

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Vývoj parametrizovaných logopedických her v Unity

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Michaela BENEŠOVÁ**
Osobní číslo: **A19B0008P**
Studijní program: **B0613A140015 Informatika a výpočetní technika**
Specializace: **Informatika**
Téma práce: **Vývoj parametrizovaných logopedických her v Unity**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s webovou aplikací BrainIn v neuroinformatické laboratoři na KIV.
2. Prostudujte problematiku logopedických her a navrhnete vhodné vstupní a výstupní parametry těchto her.
3. Dle požadavků logopedické sestry navrhnete 4 hry a použijte vstupní a výstupní parametry uvedené v bodě 2.
4. Dle návrhu v bodě 3 implementujte 4 hry a začleňte je do webové aplikace BrainIn.
5. Otestujte hry na reprezentativním vzorku testovacích subjektů a zhodnoťte dosažené výsledky práce.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Petr Brůha**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **24. srpna 2021**
Termín odevzdání bakalářské práce: **5. května 2022**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 23. června 2022

Michaela Benešová

Abstract

This bachelor's thesis was created in order to create 4 speech therapy parameterized games for the existing web application BrainIn. The aim of the games is to help patients with acquired brain damage. During neurorehabilitation, games will serve to improve cognitive functions, for example to improve memory or vocabulary. The work is divided into two parts: theoretical and practical. The theoretical part deals with the application of BrainIn, the brain and cognitive functions. It also contains games analysis. The practical part deals with the implementation, testing of games and proposals for other speech therapy games.

Abstrakt

Tato bakalářská práce vznikla za účelem vytvoření 4 logopedických parametrizovaných her pro již existující webovou aplikaci BrainIn. Cílem her je pomáhat pacientům se získaným poškozením mozku. Hry budou při neurorehabilitaci sloužit ke zlepšení kognitivních funkcí například ke zlepšení paměti či slovní zásoby. Práce je rozdělena do dvou částí teoretické a praktické. Teoretická část se zabývá aplikací BrainIn mozkem a kognitivními funkcemi. Obsahuje také analýzu her. V praktické části se řeší implementační stránka her, testování a návrhy na další logopedické hry.

Obsah

1	Úvod	7
2	Úvod do webové aplikace BrainIn	8
2.1	Neuroinformatics	8
2.2	Webová aplikace	9
2.2.1	Uživatelé webové aplikace	11
2.2.2	Základní pojmy	11
2.3	Komerční systémy	14
3	Mozek a kognitivní funkce	18
3.1	Mikroskopická anatomie mozku	18
3.2	Makroskopická anatomie mozku	18
3.2.1	Koncový mozek	19
3.3	Kognitivní funkce	19
4	Využití logopedických her při neurorehabilitaci	23
4.1	Onemocnění s vlivem na kognitivní funkce	23
4.1.1	Degenerativní změny mozku	23
4.1.2	Afázie, agrafie, alexie	23
4.1.3	Parkinsonova choroba	24
4.1.4	Psychické nemoci	24
4.2	Vliv parametrizovaných her na vývoj onemocnění	25
5	Analýza her	26
5.1	Skládání slov	26
5.1.1	Voroného diagram	28
5.2	Vodorovné křížovky	28
5.3	Doplnění slov	31
5.4	Slova, které začínají a končí na stejné písmeno	31
6	Implementace her	33
6.1	Společné prvky	33
6.1.1	Použité technologie	33
6.1.2	Šablona šablon	33
6.1.3	Lokalizační soubory	34
6.2	Skládání slov	35

6.3	Vodorovné křížovky	40
6.4	Doplnění slov	45
6.5	Slova, která začínají a končí na stejné písmeno	49
6.6	Slovník a jeho implementace	51
6.7	Úspěšnost ve hře	51
7	Testování her	53
7.1	Testování na webu	53
7.2	Testování v Unity	54
7.3	Nalezené chyby	54
8	Zhodnocení dosažených výsledků	55
9	Závěr	56
	Literatura	57

1 Úvod

K digitálnímu rozvoji dochází dnes ve všech odvětvích moderních věd a výjimkou nejsou ani obory lékařské. Velkého rozmachu se v posledních letech dočkal i obor neurorehabilitace. Vedlo k tomu několik faktorů. Pro terapeutky či zdravotní sestry je důležité, aby pacienti měli možnost léčby i v domácím prostředí, jelikož není kolikrát v jejich možnostech se věnovat pacientům plně v neurologických centrech či psychiatrických léčebnách. Zároveň potřebují mít, ale kontrolu nad splněnými úkoly, aby mohly požadavky přizpůsobovat přesně potřebám pacientů.

Proto vzniklo několik systému zabývajících se rehabilitací a tréninkem kognitivních funkcí. Tyto systémy poskytují hry vedoucí ke zlepšení celkového stavu pacienta. Pojmem hry můžeme tedy rozumět úkoly / úlohy vedoucí k pozitivnímu ovlivnění mozkové činnosti pacienta a principem těchto her by měla být převážně jejich jednoduchost a srozumitelnost.

Cílem této práce je vytvořit a naimplementovat čtyři parametrizované logopedické hry, které budou využívány v praxi s pacienty při neurorehabilitaci. Hry budou zasazeny do již existujícího systému BrainIn. Tento systém poskytuje množinu her lidem s různými poškozeními mozku. Jelikož každý pacient trpí jinak závažnými problémy je zde díky parametrům kladen důraz na individualitu. Parametry totiž slouží k ovlivnění průběhu hry a zároveň k nastavení obtížnosti.

Bakalářskou práci jsem rozdělila na teoretickou a praktickou část. V rámci teoretické části představím blíže webovou aplikaci BrainIn. Poté jsem se zaměřila na mozek a jeho kognitivní funkce využití logopedických her při neurorehabilitaci a jejich analýzu. V praktické části se zabývám implementací jednotlivých her jejich testováním a návrhy na další možné hry. V závěru jsem zhodnotila výsledky a dojmy z testování.

2 Úvod do webové aplikace BrainIn

Webová aplikace BrainIn představuje internetovou stránku, na kterou budou implementovány čtyři nové logopedické hry. Než si ale představíme samotnou aplikaci, tak si nejdříve řekneme informace o skupině, která stojí za zrodem projektu.

2.1 Neuroinformatics

Skupina výzkumníků, která stojí za zrodem webové aplikace BrainIn se nazývá Neuroinformatics a byla založena na Katedře informatiky a výpočetní techniky Západočeské Univerzity v Plzni v roce 2003. [12] Aplikace vzniká jako součást projektu s názvem *Využití moderních informačních technologií pro neurorehabilitace pacientů se získaným poškozením mozku*. Tato skupina je zároveň součástí Medicínské informatiky výzkumného centra NTIS Katedry informatiky a výpočetní techniky na Fakultě aplikovaných věd ZČU. [11]

Medicínská informatika

Medicínská informatika byla založena v roce 2015 právě sloučením několika menších skupin. Zaměřuje se na vývoj pokročilých počítačových modelů inženýrských metod a informačních technologií za účelem zvýšení rychlosti efektivity a spolehlivosti bioinženýrské činnosti v medicíně a dalších biomedicínských oborech. Výzkum se skládá z několika aktivit od analýzy vstupních dat po simulační modely včetně vizuální reprezentace. Výzkumné aktivity jsou poté řešeny v rámci nejrůznějších národních a evropských projektů. Celé oddělení medicínské informatiky má bohaté zkušenosti zejména s anotací medicínských dat extrakce geometrických informací objektů z medicínských snímků či například s kosterně svalovým modelováním.[11]

Kromě zmíněného projektu se dále skupina Neuroinformatics věnuje kupříkladu těmto tématům a aktivitám:

- měření a vyhodnocení elektrické aktivity mozku se zkoumáním lidského chování ve stresových a stereotypních situacích,

- v klinické foniatrii se podílí na vývoji moderních diagnostických postupů a klasifikačních nástrojů pro hodnocení kvality uzávěru hlasivky,
- v klinické audiometrii se zaměřují na vývoj algoritmů používaných pro automatické řízení při vyšetření prahu vnímání a pro hodnocení latence (doba mezi podnětem a zjevným projevem). [12]

V rámci projektu, ve kterém vzniká webová aplikace BrainIn, vzniká například i mozkem řízený počítačový asistenční systém pro osoby s omezenou pohyblivostí. Projekt vzniká ve spolupráci s klinikami v zahraničí, ale i s českými zdravotními ústavami například s Centrem pohybových a terénních sociálních služeb Zbůch. [16]

2.2 Webová aplikace

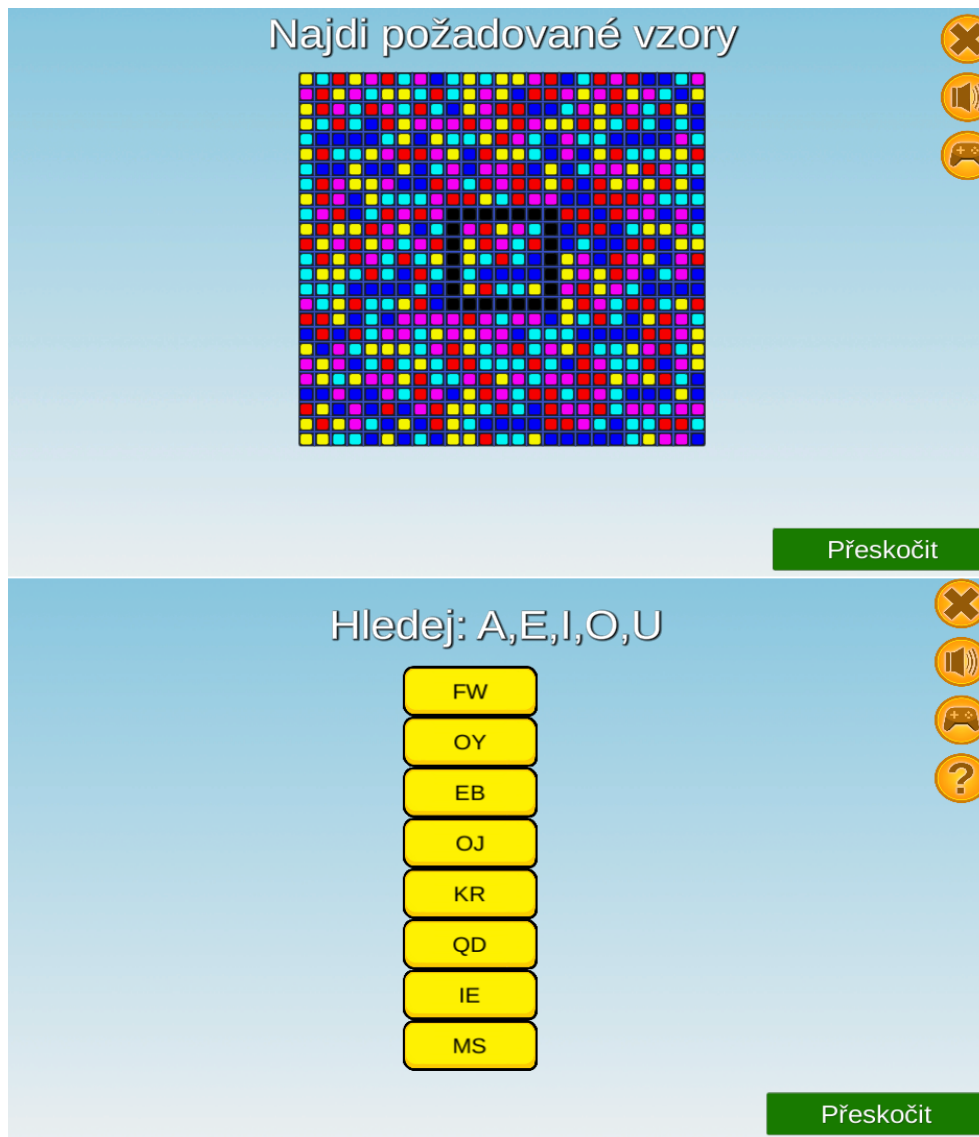
Při navštívení webové aplikace BrainIn čeká na uživatele uvítací obrazovka.



Obrázek 2.1: Uvítací obrazovka webové aplikace BrainIn

Kvůli spolupráci se zahraničními klinikami, aplikace vzniká i v německém a anglickém jazyce, tedy kliknutím levého tlačítka počítačové myši na vlaječku lze aplikaci přepnout do jiných jazyků. Při stejném kliknutí, akorát na *Ukázky úloh* se rozbálí pestrá nabídka ukázkových neurorehabilitačních her, které již byly vytvořeny v rámci projektu. Jedná se o hry, které jsou zaměřené na řečové funkce, paměť, koncentraci, logické myšlení, prostorovou orientaci či lehkou motoriku.

Každá hra obsahuje potřebné nápovědy, textové i mluvené. Zároveň hry najdeme vždy lokalizovány do třech zmíněných jazyků. Na dalším obrázku můžeme vidět ukázkou her.



Obrázek 2.2: Ukázky her Orion a Hledání znaků

Aby se na webovou aplikaci mohl uživatel přihlásit, je potřeba, aby byl zaregistrován a zaučen od terapeuta. Pokud se terapeuti chtějí účastnit projektu, provádějí registrace a žádají o práva emailem na adresu uvedené na úvodní stránce. Nyní jsme se tedy dostali k tomu, že do aplikace se přihlašuje několik typů uživatelů.

2.2.1 Uživatelé webové aplikace

- **Pacient** tato role je přidělena každému, kdo se na stránce zaregistruje. Obecně má pacient za úkol se do aplikace přihlásit, spustit a zahrát si přidělené hry, které po skončení může slovně i bodově ohodnotit.
- **Terapeut** představuje roli pro terapeuty a pracovníky rehabilitačních center. Vytváří a přiřazuje balíky úloh (soubor s hrami), které si má pacient spustit.
- **Super-terapeut** je terapeut, který má právo navíc - může nahrávat do systému nové šablony (vzory úloh, které se upravují a parametrizují, o šablonách se více dozvíme v implementačních detailech v pozdější kapitole) a konfigurovat úlohy.
- **Administrátor** přiřazuje pacienty terapeutům, většinou se jedná o sdílenou roli se super-terapeutem. Také může tvořit nové hry a spravovat uživatele.
- **Super-admin** je role určená správcům a vývojářům systému. Super-admin má všechna práva ostatních rolí a zároveň může zasahovat do nastavení prostředí aplikace.

2.2.2 Základní pojmy

V rámci představení webové aplikace je nutné se zaměřit na zvolené technologie a zmínit několik základních pojmů, se kterými se v rámci BrainInu pracuje.

Šablona

Pomocí šablony lze do systému implementovat nové hry. Každá šablona zůstává uložena v systému ve svém paměťovém prostoru, odkud se později spouští. Při vytváření šablony si musíme být vědomi toho, jaké chceme vstupní a výstupní parametry. Musíme je nadefinovat i v takovém pořadí, ve kterém je poté budou zadávat terapeuti. Parametry mohou ovlivňovat několik aspektů her - průběh, cíl a vzhled. Všechny hry, které jsou součástí webové aplikace disponují několika společnými parametry.

Obecné vstupní parametry

- **Maximální doba** udává jak dlouhý je maximální čas pro trvání hry.
- **Počet kol**
- **Nápověda** počet zobrazení nápověd.
- **Přeskočit kolo** zda má uživatel možnost přeskočit kolo.
- **Přeskočit zbylá kola** zda má uživatel možnost přeskočit všechna zbylá kola naráz.
- **Trvání semaforu** představuje dobu pauzy mezi jednotlivými koly.
- **Zobrazení správného řešení** jestli se má zobrazit správné řešení.
- **Doba zobrazení řešení** udává jak dlouho se má zobrazovat na obrazovce správné řešení.
- **Doba zobrazení hodnocení** je doba po kterou uvidí na konci hry pacient svoji úspěšnost ve hře.

Název parametru	Popis parametru	Typ dat	Hodnota
Debug	Pomocné výpisy pro testování. Zobrazení s...	Logická hodnota	
Maximální doba	Maximální doba v sekundách, kterou může...	Číslo	
Počet kol	Hodnota udávající počet kol (počet opakuj...	Číslo	
Nápověda	Kolikrát lze během úlohy použít nápovědu.	Číslo	
Přeskočit kolo	Jestli má mít uživatel možnost přeskočit ko...	Logická hodnota	
Přeskočit zbylá kola	Jestli má mít uživatel možnost přeskočit vš...	Logická hodnota	

Obrázek 2.3: Ukázka vstupních parametrů šablony

Obecné výstupní parametry

Kromě vstupních parametrů se můžeme v systému setkat i s výstupními parametry, ty máme celkové či za jedno kolo. Jedná se o výstupy jako celková doba, kterou hrál či jak hra pacientovi šla.

Úloha

Úloha představuje využití šablony, která se parametrizuje. Dojde tak k vytvoření unikátní hry přímo na míru potřebám pacienta. Často zde dochází k problému s názvoslovím, pro nás zjednodušeně můžeme říci, že úloha = hra.

Úprava úlohy (758) [Seznam úloh](#) [Otestovat úlohu](#)

[Detail](#) [Obsah](#) [Správa souborů](#)

Debug ?	Ne
Maximální doba ?	0
Počet kol ?	0
Nápověda ?	0
Přeskočit kolo ?	Ne
Přeskočit zbylá kola ?	Ne
Trvání semaforu ?	0
Zobrazení správného řešení ?	Ne
Doba zobrazení řešení ?	0
Doba zobrazení hodnocení ?	0
Šířka mřížky ?	0
Výška mřížky ?	0
Šířka vzoru ?	0

Obrázek 2.4: Vytvoření hry Orion

Balík

Vytvořené úlohy můžeme seskupovat do balíčků. Balík představuje tedy sekvenci úloh, které se jedna po druhé spouští.

Analýza

V analýze terapeut může vybrat šablonu, konkrétní úlohu a pacienta ke kterému chce vidět statistiky. Jedná se o výsledky jednotlivých úloh a aktuální postup v úlohách.

Implementační technologie

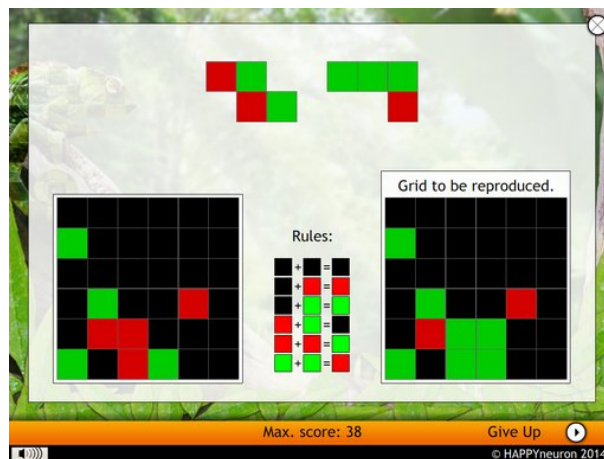
Webové hry jsou vytvářeny pomocí Unity, programovacího jazyku C# a JavaScriptu.

2.3 Komerční systémy

Momentálně jsme si představili základní informace o webové aplikaci BrainIn, která se věnuje hrám pro neurorehabilitaci mozku. Na trhu ovšem existují další systémy, které slouží ke stejnému či podobnému účelu.

HAPPYneuron

Online program ke zlepšení paměti, stimulace koncentrace, zlepšení reakcí a logiky. Disponuje sedmi jazyky a několika desítkami her. K přístupu do systému je nutná registrace, která umožní přístup do aplikace po dobu sedmi dnů. [7] Následně je nutné platit předplatné, které se cenově pohybuje od 320 korun českých měsíčně. [8]

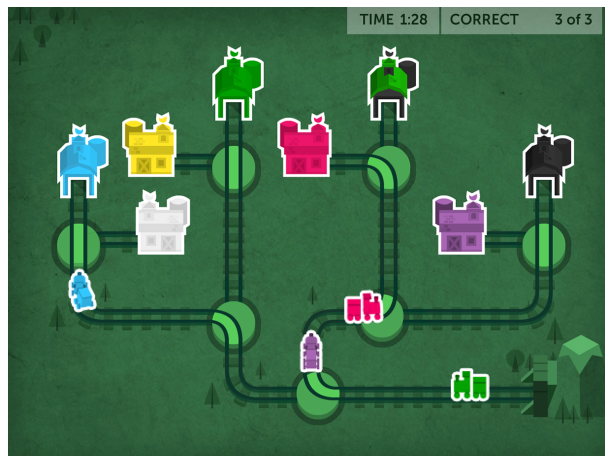


Obrázek 2.5: Hry od HAPPYneuron [7]

Lumosity

Lumosity je online program, spuštěný od roku 2007 spolkem Lumos Labs, Inc. Skládá se z webových a mobilních her zlepšujících paměť, pozornost, flexibilitu, rychlost zpracování a řešení problému. [29] Disponuje sedmi světovými jazyky více jak padesáti hrami a vlastní mobilní aplikací pro mobilní systémy Android a iOS. Při registraci zdarma může uživatel hrát pouze 3 hry

denně. Ovšem získává například možnost ověření aktuálního stavu s použitím Fit Testu a přehled dat pro osobní zlepšení. [9] Pokud by uživatel chtěl poznat veškeré funkce, které Lumosity nabízí musí odebírat předplatné. Toto předplatné začíná na minimální částce 250 korun českých měsíčně. [37]



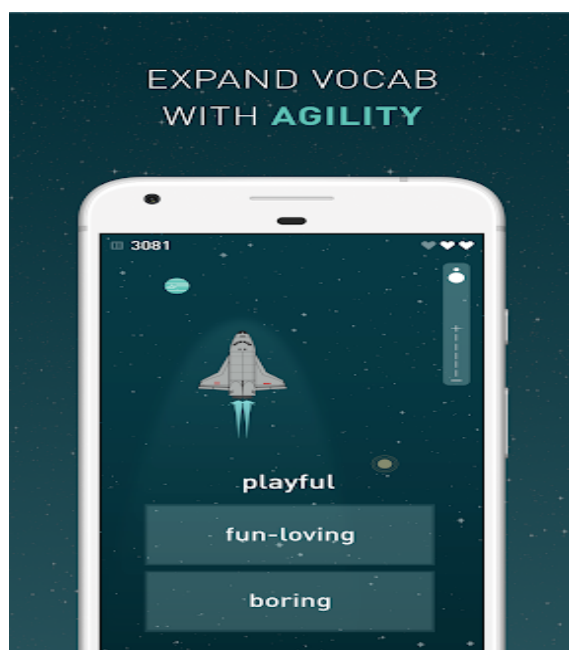
Obrázek 2.6: Hry od Lumosity [29]

Elevate

Elevate je mobilní aplikace na mobilní operační systémy Android a iOS. Slouží ke zlepšení matematických počtů logiky a komunikace. Zároveň ve spolupráci s dalším programem Balance pro meditace vytvořili inovativní a meditační systém pro lepší zvládání stresových situací a pro zlepšení kvality spánku. Obsahuje více jak čtyřicet her a v PRO verzi umožňuje vytvářet denní tréninky tří až pěti her. Cena vylepšené verze začíná na 100 korunách českých měsíčně. [37] Je ale důležité zmínit, že Elevate slouží velmi dobře jako systém pro trénink mozkové aktivity zdravého člověka. [6]

AfaSlovník

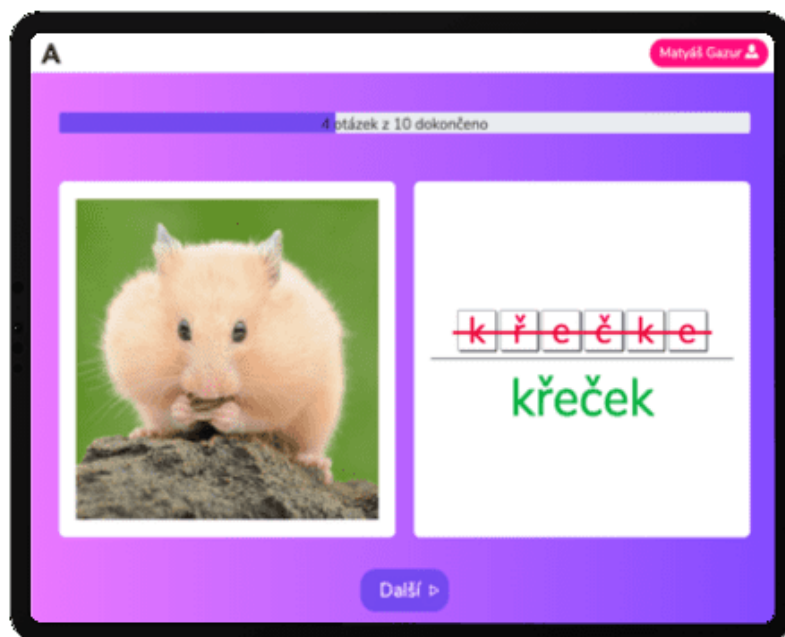
Posledním komerčním systémem, který zde chci zmínit je česká logopedická aplikace pro léčbu Afázie s názvem AfaSlovník. Afázie je porucha produkce a porozumění řeči, ale více se o tomto onemocnění dozvíme ve čtvrté kapitole. Tato aplikace slouží pro pacienty a logopedy/odborníky. Jedná se o pomůcku, která učí pacienty znovu mluvit, zábavně procvičovat a sledovat pokroky. K terapii jim dokonce mohou pomoci rodinní příslušníci. Aplikace je přístupná ve dvou verzích, placené a neplacené. Neplacená verze obsahuje pouhý slovník, placená ale dovoluje využít testové varianty, plány na míru,



Obrázek 2.7: Hry od Elevate [6]

procvičování hesel či přístup ke statistikám. Dokonce se vylepšená verze pohybuje cenově nízko a to na 99 korunách českých za měsíc. Velkou výhodou je také, to, že se dá spustit na mobilu, tabletu i počítači. [1]

Pokud bychom systémy porovnali v počtu uživatelů samozřejmě komerční systémy, mají navrch jsou více oblíbené známé, ale placené. Pokud odmyslíme finanční náklady, je jejich velkou nevýhodou neindividuální přístup k pacientům. V tomto ohledu se významně liší BrainIn a Afa Slovník. Ovšem pokud bychom měli srovnávat tyto dvě české aplikace, BrainIn je výjimečný v možnostech individuálního přístupu k pacientovi, jelikož jednotlivé hry se dají parametry upravit na míru. V shrnutí lze říci, že BrainIn nabízí nejpestřejší výběr her s největším důrazem na individualitu, a to z něj dělá výjimečný projekt a aplikaci oproti jeho konkurenci. Při zkoušení her různých systémů jsem si zároveň všimla důležitých aspektů těchto her, těmi jsou jednoduchost, srozumitelnost, přehlednost a možnost nápovědy, kterou lze vyvolávat opakovaně. Na tyto faktory bude tedy brán zřetel i při tvoření logopedických her.



Obrázek 2.8: Ukázka AfaSlovníku [1]

3 Mozek a kognitivní funkce

Hry, které se tvoří na základě zadání této práce, slouží ke zlepšení kognitivních funkcí, podíváme se tedy v této kapitole na to, jak funguje samotný lidský mozek, co jsou to kognitivní funkce a jak spolu tyto dva pojmy souvisí. V následující kapitole se poté budeme věnovat diagnózám, jejichž důsledkem je právě zhoršení kognitivních funkcí, a jak se parametrizované hry při léčbě uplatní.

Abychom se dostali k tomu, že mozková kůra je jedna z částí, která nejvíce zodpovídá za běh kognitivních funkcí, musíme se nejdříve podívat na lidský mozek jako celek a rozdělit ho na jednotlivé části.

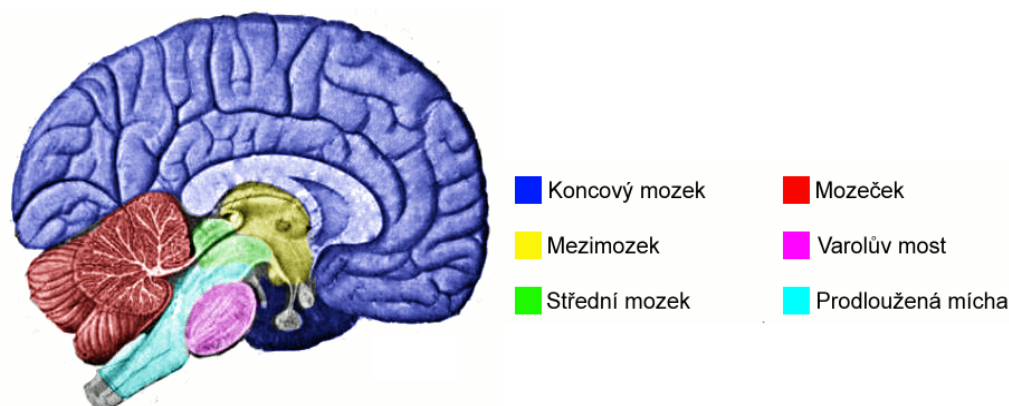
Lidský mozek je nejkomplicovanější orgán v našem těle. Je to řídicí a integrační orgán celé nervové soustavy člověka, díky které je schopný vysílat zprávy do různých částí těla. Nachází se v dutině lebeční, která ho chrání před zraněním a na jeho anatomii můžeme nahlížet ze dvou perspektiv. [36]

3.1 Mikroskopická anatomie mozku

Mikroskopické objekty nejsou viditelné pouhým okem, pokud tedy rozdělíme mozek dle mikroskopické anatomie, zjistíme, že se skládá z neuronů a glií. Neuron je základní stavební jednotkou nervové tkáně, která tvoří nervovou soustavu. Glie jsou podpůrnou tkání neuronů. Zodpovědnost neuronů spočívá ve schopnosti přijmout, vést, zpracovat a zodpovídat signály z vnitřního a vnějšího prostředí. [36]

3.2 Makroskopická anatomie mozku

Makroskopický objekt je takový fyzikální objekt, který je pozorovatelný pouhým okem. Pokud bychom nahlédli pod mozkomíšní mok, což je tekutina, která má mozek chránit před infekcí viděli bychom několik různorodých částí. Tyto části jsou například mozeček, koncový mozek nebo prodloužená mícha. Každá z těchto částí má velmi důležitou činnost v celém systému. Rozdělení v teorii najdeme vícero, jednodušších i složitějších. Já jsem si jedno základní dovolila zobrazit na obrázku 3.1.



Obrázek 3.1: Makroskopická anatomie mozku [26]

3.2.1 Koncový mozek

Koncový mozek má dvě hemisféry, pravou a levou. Zajímavostí je, že každá z hemisfér neovládá příslušnou polovinu těla, ale opačnou. Proč zde ale koncový mozek je více zmíněn? Jedná se o to, že jeho povrch tvoří mozková kůra, která je sídlem kognitivních funkcí. Ve vnitřku bychom našli bazální ganglie za které jsou zodpovědné okrsky šedé hmoty, uvnitř je bílá hmota. [34]

Mozková kůra

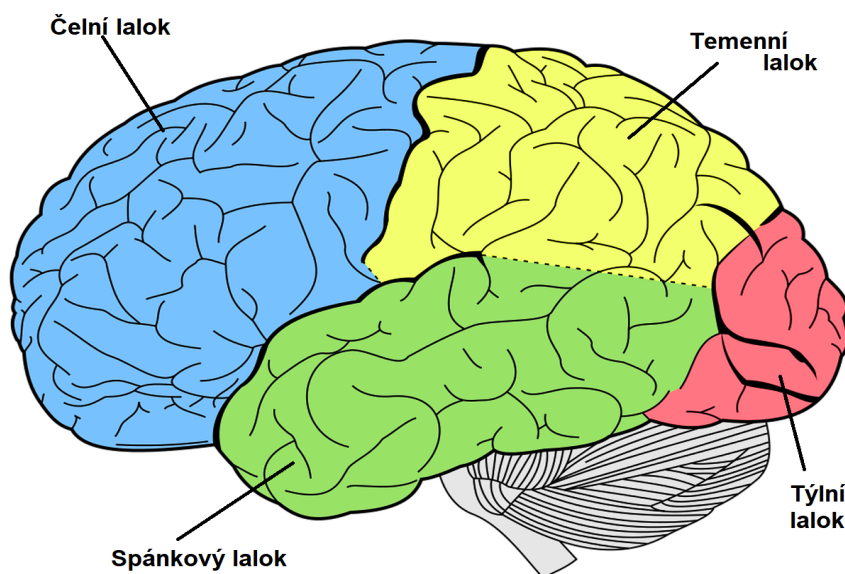
Čtyři laloky tvoří mozkovou kůru, jejich rozdělení si můžeme všimnout na obrázku níže.

Každý lalok má svoje specifické činnosti za které je zodpovědný. Jedná se o inteligenci, jazyk, čtení, chování, zrak či dokonce paměť. Tyto činnosti jsou nedílnou součástí kognitivních funkcí a jak již zaznělo, pacienti, kteří budou využívat hry vytvořené pro účel této práce se potýkají s problémy právě těchto funkcí.

3.3 Kognitivní funkce

Kognitivní systém je velmi ucelenou strukturou a tvoří ho několik funkcí včetně těch, které byly vyjmenovány v předešlé podkapitole. Můžeme říci, že jednotlivé funkce na sebe navazují a jedna bez druhé nemohou řádně fungovat. Jsme schopni je rozlišit, i když někdy rozlišení nemusí být úplně přesné.

Několik základních kognitivních funkcí zde již bylo vypsáno, teď si je popíšeme více podrobně:



Obrázek 3.2: Laloky mozkové kůry [25]

Paměť

Abychom mohli uchovávat a používat informace, potřebujeme paměť. Paměť je schopnost. Je to schopnost si informace vštěpovat, uchovávat či vybavovat. A podílí se na tom centrální nervová soustava. [38]

Jedno ze základních dělení představuje:

- krátkodobá paměť - omezená opaměť, která dokáže uchovávat informace po krátkou dobu; velkou roli zde hrají emoce či princip odměny a trestu podle toho pak dochází k přesunu do dlouhodobé paměti. Součástí krátkodobé paměti je také prostorová pracovní paměť (paměť o orientaci a umístění). [18]
- střednědobá paměť - naše informace jsou na cestě do dlouhodobé paměti, tvoří se jim paměťové stopy item dlouhodobá paměť - vyvíjí se od předškolního věku; pamatujeme si dovednosti, pravidla, vzpomínky, obecná fakta, fungování světa
- dlouhodobá paměť - vyvíjí se od předškolního věku; pamatujeme si dovednosti, pravidla, vzpomínky, obecná fakta, fungování světa

Další možné dělení [10] :

- Sémantická paměť ukládá význam informace, neboli významy pojmů, fakta, vědomosti a vztahy mezi nimi. [32]

- Mechanická paměť se osvojuje opakováním informací.
- Logická paměť se osvojuje na základ vzájemných souvislostí a smyslů informací.
- Citová paměť jejíž podstatou je zapamatování a reprodukce citů.

Pozornost

Pozornost je zaměření a soustředění duševní činnosti na určitý objekt nebo děj. Pouští do vědomí omezený počet informací a umožňuje monitorování vnějšího i vnitřního prostředí. Vybírá z něj pouze ty podněty, které si v daném okamžiku potřebujeme nebo přejeme uvědomit, ostatní ignoruje. [15]

Má pět základních vlastností [15] :

- Selektivita (výběrovost) je schopnost zaměřit pozornost na významné vnější či vnitřní podněty a ignorovat bezvýznamné opakující se podněty.
- Koncentrace (soustředěnost) je vyčlenění počtu psychických obsahů, kterými se chceme zabývat.
- Distribuce (rozdělování) je schopnost rozdělit pozornost mezi více objektů, je značně omezená.
- Kapacita (rozsah) představuje množství podnětů, které je člověk schopen zachytit v určitém časovém úseku. Přibližně odpovídá rozsahu krátkodobé paměti (pět až devět).
- Stabilita (stálost) se rovná délce časového intervalu, ve které je člověk schopen koncentrovat pozornost na určitý podnět. Přibližně tato délka není delší než pět sekund.

Rychlost

Rychlost psychických funkcí zahrnuje [19] :

- Reakční rychlost aneb jak rychle zareagujeme na nějaký podnět.
- Rychlost myšlenkových operací čili zpracovávání zadání a řešení problému. V běžném životě je dobře pozorovatelná - uskočení před projíždícím automobilem, hledání mobilu ve spěchu, zorganizování času.

- Rychlost ukládání si a znovuvybavování informací, resp. rychlost vybavení si informací z dlouhodobé paměti.
- Rychlost související s pozorností neboli přepínání pozornosti.
- Zpracovávání smyslových informací.

Exekutivní funkce včetně emocionální seberegulace (sebekontroly)

Exekutivní funkce jsou kognitivní činnosti, které zajišťují samostatné a účelné jednání a myšlení člověka. Stojí za plánováním, organizací činností, myšlením a jejich vzájemnou souhrou. [27]

Verbální dovednosti

Tyto dovednosti prokazují schopnost člověka zacházet s jednotlivými písmeny, slovy, vztahy mezi slovy a pojmy. Zahrnují porozumění slovním sdělením, schopnosti pohotově užívat znalosti jazykových pravidel, slovní zásoby, schopnosti porozumět jazykovým významům a vztahům, pochopení struktury a principiální zákonitosti a pravidla jazyka jako celku. Není zde ovšem nutná znalost pravopisné formy. [22]

Prostorovou orientaci

Orientace v prostoru znamená uvědomění si pozice nalevo, napravo, vpředu, vzadu, nahoře, dole a správné určení směru. [17]

Percepční schopnosti

Vnímání a rozlišování pomocí smyslů. [14]

Nejen zmíněné kognitivní funkce lze trénovat pomocí několika technik. Při trénování lze využívat meditaci, kreativní aktivity, multitasking či právě hry, včetně těch, které vzniknou jako výsledek této práce.

4 Využití logopedických her při neurorehabilitaci

Jak jsem již naznačila v úvodu předchozí kapitoly, zde se budeme věnovat využití logopedických her při neurorehabilitaci. Upřesním zde diagnózy u kterých mé hry naleznou uplatnění a jaký pozitivní přínos mohou hry mít na zdravotní stav pacientů.

4.1 Onemocnění s vlivem na kognitivní funkce

Na světě existuje mnoho příčin proč dochází k poruše kognitivních funkcí. Může se jednat o rozšířené psychické nemoci, ale i o vážnější stavy jako jsou mozkové příhody, nádory nebo léze. Léze jsou ložiska, ve kterém se změnila tkáň. [2] Zde bych napsala základní výčet onemocnění vedoucích k poruše kognitivních funkcí.

4.1.1 Degenerativní změny mozku

Neurodegenerativní onemocnění jsou charakterizovány progresivním zánikem neuronů, reaktivním zmnožením glií a ukládáním proteinových depozit. Základním mechanismem je změna struktury postiženého proteinu, který buňka nedokáže zpracovat a zároveň se objevuje nedostatek funkčního proteinu. Klinické projevy mohou být různorodé v závislosti na postižených strukturách. [5]

4.1.2 Afázie, agrafie, alexie

Afázie, agrafie a alexie jsou poruchy, které vznikají jako důsledek vyjmenovaných příčin a zároveň se řadí mezi nejčastější logopedické diagnózy. Každá tato porucha se může vyskytovat ojedinele, ale většinou se vyskytují dohromady v různých kombinacích. Zároveň projevy těchto poruch mohou být různého rázu. [2]

Afázie

Porucha produkce a porozumění řeči. Mezi příznaky se řadí porucha plynulosti řeči, porucha porozumění řeči, porucha pojmenování, parafázie (typy

záměn cílových slov) a perseverace (opakování stejné reakce - opakování jednoho slova na několik otázek). Mezi důsledky afázie pak tedy patří obtížné vyjadřování myšlenek, nápadů, přání či názorů v sociální interakci. [2]

Agrafie

Agrafie je ztráta schopnosti psát, přičemž není postižena hybnost ruky. Tedy motoricky i intelektově je agrafik v pořádku. [3]

Alexie

Alexie je částečná nebo úplná neschopnost číst. [4]

4.1.3 Parkinsonova choroba

Nejen genetické predispozice, ale i opakované pohmoždění hlavy může vést k rozvoji Parkinsonovy choroby. Tato choroba vzniká na základě nadměrného odumírání mozkových buněk, které za normálních okolností tvoří látku dopamin. Tato látka je stěžejní pro přenos informací mezi neurony. Co však tento proces započne, není dosud známo, a proto Parkinsonova choroba patří k nevyлéčitelným onemocněním. Onemocnění se projevuje několika motorickými i nemotorickými příznaky, mezi něž patří například zhoršení čichu, křeče, třes, problémy s pamětí, se zrakem a sluchem. Projevuje se v několika fázích a při léčbě je nutné docházet na terapie, které pomáhají pacientům se zmírněním příznaků nemoci. [13]

4.1.4 Psychické nemoci

Jedna z psychických nemocí, která se projevuje kognitivními deficity je například schizofrenie. K jejím spouštěčům patří drogy či třeba zátěžové situace. Mezi příznaky poté patří bludy, halucinace a oploštělé emoce. A jak již bylo zmíněno, dochází při ní k poruše kognitivních funkcí. [20]

Některé kognitivní funkce u schizofreniků mají míru poškození mírnou, jiné zase těžkou. Jako příklad mírné míry poškození se uvádí například dlouhodobá paměť či percepční schopnosti. Střední poškození se týká výběrové pozornosti a poté ovlivňuje několik oblastí epizodické paměti. Těžké poškození se týká například prostorové pracovní paměti a verbální pracovní paměti. Ve shrnutí tedy můžeme říci, že dochází k poruše pozornosti, pracovní paměti a exekutivních funkcí.

Léčení tedy zahrnuje různé léky, ale i odborné terapie. Ať už ke zlepšení psychického stavu či narušených kognitivních funkcí. [20]

4.2 Vliv parametrizovaných her na vývoj onemocnění

Při neurehabilitacích a využití parametrizovaných her dochází k pozitivnímu vlivu nejen na kognitivní funkce pacienta, ale i na jeho psychickou stránku. Jak již bylo zmíněno v úvodu, parametry slouží k přizpůsobení hry tak, aby pacienta motivoval jeho posun. Začíná se od nejjednoduchších úkolů po těžší. Předpokladem je zlepšení nejen narušených kognitivních funkcí, ale i vedlejších nepoškozených. Zde bych ráda uvedla obecný výčet věcí, které se zlepšují při hraní neurorehabilitačních her.

- zlepšení paměti
- zlepšená orientace na ploše -> při hraní pacienti vykonávají na obrazovce pátravý pohyb
- rychlejší zlepšení celkového zdravotního stavu -> hry jsou vytvářeny zábavnou formou, tudíž mají pacienti nutkání se ke hrám vracet i v osobním volnu
- získání nových slov -> rozšíření slovní zásoby
- zlepšování zraku, sluchu, koncentrace

A to je jen základní výčet výhod, které hry pacientům přináší. Mezi další bonusy her patří již zmíněná motivace, jelikož pacienti se chtějí posouvat k těžším úkolům a překonávat sebe samotné. Zároveň pokud je nějaký pacient indisponován, například nemůže psát dominantní rukou a papírové cvičení jsou pro něj náročné, tak klikání na tabletu či počítači pro něj představuje zjednodušenou formu terapie.

5 Analýza her

Jak již bylo několikrát zmíněno, jako výsledek této práce vzniknou čtyři parametrizované logopedické hry a v této kapitole bych je ráda představila. Náměty na hry jsem získala od paní magistry Ivany Herejkové z Fakultní nemocnice Plzeň, se kterou v rámci BP spolupracuji. Všechny hry budou přizpůsobeny pro hraní v českém, anglickém a německém jazyce, tak jak to vyžaduje mezinárodní projekt BrainIn.

5.1 Skládání slov

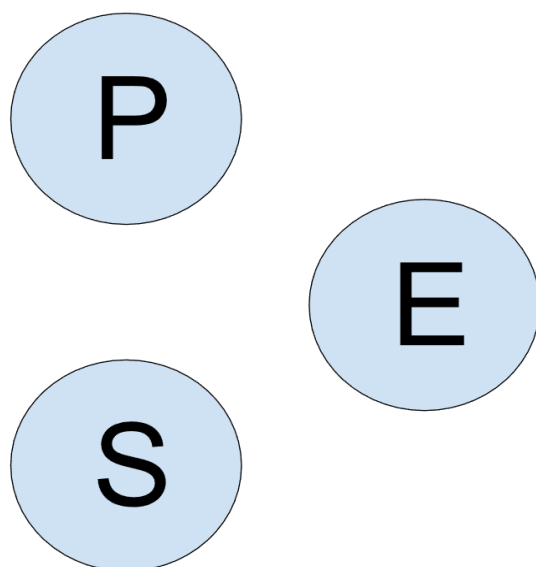
Hra, zaměřená na spojení písmen v logická slova pomocí čar.

Průběh jednoho kola:

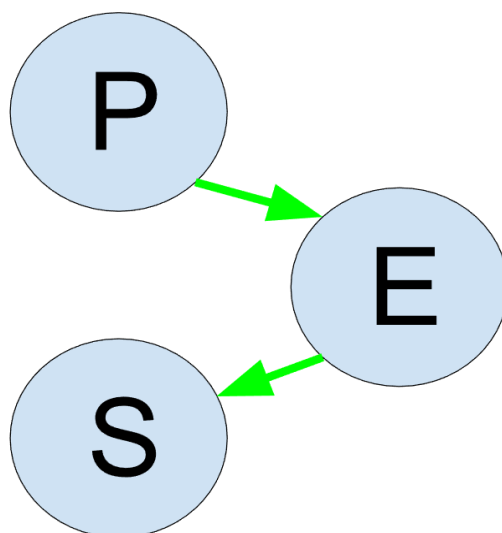
- **Vstup:**
 - **číselný vstup 7** neboli vygeneruj slovo o délce 7 v tomto kole
 - **slovo jablko** neboli v tomto kole bude slovo jablko
- *celkové vstupní parametry: základní ze šablony; slova či čísla, kdy počet se musí rovnat počtu kol*
- **Výstup:** spojené slovo od pacienta
- *celkové výstupní parametry: základní ze šablony*
- *dílčí výstupní parametry: počet chybných spojů; chybné spoje*
- **Vyhodnocení:** porovnání vstupu terapeuta a výstupu pacienta na konci každého kola

Jak tedy vyplývá ze vstupu v této hře se skrývají dvě podhry. Buď se terapeut rozhodne pro svá slova a ta se vyobrazí nebo zadá čísla, podle nichž se vyberou náhodná slova ze speciálního slovníku.

Pokud se terapeut rozhodne hru přidělit s náhodnými slovy, domluvili jsme se, že nebude zadávat číslo menší jak 3 a větší jak 11. Zároveň slova pro náhodný výběr budou ve speciálním slovníku namíru upraveném pro tuto hru. Bude čítat kolem 100 až 200 slov. Aby nedošlo k vyčerpání slov, bude možné pro každé číslo maximálně 12 kol.



Obrázek 5.1: Skládání slov - ukázka vstup



Obrázek 5.2: Skládání slov - ukázka výstup

V této hře bude hlavní problematikou rozmístění jednotlivých objektů s písmeny po ploše. Tento problém by měl být vyřešen implementací Voroného diagramu. Také zde budou muset být naimplementovány metody, které pokryjí oba druhy spuštění a pro českou lokalizaci se nesmí opomenout správné zpracování písmena *ch*.

5.1.1 Voroného diagram

Voroného diagram je rozdělení roviny n body do konvexních mnohoúhelníků tak, že každý mnohoúhelník obsahuje přesně jeden generující bod a každý bod v daném mnohoúhelníku je blíže svému generujícímu bodu než kterémukoli jinému. Nachází uplatnění v oblastech počítačové grafiky, epidemiologie, geofyziky či meteorologie. [23]

Pro výpočet Voroného diagramu můžeme použít několik algoritmů. Zde si na ukázkou některé představíme.

- **Inkrementální:** najdeme diagram pro dva body z množiny generátorů a postupně přidáváme zbylé body.
- **Fortuneho:** využíváme přímku, která se pohybuje jedním směrem a uchovává při pohybu veškeré informace potřebné pro výpočet, zároveň se při pohybu vytváří požadovaná struktura.
- **Naivní:** přímo aplikuje matematickou definici Voroného diagramu.

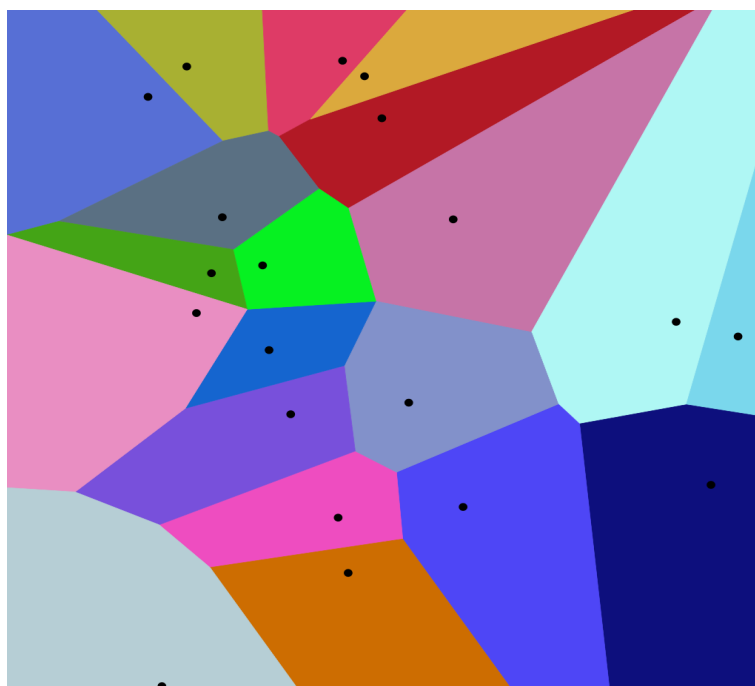
S pojmem Voroného diagramu také úzce souvisí pojmy Delaunayho triangulace a Lloydův algoritmus. Delaunayho triangulaci získáme, pokud spojíme navzájem úsečkami ty body sítě, jejichž buňky ve Voroného diagramu sousedí. Zajímavostí je, že se jedná o dualitu k Voroného diagramu tedy jedno určuje druhé. Lloydův algoritmus je iterací Voroného diagramu, který slouží k rovnoměrnějšímu rozdělení bodů v Euklidovských prostorech. [33]

5.2 Vodorovné křížovky

Hra, kdy bude pacient doplňovat vhodná slova do prázdných čtverečků ve vodorovném směru.

Průběh jednoho kola:

- **Vstup:** terapeut zadá slovo, která se má objevit v kole hry



Obrázek 5.3: 20 bodů a jejich buňky dle Voroného diagramu [24]

- *celkové vstupní parametry: základní ze šablony; slova, která se budou objevovat uprostřed, musí se rovnat počtu kol; délka slov, které má pacient vymýšlet; počet vymýšlených slov v jednom kole se určí na základě počtu písmenek slova pro příslušné kolo*
- **Výstup:** doplnění slov
- *celkové výstupní parametry: základní ze šablony*
- *dílčí výstupní parametry: výpis chybných slov; výpis správných slov*
- **Vyhodnocení:** porovnání vzniklých slov se slovníkem a následné vyhodnocení

V této hře bude hlavním problémem proměnlivá délka slov. Koncový algoritmus bude muset tvořit vhodně velké čtverce a volit jejich umístění tak, aby křížovka byla použitelná pro slova o délce tři stejně jako pro slova o délce deset. Problém také bude představovat písmeno *ch*, které bude muset v programu být ošetřeno. Kódově je totiž písmeno *ch* brané jako dva znaky *c* + *h*. Jedna z možností je v kódu nahradit toto písmeno za jiný znak, aby se s ním lépe pracovalo. Například za *@*. Příhodné bude i v příloženém slovníku pro českou lokalizaci předělat *ch* na *@*.

	P	
	E	
	S	

Obrázek 5.4: Vodorovné křížovky - ukázka vstup

Ú	P	A
L	E	S
O	S	A

Obrázek 5.5: Vodorovné křížovky - ukázka výstup

5.3 Doplnění slov

Hra, kdy bude zadáno pacientovi počáteční a koncové písmenko a pacient, bude muset doplnit písmenka tak, aby vznikla logická slova.

Průběh jednoho kola:

- **Vstup:** terapeut zadá počáteční a koncové písmeno
- *celkové vstupní parametry: základní ze šablony; dvojice vstupních a koncových písmen, z které se musí rovnat počtu kol; počet slov, které má pacient vymyslet v rámci jednoho kola*
- **Výstup:** napíše požadovaný počet slov začínající a končící na písmeno od terapeuta
- *celkové výstupní parametry: základní ze šablony*
- *dílčí výstupní parametry: seznam chybných slov; počet chybných slov; výpis správných slov*
- **Vyhodnocení:** porovnání vzniklých slov se slovníkem a následné vyhodnocení

Ukázka vstupu a výstupu:

- VSTUP: KA, 2 slova VÝSTUP: koala, kozačka

5.4 Slova, které začínají a končí na stejné písmeno

Hra, kde bude muset pacient vymýšlet slova začínající a končící na stejné písmeno.

Průběh jednoho kola:

- **Vstup:** terapeut zadá písmeno
- *celkové vstupní parametry: základní ze šablony; písmena, jejich počet se musí rovnat počtu kol; počet slov, které má pacient vymyslet v rámci jednoho kola*
- **Výstup:** napíše požadovaný počet slov začínající a končící na písmeno od terapeuta

- *celkové výstupní parametry: základní ze šablony*
- *dílčí výstupní parametry: seznam chybných slov; počet chybných slov; výpis správných slov*
- **Vyhodnocení:** porovnání vzniklých slov se slovníkem a následné vyhodnocení

Ukázka vstupu a výstupu:

- VSTUP: A, 2 slova VÝSTUP: anakonda, andulka

U her 5.3 a 5.4 bude problematika velmi podobná. Už při analýze si musíme uvědomit, že půjde o hry, které budou vyžadovat podobnou implementaci. Budeme muset vytvořit vhodné UI, které bude ukazovat náležitosti dané hry. Na obrazovce by neměl chybět výpis napsaných slov či počet slov zbývajících. Také musíme vhodně vymyslet kontroly vstupních parametrů, které jsou pro celou hru stěžejní. Zároveň pro hru 5.4 bude speciální slovník, který bude obsahovat jen výrazy, které končí a začínají na stejné písmeno.

Slovník

V rámci analýzy se několikrát objevilo slovo slovník. Pro skládání slov se bude dělat speciální slovník, stejně tak pro hru, kde se skládají slova co končí a začínají na stejné písmeno. Pro ostatní hry bude stačit jeden společný. Akorát česká lokalizace pro křížovky bude obsahovat @ místo *ch*. Mezi další lokalizace bude patřit německý a anglický jazyk. Mezi možnosti implementace se řadí databáze či textový soubor. V případě volby textového souboru se bude brát zřetel na velikost souboru, aby šly hry plynule spustit i na méně výkonných počítačích. Do slovníku se budou čerpat slova z běžně dostupných zdrojů, a to z českých národních korpusů.

6 Implementace her

V této kapitole, jak už název napovídá, se budeme věnovat implementaci jednotlivých her. Jelikož je ale BrainIn zaběhnutým systémem vyskytují se tu společné implementační prvky, které je potřeba dodržet a my se s nimi nejdříve společně seznámíme.

6.1 Společné prvky

6.1.1 Použité technologie

Technologie využité při implementaci byly zmíněny v podkapitole 2.2, zde si ale o nich řekneme bližší informace. Všechny hry, které na webu jsou dostupné nebo jsou ve vývoji, se vytvářejí v herním enginu pro tvorbu her **Unity**. Unity nabízí několik různých verzí, ovšem naše hry jsou tvořeny ve verzi 2018.4.7f1 (64 bitová verze). Tato verze je doporučena vývojáři samotné webové aplikace. Jde s ní nejsnáze předejít problémům, které vznikají při použití rozdílných verzí. Sama jsem se o tomto přesvědčila, když jsem hry zkoušela implementovat ve verzích 2019 a 2020, podrobněji tuto problematiku rozvedu v kapitole 8.

O funkčnost herních objektů se starají skripty, které jsou psány v jazyce C#.

Komunikace mezi hrou a webovým serverem je tvořena pomocí objektově orientovaného programovacího jazyka JavaScriptu.

6.1.2 Šablona šablon

Co to je šablona již víme, co se ale skrývá pod pojmem šablona šablon? Můžeme si to představit jako vzor, ze kterého vycházejí veškeré šablony tvořené na webu. Zjednodušeně se dá říci, že je to předloha pro vytváření úloh. My modifikací společného implementačního základu vytvoříme námi požadovanou hru. Díky šabloně šablon se ale při implementaci nemusíme soustředit na metody, které jsou pro všechny hry společné např. sběr dat.

Soubor Šablona šablon se strukturuje do dvou hlavních složek, kdy každá složka má na starost specifickou funkcionalitu.

Build

Zjednodušeně se dá říci, že se jedná o vyexportovanou aplikaci neboli spustitelnou aplikaci, která se nahrává na web.

Assets

Složka Assets čítá několik podadresářů, které skrývají veškeré potřebné soubory pro tvorbu hry.

- **Animations:** animace pro zabarvení okrajů; zelená animace při správné odpovědi; červená animace při špatné odpovědi
- **Fonts:** veškeré fonty, použité ve hře
- **Images:** potřebné obrázky, např. semafor, který se zobrazuje mezi jednotlivými koly
- **JsonDotNet:** uchování dat
- **Plugins:** použité knihovny
- **Prefaby = vzory objektů**
- **Scenes:** vytvořené scény pro danou hru
- **Scripts:** skripty, které se starají o zvuk, pohyby klávesnice či zmáčknutí tlačítka myši, ale také se zde nacházejí například skripty pro objekty
- **StreamingAssets -> ProgramData:** lokalizační soubory se zvukem či překlady pro errorry, vstupy a upozornění
- **TextMeshPro:** pokročilé techniky vykreslování textu

6.1.3 Lokalizační soubory

Nahrávky

Ke každé hře se musí vytvořit lokalizované nahrávky pomocí nástroje Text-ToSpeech, který poskytuje například webová stránka <https://wideo.co/text-to-speech/>. Já jsem je vytvářela ve třech požadovaných jazycích. Pro anglický jazyk byl použit hlas Alishy Howardové, pro německý Lisy Fisherové a český jazyk prezentuje hlas Marie Hendersonové. Tyto mluvené soubory slouží například k přečtení zadání dané hry.

Překlady

Stejně jako je nutné mít hlasové nahrávky ve třech jazycích je zapotřebí, aby veškeré texty ve hře byly přeložené do němčiny a angličtiny. Texty jsou uchovávané v xml souborech ve složce ProgramData. Do těchto souborů vkládáme veškeré použité textové řetězce pro jednotlivé hry. Další překlady je nutné vytvořit při nasazení her do webové aplikace, kde se tvoří popis hry či popisy parametrů.

6.2 Skládání slov

Nyní se zaměříme na implementování první ze čtyř her. Tato hra má za cíl spojení písmen v logická slova a díky svému vzhledu získala v průběhu tvorby pracovní název Bublinky.

Při spuštění hry dochází v kódu ke zpracování vstupních parametrů, kde pro nás je nejdůležitější parametr na posledním místě. Ten může obsahovat dva druhy vstupu podle toho jaký typ podhry chceme spustit.

Po kontrole parametrů přichází rozdělení dle druhu vstupu. Volá se metoda *CreateRandomRounds()* pro číselný vstup či *CreateRounds()* pro vstup se slovy. Než metoda *CreateRandomRounds()* přejde do stejného momentu tvoření kol, musí nejdříve pomocí metody *PickWord()* (ukázka viz Kód 6.1) najít vhodná slova, která se zobrazí. Předtím ale ještě kontroluje zda zadaná délka je větší jak 3 a menší jak 11.

```
1      List<string> wordlist = new List<string>(txt.  
2      text.Split('\n'));  
3  
4      List<String> acceptableWords;  
5      acceptableWords = new List<string>();  
6      foreach (string line in wordlist)  
7      {  
8          string word = Regex.Replace(line, @"\s+", "");  
9  
10         if (Locale.GetCode() == Constants.CS_LOCALE)  
11         {  
12             int numberOfCh = Regex.Matches(word, "ch").  
13             Count;  
14             int numberOfCh2 = Regex.Matches(word, "Ch").  
15             Count;  
16  
17             if (word.Length - numberOfCh -  
18             numberOfCh2 == length)  
19             {
```

```

20         acceptableWords.Add(word);
21     }
22 }
23 else
24 {
25     if (word.Length == length)
26     {
27         acceptableWords.Add(word);
28     }
29 }
30 }
31 Random random = new Random();
32 int rnd = random.Next(0, acceptableWords.Count);
33 string pickWordForRound = acceptableWords[rnd];

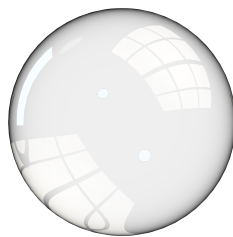
```

Kód 6.1: Ukázka z metody PickWord()

Poté následuje v obou typech spuštění stejná implementace, která začíná tvorbou kol s požadovanými slovy. Aby celá naše hra mohla fungovat, potřebujeme mít vytvořenou bublinku.

Tvorba bubliny

Bublina je objekt vytvořený pomocí transparentního obrázku. K objektu je přiřazen skript *Bubble.cs*. Ten v sobě skrývá tři parametry, z nichž velmi důležitý je parametr *finished*, který říká, zda do bublinky může vést spoj. Jakmile je bublinka dokončena *finished = true*, je nemožné do ní spoj vést.



Obrázek 6.1: Objekt Bublina

Rozmístění bublin

K implementaci rozmístění bublin se využívá volně dostupná hotová knihovna Delaunay [30], která nabízí metody pro tvorbu Voroného diagramu. Jako výsledek získáme body pro umístění bublin, viz Kód 6.2.

```

1     public Dictionary<Vector2f, Site> GetPoints(int
2     numberOfPoints) {
3         //centra polygonu
4         List<Vector2f> points =
5         CreateRandomPoint(numberOfPoints);
6         //hrany voroneho diagramu
7         GameObject canvas = GameObject.Find("Canvas Game");
8         RectTransform rt = (RectTransform)canvas.transform;
9         float bottomLeftX = canvas.transform.position.x -
10        (rt.rect.width / 2);
11        float bottomLeftY = canvas.transform.position.y -
12        (rt.rect.height / 2);
13        Rectf bounds = new Rectf(bottomLeftX, bottomLeftY,
14        rt.rect.width - (rt.rect.width/100)*10,
15        rt.rect.height -
16        (rt.rect.height/100) *10);
17        //lloydova iterace
18        Voronoi voronoi = new Voronoi(points, bounds, 5);
19        //finalni body
20        sites = voronoi.SitesIndexedByLocation;
21        return sites;
22    }

```

Kód 6.2: Ukázka metody, ze které získáme body pro umístění bublin

Po zobrazení bublin na herní scéně, můžeme začít objekty spojovat. K tomu je zapotřebí vytvořit prefab čáry. V základu bude mít čára modrou barvu. Poté co pacient provede spojení, se spoj zkontroluje a na základě vyhodnocení se obarví. Po spojení celého slova se pokračuje do dalšího kola.

Kreslení čáry, kontrola spoje

Aby pacient mohl čáru nakreslit je implementována metoda *drawLine()*, ta obsahuje jednotlivé listenery, které reagují na chování myši. Pro potřeby hry sledujeme stisk levého tlačítka, tah myši či puštění levého tlačítka.

Listener pro stisk levého tlačítka rozezná zda se kliknutí provedlo na objektu bubliny nebo mimo objekt. Při kliknutí mimo se nic nestane, při kliknutí na bublinu se začne vykreslovat čára a zároveň se uloží startovní objekt. Po puštění levého tlačítka se zkontroluje zda čára končí na objektu bubliny. Pokud ne, čára zmizí. Pokud ano, porovnájí se písmena na startovní a koncové bublině spolu s požadovaným spojením. K tomu slouží metoda *checkConnection()* viz Kód 6.3. Abychom mohli sledovat pohyb čáry je zaveden ještě jeden listener navíc.

```

1     public bool checkConnection(string a, string b) {
2         //offset slouzi k urceni spravne pozice ve slove
3         string subWordA = InputParams.Rounds[round].
4         Word_arr[0+offset];
5         string subWordB = InputParams.Rounds[round].
6         Word_arr[1+offset];
7
8         if(subWordA == a && subWordB == b) {
9             offset += 1;
10            return true;
11        }
12        return false;
13    }

```

Kód 6.3: Ukázka metody, která kontroluje spojení

Při správném spojení se čára obarví zeleně, při špatném se obarví červeně.

Výstupní celková i dílčí data

Zároveň jsou ve výše zmíněném listeneru na puštění tlačítka myši špatné spoje zanášeny do dílčích výstupních dat pomocí metody *updateWrongConnectionNumber()* a *updateWrongConnections()* viz Kód 6.4. Jak už názvy metod napovídají, jedná se o počet špatných spojů a list špatných spojů. O počet správných spojů se stará *updateCorrectConnectionNumber()*.

```

1     public void updateWrongConnections(string a, string b) {
2         string newWrongConnection = a + "," + b;
3         currentRound.WrongConnections.
4         Add(newWrongConnection);
5     }

```

Kód 6.4: Ukázka metody, která přidá špatný spoj do dílčích výstupních dat

Pokud by nás zajímala celková výstupní data museli bychom se podívat do skriptu *GameResults.cs*. Ten v příslušné metodě *ToString()* tvoří výstupní řetězec, který převádí do JSON formátu.

Doplňující metody

Kromě drobných zásahů do existující šablony šablon bylo potřeba vytvořit ještě několik doplňujících metod.

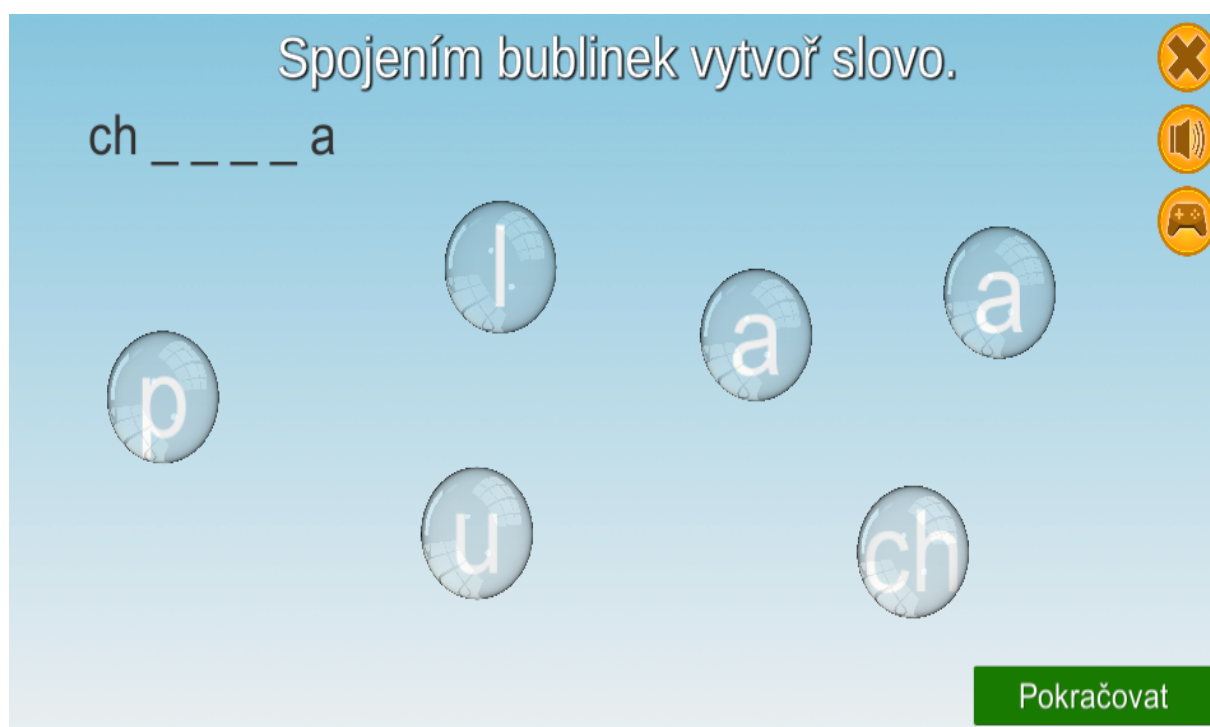
- Před začátkem dalšího kola je potřeba zničit veškeré objekty zobrazené na herní scéně, k tomu slouží metoda *Destroy()*.

- Pro zničení čáry při špatném spoji je implementována metoda *deleteLine()*.
- Při tvorbě nového kola se v konstruktoru kromě Voroného diagramu rozkládá vstupní slovo na jednotlivá písmena a je zde zajištěno aby v české lokalizaci *ch* bylo brané jako jedno písmeno. Princip spočívá v tom, že se testuje, zda vstupní slovo obsahuje posloupnost písmen *c* a *h*. Nesmí se opomenout snížení počtu bublinek o jedna.

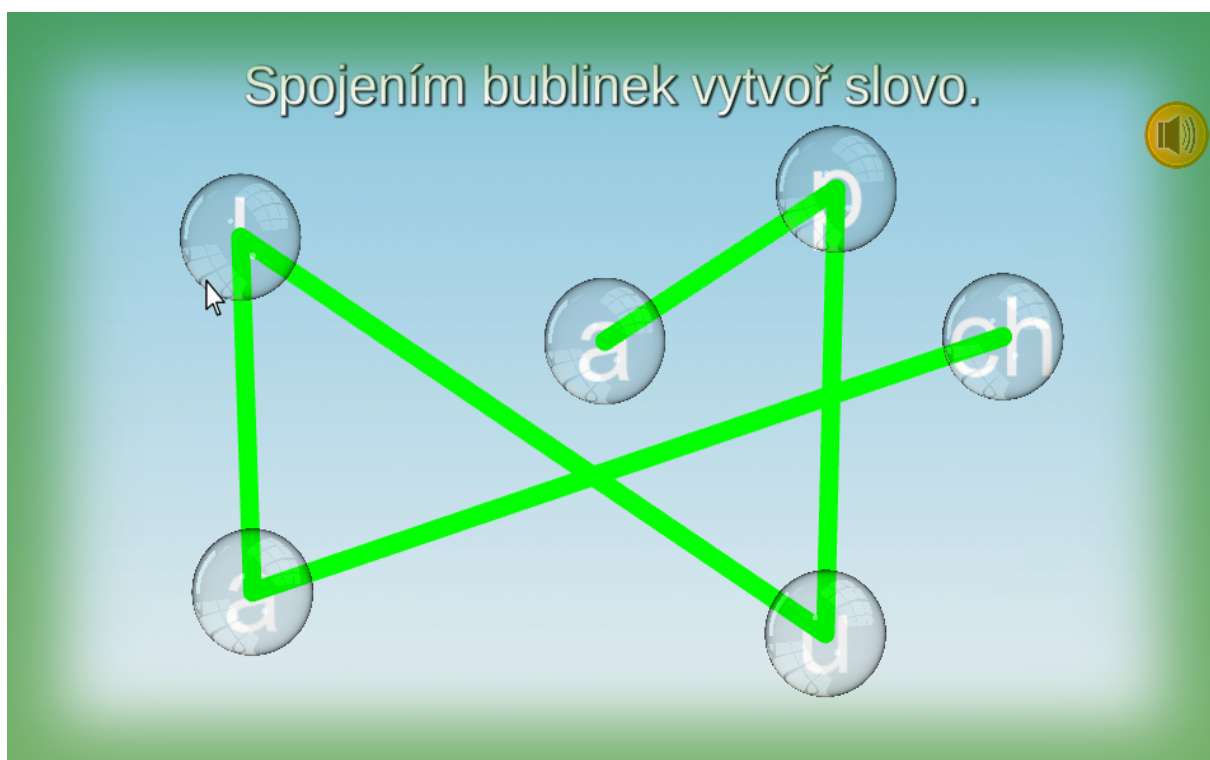
Nápověda

V této hře můžeme najít nápovědu o kterou se stará metoda *printHelp()*. Ta je implementována tak, že se na herní scéně objeví text počátečního a koncového písmenka. Mezi nimi jsou symbolizující písmena. Pro českou lokalizaci se musí sledovat počet *ch*, aby se nápověda zobrazila správně.

Nápovědu můžeme vidět na obrázku 19, kde je zároveň vidět vstup hry. Výstupu si můžeme všimnout na obrázku 20.



Obrázek 6.2: Skládání slov - výsledná hra na vstupu, včetně nápovědy



Obrázek 6.3: Skládání slov - výsledná hra, výstup slovo chalupa

6.3 Vodorovné křížovky

Druhou implementovanou hrou se staly Křížovky.

Vstupní parametry + kontrola

Při zpracování vstupních parametrů, jsou pro nás nejdůležitější dva parametry od konce. První z nich představuje slova, která se zobrazí jako tajenka. Druhý znázorňuje délku vymyšlených slov.

Jakmile máme zadané tajenky a délky, je důležité zkontrolovat několik bodů. Mezi tyto body patří vyhledání *ch* v tajenkách pro českou lokalizaci a jeho nahrazení za znak *@* viz Kód 6.5 (důvody byly popsány již v analýze úlohy), zda se počet slov shoduje s počtem zadaných délek či zda žádná délka není menší než tři. Podmínkou je totiž vymyslet slova o minimální délce třech znaků.

```

1   for (int i = 0; i < words.Length; i++) {
2       if (Locale.GetCode() == "cs") {
3           words[i] = words[i].Replace("ch", "@");
4       }
5   }

```

Kód 6.5: Nahrazení ch

Poté se začnou předpřipravovat jednotlivá kola. S přípravou přichází kontrola zda vstup od terapeuta je splnitelný. Například pokud bychom měli vymýšlet slovo o délce tři a na prostřední pozici by byl znak q, pravěpodobně bychom žádné slovo nevymysleli. Proto raději dochází v kódu k logické kontrole na které spolupracují dvě metody viz Kód 6.6 a Kód 6.7.

```
1 public void RightWords() {
2     //abychom mohli hledat vhodna slova pro nase zadani,
3     musime podle lokazilace zvolit spravny slovník
4     TextAsset txt = null;
5     switch (Locale.GetCode()) {
6         case "cs":
7             txt = (TextAsset)Resources.Load("slovník_cs",
8             typeof(TextAsset));
9             break;
10        case "en":
11            txt = (TextAsset)Resources.Load("slovník_de",
12            typeof(TextAsset));
13            break;
14        case "de":
15            txt = (TextAsset)Resources.Load("slovník_en",
16            typeof(TextAsset));
17            break;
18    }
19
20    List<string> lines = new List<string>
21    (txt.text.Split('\n'));
22
23    foreach (string line in lines) {
24        string word = Regex.Replace(line, @"\s+", "");
25        if (word.Length <= 2) {
26            continue;
27        }
28        // pokud slovo ve slovníku ma pozadovanou delku,
29        kontrola zda obsahuje pismeno z tajenky na
30        prostredni pozici
31        if (word.Length == columns) {
32            for (int i = 0; i < secret.Length; i++) {
33                if (word[columns / 2] == secret[i]) {
34                    rightWords.Add(word);
35                    letterFromSecret[i] = 1;
36                }
37            }
38        }
39    }
40 }
```

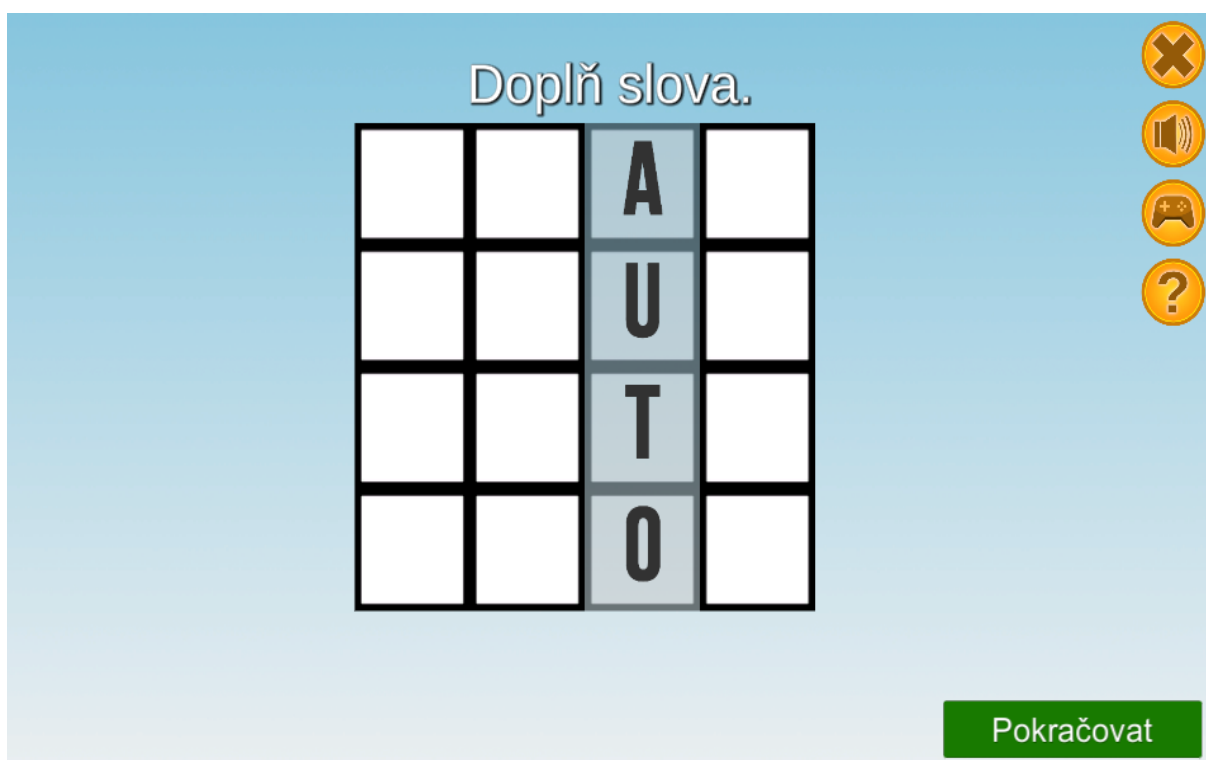
Kód 6.6: Metoda, která hledá vhodná slovo pro doplnění a kontroluje výskyt písmene z tajenky

```
1 public bool itsPossible() {  
2     for (int i =0; i < letterFromSecret.Length; i++) {  
3         if(letterFromSecret[i] == 0) {  
4             return false;  
5         }  
6     }  
7     return true;  
8 }
```

Kód 6.7: Metoda, která rozhoduje o tom, zda je možné splnit zadání

Ukázka hry

Po kontrole všech bodů dochází k zobrazení prvního zadání na obrazovce. Velká písmena byly požadavkem paní Herejkové, jelikož je v této hře pro pacienty lepší nerozeznávat malá/velká písmena.



Obrázek 6.4: Křížovky - výsledná hra na vstupu

Ovšem abychom mohli takového výstupu na obrazovce docílit, je v kódu zapotřebí naimplementovat několik objektů a metod.

Tvorba těla křížovky

Nejdříve je zapotřebí si vytvořit jeden čtverec, k tomu nám poslouží obrázek, který si uložíme jako prefab. Pokud máme připravený jeden tvar, je namísto vytvořit tělo křížovky.

K vytvoření těla křížovky jsou zapotřebí tři metody: *calculateRatio()*, *calculateStartPosition()* a *generate()* viz Kód 6.8. Tyto tři metody byly implementovány na základě matematických výpočtů a jsou volány před spuštěním každého kola.

```
1 for (int x = 0; x < InputParams.Rounds[round].
2   columns; x++) {
3     GameObject block = Instantiate(rect,
4     new Vector2(startPosition.x +
5     (x * offset), startPosition.y - (y * offset)),
6     Quaternion.identity);
7     Vector3 scale = block.transform.localScale;
8     block.transform.parent = canvas.transform;
9     block.transform.localScale = new Vector3(
10    scale.x * ratio,
11    scale.y * ratio,
12    scale.z);
13    block.tag = "Input";
14 }
```

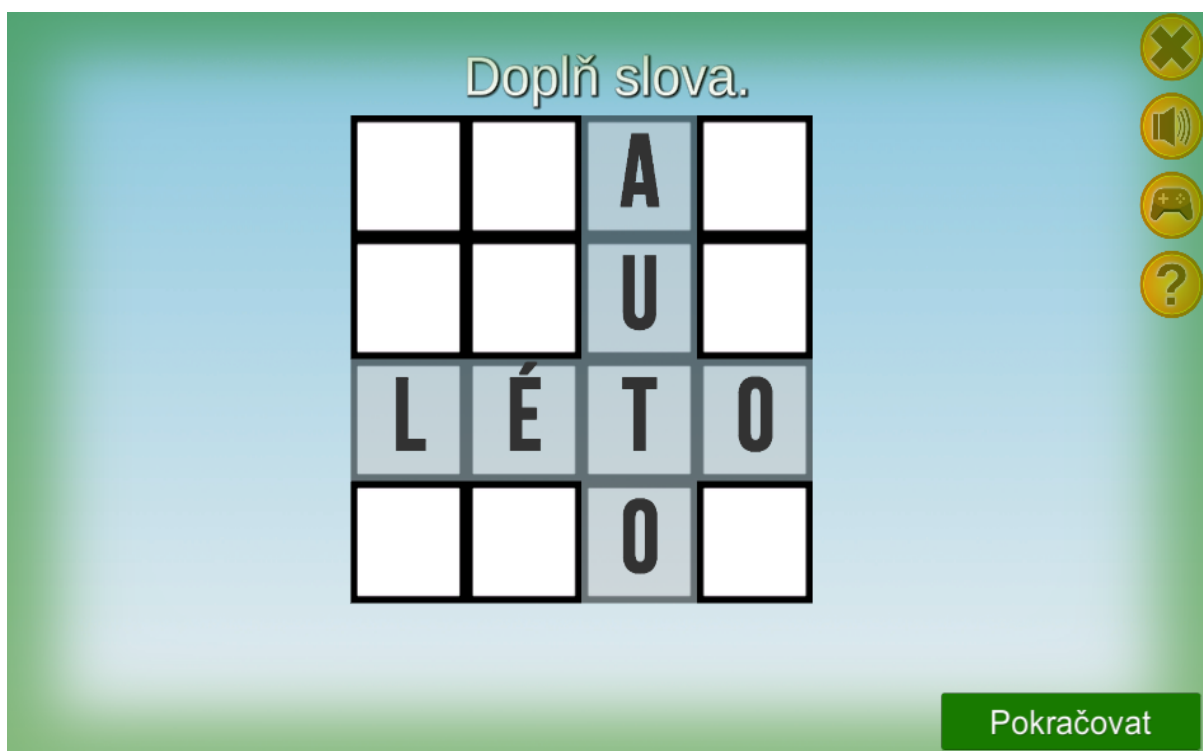
Kód 6.8: Ukázka z metody, která generuje tělo křížovky

Kontrola vyplněných slov

V tento moment už jsme docílili ukázkové scény a můžeme do těla křížovky vyplňovat slova.

Ke kontrole slov v této hře vznikl nový skript, který nese název *Checker.cs*. Tento skript obsahuje metodu, která se zavolá pokaždé, když se napíše nový znak do křížovky. Unity příhodně disponuje funkcí *On Value Changed* do které můžeme přidat vlastní metodu, která se zavolá při jakékoliv změně. Metoda *checkWord()* tedy rozhodne, zda je v řádku smyslupné slovo, které splňuje zadání či ne. Také zkontroluje zda se dané slovo již v křížovce nevyskytuje. Je důležité nezapomenout převést @ zpět na *ch*. Pokud pacient zadal správné slovo, nastaví se buňky jako neinteraktivní, tedy nemůže je měnit. Jinak může pacient přepisovat dosavadní písmena a skládat nové slovo.

```
1 if (sb.ToString() == round.rightWords[k]) {
2     main.Audio.GetComponent<AudioPlayer>().
3     Play(true);
4 }
```



Obrázek 6.5: Křížovky - průběh hry, zápis správného slova

```

5   GameObject.FindObjectOfType<Main>().
6   Results.addRightWord
7   (sb.ToString());
8   for (int l = 0; l < round.columns; l++) {
9       round.cells[l, i].
10      interactable=false;
11  }
12 }

```

Kód 6.9: Ukázka z checkWord(), která rozhodne o správnosti slova

Dílčí výstupní parametry

V metodě CheckWord() se zároveň volají metody pro uchování správných či nesprávných slov. Tyto dva údaje tvoří dílčí výstup této hry.

Celá implementace je ale trochu složitější, jelikož zmíněné metody můžeme najít ve skriptu *Results.cs* a v jejich těle se modifikují parametry, které jsou součástí skriptu *Round.cs* (reprezentant jednoho kola). Zde pak i nacházíme řetězec pro získání dílčích výsledků.

Celkové výstupní data bychom získali stejně jako v předešlé hře.

Další implementační záležitosti

Metody, které jsou využívány pro správný běh hry jsou popsány výše. V kódu se ale objevuje několik drobnějších zásahů do šablonních skriptů. Například před začátkem každého kola se v metodě *BeforeNextRound()* musíme postarat o zničení všech čtverců na scéně, aby se mohlo nové plátno vytvořit správným způsobem.

Křížovky máme naimplementované a můžeme se přesunout k implementaci předposlední hry.

6.4 Doplnění slov

Vstupní parametry + kontrola

Před vykreslením herní scény a začátkem hry se provádí důležitý krok, který je nám již dobře známý, tím je zpracování parametrů. Nejdůležitější jsou pro nás parametry na předposledním a posledním místě (dvojice písmen a počet slov). Po získání parametrů se zkontroluje v metodě *CheckAssignment()* zda terapeut zadal splnitelné požadavky. Kontroluje se jestli počet existujících slov není roven 0 pro zadanou dvojici počátečního a koncového písmena, ukázkou z metody můžeme vidět v Kódu 6.12.

```
1 for (int i = 0; i < numOfRounds; i++) {
2     Rounds[i] = new RoundData(letters[i][0].ToString(),
3     letters[i][1].ToString(), int.Parse(countWordsInRounds[i]
4     ));
5     if (Rounds[i].acceptableWords.
6     Count == 0) {
7         throw new InputParamsException(String.
8         Format(ConfigFiles.Errors.Data.ZeroWordsFound.
9         ToString(), letters[i]));
10 }
```

Kód 6.10: Ukázka z *CheckAssignment()*

Mohli jsme si všimnout, že v metodě máme znalost o tom, jak dlouhý je seznam akceptovaných slov. Akceptovaná slova se získávají z metody, která se volá v konstruktoru skriptu *Round.cs*, jehož instance existuje pro každé vzniklé kolo. Dojde v něm k zavolání metody *getWordsFromVocab()*. Tato metoda získá veškerá slova z příslušného slovníku dle lokalizace a každé slovo zkontroluje, zda začíná a končí na zadaný vstup od terapeuta. Pokud ano, přidá ho do listu. Ukázkou obsahuje Kód 6.13.

```

1 List<string> lines = new List<string>(txt.text.Split('\n'));
2     foreach (string line in lines) {
3         string word = Regex.Replace(line, @"\s+", "");
4         if (word.Length < 3) {
5             continue;
6         }
7         if (word[0] == this.startLetter[0] && word[word.
8             Length - 1] == this.endLetter[0]) {
9             acceptableWords.Add(word);
10        }
11    }

```

Kód 6.11: Ukázka z `getWordsFromVocab()`

Vykreslení scény

Po úspěšné kontrole se přechází k vykreslení herní scény. Tentokrát nám postačí pouze základní objekty, které nám nabízí herní engine Unity. Pomocí řídicí struktury *switch*, která rozhoduje na základě lokalizace, se nastaví na scéně vhodné doplňující texty viz Obrázek 23. Správné zadání ze získaných vstupních parametrů zobrazí metoda *NextRound()*, která se provede vždy na začátku kola.

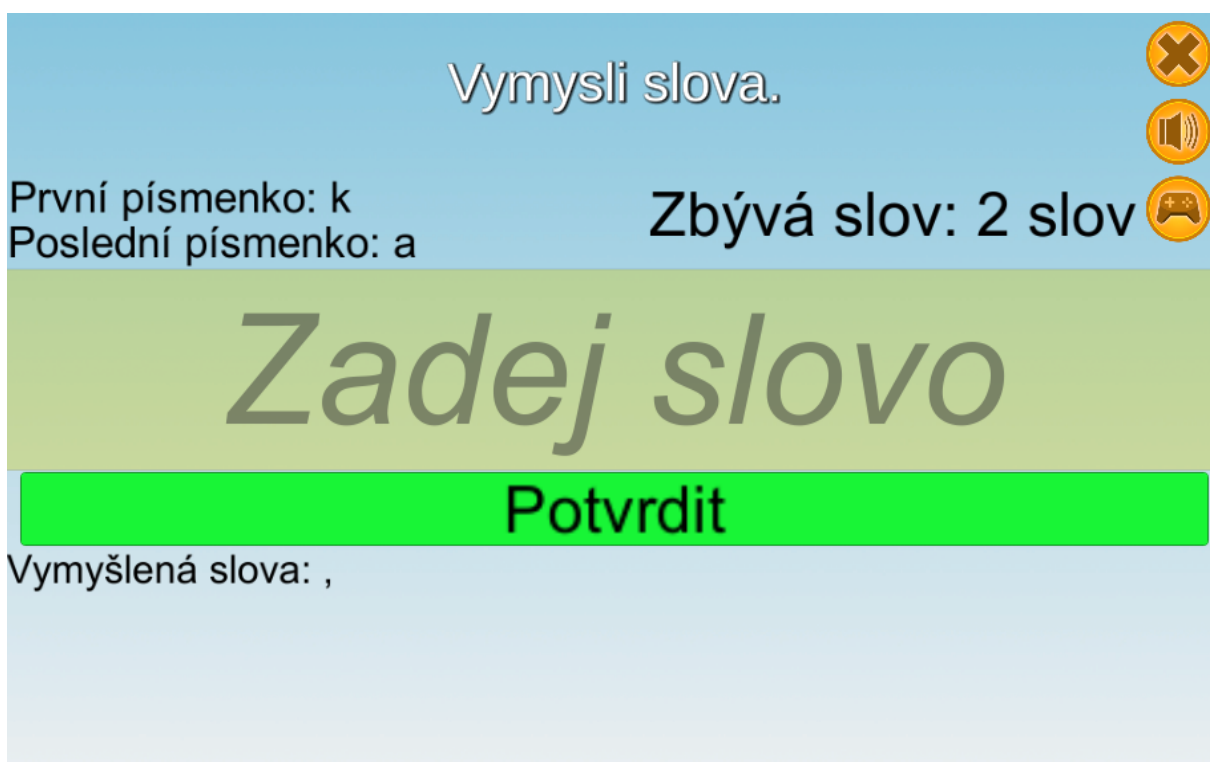
Kontrola napsaného slova

Po vykreslení scény můžeme začít hrát. Jakmile nás napadne vhodné slovo zapíšeme ho do prostředního objektu *Zadej slovo* a klikneme na tlačítko *Potvrdit*. Správné slovo se poté zobrazí v sekci *Vymyšlená slova*.

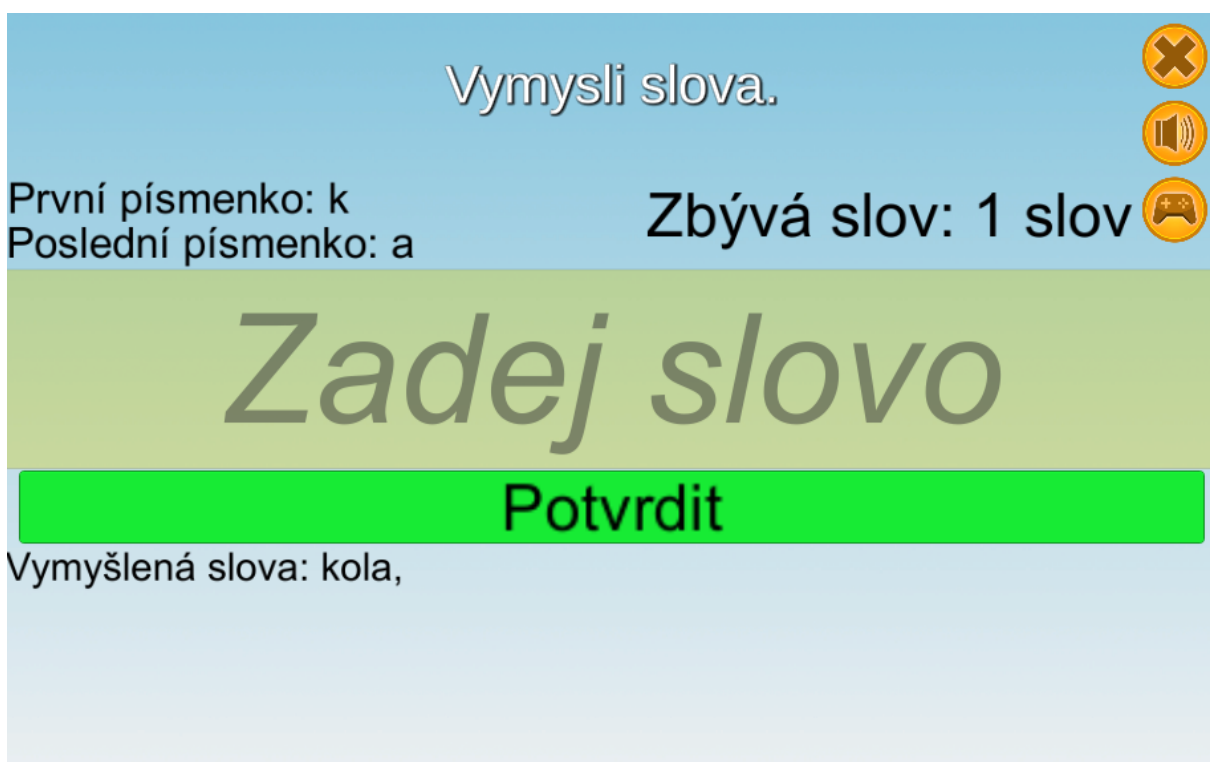
Abychom vypsali správné slovo na herní scénu potřebujeme metodu *confirmWord()* viz Kód 6.14. Tato metoda se volá při každém kliknutí na tlačítko *Potvrdit* díky funkci *On Click()*.

ConfirmWord() v těle volá metodu pro kontrolu slova *CheckWord* (viz Kód 6.15). Na základě správnosti ho zobrazí na obrazovku a nebo jen předá do výstupní proměnné. Volání metod, které modifikují výstupní proměnné si též v Kódu 6.15 můžeme povšimnout.

V *CheckWord* se také nachází implementační drobnost, a to převod na malá písmena. Ten se provádí kvůli tomu, že slovník nerozlišuje velká a malá písmena, to znamená, že pacient by například při psaní jmen mohl být mylně hodnocen. Ovšem ukládání do výstupních parametrů je přesně v tvaru v jakém bylo slovo napsané, pokud by se tedy pacient rozhodl psát něco takového aHoJ, terapeut to z dílčího výstupu zjistí.



Obrázek 6.6: Doplnění slov - výsledná herní scéna



Obrázek 6.7: Doplnění slov - průběh hry


```

1 public void confirmWord() {
2     Text text = GameObject.Find("word").GetComponent<Text>();
3     string word = text.text;
4     bool result = InputParams.Rounds[round].checkWord(word);
5     Audio.GetComponent<AudioPlayer>().Play(result);
6
7     if (result) {
8         Results.addRightWord(word);
9         inputField.GetComponent<InputField>().text = "";
10
11         remaining.text = remainingText +
12             (InputParams.Rounds[round].NumberOfWords -
13             InputParams.Rounds[round].AnsweredWordsCount) +
14             wordText;
15         rightWords.text = correctWordsText + String.
16             Join(",", InputParams.Rounds[round].AnsweredWords);
17
18         if (InputParams.Rounds[round].AnsweredWordsCount ==
19             InputParams.Rounds[round].NumberOfWords) {
20             inputField.GetComponent<InputField>().placeholder
21                 .
22                 GetComponent<Text>().text = waitText;
23                 StartCoroutine(startNewRound());
24         }
25     }
26     else {
27         Results.addWrongWord(word);
28         inputField.GetComponent<InputField>().text = "";
29     }
30 }

```

Kód 6.12: confirmWord()

```

1 public bool checkWord(string word) {
2     word = word.ToLower();
3     if (Array.IndexOf(answeredWords, word) > -1) {
4         return false;
5     }
6
7     if (acceptableWords.Contains(word) == false) {
8         return false;
9     }
10    answeredWords[answeredWordsCount] = word;
11    answeredWordsCount++;
12    return true;
13 }

```

Kód 6.13: checkWord()

Dílčí a celkové výstupní parametry

Po odehrání všech kol nám zbývá jen získání výstupu pro terapeuta. Ani v této hře se implementace dílčích výstupních parametrů nebude markantně lišit. Opět využíváme dvě metody *addRightWord()* a *addWrongWord()*, které se chovají jako stejnojmenné metody v předešlé hře.

Také celkové výstupní parametry se nijak neliší.

Nápověda

Využívá existujícího seznamu akceptovaných slov pro dané kolo, vybere jedno slovo náhodně a zobrazí do textového objektu uprostřed obrazovky.

6.5 Slova, která začínají a končí na stejné písmeno

Jelikož tato hra je implementována na základě hry z předchozího bodu, uvedeme si zde jen malou rozlišnost, kterou v kódu najdeme.

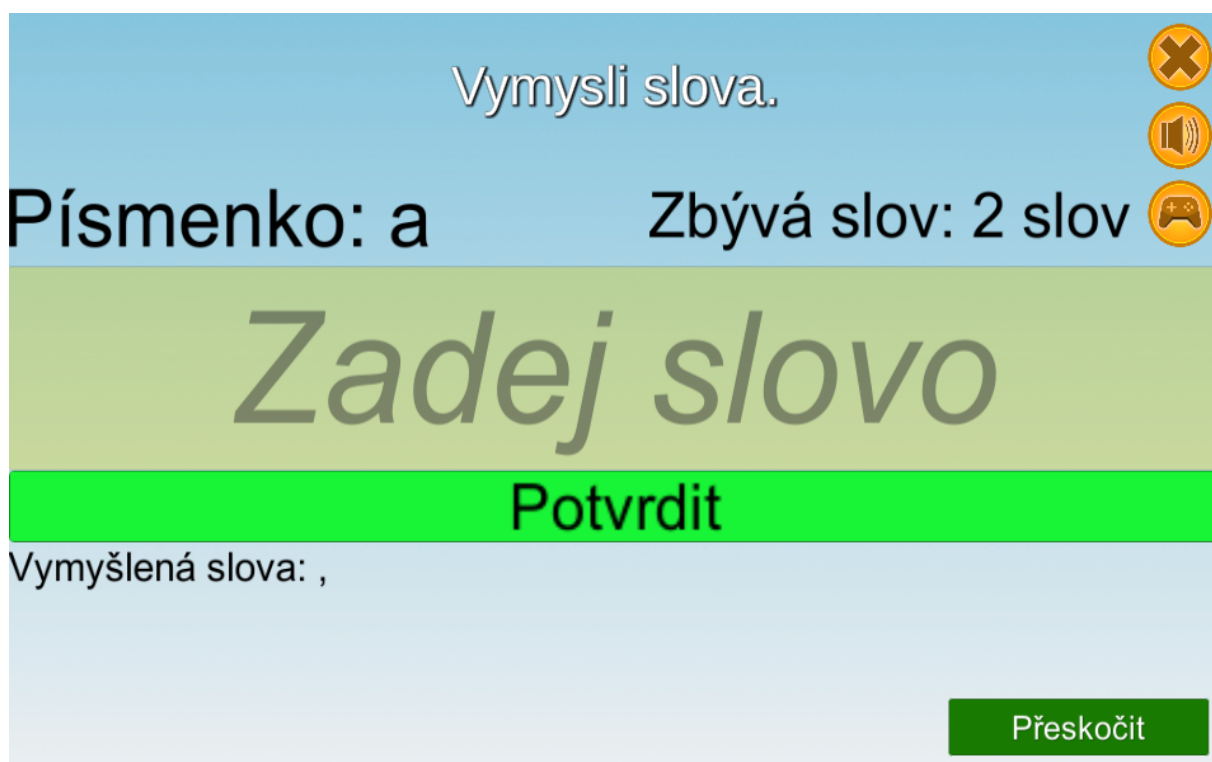
Změny oproti doplnění slov

Jediné změny, které bylo třeba provést v této hře, se týkaly tvorby kola. Nutné bylo změnit podmínku pro vytvoření listu akceptovaných slov.

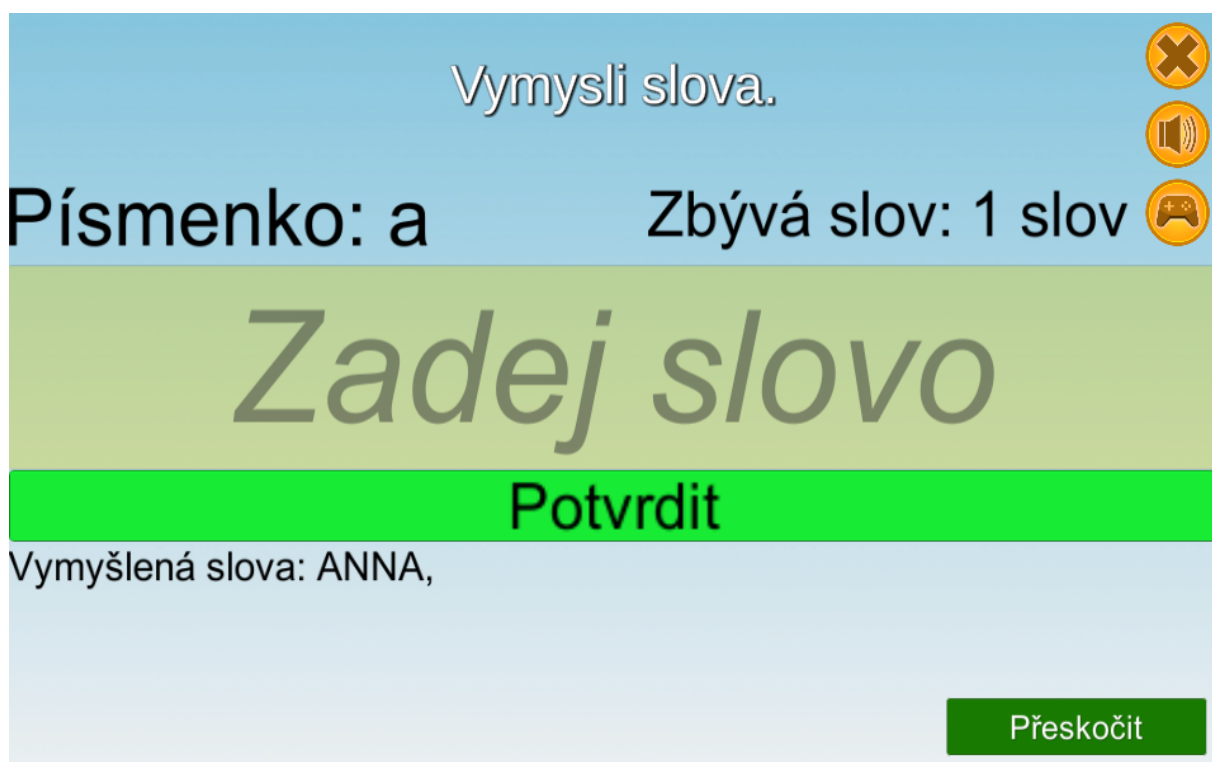
```
1 List<string> lines = new List<string>(txt.text.Split('\n'));
2     foreach (string line in lines) {
3         string word = Regex.Replace(line, @"\s+", "");
4         if (word.Length < 3) {
5             continue;
6         }
7
8         if (word[0] == this.letter[0] && word[word.Length -
9 1] == this.letter[0]) {
10             acceptableWords.Add(word);
11         }
```

Kód 6.14: Ukázka `getWordsFromVocab()`

Na závěr této hry se podíváme na to jak ve výsledku vypadá. Už na první pohled je nám ale zřejmý již zmíněný stejný implementační základ.



Obrázek 6.8: Stejně písmeno - začátek hry



Obrázek 6.9: Stejně písmeno - průběh hry

6.6 Slovník a jeho implementace

Textové soubory symbolizující slovníky jsou ve třech lokalizacích přikládány do složky *Resources* pro každou hru.

CZ Slovník

Český slovník byl vytvořen z dostupných internetových zdrojů. Byl použit speciální seznam podstatných jmen [21] a korpusy¹ současné psané češtiny [28]. Tyto seznamy byly pomocí programovacího jazyka Python sloučeny do jednoho textového souboru, zbaveny duplicitních slov a upraveny pro potřeby bakalářské práce.

EN Slovník

Anglický seznam slov [31] byl nalezen nelicencovaný a volně dostupný, nebylo potřeba ho nijak upravovat.

DE Slovník

Německý slovník [35] též nebylo třeba přetvořit a byl převzat v původní podobě.

Speciální přístup

V analýze byly zmíněny tři speciální slovníky, všechny byly vytvořeny a jsou součástí příslušných her. Pro připomenutí se jedná o hry skládání slov, vodorovné křížovky a slova, která začínají a končí na stejné písmenko.

6.7 Úspěšnost ve hře

Při nasazení her na web se zadává vzoreček, jak se vypočítá úspěšnost z výstupních parametrů. U žádných z her se mi nepodařilo vzorce na webu úspěšně zprovoznit, ovšem v kódu vždy fungovali. Zavedla jsem tedy nový výstupní parametr, který poté předám na místo očekávaného výpočtu. Výsledek získaný z parametru se interpretuje jako úspěšnost na konci dané hry.

Zde bych ráda uvedla vzorce jednotlivých her.

¹soubor zpracovaných jazykových dat

Skládání slov

$$success_rate = a/(a + b) \quad (6.1)$$

a = správné spojení b = špatné spojení

Vodorovné křížovky

$$success_rate = a/(a + b) \quad (6.2)$$

a = celkový počet kol b = přeskočený počet kol

Doplnění slov

$$success_rate = (a - b)/(c - b) \quad (6.3)$$

a = počet správných slov b = počet využitých nápověd c = počet všech slov

Slova, která začínají a končí na stejné písmeno

$$success_rate = (a - b)/(c - b) \quad (6.4)$$

a = počet správných slov b = počet využitých nápověd c = počet všech slov

7 Testování her

Aby se minimalizoval počet chyb v softwaru, je nutné ho podrobit testování. Já jsem testování rozdělila do dvou komplexních částí.

7.1 Testování na webu

Testování v nemocnici

Po nahrání testovacích verzí na web jsem se sešla s paní Herejkovou, kde jsme společně prošli všechny hry. Zjistili jsme, že někde chybí velká písmenka, drobné texty či je špatně zobrazované písmenko *ch*. Tyto chyby byly po schůzce ihned opraveny a následně začalo testování na zvolených subjektech.

Paní Herejková vybrala pacienta, kterému jsme hry přiřadily. Jelikož jsem s pacientem byla v kontaktu i osobně, mohla jsem si od něj vyslechnout připomínky, například jaké nápovědy by uvítal ve hře. Na základě jeho námětů a po diskuzi s paní Herejkovou jsem nápovědy do her implementovala. Původně byly hry bez nápověd.

Testování se scénáři

Ostatním subjektům, kamarádům ve věku 15-20 let, jsem připravila předem připravené scénáře. Při první hře jsem je nijak neseznámila s podrobnostmi a čekala jsem na to, zda intuitivně přijdou na to jak se hra má hrát. To mi ukázalo, že jsou hry přehledně vytvořeny a nápovědy ve hře jsou dostačující. Následně jsem předala scénář s bezchybným průchodem hrou, chybovým průchodem či průchod s využitím nápověd a přeskokování. Získala jsem tak cennou odezvu. Například se odhalilo nesprávné spojování ve hře Doplňování slov.

Testování parametrů

Poté jsem hry testovala osobně. Nastavovala jsem na webu různé vstupní parametry, ať už smyslné či nesmyslné a sledovala vyhození výjimek. Hry se chovaly dle očekávání.

7.2 Testování v Unity

Pro každou hru bylo vytvořeno několik jednotkových testů pomocí nástroje **Unity Test Runner**. Tyto testy sloužily ke kontrole nejdůležitějších metod pro hru. Šlo především o rozdělení vstupních parametrů či tvorbu herní scény.

7.3 Nalezené chyby

Při testování bylo nalezeno několik chyb, všechny byly opraveny a zaneseny do implementace.

Doplňování slov

- Nefungoval druhý typ spuštění podle čísel.
- Náповěda správně nezobrazovala písmenko ch.
- Šlo nepsrávně spojovat bubliny pokud se ve slově objevila stejná písmenka.

Vodorovné křížovky

- V jednom kole šlo použít vícekrát stejné slovo
- Špatně se ukládaly výstupní parametry.

Doplnění slov

- Špatný text, místo "Vyčkej na další kolo" bylo vypsáno "vyčkej na další"

8 Zhodnocení dosažených výsledků

Při zahájení práce na hrách jsem začala s tvorbou *Skládání slov*, které mělo jeden typ spuštění a to se zadanými slovy od terapeuta. Po setkání s paní Herejkovou proběhla větší předělávka na bázi jejího podnětu druhého typu spuštění. Jeden z prvních problémů představoval slovník, ze kterého se budou vybírat náhodná slova. Nakonec jsme po domluvě došli k závěru, že bude nejlepší udělat slovník vlastní a speciální pro tuto hru. V implementaci jsem řešila dilema jak rozpoznávat typy spuštění, tak abych nemusela přidávat vstupní parametry navíc. K řešení toho problému se nejvíce osvědčil postup, který je ve výsledku naimplementovaný, tudíž že na základě vstupu hra sama rozezná o jaký typ se jedná. V pozdější fázi testování se objevila nečekaná chyba ve spojování bublin, ta se nakonec podařila vyřešit zavedením nové proměnné *finished* ve skriptu. Plán práce pokračoval tvořením Vodorovných křížovek. Největší obtíží bylo zvolení správných vzorců pro vygenerování vhodně velké a umístěné křížovky. Také jsem musela přistoupit v kódu k nahrazení *ch* za jiný znak, aby se lépe prováděli jednotlivé metody. Poté jsem přešla ke zbývajícím dvou hrám. Zde jsem měla potíže s vhodným rozmístěním komponent, přehledností a s kontrolou slov.

Po vytvoření her se hry testovaly. Mezi kamarády, rodinou, ale i v nemocnici. Zde se ale dostávám k největší nevýhodě, jaké byla práce vystavena, a to že mezi pacienty se našel pouze jeden vhodný subjekt k testování. Do budoucna by hry určitě vyžadovaly větší počet testovacích subjektů v nemocničních zařízeních, aby se pacientům uzpůsobily co nejvíce na míru.

V průběhu práce a zjišťování problematiky vznikly i nápady na další hry, které by mohly být realizovány. Na slovní zásobu by se hodila například hra *Pyramida*, která by přijímala v parametru jedno písmenko a pacient by měl za úkol vymyslet slova v pyramidě. Na prvním místě by bylo slovo o délce tři, na posledním o délce deset.

Kromě nápadů na nové hry, jsem se zaručila k tomu, že pokud se i po odevzdání práce objeví ve hrách chyby, budou napraveny. Zároveň pokud bude potřeba zasáhnout do struktury jednotlivých her, budu se snažit co nejvíce vyhovět.

9 Závěr

V úvodu práce jsem se seznámila se skupinou výzkumníků *Neuroinformatics*, kteří stojí za zrodem projektu *BrainIn*. Následně jsem prozkoumala webovou stránku projektu a jednotlivé možnosti, které nabízí. V systému se skrývalo několik nových pojmů, se kterými bylo potřeba se obeznámit. Především bylo nutné se seznámit se Šablonou šablon či funkčností vstupních a výstupních parametrů. Poté mě zajímalo zda se na trhu vyskytují podobné služby, a tak jsem prozkoumala komerční systémy. Na základě průzkumu jsem si dovolila vytvořit i malé srovnání. BrainIn je mezi systémy výjimkou především parametrizací, která dovoluje mnoho možností jednotlivých her.

Abych mohla přejít k analýze her a vytvořila si ucelenou představu vhodných her, bylo zapotřebí poznat funkci mozku a problematiku kognitivních funkcí. Tato témata jsou velice obsáhlá a já se snažila v nich co nejlépe zorientovat z dostupné literatury. Na základě těchto informací jsem se přesměrovala k jednotlivým diagnostikám a důvodům proč je vhodná pro pacienty neurorehabilitace. Získala jsem znalosti, jaký typ pacientů bude hrát mé hry a co je v daných hrách klíčové.

V průběhu studování problematiky proběhla schůzka s paní Herejkovou z nemocnice FN Plzeň, která mi sdělila svojí vizi. Na základě jejího podnětu jsem vymyslela čtyři hry, které by mohly splňovat její očekávání. Také jsem navrhla vhodné vstupní a výstupní parametry tak, aby je ona či jiný terapeut mohly přizpůsobit co nejlépe pacientovi.

Hry jsem naimplementovala a začlenila do webové aplikace. Následovalo testování dvojího typu. Jak jsem již zmínila v minulé kapitole, zde byla největším problémem nedostupnost vhodných testovacích subjektů.

Ovšem i přesto se podařilo z her odstranit několik chyb a do webové aplikace připravit šablony, které co nejvíce odpovídají zadání. Na závěr bych jen ráda dodala, že doufám v úspěchy projektu BrainIn a snad se v budoucnu s tímto skvělým projektem opět setkáme.

Literatura

- [1] Afa slovník, . Dostupné z: <https://afaslovník.cz/>.
- [2] *Afázie* [online]. Asociace klinických logopedů České republiky, . [cit. 2022/01/11]. Dostupné z: <https://www.klinickalogopedie.cz/index.php?pg=verejnost--co-je-to--afazie>.
- [3] *Agrafie (agrafik, agrafický)* [online]. Psychoweb. [cit. 2022/01/13]. Dostupné z: <https://www.psychoweb.cz/slovník/agrafie-agrafik-agraficky-co-to-je-vyznam-podle-slovníku/>.
- [4] *Agrafie (agrafik, agrafický)* [online]. WikiSkripta. [cit. 2022/01/13]. Dostupné z: <https://www.wikiskripta.eu/w/Alexie>.
- [5] *Neurodegenerativní onemocnění* [online]. WikiSkripta. [cit. 2022/01/11]. Dostupné z: https://www.wikiskripta.eu/w/Neurodegenerativn%C3%AD_onemocn%C4%9Bn%C3%AD.
- [6] *About Elevate Labs* [online]. Elevate. [cit. 2022/01/14]. Dostupné z: <https://elevateapp.com/about>.
- [7] [online]. Happy neuron, . [cit. 2022/01/14]. Dostupné z: <https://www.happy-neuron.com/>.
- [8] [online]. Happy neuron, . [cit. 2022/01/14]. Dostupné z: <https://www.happy-neuron.com/help?question=26>.
- [9] *Lumos Labs* [online]. Lumosity. [cit. 2022/01/14]. Dostupné z: <https://www.lumosity.com/en/resources>.
- [10] Paměť a její druhy, 2007. Dostupné z: <https://nakladatelstvi.portal.cz/nakladatelstvi/aktuality/81277>.
- [11] *O nás* [online]. Medical informatics. [cit. 2022/01/14]. Dostupné z: <https://mi.kiv.zcu.cz/cz/o-nas/>.
- [12] *About us* [online]. Neuroinformatics. [cit. 2022/01/14]. Dostupné z: <http://neuroinformatics.kiv.zcu.cz/menus/about-us>.
- [13] Parkinsonova choroba — onemocnění známé od starověku, jehož příčina dosud nebyla odhalena. Dostupné z: https://www.neztratitsevestari.cz/clanky/parkinsonova-choroba?gclid=Cj0KCQiAuP-OBhDqARIsAD4XHpcptnrTYz0ZgTTDTTLp71Zuv-Q1p-39-79PfdG2Utf_KUplnMqp0bYaAm0YEALw_wcB.

- [14] Vlohy, schopnosti, dovednosti, 2021. Dostupné z:
https://moodle.czu.cz/scorms/EPE71E/PS003/resources/12.Vlohy_schopnosti_dovednosti.htm.
- [15] Pojem sémantická paměť. Dostupné z: https://wikisofia.cz/wiki/Pozornost,_jej%C3%AD_druhy_a_vlastnosti.
- [16] *Využití moderních informačních technologií pro neurorehabilitace pacientů se získaným poškozením mozku?* [online]. Neuroinformatics, . [cit. 2022/01/14]. Dostupné z: shorturl.at/gwTW2.
- [17] Prostorová orientace, . Dostupné z:
<https://www.tablexia.cz/cs/encyklopedie/13/>.
- [18] Co je to prostorová paměť?, . Dostupné z:
<https://www.netinbag.com/cs/health/what-is-spatial-memory.html>.
- [19] Rychlost. Dostupné z: <https://www.mentem.cz/blog/rychlost/>.
- [20] *Schizofrenie – příčina, příznaky a léčba* [online]. EUC klinika. [cit. 2022/01/13]. Dostupné z: <https://euc.cz/clanky-a-novinky/clanky/schizofrenie-pricina-priznaky-a-lecba/>.
- [21] *Česká podstatná jména* [online]. [cit. 2021/09/01]. Dostupné z:
<https://pastebin.com/7uPetKWh>.
- [22] Verbální dovednosti. Dostupné z:
<https://www.mentem.cz/blog/mysleni/>.
- [23] *Voronoi Diagram* [online]. Wolfram, . [cit. 2022/05/05]. Dostupné z:
<https://mathworld.wolfram.com/VoronoiDiagram.html>.
- [24] *Voronoi Diagram* [online]. Balu Ertl, . [cit. 2022/05/05]. Dostupné z:
https://en.wikipedia.org/wiki/Voronoi_diagram#/media/File:Euclidean_Voronoi_diagram.svg.
- [25] ARTSYBEE. *Brain-lobes-neurology-human-body* [online]. Pixabay. [cit. 2022/01/01]. [Obrázek upraven pro potřeby BP.]. Dostupné z:
<https://pixabay.com/illustrations/brain-lobes-neurology-human-body-1007686/>.
- [26] COMMONS, W. *File:Mozek obratlovců – oblasti.png* — *Wikimedia Commons, the free media repository* [online]. Wikipedie, 2020. [cit. 2022/01/01]. [Obrázek upraven pro potřeby BP.]. Dostupné z:
https://commons.wikimedia.org/w/index.php?title=File:Mozek_obratlovc%C5%AF_%E2%80%93_oblasti.png&oldid=438342340.

- [27] DURKÁČOVÁ, M. K. *Exekutivní funkce* [online]. Memtem. [cit. 2022/01/13]. Dostupné z: <https://www.mentem.cz/blog/exekutivni-funkce/>.
- [28] KŘEN, M. seznamy:srovnacni seznamy Příručka ČNK, 2016. Dostupné z: https://wiki.korpus.cz/doku.php/seznamy:srovnacni_seznamy. [Online; accessed 14-September-2021; licence CC BY 4.0, ze souborů SYN2010 a SYN2015 (slovní tvary + lemmata) byla vyjmuta jednotlivá slova a byla využita pro tvorbu vlastního slovníku].
- [29] LUMOSITY. *Lumosity's Human Cognition Project* [online]. Lumosity, 2021. [cit. 2022/01/14]. Dostupné z: <https://www.lumosity.com/hcp>.
- [30] POULEFRIT. *csDelaunay* [online]. [cit. 2022/06/20]. Dostupné z: <https://github.com/PouletFrit/csDelaunay>.
- [31] PRICE, E. Wordlist 10000. Dostupné z: <https://www.mit.edu/~ecprice/wordlist.10000>. [Online; accessed 14-September-2021].
- [32] KOHOUTEK, C. Pojem sémantická paměť. Dostupné z: <https://slovník-cizich-slov.abz.cz/web.php/slovo/semanticka-pamet>.
- [33] SAMKOVÁ, K. Voroneho diagramy a jejich aplikace. PDF download.
- [34] SAND, R. *Split brain syndrome* [online]. Britannica, 2019. [cit. 2022/01/01]. Dostupné z: <https://www.britannica.com/science/split-brain-syndrome>.
- [35] SCHREIBER, J. German Dictionary. Dostupné z: <https://sourceforge.net/projects/germandict/>. [Online; accessed 12-June-2022].
- [36] TONYA HINES, C. *Anatomy of the Brain* [online]. Mayfield Clinic, 2021. [cit. 2022/01/01]. Dostupné z: <https://mayfieldclinic.com/pe-anatbrain.htm>.
- [37] WHITEMAN, H. *Five of the best apps to train your brain* [online]. Medical News Today, 2017. [cit. 2022/01/14]. Dostupné z: <https://www.medicalnewstoday.com/articles/316684#Lumosity:-Colorful-and-fun>.
- [38] WIKIPEDIE. Paměť — Wikipedie: Otevřená encyklopedie, 2021. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Pam%C4%9B%C5%A5&oldid=20135678>.

Seznam obrázků

2.1	Uvítací obrazovka webové aplikace BrainIn	9
2.2	Ukázky her Orion a Hledání znaků	10
2.3	Ukázka vstupních parametrů šablony	12
2.4	Vytvoření hry Orion	13
2.5	Hry od HAPPYneuron [7]	14
2.6	Hry od Lumosity [29]	15
2.7	Hry od Elevate [6]	16
2.8	Ukázka AfaSlovníku [1]	17
3.1	Makroskopická anatomie mozku [26]	19
3.2	Laloky mozkové kůry [25]	20
5.1	Skládání slov - ukázka vstup	27
5.2	Skládání slov - ukázka výstup	27
5.3	20 bodů a jejich buňky dle Voroného diagramu [24]	29
5.4	Vodorovné křížovky - ukázka vstup	30
5.5	Vodorovné křížovky - ukázka výstup	30
6.1	Objekt Bublina	36
6.2	Skládání slov - výsledná hra na vstupu, včetně nápovědy	39
6.3	Skládání slov - výsledná hra, výstup slovo chalupa	40
6.4	Křížovky - výsledná hra na vstupu	42
6.5	Křížovky - průběh hry, zápis správného slova	44
6.6	Doplnění slov - výsledná herní scéna	47
6.7	Doplnění slov - průběh hry	47
6.8	Stejně písmeno - začátek hry	50
6.9	Stejně písmeno - průběh hry	50

Uživatelská příručka

Sestavení hry v operačním systému Windows

- Přes **Unity Hub** stáhneme a nainstalujeme **Unity ve verzi 2018.4.7f1**.
- Klikneme na v záložce **Projects** na tlačítko **Open**, následně zvolíme **Add project from disk**. Přidáme cestu ke kořenové složce požadované hry.
- Po otevření projektu klikneme na **File, Build Settings**.
- Platformu volíme **WebGL** a klikneme na **Build**.

Nastavení ve webové aplikaci

- V internetovém prohlížeči přejdeme na adresu **https://brainin.kiv.zcu.cz/**. Přihlásíme se, či požádáme o registraci a přidělení práv na adrese `neuro@ntis.zcu.cz`.
- Přejdeme do sekce **Šablony** -> **Správa šablon** -> **Vytvoření šablony**. Zvolíme název a klikneme na **Vytvořit**.
- V popisu zvolíme **Změnit, Vytvořit překlad** a vytvoříme popis naší hry. Přiřadíme klíčová slova a kategorii.
- Ve **Vstup do šablony** a **Výstup do šablony** navolíme veškeré parametry, které jsou součástí naší hry. Opět vytvoříme jejich popis.
- Poté přejdeme do **Správa souborů**. Zde nahrajeme veškeré složky naší hry. V **Zdroje k šabloně** je nutné dodržet adresářovou strukturu *Build, StreamingAssets* -> *ProgramData, TemplateData*.
- Klikneme na **Uložit**.

Spuštění ve webové aplikaci

Abychom mohli hru testovat potřebujeme parametrizovat námi vytvořenou šablonu.

- Přejdeme do sekce **Úlohy** -> **Vytvořit úlohu**. Napíšeme název a zvolíme šablonu ze které hra vychází.

- Vytvoříme popis hry.
- V obsahu nastavíme jednotlivé parametry.
- Klikneme na **Otestovat úlohu**.
- Nyní máme hru připravenou na přiřazení pacientovi.

Obsah odevzdaného souboru

Veškeré soubory spjaté s prací byly odevzdány v souboru A19B0008P_prilohy.zip. Soubory obsahuje následující položky:

- Adresář **Text_prace** obsahuje soubor .tex s tímto textem + PDF soubor.
- Adresář **Aplikace_a_knihovny** obsahuje veškeré zdrojové soubory ke hrám, Delaunay knihovnu, uživatelskou příručku.
- Dále najdeme v .zip textový soubor s popisem adresářové struktury **README.txt**.