

# Simulace lidské tváře ovládaná textem

*Martin Čáp, Šimon Sedláček*

B4M39MMA  
FEL ČVUT  
1. ledna 2018  
Praha

## Obsah

Úvod	1
Výběr techniky a její aplikace na český jazyk	2
Model	3
Animace a ovládací body	4
Aplikace v Unreal Engine	7
Uživatelské rozhraní a ovládání	10
Závěr	12
Reference	13

## Úvod

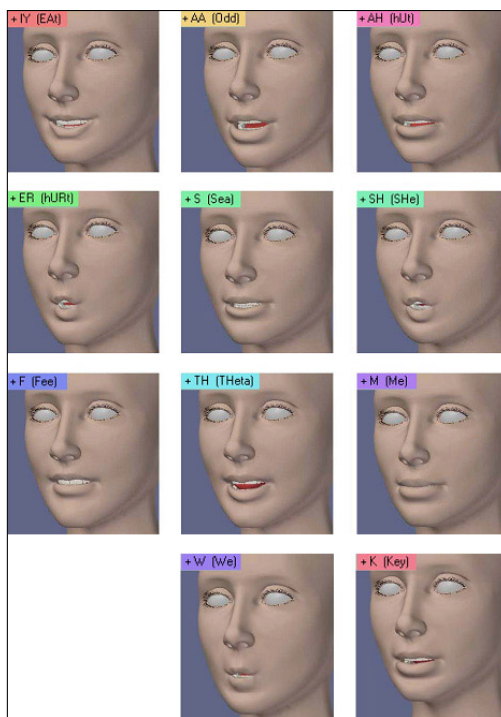
Tento projekt vznikl v rámci předmětu Multimedia a počítačová animace. Cílem bylo vytvořit interaktivní aplikaci, která dokáže převést napsaný text do animace řeči.

V práci jsme se věnovali animování 3D objektů pomocí tzv. kostí a vah a také jejich rozložení na modelu. Dalším důležitým bodem byl převod lidské řeči v českém jazyce do animací. Zde jsme použili techniku převodu fonémů na vizémy.

Posledním krokem byla kompozice daných animovaných vizémů v herním enginu (byl použit Unreal Engine, zkráceně UE) a vytvoření finální interaktivní aplikace s přívětivým uživatelským rozhraním.

## Výběr techniky a její aplikace na český jazyk

Při tvorbě naší aplikace jsme se snažili inspirovat tím, jak se takovýto problém řeší v praxi. Naším hlavním zdrojem byl článek z manuálu CryEngine 3 [1]. Jak je vidět na obrázku, jazyk lze rozložit pouze do pár vizémů, které po složení do sekvence vyjadřují libovolné slovo.



Obrázek 1: Přehled vizémů v anglickém jazyce [1].

Popis vizémů byl bohužel pouze pro anglický jazyk. A jak se ukázalo, angličtina je v tomto směru značně jednodušší jak čeština. Nicméně díky tomuto článku jsme si udělali tabulku svých požadovaných vizémů, které jsme naanimovali a zakomponovali do výsledného projektu.

Vizém	A	B	C	E	F	H	I	J	O	K	L	S	U
Skupina písmen	a	b, m, p	c, d, g, ch	e	f, q, v	h, n	i, y	j	o	k	l, r, t	s, x, z	u

Tabulka 1: Tabulka vizémů a jejich korespondujících písmen v českém jazyce.

Abychom dosáhli vysoké autenticity a naturálního vzhledu řeči, byly vytvořeny pro všechna písmena abecedy samostatné animace, vycházející ze společných charakteristik skupin vizémů. Pro některá písmena byly navíc vytvořeny různé verze animace pro zvýšení variability vzhledu řeči.

## Model

Pro tvorbu modelu a animací byl zvolen 3D modelář Blender. Blender je výkonný free source software, se kterým jsme oba obeznámeni. Model byl vytvořen ve volném čase před kurzem, ale nebyl doposud použit v žádné školní ani komerční práci. Byl navíc upraven pro jednoduchý přenos do Unreal Engine změnou jeho velikosti a měřítka scény. Blender totiž používá jako výchozí jednotku metr, zatímco Unreal Engine má cm.

Texturování postavy bylo též provedeno manuálně a to speciální technikou kresby textur na samotný 3D model v Blenderu. Pro vytvoření realistických výsledků byla použita tzv. Stencil textura, jež umožňuje kreslit projekci 2D textury na 3D model [2]. Pro vyhlazení různých ostrých přechodů a dalších problematických částí pak bylo možné udělat další manuální úpravy pomocí rozmazání nebo např. pomocí Clone tool.

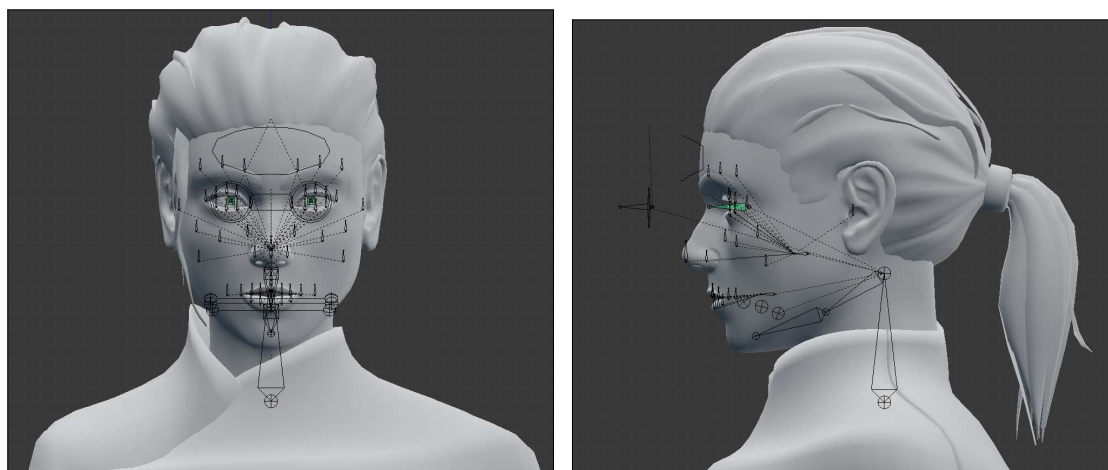


Obrázek 2: Render vytvořeného modelu.

## Animace a ovládací body

Pro tvorbu animací bylo vytvořeno 63 ovládacích bodů (v Blenderu klasicky zvané bones - kosti) uspořádaných v jedné skupině zvané armature [3]. Rozložení bodů bylo zvoleno především pro tvorbu přesvědčivých výrazů lidské tváře (vizémů, jež reprezentují jednotlivé výrazy při vyslovování slabik či jednotlivých písmen). Pro tvorbu animací bylo nutné rozhodnout, zda bude postava mluvit anglicky nebo česky. Protože je kurz MMA vyučován v češtině, byla zvolena jako výchozí jazyk pro tento projekt.

Každý řídicí bod ovládá část obličeje. Tato část je určena tzv. váhami (weights) [5], které náleží každému bodu (vertex) sítě tváře. Pro každý řídicí bod Blender navíc automaticky vytváří tzv. vertex groups, jež jsou skupiny bodů, které jsou do určité míry ovlivňovány tímto řídicím bodem. Všechny váhy (a tím i vertex groups) byly vytvořeny manuální kresbou na 3D model v Blenderu. Tento proces byl iterativní, kde po každé iteraci bylo nutné kvalitu řídicích bodů otestovat pózováním obličeje.

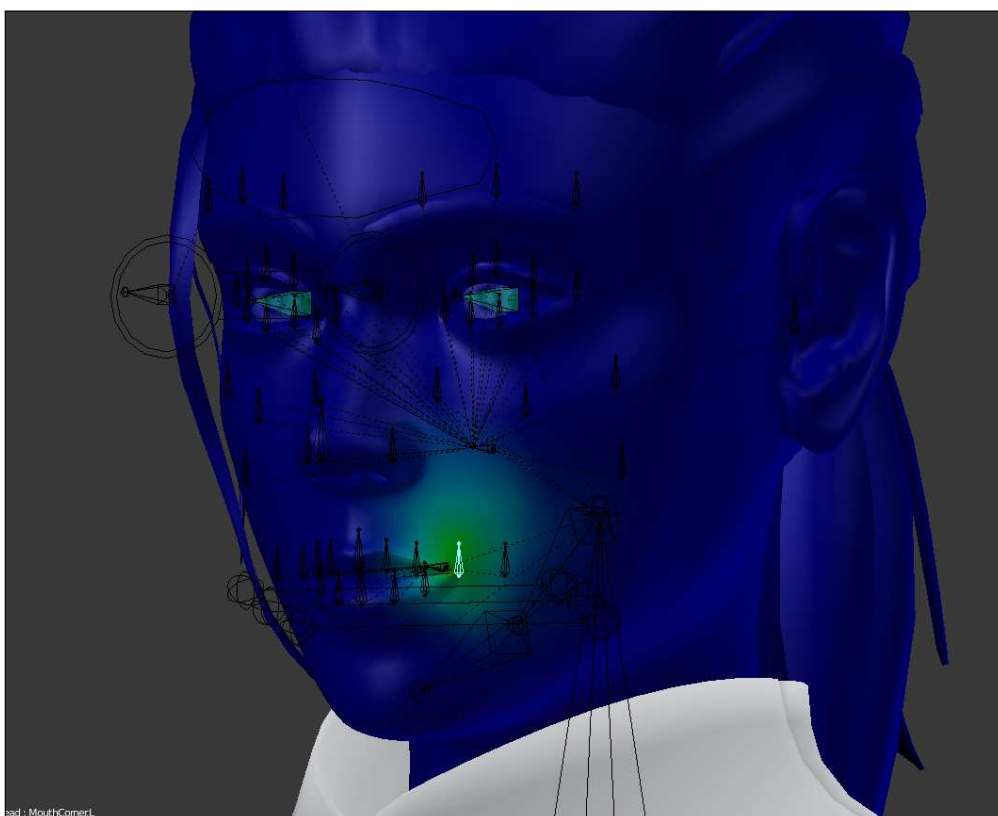


(a) Pohled zepředu.

(b) Pohled z boku.

Obrázek 3: Řídící body modelu obličeje.

Důležité je zmínit, že pro symetrické body byly váhy vytvořeny jen pro jednu stranu a automaticky přeneseny na druhou stranu obličeje pomocí operace Armature → Symmetrize. Čtenáře je třeba upozornit, že pro úspěšné provedení této operace je nutné řídicí body jmenovat podle konvence Blenderu tak, že pokud je na levé straně modelu, končí suffixem “.L”, nebo naopak, pokud je na pravé straně, končí suffixem “.R”. Bohužel tento postup dále vyžaduje kopii vertex grupy symetrizovaného ovládacího bodu, její



Obrázek 4: Váhy levého koutku úst postavy.

symetrizace (Mirror Vertex Group) a následně manuální přejmenování. Např. pro grupu zvanou MouthCorner.L je nutné přejmenovat kopii MouthCorner.L\_Copy na MouthCorner.R. Blender si reference totiž udržuje podle názvů těchto grup. Tímto procesem lze úspěšně symetrizovat všechny řídicí body a jejich váhy vlivu na obličej.

Za povšimnutí stojí, že většina řídicích bodů je situována okolo úst postavy. Zde byl kladen největší důraz na dobrou kontrolu nad animací modelu. Pro vyjádření výrazů ve tváři má pak postava ovládací prvky i na čele, obočí, očích, očních víčkách atp.. Jednotlivé ovládací prvky budou probrány dále. Ovládací prvky jsou symetrické, ale je možné každý z nich editovat zvlášť.

Co se týče různých pohybových omezení řídicích bodů (constraints), nebylo jich aplikováno mnoho. To je způsobeno především tím, že námi zvolená platforma pro tvorbu finální aplikace (Unreal Engine) omezení v Blenderu nepodporuje. Omezení byla tedy použita především v případech, kdy jejich přítomnost umožní jednodušší práci s modelem.

Na druhou stranu jsme chtěli nechat pro práci s modelem co největší volnost, a proto byla omezení uplatněna pouze na jazyk (pro jednoduchou manipulaci, neboť je schován v ústech) a na oči, kde je použito pozorování oka jiného řídicího bodu.

Jak již bylo řečeno, model má 63 řídicích bodů. 13 z těchto bodů je umístěno na ose z (v kartézském systému Blenderu) a nemají k sobě symetrický bod. Zbylých 50 je 25 symetrických párů. Z 13 asymetrických bodů tvoří 3 základ hlavy (krk, čelisti), 2 jsou pomocné pro míchání animací v UE a obecně organizaci (jeden pod sebe uspořádává body pro ovládání úst, druhý slouží pro seskupení řídicích bodů zbytku obličeje), 3 jsou pro ovládání jazyka, jeden pro ovládání obou očí a jejich směru pohledu a zbylé slouží pro ovládání obočí, špičky nosu a středu horního a dolního rtu.

Každé oko (a jeho okolí) postavy má 10 řídicích bodů (+ již zmiňovaný bod pro pohled obou očí). 8 bodů řídí pohyb víček a krajů očí (2 v krajích, 3 na horní víčko, 3 na dolní víčko). Jeden bod slouží pro pohled samotného oka a jeden je pomocný. Pomocný bod slouží pro zachování informace orientace oka, neboť, jak již bylo zmiňováno, Unreal Engine nepodporuje omezení z Blenderu, která byla použita pro pohled oka na řídicí bod (Constraint: Damped Track).

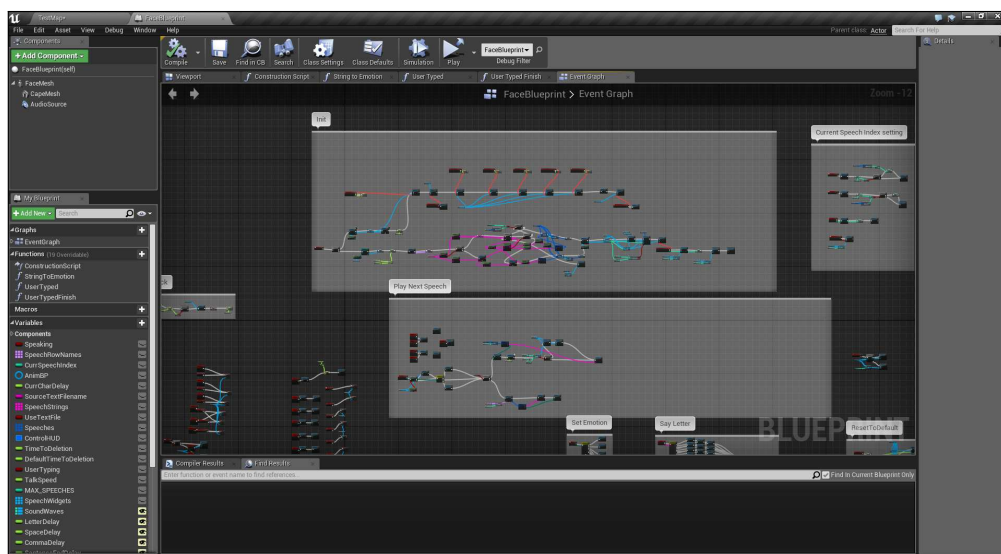
Kritickou částí obličeje jsou především ústa, která jsou hlavním aktérem naší aplikace. Pro jejich ovládání bylo vytvořeno 14 řídicích bodů nepočítaje kost pro ovládání čelisti a 3 body pro pohyb jazyka. Pohyb se rty je přímočarý, stačí s body manipulovat (posun, rotace) pro deformaci jednotlivých částí horních a dolních rtů. Důležité je také zmínit, že poloha zubů je pouze závislá na pozici čelisti postavy a není pro ní poskytnut žádný další ovládací bod, který by umožňoval provádět nerealistické operace s tváří.

Pro některé důležité řídicí body byla vytvořena vlastní grafická reprezentace jejich ovladačů (“tahadel”) v Blenderu. Toto nastavení je možné kdykoliv změnit v tabulce pro nastavení kosti pod Bones → Display → Custom Shape.



## Aplikace v Unreal Engine

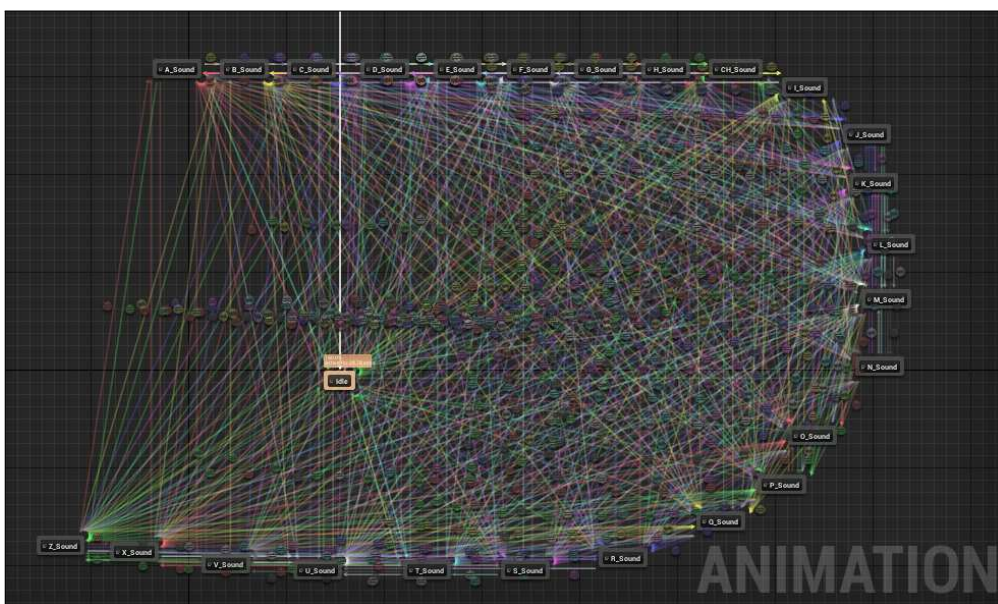
V Unreal Engine bylo použito vizuální skriptování (tzv. Blueprints). Tvář postavy je tzv. skeletální model, jež má k sobě přiřazený animační blueprint [4]. V animačním blueprintu se vyskytuje stavový automat, který nám umožňuje přecházet mezi jednotlivými vizémi. Finální animace je mix dvou stavových automatů. První je již zmiňovaný automat pro přepínání vizémů (nazvaný Visemes) a druhý pro změnu nálady a výrazu postavy (zvaný Emotions). Míchání je provedeno pomocí uzlu Layered blend per bone [6].



Obrázek 5: Část blueprintu pro zpracování textu a vstupů.

Tento přístup nám povoluje specifikovat dobu a křivku přechodu mezi jednotlivými vizémi. To samé platí pro obecné výrazy. Navíc můžeme v každém stavu náhodně (nebo podle nějakého pravidla) vybrat animaci z definované množiny animací. Díky tomu lze vytvořit přesvědčivou reprezentaci řeči, kde se neopakují pořád tytéž animace.

Důležitou součástí práce je také zadávání vstupního textu postavě. Zde jsme se rozhodli umožnit uživateli jak dynamicky zadávat text přímo na klávesnici, tak předpřipravit určité textové řetězce do samostatného souboru, který se nalézá v adresáři hry a nazývá se speeches.txt. Původně byla též implementována možnost použít speciální tabulky Unreal Engine zvané Datables, ovšem jejich použití vyžaduje, aby byly součástí již zkompilevaného projektu, a proto byla nakonec dána přednost klasickým textovým souborům. Textový soubor má jednoduchý formát, nejdříve se zadá emoce postavy (např. Neutral, Angry), následuje ukončující znak `\e` a poté lze zadat vlastní text (na libovolný



Obrázek 6: Stavový automat pro vizémy.

počet řádků). Pokud není zadána validní emoce nebo chybí oddělovač `\e`, je použita ta výchozí (Neutral). Pro oddělení jednotlivých textových položek je nutné použít `\\` podobně jako v  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u.

Co se týče čtení textu je nutné brát v úvahu více proměnných. Základní z nich je prodleva mezi jednotlivými písmeny (a tedy i vizémy) textu nazvaná v aplikaci jako `LetterDelay` (případně `DefaultLetterDelay` jakožto výchozí nastavení). Dále mezi ně patří prodleva při výskytu mezery v textu (`SpaceDelay`), čárky v textu (`CommaDelay`) nebo ukončovacího znaku věty (`., ? , ! - SentenceEndDelay`). Důležité je také zmínit, že člověk nečte každý text naprosto pravidelně. Pro simulaci tohoto faktu je do prodlev přidán randomizační faktor zvaný `TalkSpeedRandomizationRange`. Ten může nabývat hodnot z intervalu  $[0, 1]$ , kde 0 není žádná randomizace a naprosto pravidelné čtení jednotlivých znaků textu (každý je vyslovován stejnou dobu) a 1 způsobuje buď zkrácení délky trvání znaku k nule, nebo jeho dvojnásobku (vyšší hodnoty by působily nerealisticky). Na závěr byla navíc vytvořena proměnná `TalkSpeed`, jež ovlivňuje všechny tyto proměnné globálně a je zde spíše pro uživatele, který se nechce zabývat laděním jednotlivých prvků.

K textům lze též přiřadit zvukový doprovod v podobě tzv. SoundCues v Unreal Engine. Ty jsou ve skriptu uloženy v mapě podle jejich pořadového čísla (tedy první text v souboru speeches.txt má pořadové číslo 1 - indexace od 1, neboť Unreal Engine mapa vyhraňuje hodnotu klíče 0 jako výchozí hodnotu).

Jak pro předpřipravené texty, tak pro znaky zadané přímo z klávesnice uživatele byly vytvořeny titulky. Titulky se vytváří podle zrovna přečteného znaku s tím, že pokud je uživatel zadává z klávesnice, tak jsou ve vhodný moment (na konci slova, věty, nebo pokud uživatel dlouho nic nezadal) smazány. Tuto prodlevu lze opět nastavit ve skriptu a nazývá se DefaultTimeToDeletion. Pro automatický text se mažou po dosažení určitého limitu znaků (aby nepřesáhly 3 řádky textu) opět pouze na konci slova nebo nejlépe věty.

U postavy je též možné měnit její texturu za běhu hry. Byly pro to vytvořeny dvě textury (základní a s makeupem). Pro přepínání textury byla v Unreal Engine použita tzv. DynamicMaterialInstance [7], která je vytvořena v ConstructionScript (skript, který se spustí při každé změně jeho majitele i v editoru) [8].

```
Neutral \e
Ahoj, tohle je test.
\\

Angry \e
Jsem velmi nastvana!!!
\\

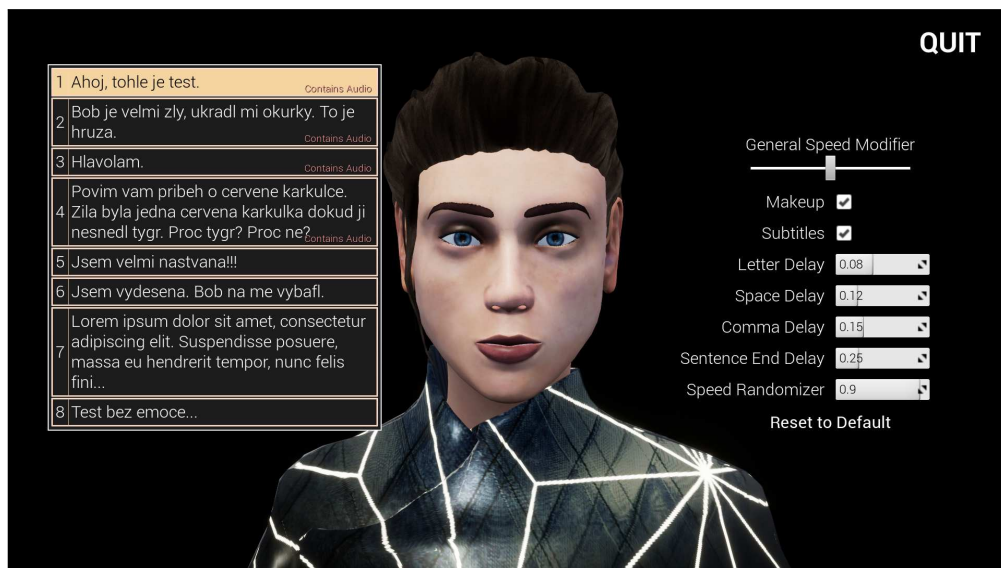
Scared \e
Hlavo!am.
\\
```

Obrázek 7: Ukázka textového souboru speeches.txt a jeho korektního formátu.

## Uživatelské rozhraní a ovládání

Pro aplikaci bylo vytvořeno jednoduché uživatelské rozhraní, ke kterému se lze dostat buď pomocí klávesy Escape nebo /, která byla použita v rámci testování aplikace (klávesnice Escape ukončí v Unreal Engine editoru hru, zatímco nemá žádný význam ve standalone aplikaci). Menu obsahuje přehled všech textů a jejich pořadí s tím, že pokud text přesahuje opět nějakou definovanou délku, tak je v tomto menu ořezán. Každá textová položka, která obsahuje zvukový klip, je navíc označena textem “Contains Audio”.

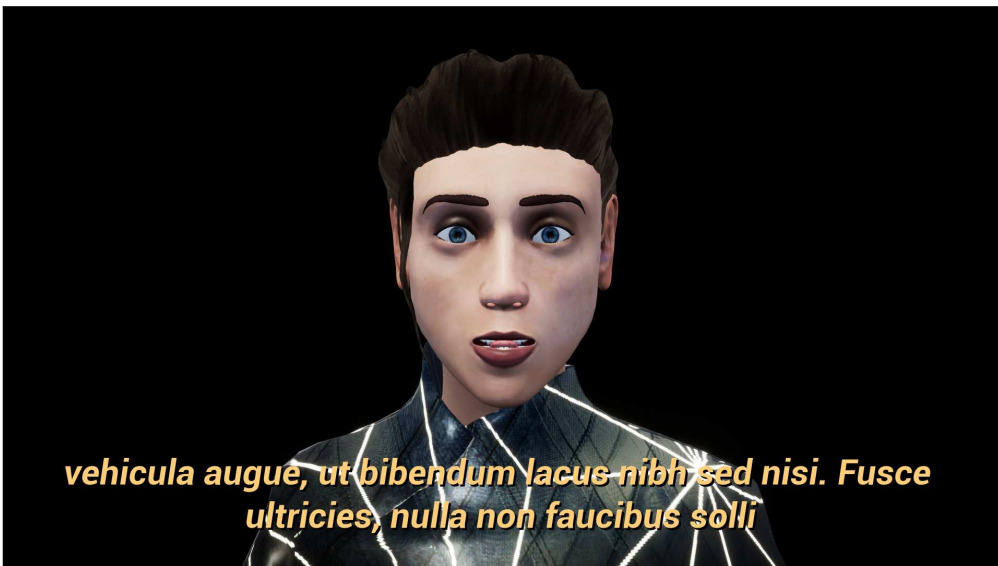
Kromě seznamu textů lze v hlavním menu nastavovat všechny zmiňované proměnné týkající se řeči jako jsou prodlevy mezi znaky, mezerami atd.. Navíc byl přidán posuvník pro změnu globálního nastavení rychlosti řeči. V uživatelském rozhraní lze navíc vypnout nebo zapnout makeup a titulky. Všechny tyto položky lze navíc navrátit do výchozího nastavení tlačítkem “Reset to Default”.



Obrázek 8: Uživatelské rozhraní aplikace.



Obrázek 9: Postava bez makeupu.



Obrázek 10: Titulky a postava právě vyslovující znak l s vyděšeným výrazem.

## Závěr

V rámci semestrálního projektu předmětu MMA byl použit již hotový model hlavy, pro který byl vytvořen rig v modeláři Blender (zde nazvaný Armature) a textury. Za použití tohoto rigu bylo vytvořeno 35 animací, jak pro vizémy českého jazyka, tak pro základní emoce. Model byl exportován společně s animacemi do Unreal Engine, kde byla vytvořena aplikace pro jeho ovládání pomocí zadaného textu uživatelem za běhu i pro předpřipravený textový soubor s vlastním formátem. Byly použity animační blueprint skripty Unreal Engine a v nich obsažené stavové automaty. V rámci práce bylo též vytvořeno uživatelské rozhraní pro jednoduché ovládání aplikace. V aplikaci je též možné dodat k jednotlivým předpřipraveným textům zvukové soubory, které budou při čtení přehrány. Ve výsledku jsme dosáhli přesvědčivé reprezentace řeči českého jazyka, především díky velkému počtu animací (a náhodnému výběru z nich) a díky použitým odchylkám v rychlosti řeči.

## Reference

- [1] *Phonemes and Visemes*, [online dokumentace], publikováno 14.8.2013 [cit. 31.12.2017], Dostupné z: <http://docs.cryengine.com/display/SDKDOC2/Phonemes+and+Visemes>
- [2] *Texture & Texture Mask*, [online dokumentace], [cit. 31.12.2017], Dostupné z: [https://docs.blender.org/manual/en/dev/sculpt\\_paint/texture\\_tex\\_mask.html](https://docs.blender.org/manual/en/dev/sculpt_paint/texture_tex_mask.html)
- [3] *Armatures*, [online dokumentace], [cit. 31.12.2017], Dostupné z: <https://docs.blender.org/manual/en/dev/rigging/armatures/index.html>
- [4] *Animation Blueprints*, [online dokumentace], [cit. 31.12.2017], Dostupné z: <https://docs.unrealengine.com/latest/INT/Engine/Animation/AnimBlueprints/>
- [5] *Weight Paint Mode*, [online dokumentace], [cit. 31.12.2017], Dostupné z: [http://blender-manual-i18n.readthedocs.io/ja/latest/modeling/meshes/vertex\\_groups/weight\\_paint.html](http://blender-manual-i18n.readthedocs.io/ja/latest/modeling/meshes/vertex_groups/weight_paint.html)
- [6] *Blend Nodes*, [online dokumentace], [cit. 31.12.2017], Dostupné z: <https://docs.unrealengine.com/latest/INT/Engine/Animation/NodeReference/Blend/>
- [7] *Instanced Materials*, [online dokumentace], [cit. 31.12.2017], Dostupné z: <https://docs.unrealengine.com/latest/INT/Engine/Rendering/Materials/MaterialInstances/>
- [8] *Construction Script*, [online dokumentace], [cit. 31.12.2017], Dostupné z: <https://docs.unrealengine.com/latest/INT/Engine/Blueprints/UserGuide/UserConstructionScript/>