod animací řídit pomocí JavaScriptu změnou atributu třídy a tím animaci například pozastavit nebo vyměnit za jinou.

### 5.2 GIF animace

Animační formát GIF disponuje paletou 256 barev, která je vhodná pro jednodušší grafiku, a z téhož důvodu nevhodná například k reprodukci plynulých gradientů. Hlavní předností formátu je jeho kompatibilita a možnost virálního rozšíření. Formát není dostatečně flexibilní, aby v něm mohlo být smysluplně realizováno například mixování a blendování animací (viz kritéria 4.1). V porovnání s SVG (viz 5.4 má mnoho nevýhod, mezi které patří hlavně obtížný způsob manipulace a nároky na paměť.

### 5.3 Video

Video formát poskytuje výhodu relativně dobré kompatibility, nicméně má zvýšené nároky na datový přenos. Mnoho webových stránek cílících na dojem videa integruje, nicméně úroveň interaktivity bývá často limitována. Videoformát nevyhovuje předpokladům z kapitoly 4.1, neboť neumožňuje použití mixovaných a blendovaných animací.

Ve výsledných webových stránkách má však video svůj prostor, například ve formě integrovaného videa pro trailer z YouTube.

### 5.4 SVG animace

SVG animace slouží k manipulaci s SVG formátovanými 2D vektorovými obrázky, které jsou díky XML notaci snadno čitelné lidským okem a vhodné i pro indexování webovými vyhledávači. Vhodné použití obsahuje animace jednodušších vektorových ilustrací, prvků v uživatelském rozhraní a ikon. Nicméně pomocí SVG animací lze animovat i rastrovane obrázky vložené v textové podobě do SVG formátu například pomocí kódování base64. Čistě vektorové SVG animace jsou úplně nezávislé na rozlišení. Dokážou proto zobrazit animaci ostře na jakémkoliv monitoru. Další výhodou je schopnost efektivně si poradit s částečnou průhledností barev. V animacích lze snadno docílit plynulosti přechodů mezi jednotlivými barvami – tím se liší např. od formátu GIF, kde je plynulost přechodu určena počtem snímků. Malá velikost SVG animací také snižuje načítací čas stránky.

Použití SVG animací pro zadanou úlohu (tedy animované postavy tvořené rastroványmi obrázky) nepřináší téměř žádné výhody. Konečným důvodem, proč se této možnosti v případě animovaných postav vyhnout, je předpoklad použití programu Spine, který nenabízí možnost exportu do SVG formátu. Komplexní animovaná grafika ve formátu SVG také nemusí mít příliš dobrý výkon.[1][13]

SVG materiály však pro výslednou webovou stránku mohou mít význam pro ikony a další jednoduchou grafiku v rámci grafického rozhraní.

### 5.5 Flash animace

Flashové animace značně ovlivnily vývoj webových animací, neboť byly prakticky první existující možností, jak animovaný obsah na webu vytvářet. Nyní z webového světa mizí především díky ucelené standardizaci HTML5 umožňující do webových stránek snadno zakomponovat video napříč webovými prohlížeči. Výhodou HTML5 je také jeho otevřený formát. Ve srovnání s HTML5 má také problémy s přístupností a s responzivitou, s čímž souvisí i chybějící podpora na mobilních zařízeních.[9]

Technologie je již na ústupu a chybějící podpora na mobilních zařízeních zamezuje jejímu použití pro danou úlohu.

## 5.6 Unity export a WebGL

Dřívější možnost exportu Unity na web pomocí Unity Webplayer přestala být podporována[7]. Nyní je nahrazena exportem do WebGL. V praxi se tak jedná o JavaScriptový program, který používá HTML5 technologii a využívá pro renderování WebGL. Pro přístup k DOM elementům ve stránce je k dispozici speciální API. Na překlad kódu z jiných jazyků (např. skriptů v C#) do JavaScriptu je Unity připravená a tuto možnost poskytuje.

Dle definovaných cílů v kapitole 1.2 není tato možnost preferovaná vzhledem k tomu, že by samotná hra neměla být na webu zpřístupněna.

## 5.7 JavaScriptové animace

JavaScriptové animace mohou být prováděny na plátně sloužícím k vykreslování grafiky, tedy elementu známém jako canvas. Jsou vhodné pro komplexní pohyby prvků, které nelze poskládat z CSS animací. Předností JavaScriptových animací je možnost kontrolovat jejich proces a interaktivně reagovat na různé události. Ve srovnání s CSS je zde odlišný proces tvorby animací, který není svázán pravidly tvorby CSS stylů, tj. není například nutné časovat klíčové body v procentech. Rovněž je možné za běhu například vypnout jednu animovanou vlastnost při zanechání zbytku animace v původním stavu, a to s nadefinovanou CSS animací prakticky nelze realizovat. JavaScriptové animace umožňují snadné použití relativních hodnot a u komplexních animací je velkou výhodou taktéž stručnost zápisu.[16]

Za nevýhodu – ve srovnání s CSS animacemi – se pak obvykle považuje nižší výkonnost JavaScriptových animací, nicméně toto všeobecné mínění je často ovlivněno srovnáním CSS animací s jQuery animacemi, u kterých je výkon skutečně nižší. Samotný JavaScript je tak na tom znatelně lépe. Navíc optimalizované chování prohlížečů s CSS elementy využívající výkon grafické karty lze vynutit pomocí několika technik, jednou z nich je například používání funkce `requestAnimationFrame()`, která je součástí standardu HTML5.[10] [4]

Animační software Spine, pracující s exportovaným formátem JSON, poskytuje několik knihoven v JavaScriptu, které slouží k manipulaci s animacemi. Některým knihovnám, jako např. Pixi.js, chybí klíčové vlastnosti (v tomto případě meshování textur), ostatní ale mají podporu prakticky všech vlastností z požadavků na animace (kapitola 4.1. Mezi tyto knihovny patří Spine StageXL v jazyce Dart, Embeddable Widget postavený na prvcích HTML5 a Spine knihovna pro WebGL. Podrobně jsou analyzovány v další kapitole.

## Kapitola 6

# Analýza technologií

### 6.1 Spine StageXL

Knihovna je napsána v jazyce Dart. Jedná se o programovací jazyk vytvořený společností Google. Dart je založen na syntaxi jazyka Javy a dědí od něj většinu charakteristik, jako je objektovost a přímá dědičnost. Mezi další zajímavé vlastnosti patří volitelná typovatelnost. Dart kód je interpretován za běhu pouze prohlížečem Dartium a nebo po kompilaci do JavaScriptu ostatními prohlížeči, které podporují interpretaci JavaScriptu.[2]

StageXL je knihovna pro jazyk Dart sloužící k vykreslování grafických prvků. Knihovna vznikla pro zjednodušení migrace z Flashe do HTML5, a má tedy stejnou hierarchii tříd. Třídní metody a vlastnosti odpovídají těm z Flash API, přičemž obsahují některá rozšíření. Jedním z příkladů je například možnost posunu počátku vykreslených obrázků. Třída Stage pak reprezentuje hlavní vykreslovanou oblast, která obaluje HTML5 element canvas. Knihovna má podporu WebGL.[14]

Knihovna Spine StageXL byla vytvořena Spine komunitou a podporuje všechny vlastnosti zmíněných v požadavcích na animace. Její přednost vychází z vlastností jazyka Dart, u větších projektů tedy udržení strukturovanosti a přehlednosti kódu. Neméně podstatnou výhodou je systém chybových hlášek.

### 6.2 Spine WebGL

Oficiální knihovna, podporovaná vývojáři Spine, je napsána v jazyce TypeScript, který lze snadno překompilovat do JavaScriptu. Splňuje všechny požadavky na animace a navíc obsahuje mnoho příkladů vzorových příkladů, ze kterých lze porozumět použití knihovny.



## 6.3 HTML5 Embeddable Widget

Další oficiální knihovna, rovněž napsána v TypeScriptu. Knihovna podporuje HTML prvky se speciálně nakonfigurovanými atributy, jejichž hodnoty slouží k jednoduché parametrizaci animací. Pro komplexnější animace je pak zapotřebí použít skriptování. Knihovna odstiňuje volání zobrazovacích funkcí. Snaží se tak zjednodušit manipulaci s animacemi.

Na rozdíl od Spine WebGL obsahuje minimum vzorových příkladů, nicméně základní použití je velmi intuitivní.

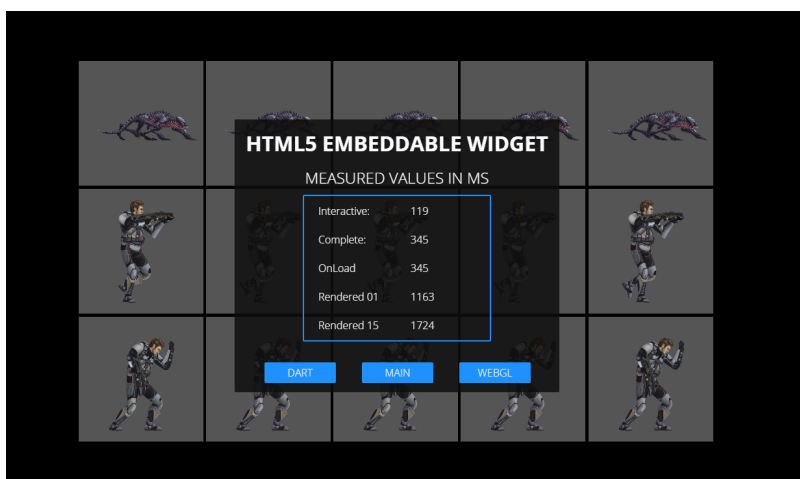
## 6.4 Srovnání technologií

### 6.4.1 Příprava výkonostního testu

Vzhledem k tomu, že jedním z kritérií je optimalizace stránky, bylo nutné provést testy výkonnosti zobrazování jednotlivých knihoven. Pro každou knihovnu byly připraveny webové stránky o stejné šabloně, do které bylo umístěno 15 pláten pro vykreslení animace. Na každém z pláten se objeví jedna ze dvou připravených animovaných postav z předpřipravených materiálů. Takové množství zobrazených elementů plátna, respektive postav, není v plánu realizovat na webové stránce v rámci jednoho výřezu, nicméně pro vyvozování závěrů je výhodné otestovat tuto náročnější variantu.

Samotné testovací stránky jsou k dispozici na níže uvedených adresách:

- **Spine StageXL:**  
`http://digital-shard.com/test/performance/build/web/dart.html`
- **Spine WebGL:**  
`http://digital-shard.com/test/performance/spine-ts/webgl/example/webgl.html`
- **Spine Embeddable Widget:**  
`http://digital-shard.com/test/performance/spine-ts/widget/example/widget.html`



Obrázek 6.1: Ukázka stránky, na které bylo testování prováděno.

Stránky zobrazují výsledky aktuálního měření. Měření zachycuje čas nutný pro interakci se stránkou, trvání jejího plného načtení a čas prvního a posledního vykresleného plátna s animací. Načtení stránky vypovídá o knihovně málo, a proto nejceněnější informací jsou časy vykreslení, jejichž interpretace je uvedena v následujících podkapitolách. Načtení stránky bylo v každém prohlížeči provedeno desetkrát a výsledek zprůměrován. Soubor všech naměřených hodnot se nachází v příloze.

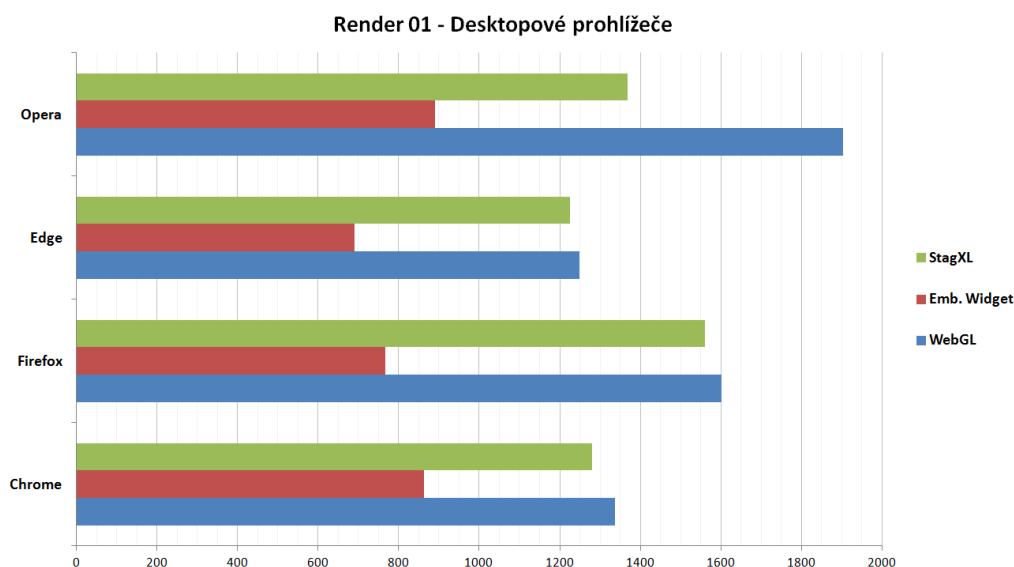
Test byl vytvořen se základní znalostí knihoven na bázi vzorových projektů. Lze tedy předpokládat, že by bylo možné při hlubším studiu technologií výkon ovlivnit dílčími optimalizačními kroky, které nebyly na první pohled zřejmé.

#### 6.4.2 Porovnání výkonu na desktop prohlížečích

Výsledky měření jsou uvedeny zobrazeny v grafech níže a naměřené hodnoty jsou k dispozici v přílohách. Měření bylo prováděno na prohlížečích Chrome 58, Firefox 50, Microsoft Edge 14 a Opera 45 ve verzích uvedených v tabulce. Render 01 udává dobu, po které se zobrazila první vykreslená animace. Render 15 udává dobu, po které již byly všechny animace na plátně vykresleny.

Velmi dobře si v testu vede knihovna Embeddable Widget. Jak se později během implementace ukázalo, nenabízí však dostatečně otevřený přístup k vykreslovacím funkcím a dalším relevantním problémem bylo plátno, na kterém se mohla vyskytovat maximálně jedna Spine animace. Větší rozdíly časů se objevují u prvního vykreslení animací v prohlížeči Firefox pro technologie Spine WebGL a StageXL v jazyce Dart. Technologie WebGL zaostává za ostatními převážně v prohlížeči Opera, v němž celkové vykreslení trvá mnohem déle.

Poznamenejme ještě, že u Microsoft Edge je rozdíl mezi Embeddable Widget a konkurenčními knihovnami rovněž téměř dvojnásobný, nicméně ve všech těchto případech má Microsoft Edge lepší časy než všechny ostatní prohlížeče.



Obrázek 6.2: Graf naměřených hodnot rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Opera 42. Render 01 reprezentuje čas vykreslení první animace.

### 6.4.3 Porovnání výkonu na Android prohlížečích

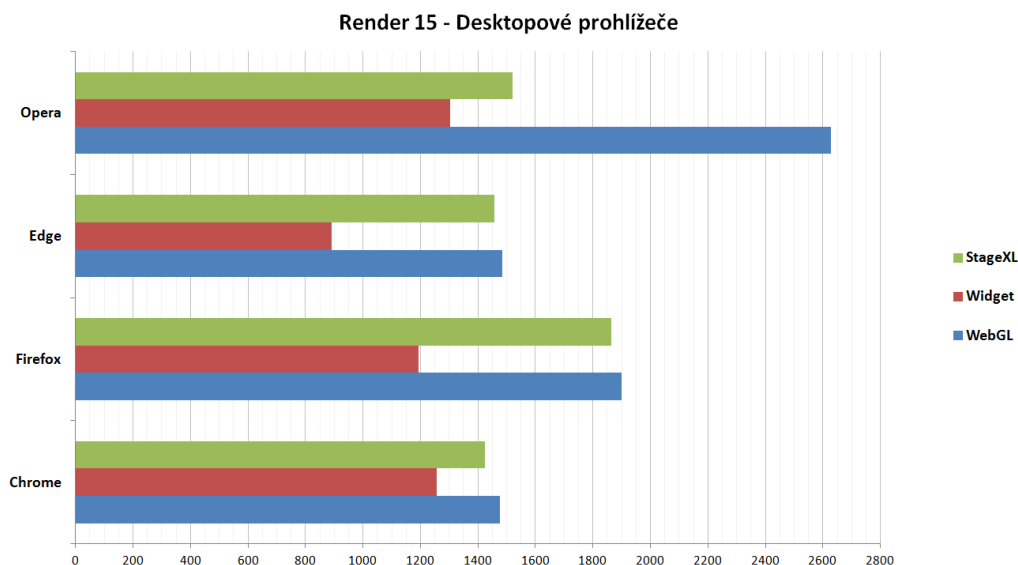
První testování probíhalo na telefonu Sony Xperia E5 s operačním systémem Android 6.0.0. Testoval se pouze nejpoužívanější prohlížeč Chrome ve verzi 58.

Ve všech testech však došlo k problémům se zobrazením prvních 8 plátén, pravděpodobně kvůli přílišné HW náročnosti. Druhé testování proběhlo na telefonu Huawei P8 Lite se stejnou verzí systému Android, tedy 6.0.0. Verze prohlížeče Chrome byla opět 58. K problémům se zobrazením zde nedocházelo, a hodnoty měření jsou tedy směrodatnější.

Velmi dobrých výsledků dosahoval především Spine Embeddable Widget. Časy zobrazení první animace pomocí Spine WebGL byly pak téměř dvakrát větší. Nejhuře si u prvního vykreslení vedla knihovna StageXL. Čas celkového načtení však měla se Spine WebGL srovnatelný. Nutno podotknout, že dokud se nenačetly všechny animace u Spine WebGL, animace měly tak nízké snímkování, že obrázky působily chvíli jako statické.

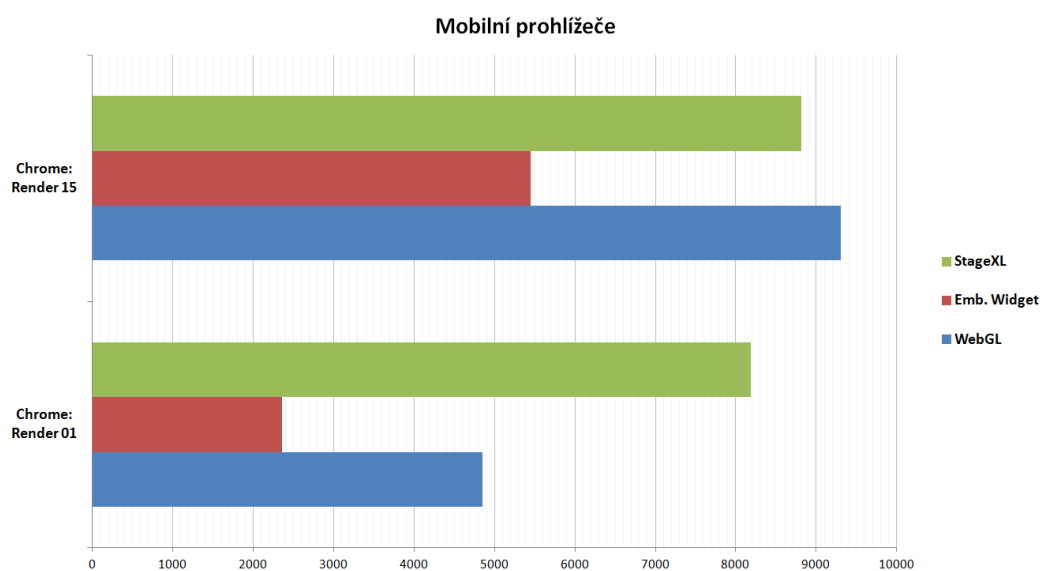
### 6.4.4 Verdikt

Z výsledků měření vyplynulo, že nejvhodnějším kandidátem je Spine Embeddable Widget. Během implementace se však ukázalo, že jedním z předpokladů této knihovny je, že na jednom plátně se vyskytuje právě jeden Spine objekt. Další komplikací byl omezený přístup k vykreslovacím funkcím. Kvůli těmto vlastnostem, které značně komplikovaly implementaci, nebyla technologie vybrána pro finální použití.



Obrázek 6.3: Graf naměřených hodnot rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Opera 42. Render 15 reprezentuje čas vykreslení poslední animace.

V testech pro desktopové prohlížeče si vedlo Spine WebGL srovnatelně jako StageXL (jen u Opery mělo StageXL lepší výsledky). Mobilní prohlížeče naopak ukázaly, že StageXL má značně zpožděné vykreslení prvního plátna, i když u Spine WebGL kvůli rychlejšímu vykreslení prvních pláten trpěla počáteční plynulost. I přes tento aspekt byla nakonec vybrána knihovna StageXL, a to z důvodů lepší organizovanosti kódu. Při implementaci byly oceněny především Dartem podporované výjimky, které napomáhají efektivně debugovat kód. V neposlední řadě toto rozhodnutí ovlivnila typovavost, která zjednodušovala práci s objekty.



Obrázek 6.4: Graf zprůměrovaného měření rychlosti načítání v milisekundách na telefonu Huawei P8 Lite. Render 01 zastupuje načtení první animace, Render 15 pak té poslední.

## Kapitola 7

# Nástroj pro tvorbu scén

### 7.1 Úlohy nástroje

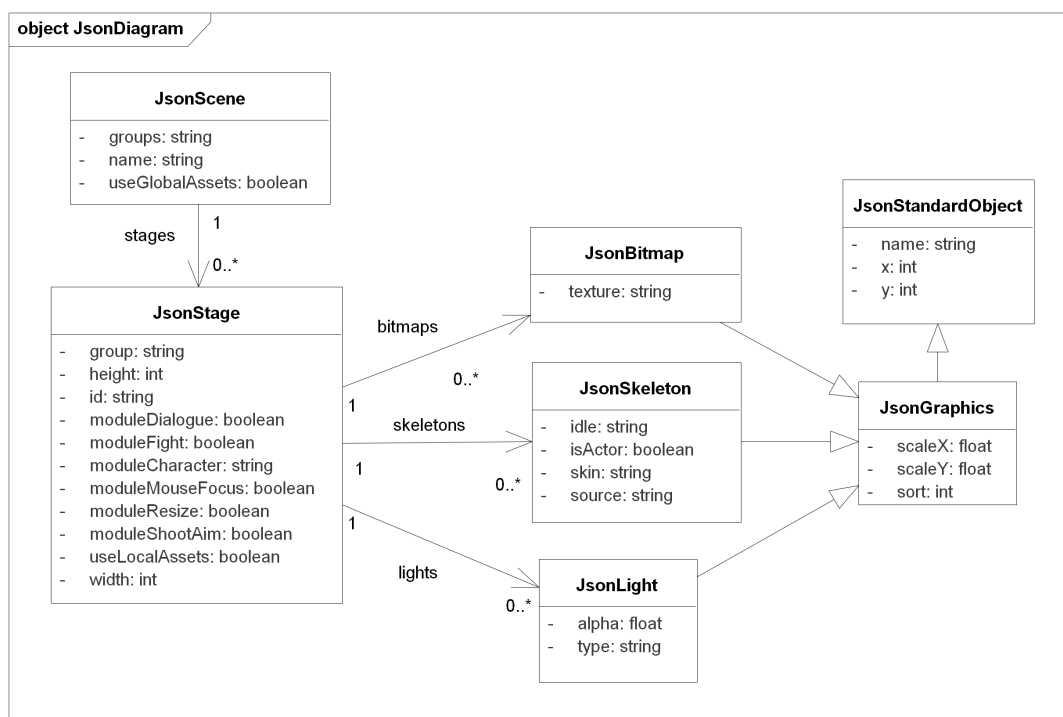
Z požadavků specifikovaných v cílech 1.2 vyplývají nároky na nástroj, který měl vzniknout pro snadné úpravy a nastavitelnost jednotlivých scén. Nejprve bylo nutné vyřešit formát konfiguračních souborů, ve kterém budou jednotlivé scény a jejich obsah popsány. Další záležitostí byl výběr způsobu editace těchto konfiguračních souborů pro scény. Posledním úkolem bylo rozmyslet, jaký typ práce by měl tento nástroj pokrýt.

### 7.2 Výběr formátu scény

Vzhledem k možné variantě, že by konfigurační nastavení chtěl upravit i uživatel, který nedisponuje nástrojem na editaci scén, bylo rozhodnuto, že bude vhodné zvolit nějaký člověkem čitelný formát. Nabízí se varianty jako XML, YAML a JSON. XML lze pro tyto účely považovat za méně přehledný, neboť by konfigurační soubory mohly být zahlceny značkami. YAML formát mě připadal v úvahu pro svou čistotu a zároveň možnost přidávat komentáře, nicméně JSON nakonec zvítězil nejen proto, že existuje mnoho knihoven pro manipulaci nad tímto formátem, ale také proto, že tento formát je preferován programem Spine (a to nejen v případě exportu animace do formátu JSON, ale i při uložení nastavení nástroje TexturePacker). Protože se s tímto formátem díky exportu animací setkávají i ostatní členové týmu, byla tato volba upřednostněna.

Jedním z prvních rozhodnutí bylo sloučit všechny konfigurační soubory jednotlivých scén do jednoho velkého a tím zlepšit orientaci v adresářích. Při nasazení tento krok navíc přispívá i k minimalizaci počtu HTTP požadavků na server. V přiloženém diagramu se nachází celá webová stránka reprezentována strukturou JsonScene, a jednotlivé scény jsou pojmenovány JsonStage.

V diagramu nejsou zahrnuty dvě další struktury. Jedná se o JsonImage, která slouží převážně k debugování – díky ní lze do scény snadno nahrát obrázek, aniž by byl součástí seskupených balíků textur. Dalším strukturou je JsonTrigger, který slouží ke spouštění událostí postavíčkou. Tato struktura nebyla nakonec implementována kvůli změnám v konceptu interaktivních scén.

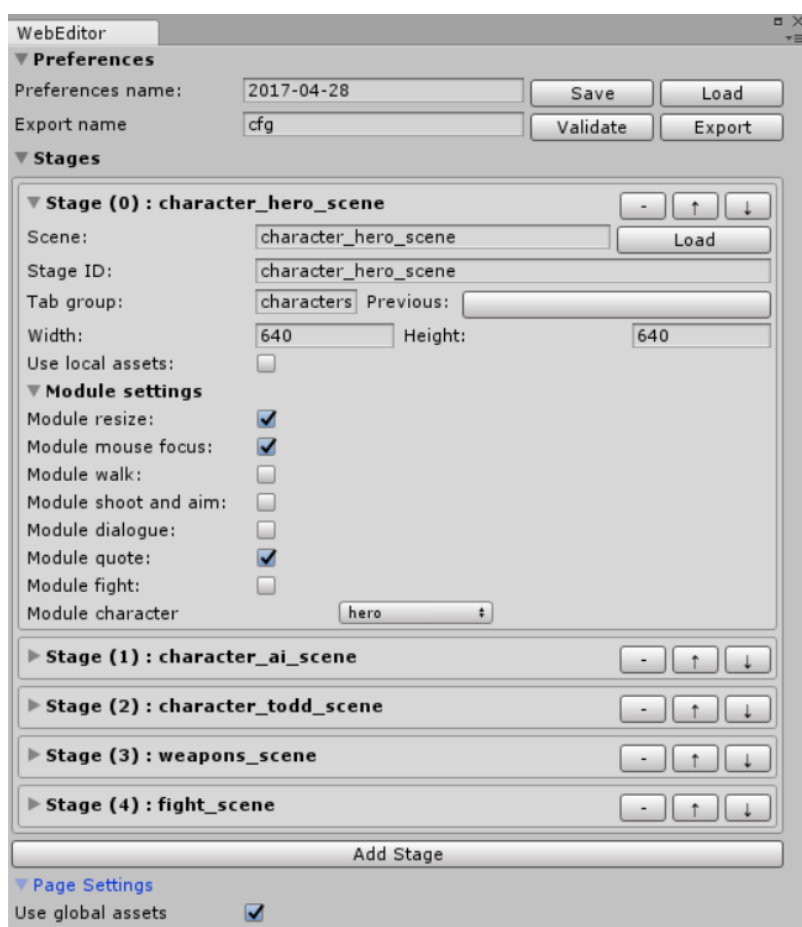


Obrázek 7.1: Schéma popisuje parametry nastavitelné v konfiguračním souboru pro webovou stránku v UML formátu jako diagram tříd.

## 7.3 Výběr technologie editoru

Z rozsáhlosti konfiguračních souborů vyplývá, že tvorba takových souborů – od základu v textovém editoru – by byla uživatelsky nepřívětivá. Hlavní otázkou tedy bylo, na jakém základu konfigurační nástroj postavit. První úvahy vedly k statické JavaScriptové stránce a k desktopové aplikaci v Javě. Po zralé úvaze byla ovšem přijata myšlenka, že bude nejvhodnější vytvořit nástroj jako editorové okno v Unity. Níže jsou uvedené hlavní důvody:

- Scény mají velmi podobný charakter jako úrovně (ve smyslu lokací). Obsahují statickou grafiku pozadí, animované objekty a světla. U těchto objektů lze navíc snadno nastavit množství parametrů jako pozice aj., a případně přidat další parametry v rámci skriptů.
- Bude možné zobrazit si náhled scény dříve, než bude otestována na webové stránce.
- Členové týmu jsou obeznámeni s prací v Unity a jsou zvyklí v rozhraní editoru pracovat. Nemalou výhodou je, že nikdo nebude nucen instalovat další externí nástroj, se kterým by se byli nuceni naučit zacházet a všechny prvky budou v blízkých adresářových strukturách.



Obrázek 7.2: Grafické rozhraní pro úpravu skriptů je vyřešeno Unity editorovým oknem.

## 7.4 Klíčové vlastnosti editoru

Bylo záměrem, aby po nastavení všech parametrů vygeneroval editor funkční webovou šablonu, kterou pak lze spravovat i se základní znalostí HTML. Níže jsou uvedené body, které specifikují, s čím dokáže editor uživateli pomoci:

- **Vizuál scény:** Uživatel v Unity Engineu vytvoří scénu tak, jako byl zvyklý tvořit úroveň. Může se tím pádem soustředit hlavně na cílový dojem. Ve scéně má možnost použít textury v pozadí a řídit jejich pořadí vykreslování. U animovaných objektů může nastavit, jaká je jejich základní animace, která se posléze bude ve smyčce přehrávat. Do scény může uživatel přidat světla a k nim příslušný skript, kde definuje jejich typ. Centrum scény, podle kterého bude scéna zobrazena na webové stránce, je dáno pozicí kamery.
- **Úpravy již vytvořených scén:** Jednotlivé scény lze uložit jako úrovně. Nastavení exportu scén na web lze uložit ve formátu JSON, a tím pádem lze změny propagovat jedním kliknutím.



- **Nastavení vlastností scény:** Uživatel může nastavit, které vlastnosti implementované ve skriptu webové stránky chce u dané scény použít. Stejně pak nastavuje základní konfiguraci, jakou je například výška a šířka scény.
- **Export do HTML:** Výsledné scény lze v grafickém rozhraní editoru seřadit a posléze exportovat. Vygenerované materiály se uloží do adresářové struktury, kterou lze přímo použít pro webovou prezentaci. Export obsahuje – mimo HTML soubory – také optimalizované textury pro načtení (viz další bod) a konfigurační soubor ve formátu JSON.
- **Optimalizace textur:** Editor seskupí všechny obrázky z dané scény a vytvoří z nich seskupený balík textur ve formátu PNG s příslušným atlasem, a to pomáhá rychlejšímu vykreslování na webové stránce. Preferovanou možností je seskupit všechny textury napříč scénami do jednoho balíku s názvem *all\_assets0.png* s příslušným atlasem *all\_assets.atlas*.

Editor se nezabývá tvorbou textových článků, které jsou v částech vložených mezi jednotlivými scénami na výsledné stránce. Jejich vzhled bývá zpravidla rozumnější doladit úpravou CSS, a tak je i jejich přidání do stránky řešeno manuálně. Šablony pro tvorbu těchto článků jsou ovšem vytvořeny, a jedná se tak ve většině případů o jejich přkopírování, modifikaci textu a úpravu drobností v CSS.

Zdrojový kód editoru je k dispozici na GitHubu:

<https://github.com/trnenmic/Spine-Unity-Tool>

## Kapitola 8

# Webová stránka

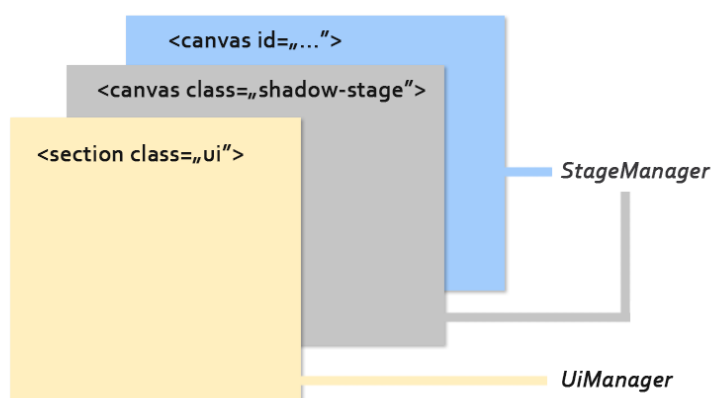
### 8.1 Scény

#### 8.1.1 Obecný popis scén

Každá scéna se skládá ze tří vrstev. První vrstvou je hlavní plátno, na němž se odehrávají animace postav a na kterém jsou zobrazeny textury pro pozadí, které se v případě aktivního modulu pro chůzi mohou posouvat. Nad ním je vrstva stínového plátna, které má v základu černou barvu s nízkou průhledností. Od tohoto plátna jsou pak pomocí módu prolnutí typu ERASE odečteny textury světla. Intenzita světla je tak dána svým alfa kanálem. Nad touto vrstvou je vrstva ovládacích prvků.

Výsledná kompozice stránky je k nahlédnutí zde:

<https://www.digital-shard.com/test/2017-05-11/template.html>



Obrázek 8.1: Vrstvy scény jsou zobrazeny přes sebe pomocí absolutního pozicování. Přední vrstvu obaluje třída `UiManager` a dvě zadní zase třída `StageManager`.

Scéna je přiřazena vykreslovací smyčce jen tehdy, pokud je alespoň nějaká její část vidět v okně prohlížeče. Plátno se světlý se navíc aktualizuje jen při specifických událostech, jakými jsou například chůze hlavní postavy nebo změna velikosti okna, protože tyto události vynucují překreslení světla. V případě, že se tak nedeje, k překreslování světla nedochází.

Pro většinu scén je vhodné použít modul pro změnu velikosti okna. Ten zajišťuje přepočty pozic objektů na hlavním plátně tak, aby hlavní postava byla neustále ve středu plátna, nehlédě na rozlišení okna prohlížeče.

Dalším významným modulem, který je vhodné použít téměř u jakékoliv scény, je modul pro focus myši (focus zde reprezentuje zaměření HTML elementu). Ten zajišťuje, že hlavní plátno získá focus i po pouhém přejetí myši, čímž se umožní ovladatelnost plátna pomocí kláves. Získáním focusu se mění také ztmavení okrajů, které je tvořeno pomocí radiálního CSS gradientu.

Pro všechna plátna také platí, že během načítání se na nich přehrává CSS animace ve formě točícího se kruhu. V této chvíli je pozadí scény nahrazeno obrázkem, který odpovídá scéně, ve které by byly postavy statické. Obrázek je z logiky věci nutné vytvořit manuálně a přiřadit jej jako pozadí hlavního plátna pomocí ID v CSS stylu.

### 8.1.2 Souboj s monstrem

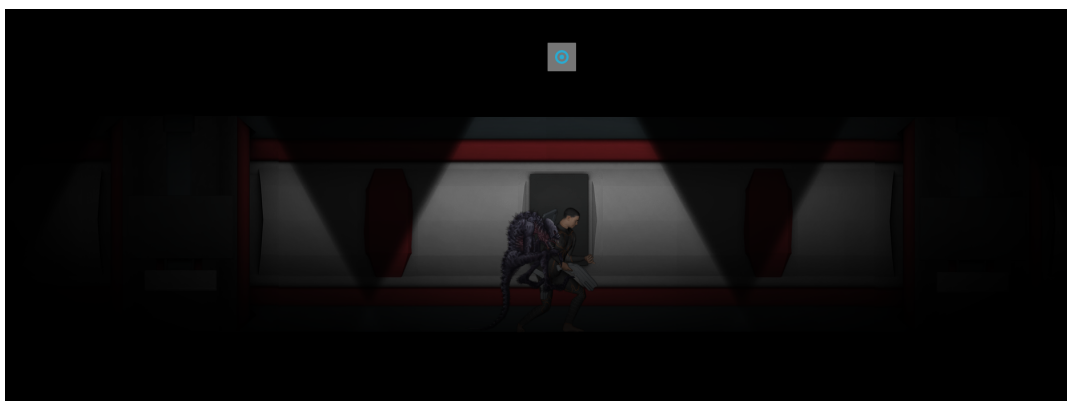
Na počátku je namísto scény zobrazené okno, ve kterém je uživatel informován o tom, že stiskem speciální ikony bude moci bránit hlavní postavu. Po stisku červeného tlačítka je pak puštěn do scény samotné, kde je protagonista napaden monstrem. Návštěvník může monstrem odrazit kliknutím na blikající tlačítko, anebo kliknout nestihne, a tím pádem monstrem protagonistu usmrtí. Pak se na obrazovce objeví další okno s hláškou reflektující výsledek. Situace se opakuje třikrát, pořadí pokusu je indikováno množstvím modrých obdélníků. Každá situace využívá odlišné animace. Třetí opakování je zamčené. Pro jeho odemknutí je zapotřebí zadat devítimístný kód, který lze zjistit v dialogové scéně. Po závěrečném kole se zobrazí zpráva s blahopřáním.

Ovládání scény je postaveno na odchyťávání událostí, které jsou ve Spine přidány přímo do animace.

Scéna je ovládána modulem souboje. Pro upozornění na tlačítko, které slouží k odražení monstra, je použita CSS animace. Odemknutí poslední scény je podmíněno zadáním správného kódu do vstupního pole. Ikona pro odražení je typu SVG, a tak se i mění její barevnost.

### 8.1.3 Demonstrace zbraní

Scéna demonstruje výměnu hlavních zbraní. Při kliknutí na jednu z nich dojde k výměně zbraně aktuální za nově vybranou a tato změna se projeví jak na protagonistovi, který u této situace přehraje animaci, tak na horním pruhu elementu výběru zbraně,



Obrázek 8.2: Ukázka interaktivní scény, ve které uživatel může odrazit nepřítele stisknutím tlačítka.

a to zbarvením do modra. Další interakcí je míření, které se děje při najetí myši na plátno a střílení, které je realizováno kliknutím. Tato funkcionalita je implementována v modulu střílení a míření. V této scéně je povolena také chůze, která je ovládána šipkami. Ta je kontrolována modulem chůze.



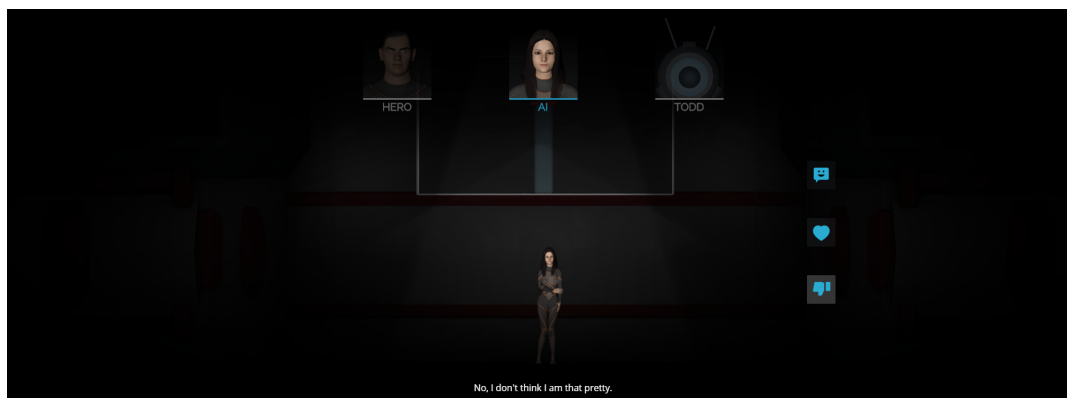
Obrázek 8.3: Ukázka scény se zbraněmi. Animace střelby je realizována pomocí inverzní kinematiky.

Uvedená scéna staví na mixování animací. Postava totiž zároveň přehrává animaci dýchání, chůze a míření, respektive střílení.

O zbraních lze zjistit více kliknutím na speciální tlačítka. Pro upoutání pozornosti na jejich přítomnost je na ně aplikovaná CSS animace změny barvy pozadí. Při najetí na ně se současně mění jejich velikost pomocí CSS vlastnosti transform. Tato tlačítka vynutí přítomnost modálního okna, které zakryje zbytek obrazovky. Panely se zbraněmi jsou responzivní a při malém rozlišení se změní počet aktuálně zobrazených zbraní na jednu a přepínají se pomocí šipek na stranách.

### 8.1.4 Dialogy s postavami

Více dialogových scén tvoří obvykle skupinu, která je řízena přepínačem postav. Přepnutím postavy tak dojde ke skrytí aktuální scény a zobrazení scény nové. Přepínač postav obsahuje portréty a jména protagonistů. Aktuální postava je indikována barevným pruhem a změnou průhlednosti. Každá postava pak přehrává svou stojící animaci.



Obrázek 8.4: Ukázka dialogové scény. K dispozici jsou tři možnosti dialogové interakce.

U každé postavy je k dispozici rozhraní pro komunikaci, které je založeno na třech tlačítkách s různými SVG ikonami. Interaktivita těchto tlačítek je podpořená CSS animací měnící barvu pozadí. Tlačítka spouští animace, které postavy posléze přehrají. Zde je využito blendování animací. Navíc se také zobrazí replika postavy, která je realizována jako HTML paragraf element, a je tedy snadno upravitelná v rámci HTML kódu. Dle CSS tříd, které návrháři dialogu přiřadí danému paragrafu, může dodat replikám ještě další funkcionalitu, která je z části závislá na tom, pro jakou postavu je aktuálně modul nastaven. Postava tak může například odejít z plátna a nadále se jen uraženě hádat s návštěvníkem.

Dalším rozšířením interaktivity je reakce postav na kliknutí, které posléze postavy začnou rovněž komentovat. Navíc je implementováno počítadlo kliknutí, takže po vytrvalém klikání lze od postav získat ještě speciální reakci. Postavy také přehrají animaci (a případně přidají repliku), pokud uživatel používá posuvník. Poslední specialitou je využití techniky Session Storage, které lze využít (podobně jako se v jiných případech používá Cookies) k uchování informace o procesech dějících se na stránce. Tímto způsobem postavy reagují na stav dialogu, navzdory tomu, že stránka byla obnovena.

### 8.1.5 Další scény

Prostor pro tvorbu dalších sekcí je vymezen pomocí modulů. Existuje modul dialogů, který umožňuje vést konverzaci, ve které uživatel vybírá z připravených odpovědí, a celý dialog se tedy podle toho větví. Modul byl původně zamýšlen jako úvodní scéna,

ve které by hlavní protagonista procházel místností, bavil se s dalšími postavami a řešil drobnou hádanku. Celá takto navržená scéna by však byla pro neznalého návštěvníka buďto příliš komplexní, anebo by po jejím zjednodušení byla zavádějící (tj. lišila by se od principů zavedených ve hře End of Line). V případě, že by však vznikl smysluplný návrh pro tento typ scény, dialogový modul by k těmto účelům dobře posloužil.

## 8.2 Přehled článků

### 8.2.1 Úvodní článek

Úvodní článek slouží k představení projektu. Návštěvník je informován o klíčových vlastnostech projektu a o možnosti si produkt koupit. Tlačítko pro koupi využívá CSS animaci, kvůli které tlačítko jednou za několik vteřin zapulsuje. Článek obsahuje přichycené pozadí, které je řešeno pomocí techniky will-change v kombinaci s CSS pseudoelementem before, čímž bylo dosaženo plynulejšího přesunu stránce.[15] Jedním z designových experimentů je využití trojúhelníkových tvarů namísto standardních obdélníkových, které jsou pro webové stránky typičtější.

Úvodní článek obsahuje zakomponované YouTube video. Celý článek je responzivní a změna velikosti jednotlivých prvků při změně rozměrů okna je zpravidla plynulá díky využití CSS transition.

### 8.2.2 Článek s baterkou

Článek s baterkou představuje světla, která bylo zapotřebí prezentovat jako klíčovou vlastnost nadefinovanou v cílech (kapitola 1.2). Pozadí článku je tvořeno dvěma vrstvami, kde zadní vrstva zahrnuje grafiku baterky a přední vrstva obsahuje světelný paprsek tvořený průhlednou rastrovanou grafikou. Průhlednost světelného paprsku je při přesunu po stránce animována v jazyce Dart, respektive po jeho kompilaci JavaScriptem. Animuje se také přilet textu. Článek je responzivní a prvky tak mění svou pozici podle velikosti rozlišení.

### 8.2.3 Článek s vesmírnou stanicí

Článek obsahuje přichycené pozadí s pohledem na vesmírnou planetu. Přechod do článku je oživen trojúhelníkovým překrytím seshora a lineárním gradientem zezdola. Samotná vesmírná stanice je animována pomocí CSS animace.

## 8.3 Načítání stránky

Je žádoucí, aby byl uživatel informován o tom, že stránka ještě není plně funkční. Pro indikaci načítání slouží horní lišta podobná té, která je na stránkách YouTube. Aby uživatel nepřicházel o zážitek během načítání, plátna obsahují náhradní dočasný

obrázek a načítací rotující kruh.

Další strategií je urychlit načítání snížením HTTP požadavků. K tomu napomáhá seskupování textur. Spine animace takto exportuje grafiku rovnou, grafika v pozadí pak byla takto exportována pomocí vytvořeného Unity nástroje. K dalším sdružováním došlo například u SVG ikon a u konfiguračního souboru scény.

Finální redukce se pak týkala samotných materiálů. Pro redukci velikosti obrázků byla použita webová služba TinyPNG, soubory typu JSON byly exportovány ze Spinu přímo v redukované, ovšem hůře čitelné, podobě. K odstranění bílých znaků pak došlo i v CSS souboru.

## 8.4 Struktura kódu

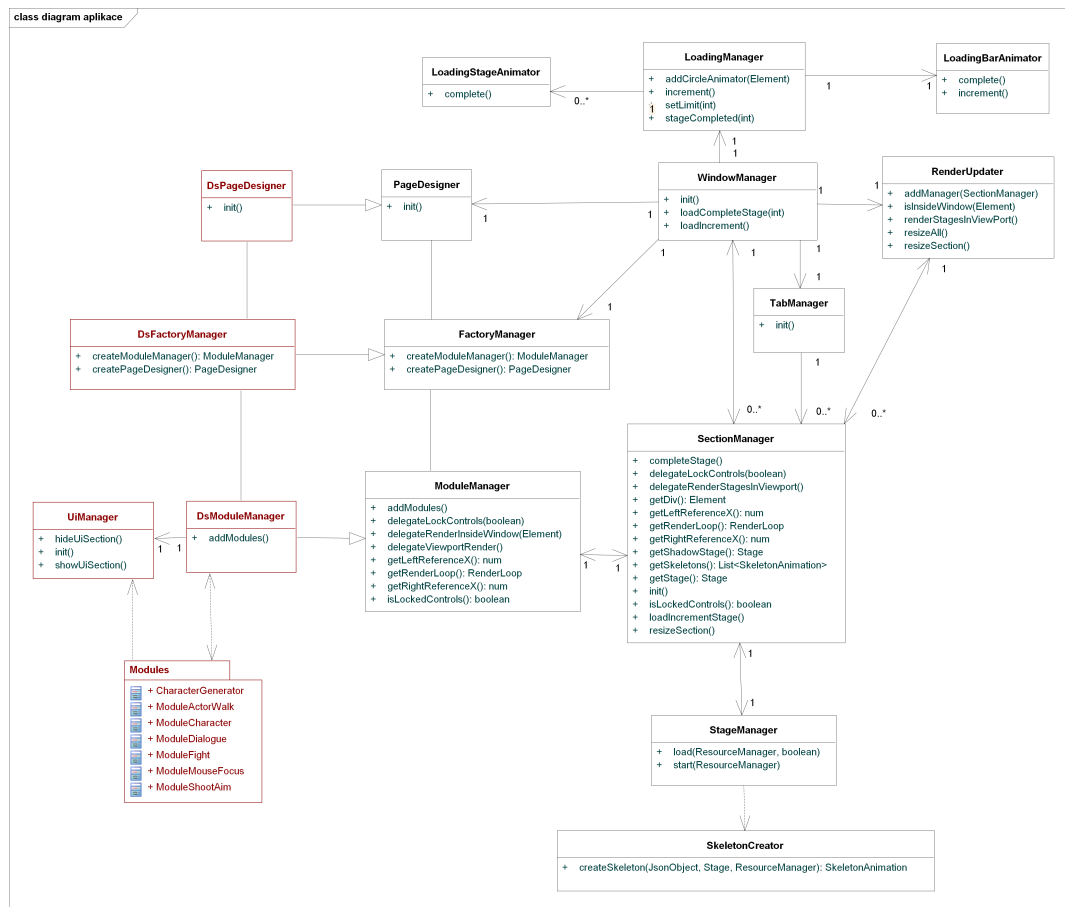
Skript ovládající scény je k dispozici na GitHubu.

<https://github.com/trnenmic/Spine-Web-Scenes>

Třídy napsané v jazyce Dart, které tvoří základní strukturu stránky, jsou sepsány níže. Jejich návaznosti jsou rozebrány ve zjednodušeném diagramu tříd, který pro větší přehlednost neobsahuje atributy. Seznam hlavních tříd:

- **WindowManager:** Zastřešuje všechny operace na webové stránce.
- **RenderUpdater:** Řídí vykreslování scén, které jsou ve viditelném výřezu.
- **PageDesigner:** Je abstraktní třídou. Po implementaci by měl řídit animace a změny CSS tříd, které se týkají článků (a nikoliv scén).
- **ModuleManager:** Je abstraktní třídou. Po implementaci by měl zodpovídat za interaktivitu ve scénách se skeletálními animacemi.
- **TabManager:** Zařizuje přepínání, respektive změnu viditelnosti u sdružených scén.
- **SectionManager:** Řídí celou scénu, je mostem mezi HTML elementy, logikou definovanou v modulech a vykreslovacím plátnem.
- **StageManager:** Zastřešuje stínové plátno a hlavní plátno, na které přidá všechny objekty k vykreslení.
- **LoadingManager:** Tvoří fasádu pro všechny animace aktivní během načítání.
- **LoadingBarAnimator:** Zařizuje, aby docházelo ke korektnímu přičítání hodnot v načítací liště.
- **LoadingStageAnimator:** Stará se o zobrazování rotujícího kruhu při načítání.

- **SkeletonCreator:** Poskytuje statickou metodu pro vytvoření nového animovaného objektu, který je posléze přímo připraven pro přidání na plátno.



Obrázek 8.5: Zjednodušený diagram demonstruje návaznosti tříd. Červeně jsou označeny třídy definující interaktivitu scén, která je specifická pro projekt End of Line.

Seznam modulů:

- **ModuleActorWalk:** Přidává funkcionalitu po stisku klávesy. Zastřešuje pohyb hlavního animovaného objektu.
- **ModuleDialogue:** Přidává funkcionalitu po stisku klávesy. Vede na zobrazení dialogů.
- **ModuleFight:** Řídí sekvenci animací a reaguje po kliknutí na tlačítko specifickou animací.
- **ModuleMouseFocus:** Reaguje na přítomnost myši a řídí vzhled plátna.
- **ModuleShootAim:** Umožňuje výměnu zbraní a střelbu u hlavního animovaného objektu.



- **ModuleCharacter:** Zařizuje zobrazování textových hlášek po kliknutí na tlačítka sloužící pro komunikaci s postavami reprezentovanými animovanými objekty.
- **CharacterGenerator:** Přísluší předchozímu modulu a zařizuje nastavení správných vlastností animovaných objektů.

Statické konstanty a nastavení, která jsou v rámci kódu nezměnitelná, jsou roztržena do těchto tříd:

- **ConstAnim:** Poskytuje názvy určené pro animace objektů.
- **ConstSettings:** Obsahuje nastavení cest k souborům a výchozí hodnoty některých nastavení.
- **ConstCss:** Obsahuje názvy CSS tříd, které jsou v kódu na některých místech použity k zacílení elementů.
- **ConstKeys:** Obsahuje názvy klíčů, které jsou v konfiguračním souboru ve formátu JSON.
- **ConstValues:** Obsahuje hodnoty klíčů pro konfigurační soubor ve formátu JSON.

# Kapitola 9

## Závěr

### 9.1 Shrnutí

Jedním z cílů práce bylo vybrat technologie pro manipulaci se Spine animacemi. Po analýze různých animačních technik byly předvybrány knihovny operující s animacemi ve formátu JSON pomocí JavaScriptu. Podrobněji byly rozebrány knihovny Spine StageXL, Spine WebGL a Spine Embeddable Widget. U těchto knihoven bylo následně provedeno měření výkonnosti. Nejrychleji docházelo k načítání animací u Spine Embeddable Widget, ale knihovna podporovala pouze jednu animaci v rámci plátna. Technologie Spine StageXL a Spine WebGL si vedly ve většině testů podobně – s tím rozdílem, že Spine WebGL bylo rychlejší na mobilních zařízeních. I přesto byla nakonec pro implementaci vybrána knihovna Spine StageXL, neboť poskytovala nejlepší nástroje pro údržbu kódu.

Hlavním cílem práce pak bylo vytvořit generátor pro skeletální animace, respektive jejich export na web. Výsledný nástroj je rozšířením Unity na bázi editorového okna. Tvorba interaktivních scén se skeletálními animacemi je tak velmi podobná tvorbě herních úrovní. Ovládání nástroje zajišťuje standardizované grafické rozhraní Unity editoru, ve kterém návrhář může nastavit scénám různé parametry. Nástroj umožňuje poskládat z dílčích úseků HTML kompletní šablonu stránky. Dále nástroj připraví seskupené obrázky a konfigurační soubory scén ve formátu JSON. Užitečnost nástroje se ukázala už během implementace zárodků stránky – především kvůli jednoduché úpravě konfiguračních souborů. Nad vyexportovanými soubory pracuje univerzální jádro webové stránky napsané v jazyce Dart (posléze zkompilevané do JavaScriptu). Jádro zajišťuje správnou inicializaci scén z konfiguračních souborů. Pro implementaci interaktivity vznikly nezávislé moduly specifické pro hru End of Line.

Celkem bylo vytvořeno několik modulů definujících interaktivitu. Z nich byly poskládány tři plně funkční interaktivní scény obsahující 2D skeletální animace postav. Postavy byly pro tyto účely upraveny v animačním programu Spine. Mimo samotné scény byly také připraveny tři články s animovanými efekty (typickými pro webové stránky). Těchto šest celků dohromady splňuje požadavky na prezentaci projektu End

of Line a stránka postavená na vygenerovaném základu je tak funkčním prototypem, jehož části lze v případě změn v projektu snadno upravit.

## 9.2 Budoucnost nástroje a stránky

Grafika animovaných prvků vyskytujících se na vygenerované stránce bude během následujících dvou měsíců nejspíše vyměněna za novou. Na tuto změnu je nástroj dobře připraven, neboť k podobné situaci došlo už během jeho vývoje. Během implementace generátoru jeho vývoj v některých oblastech předběhl vývoj hry, a tak přichází v úvahu varianta, že bude nutné nástroj postupně upravovat. Pro tyto účely se počítá s přidáváním dalších modulů do ekosystému stránky. Je také v plánu postupně zlepšovat automatizaci procesu postupným zvyšováním počtu nastavitelných parametrů v generátoru a tím docílit větší flexibility.

Export konfiguračních souborů je plně připraven pro použití i v rámci jiných projektů postavených na bázi Spine a Unity. Podobně jsou na tom i moduly, které jsou navrženy tak, aby jejich funkcionalita mohla být jednoduše změněna pro účely jiných projektů. Aby byl nástroj dostatečně univerzální, bude ovšem ještě zapotřebí rozšířit parametry generátoru a umožnit tvůrci stránek jednodušší manipulaci se zdrojovými soubory pro HTML export. Aktuální verze generátoru budou nadále přidávány na GitHub.

U vygenerované stránky dojde před samotným vydáním hry ještě k úpravě stylu. Finální podoba stránky bude uzavřena až po dokončené betaverze hry. Teprve v tomto období je v plánu stránku optimalizovat pro mobilní prohlížeče – je v úmyslu připravit pro ni odlišný stylopis, který zjednoduší čitelnost textů a ovladatelnost scén a případně omezí náročné animace.

# Literatura

- [1] *PNG vs SVG for sprites* [online]. [cit. 6.5.2017]. Dostupné z: <http://codepen.io/adrianosmond/pen/LCogn>.
- [2] *Dart programming language* [online]. [cit. 6.5.2017]. Dostupné z: <https://www.dartlang.org/>.
- [3] *Using CSS animations - CSS / MDN* [online]. [cit. 6.5.2017]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Animations/Using\\_CSS\\_animations](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Animations/Using_CSS_animations).
- [4] *window.requestAnimationFrame() - Web APIs / MDN* [online]. [cit. 6.5.2017]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame>.
- [5] *Spine: In Depth* [online]. [cit. 6.5.2017]. Dostupné z: <http://esotericsoftware.com/spine-in-depth>.
- [6] BECKER, A. *12 Principles of Animation* [online]. [cit. 6.5.2017]. Dostupné z: [https://www.youtube.com/playlist?list=PL-b0h8btec4CXd2ya1NmSKpi92U\\_16ZJd](https://www.youtube.com/playlist?list=PL-b0h8btec4CXd2ya1NmSKpi92U_16ZJd).
- [7] BERGER, T. *Say goodbye to Unity Web Player • Nordic Game Community* [online]. Nov 2015. [cit. 6.5.2017]. Dostupné z: <http://nordicgame.com/say-goodbye-to-unity-web-player/>.
- [8] BRUNI, E. *The ultimate guide to Web animation* [online]. May 2015. [cit. 5.5.2017]. Dostupné z: <http://www.webdesignerdepot.com/2015/05/the-ultimate-guide-to-web-animation/>.
- [9] CULLEN, E. *HTML5 vs Adobe Flash* [online]. Jun 2016. [cit. 6.5.2017]. Dostupné z: <http://hub.bannerflow.com/online-display-advertising-hub/html5-vs-adobe-flash>.
- [10] DOYLE, J. *Myth Busting: CSS Animations vs. JavaScript* [online]. Jan 2014. [cit. 6.5.2017]. Dostupné z: <https://css-tricks.com/myth-busting-css-animations-vs-javascript/>.
- [11] IVANOV, I.-A. *Gamasutra - Practical Texture Atlases* [online]. [cit. 6.5.2017]. Dostupné z: [http://www.gamasutra.com/view/feature/2530/practical\\_texture\\_atlases.php](http://www.gamasutra.com/view/feature/2530/practical_texture_atlases.php).

- [12] MECHNER, D. – MECHNER, J. *Prince of Persia rotoscopy* [online]. YouTube, Jan 2009. [cit. 6.5.2017]. Dostupné z: <https://www.youtube.com/watch?v=WAjRNU3DbSY>.
- [13] PATHAN, J. *5 reasons you should be using SVG's over PNG's » WATB Creative* [online]. Jun 2016. [cit. 6.5.2017]. Dostupné z: <https://watb.co.uk/5-reasons-you-should-be-using-svg-s-over-png-s/>.
- [14] PICHLER, B. *StageXL for Dart* [online]. [cit. 6.5.2017]. Dostupné z: <http://www.stagexl.org/>.
- [15] RUPPEL, C. *Fix scrolling performance with CSS will-change property - Four Kitchens* [online]. Jul 2015. [cit. 8.5.2017]. Dostupné z: <https://www.fourkitchens.com/blog/article/fix-scrolling-performance-css-will-change-property/>.
- [16] SHAPIRO, J. *Web animation using JavaScript: develop & design*. San Francisco : Peachpit Press, 2015.
- [17] SIMON, M. *Producing independent 2D character animation: making and selling a short film*. Boston : Focal P., 2003.
- [18] THOMAS, F. – JOHNSTON, O. *The illusion of life Disney animation*. New York : Hyperion, 1995.

## Příloha A

# Seznam použitých zkratek

API Application programming Interface

CSS Cascading Style Sheets

GIF Graphics Interchange Format

HW Hardware

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation

PNG Portable Network Graphics

UI User Interface

XML Extensible Markup Language

YAML YAML Ain't Markup Language

## Příloha B

# Tabulky

### B.1 Měření na desktopovém zařízení

Tabulka B.1: Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Chrome 55. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

Render 01			Render 15			Render delta			Load		
<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>
1329	981	1208	1467	1363	1363	138	382	155	566	215	528
1390	900	1255	1537	1326	1404	147	426	149	660	163	542
1271	826	1237	1399	1262	1366	128	436	129	583	185	543
1372	812	1280	1517	1107	1412	145	295	132	609	202	521
1456	832	1300	1612	1195	1433	156	363	133	684	153	575
1371	828	1268	1505	1237	1403	134	409	135	632	183	549
1292	849	1349	1428	1252	1500	136	403	151	611	158	548
1321	818	1345	1456	1238	1499	135	420	154	604	163	597
1276	853	1330	1413	1243	1477	137	390	147	599	184	629
1300	939	1229	1436	1341	1378	136	402	149	543	180	530

Tabulka B.2: Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Firefox 50. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

Render 01			Render 15			Render delta			Load		
<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>
1581	860	1528	1874	1316	1811	293	456	283	1515	60	1465
1552	732	1532	1848	1168	1838	296	436	306	1485	64	1467
1548	719	1548	1845	1199	1853	297	480	305	1482	62	1483
1590	849	1562	1885	1262	1865	295	413	303	1522	70	1494
1609	743	1577	1902	1186	1879	293	443	302	1545	63	1494
1688	720	1589	1986	1128	1890	298	408	301	1620	57	1524
1549	718	1593	1862	1108	1901	313	390	308	1467	60	1510
1611	885	1555	1910	1285	1859	299	400	304	1547	71	1477
1642	719	1559	1939	1136	1862	297	417	303	1578	63	1492
1642	731	1565	1941	1138	1869	299	407	304	1576	61	1503

## PŘÍLOHA B. TABULKY

Tabulka B.3: Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Microsoft Edge 14. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

Render 01			Render 15			Render delta			Load		
<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>
1241	685	1225	1473	859	1452	232	174	227	470	90	505
1262	677	1196	1493	878	1419	231	201	223	480	85	493
1237	686	1215	1489	927	1453	252	241	238	464	90	507
1229	703	1251	1466	895	1478	237	192	227	482	112	455
1249	710	1235	1505	921	1473	256	211	238	453	106	459
1230	693	1261	1478	890	1499	248	197	238	480	92	480
1330	681	1208	1535	887	1446	205	206	238	448	86	454
1217	707	1179	1449	909	1413	232	202	234	483	92	450
1226	682	1247	1459	870	1455	233	188	208	462	93	483
1264	682	1243	1504	870	1474	240	188	231	472	93	460

Tabulka B.4: Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Lenovo Thinkpad E540 s Windows 10 v prohlížeči Opera 42. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

Render 01			Render 15			Render delta			Load		
<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>	<i>GL</i>	<i>Emb.</i>	<i>XL</i>
1848	845	1448	2543	1226	1602	695	381	154	1664	184	744
1994	1034	1656	2706	1403	1804	712	369	148	1730	184	754
1999	955	1305	2775	1343	1457	776	388	152	1795	183	568
1888	856	1388	2625	1281	1561	737	425	173	1694	196	654
1858	847	1289	2590	1270	1452	732	423	163	1666	180	594
1920	921	1326	2675	1350	1469	755	429	143	1668	191	620
1959	842	1305	2647	1261	1452	688	419	147	1739	178	587
1735	873	1302	2502	1324	1446	767	451	144	1525	195	596
1923	834	1307	2540	1251	1470	617	417	163	1734	188	600
1912	902	1362	2662	1312	1504	750	410	142	1727	210	615

Tabulka B.5: Zprůměrované měření rychlosti načítání na desktopovém zařízení v milisekundách. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

	Render 1			Render 15			Render delta		
	GL	Emb.	XL	GL	Emb.	XL	GL	Emb.	XL
<b>Chrome</b>	1338	864	1280	1477	1256	1424	139	392,6	143
<b>Firefox</b>	1601	768	1561	1899	1193	1863	298	425	302
<b>Edge</b>	1249	691	1226	1485	891	1456	236	200	230
<b>Opera</b>	1904	891	1369	2627	1302	1522	723	411	153



## B.2 Měření na mobilních zařízeních

Tabulka B.6: Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Sony Xperia E5 se systémem Android 6.0.0 v prohlížeči Chrome 58. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

Render 01			Render 15			Render delta			Load		
GL	Emb.	XL	GL	Emb.	XL	GL	Emb.	XL	GL	Emb.	XL
2767	1624	6398	6092	4087	6803	3325	2463	405	2191	313	1881
2636	1627	6446	6422	3841	6893	3786	2214	447	2010	337	1660
2672	1558	6365	6446	3887	6735	3774	2329	370	2085	346	1704
2609	1575	6493	6121	3974	6857	3512	2399	364	2016	362	1526
2523	1546	6653	6003	4065	6989	3480	2519	336	1996	310	1563
2560	1551	6112	6091	4144	6614	3531	2593	502	1918	385	1665
2619	1508	6126	6275	4048	6518	3656	2540	392	1968	330	1473
2532	1500	6178	6336	4017	6562	3804	2517	384	2006	303	1492
2622	1482	5956	6049	3791	6438	3427	2309	482	1955	316	1459
2583	1508	6133	6131	4088	6510	3548	2580	377	1937	316	1470

Tabulka B.7: Naměřené hodnoty rychlosti načítání v milisekundách. Měření provedeno na zařízení Huawei P8 Lite se systémem Android 6.0.0 v prohlížeči Chrome 58. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

Render 01			Render 15			Render delta			Load		
GL	Emb.	XL	GL	Emb.	XL	GL	Emb.	XL	GL	Emb.	XL
4974	2635	8340	9474	5820	9026	4500	3185	686	3951	516	2051
5009	2363	7813	9955	5666	8425	4946	3303	612	4063	355	1961
4810	2904	8303	9430	5944	8917	4620	3040	614	3951	892	1997
4970	2225	8103	9168	5296	8716	4198	3071	613	3929	324	1888
4707	2255	8301	8651	5109	8954	3944	2854	653	3872	258	1930
4402	2245	8212	9117	5170	8817	4715	2925	605	3676	212	1785
5247	2278	8092	9402	5301	8708	4155	3023	616	4198	330	1814
4932	2259	8225	9256	5427	8834	4324	3168	609	3789	209	1767
4790	2145	8085	9117	5326	8697	4327	3181	612	3873	256	1943
4638	2287	8411	9507	5413	9011	4869	3126	600	3804	325	1899

Tabulka B.8: Zprůměrované měření rychlosti načítání v milisekundách na telefonu Sony Xperia E5. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

	Render 1			Render 15			Render delta		
	GL	Emb.	XL	GL	Emb.	XL	GL	Emb.	XL
<b>Chrome</b>	2612	1548	6286	6197	3994	6692	3584	2446	406

Tabulka B.9: Zprůměrované měření rychlosti načítání v milisekundách na telefonu Huawei P8 Lite. Zkratka GL je použita pro WebGL, XL pro StageXL a Emb. pro Embeddable Widget.

	Render 1			Render 15			Render delta		
	GL	Emb.	XL	XL	Emb.	XL	GL	Emb.	XL
<b>Chrome</b>	4848	2360	8189	9308	5447	8811	4460	3088	622