

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

Bakalářská práce

**Vývoj parametrizovaných her v
Unity pro pacienty trpící poruchou
pozornosti s hyperaktivitou**

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan BARTOŠEK**
Osobní číslo: **A18B0170P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Téma práce: **Vývoj parametrizovaných her v Unity pro pacienty trpící poruchou pozornosti s hyperaktivitou**
Zadávací katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s webovou aplikací BrainIn v neuroinformatické laboratoři na KIV.
2. Prostudujte problematiku kognitivních her pro pacienty s ADHD a navrhňte vhodné vstupní a výstupní parametry.
3. Dle požadavků lékařů navrhňte 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**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

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

Datum zadání bakalářské práce: **5. října 2020**
Termín odevzdání bakalářské práce: **6. května 2021**

L.S.

Doc. Dr. Ing. Vlasta Radová
děkanka

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

Poděkování

Touto cestou bych chtěl poděkovat svému vedoucímu bakalářské práce Ing. Petru Brůhovi za odborné vedení a cenné rady. Dále bych chtěl poděkovat PhDr. Karolíně Malé za odbornou pomoc v oblasti psychoterapie a Nikole Samcové za pomoc při vytváření grafických prvků.

Prohlášení

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

V Plzni dne 2. května 2021

Jan Bartošek

Abstract

This bachelor thesis aims to create games that could be used in the treatment of patients suffering from attention deficit hyperactivity disorder. The theoretical part of the thesis briefly describes the theory of cognitive functions and ADHD. The theoretical part also examines existing cognitive games and designs 4 games with potential parameters affecting the game. The practical part describes the implementation of 4 games that could help patients in their treatment.

Abstrakt

Tato bakalářská práce cílí na vytvoření her, které by mohly být využívány při léčbě pacientů trpících poruchou pozornosti s hyperaktivitou. V teoretické části práce je krátce popsána teorie kognitivních funkcí a ADHD. Následně jsou prozkoumány již existující kognitivní hry a jsou navrženy 4 hry včetně parametrů ovlivňujících jejich průběh. V praktické části je popsána implementace 4 her, které napomáhají pacientům při jejich léčbě.

Obsah

1	Úvod	9
2	Webová aplikace BrainIn	10
2.1	Prostředí aplikace	10
2.1.1	Uživatelé	10
2.1.2	Šablona	11
2.1.3	Úloha	12
2.1.4	Balík	12
2.1.5	Analýza	12
2.1.6	Překlad	13
3	Teorie kognitivních funkcí	14
3.1	Kognitivní funkce	14
3.1.1	Paměť	14
3.1.2	Pozornost	15
3.1.3	Exekutivní funkce	17
3.1.4	Řeč	18
3.1.5	Zrak a prostor	18
3.2	ADHD	19
3.2.1	Symptomy	19
3.2.2	Příčiny vzniku	20
3.3	Kognitivní trénink	21
3.3.1	Komerční systémy	21
3.3.2	Kognitivní hry	23
4	Návrh her	24
4.1	Návrh parametrů	24
4.1.1	Vstupní parametry	24
4.1.2	Výstupní parametry	24
4.1.3	Psychiatrická nemocnice v Dobřanech	25
4.2	Systém odměn	25
4.2.1	Strom	25
4.2.2	Věž	26
4.2.3	Měšec	26
4.3	Hry	27
4.3.1	Harvest	27

4.3.2	TestTubes	29
4.3.3	Calc	31
4.3.4	Grid	32
4.3.5	Společné parametry	34
5	Specifika implementace	35
5.1	Vytváření šablony	36
5.2	Herní engine Unity	36
5.2.1	Prvky vývojového prostředí	36
5.2.2	Průběh hry	37
5.2.3	Šablona šablon	37
6	Implementace	38
6.1	Vstupní parametry	38
6.2	Generování hry	38
6.2.1	Seed	38
6.2.2	Harvest	39
6.2.3	TestTubes	40
6.2.4	Calc	40
6.2.5	Grid	41
6.3	Herní průběh	42
6.3.1	Harvest	42
6.3.2	TestTubes	43
6.3.3	Calc	43
6.3.4	Grid	44
6.4	Grafické prvky	44
6.4.1	Způsob grafické reprezentace	44
6.4.2	Harvest	45
6.4.3	TestTubes	45
6.4.4	Odměny	46
6.4.5	Přístup ke grafickým prvkům	47
6.5	System odměn	48
6.6	Tutoriál	49
6.7	Lokalizace	49
6.8	JavaScriptové rozhraní	49
6.9	Nápověda	49
7	Testování	50
7.1	Jednotkové testování	50
7.1.1	Odhalené chyby	50

7.2	Beta testování	50
7.2.1	Odhalené chyby	51
7.2.2	Rychlost načítání hry	52
7.3	Testování v PN v Dobřanech	52
8	Zhodnocení dosažených výsledků	53
9	Závěr	55
	Literatura	56
10	Uživatelská dokumentace	61
10.1	Funkce tlačítek	61
10.2	Ukázka her	62
11	Návod k sestavení	64

1 Úvod

Neurologické rehabilitace, které obvykle probíhají v různých neurologických centrech nebo psychiatrických léčebnách, mohou být z části nahrazeny domácí léčbou bez nutnosti přítomnosti terapeuta nebo zdravotní sestry. Tato skutečnost dává pacientům možnost léčit se každý den, a to i z pohodlí domova. Cílem této práce je vytvořit čtyři hry, které svým obsahem napomohou při různých neurologických terapiích. Hry budou navrženy tak, aby pomáhaly hráčům s trénováním mozku a kognitivních funkcí. Každý pacient trpí jinými, různě vážnými problémy, proto bude kladen důraz na vytvoření možnosti konfigurace průběhu jednotlivých her, aby mohly být upraveny subjektivně pro každého pacienta. Tyto úpravy by měly nejen správně ovlivňovat náročnost her, ale i zvyšovat míru zábavy. Výsledek a průběh hry by měl být vhodně interpretován terapeutovi nebo příslušné zdravotní sestře.

Hlavními pozitivy při zpracování tohoto tématu shledávám nejen v potencionální pomoci lidem s různými neurologickými potížemi, ale i v nahlédnutí do různých odvětví medicínské informatiky a prostudování nových technologií.

Bakalářská práce je rozdělena na teoretickou a praktickou část. Teoretická část se zabývá teorií kognitivních funkcí mozku a blíže popisuje poruchu pozornosti s hyperaktivitou. Dále obsahuje část věnovanou kognitivním hrám, návrh vstupních a výstupních parametrů her a návrh jednotlivých her. Praktická část je věnována implementaci všech čtyř her, detailnějšímu popisu vývoje a testování.

2 Webová aplikace BrainIn

BrainIn je interaktivní webová aplikace sloužící pro neurorehabilitaci pacientům, kteří prošli závažnými neurologickými problémy a napomáhá jim při návratu do běžného života. V této webové aplikaci, která je vyvíjená na KIV, mohou být vytvářeny úlohy pro každého pacienta na základě jeho potřeb, zájmů a neurologických onemocnění. Konkrétní úlohy svým pacientům vytváří terapeut a pacienti mohou rehabilitovat bez nutnosti terapeutovy přítomnosti. Každá z her produkuje informace o pacientově snažení, které si terapeut může prohlédnout a zhodnotit tak pacientův pokrok. [18]

2.1 Prostředí aplikace

Webová aplikace je rozdělena do několika částí.

2.1.1 Uživatelé

Každý registrovaný uživatel disponuje konkrétní rolí. Uživatelům je možné přiřadit celkem 6 rolí, které upravují jejich možnosti a chování ve webové aplikaci. Specifikace konkrétních rolí jsou vypsány níže:

- **Super-Administrátor** je role určená především správcům a vývojářům systému. Super-Administrátor vlastní práva všech ostatních rolí, zajišťuje nastavení prostředí aplikace.
- **Administrátor** se stará o chod aplikace jako takové. Vytváří nové hry, spravuje uživatele, přiřazuje pacienty terapeutům apod.
- **Terapeut** je uživatelská role připravená pro terapeuty a rehabilitační pracovníky. Každému terapeutovi je možné přiřadit pacienty, které poté může spravovat a přiřazovat jim různé úlohy.
- **Super-Terapeut** je mocnější verzí terapeuta. Tato uživatelská role disponuje možností vytvářet vlastní úlohy. V praxi může reprezentovat např. nadřízeného, který vytvoří úlohy, a jeho podřízený terapeuté je využívají.
- **Pacient** má sice omezenou funkcionalitu, ale z hlediska charakteru aplikace je vlastně nejdůležitější rolí. Přihlašuje se do aplikace a zde

hraje hry, které mu byly přiděleny. Po odehrání může slovně a bodově tyto hry ohodnotit.

- **Nepřiřazeno** je role, která je přiřazena každému uživateli po registraci. Po validaci registrovaného uživatele administrátorem mu typicky bývá přidělena příslušná role.

2.1.2 Šablona

Šablona je sekce, ve které je možné přidávat do systému zcela nové hry. Každá šablona má přidělený svůj vlastní paměťový prostor, na který se hra nahraje a odkud je později také spouštěna. Při vytváření nové šablony je nutné znát přesný formát a pořadí vstupních a výstupních parametrů dané hry (viz obr. 2.1). Podle formátu a pořadí parametrů jsou vytvořeny např. textové vstupy, do kterých se v pozdější fázi doplňují vstupní parametry, jež ovlivňují průběh, cíl nebo vzhled hry. Po dokončení hry jsou vyprodukované statistiky a výstupní parametry zpracovány právě na základě formátu výstupních parametrů ze šablony. Všechny hry začleněné do tohoto systému obsahují několik stejných parametrů. Těmto parametrům jsou věnovány sekce 4.1.1 a 4.1.2.

BrainIn Uživatelé ▾ Překlady ▾ Šablony ▾ Úlohy ▾ Balíky ▾ Analýzy ▾ Účet Správce! ▾

Náhled šablony (207) Seznam šablon

Popis šablony Vstup do šablony Výstup z šablony Správa souborů Výpočty

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	
Trvání semaforu	Koeficient (reálné číslo, zadané s desetinn...	Číslo	
Zobrazení správného řešení	Jestli se má zobrazit správné řešení.	Logická hodnota	
Doba zobrazení řešení	Doba v sekundách, po kterou bude zobraz...	Číslo	
Doba zobrazení hodnocení	Doba v sekundách, po kterou bude zobraz...	Číslo	

Uložit

BrainIn, Copyright © 2020, 1.1.0.18148

Obrázek 2.1: Ukázka sekce „šablony“ ve webové aplikaci BrainIn.

2.1.3 Úloha

Po vytvoření šablony je možné vytvořit konkrétní úlohu dané hry (viz obr. 2.2). Do vytvořených parametrů jsou zadávány takové hodnoty, které danou hru ovlivňují. Konkrétní smysl parametrizace spočívá v tom, že terapeut může vytvořit úlohu, jež bude upravená subjektivně pro každého pacienta.

The screenshot shows the 'Úprava úlohy (592)' page in the BrainIn application. The page has a dark header with navigation links: 'BrainIn', 'Uživatelé', 'Překlady', 'Šablony', 'Úlohy', 'Balíky', 'Analýzy', and 'Účet Správce!'. Below the header, there are two buttons: 'Seznam úloh' (green) and 'Otestovat úlohu' (blue). The main content area has three tabs: 'Detail' (selected), 'Obsah', and 'Správa souborů'. The 'Detail' tab contains a form with the following fields:

Debug	Ne
Maximální doba	25
Počet kol	5
Nápověda	0
Přeskočit kolo	Ne
Přeskočit zbylá kola	Ne
Trvání semaforu	1
Zobrazení správného řešení	Ne
Doba zobrazení řešení	0
Doba zobrazení hodnocení	0

At the bottom of the form is a blue 'Uložit' button.

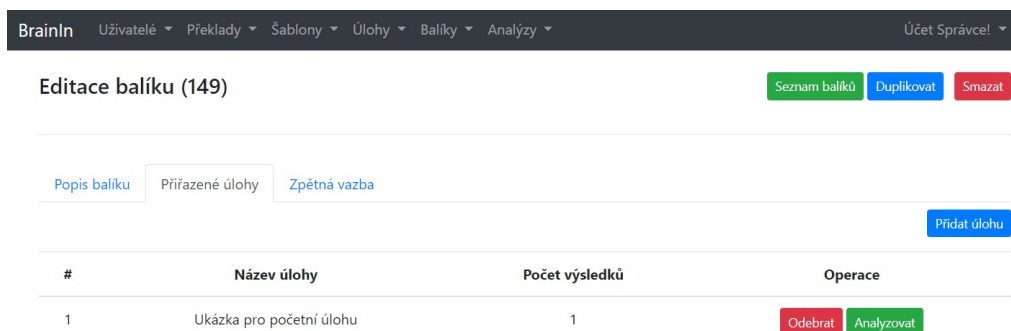
Obrázek 2.2: Ukázka vytváření úlohy ve webové aplikaci BrainIn.

2.1.4 Balík

Vytvořené úlohy mohou být přidány do tzv. balíků. Tyto balíky obsahují sekvenci úloh, které jsou postupně spouštěny. Terapeut obvykle vytvoří nějaké úlohy, přidá je do balíku a ten poté přidělí svému pacientovi.

2.1.5 Analýza

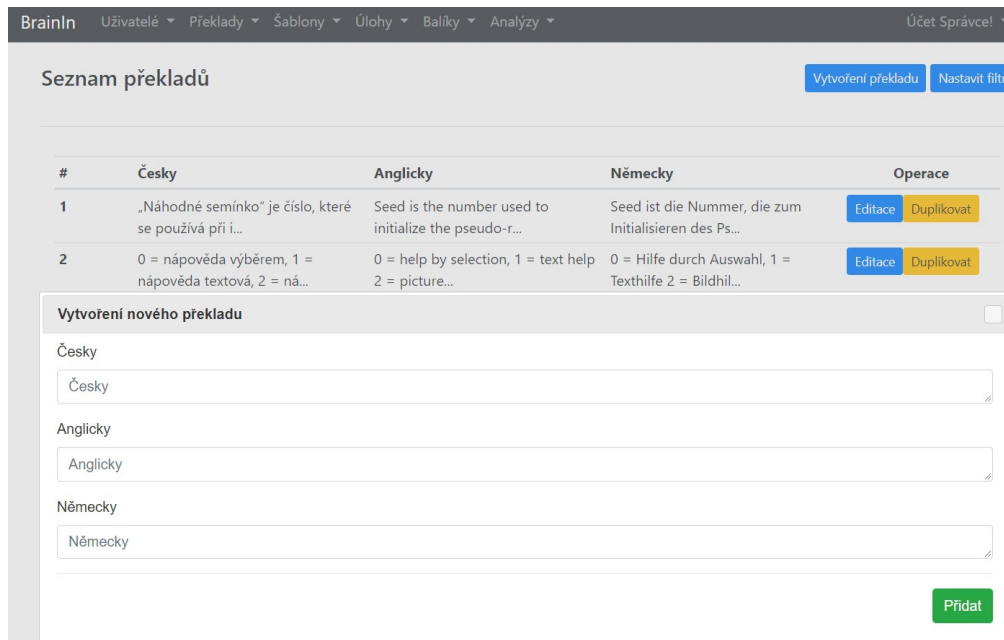
V této sekci vidí terapeut všechny statistiky odehraných her přidělených pacientům. Většinou se tyto statistiky týkají odehraného času, dokončení úlohy či procentuální úspěšnosti.



Obrázek 2.3: Ukázka balíku a jemu přiřazených úloh ve webové aplikaci BrainIn.

2.1.6 Překlad

Při vývoji této aplikace byl kladen důraz na multijazyčnost (viz obr. 2.4), proto je možné celou aplikaci prezentovat ve třech jazycích – češtině, angličtině a němčině. Tato skutečnost se promítá i do konkrétních her, které jsou prezentovány právě zvoleným jazykem. Každá hra je tedy obohacena o soubor přeložených textů, jež jsou na základě zvoleného jazyka do hry načteny a zobrazovány.



Obrázek 2.4: Ukázka seznamu překladů a vytváření překladu ve webové aplikaci BrainIn.

3 Teorie kognitivních funkcí

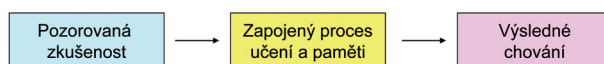
3.1 Kognitivní funkce

Kognitivní, nebo také poznávací funkce jsou vlastně psychické procesy odehrávající se v nervové soustavě, díky nimž člověk dokáže rozpoznávat, pamatovat si, učit se a nebo se přizpůsobovat měnícím se podmínkám okolního prostředí. Do kognitivních funkcí spadá koncentrace, pozornost, paměť, rychlost myšlení a nebo také porozumění informacím. Speciálním typem poznávacích funkcí jsou funkce exekutivní (vyšší). Ty reprezentují schopnosti řešení problémů, plánování, organizování apod. [17][24]

Tyto funkce jsou umístěny v různých částech mozku a jeho poranění tedy může poškodit všechny nebo jen některé z nich. Při poškození kterékoliv poznávací funkce dochází ke ztížení každodenního života. Poruchy kognitivních funkcí (kognitivní deficit) jsou velmi časté, ale z počátku nebývají zcela nápadné. Příčinou vzniku kognitivního deficitu může být traumatické poškození mozku, centrální mozková příhoda, infekční nebo toxické poškození mozku, neurodegenerativní onemocnění, ale i psychiatrické onemocnění (jako například deprese). [24][1]

3.1.1 Paměť

Prostřednictvím paměti člověk přijímá, uchovává si a následně si vybavuje nové informace, vjemy, vzpomínky, zážitky. Paměť dává člověku možnost vědět, znát a pamatovat si. Paměťový proces se dá rozdělit do tří fází – vstřípení, uchování a vybavení. Každý jedinec se různé věci učí a pamatuje jinak. Schéma vztahu mezi učením a pamětí je však u všech stejné (obrázek 3.1). [7][6]



Obrázek 3.1: Komponenty učení a paměti. [6]

V minulosti převládal názor, že pro různé typy informací (např. motorické, verbální, sensorické) existuje pouze jeden druh paměti. V současnosti se vědci shodují, že těchto kategorií je více. Ty se dělí podle způsobu ukládání informací, podle způsobu jejich zpracování a podle délky uchování. Podle

délky uchování paměťových stop bývá paměť rozdělována na krátkodobou a dlouhodobou. [7]

Krátkodobá paměť

Součástí krátkodobé paměti je senzorická (ultrakrátká) paměť, která přijímá informace přenesené smysly. Krátkodobá paměť působí, jak je patrné už z jejího pojmenování, jen krátkou dobu. Slouží k uchování informací pouze po dobu několika sekund. Většina psychických procesů se uskutečňuje právě v této paměťové části. Pro vytvoření paměťové stopy a zabránění ztráty informací je třeba získané informace dostatečně opakovat (viz obrázek 3.2). Jednou z variant kognitivní paměti je tzv. pracovní paměť, která vzniká opakováním informací, jež jsou využívány k vyřešení nějakého aktuálního úkolu. Pracovní paměť postupně přechází v paměť dlouhodobější. Přenos dat do dlouhodobější paměti však není stoprocentní. Úspěšné dlouhodobé zapamatování ovlivňuje několik faktorů, jako jsou míra opakování, významnost a nebo porozumění informacím. [24][7][6]

Dlouhodobá paměť

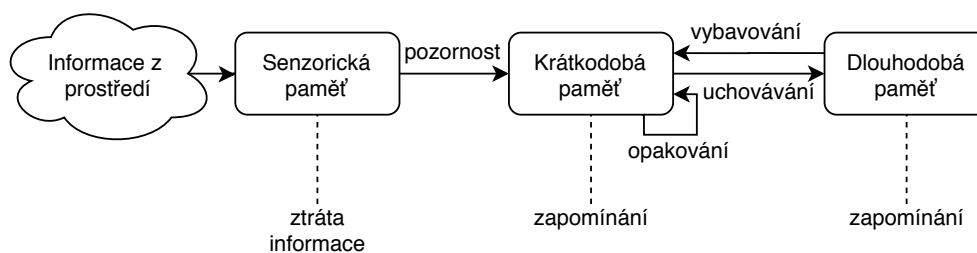
Dlouhodobá paměť ukládá významné informace a poznatky. Informace a data do ní lze zaznamenat opakováním, ale i mimovolně. Délka zápisu trvá od několika minut až po dobu mnoha let a čím je starší, tím je rezistentnější k zapomínání (je málo zranitelná). [24][6]

Implicitní paměť (někdy také nedeklarativní) je jednou z druhů dlouhodobé paměti. Dovoluje člověku osvojovat si dovednosti a zvyky i bez jeho uvědomění. K využívání takových znalostí není třeba si cokoli vybavovat, proto se do ní řadí jízda na kole, schopnost čtení, psaní apod. [6][7]

Explicitní paměť (deklarativní) umožňuje uchovávat fakta, informace z minulosti nebo osobní vzpomínky. Člověk si na tyto informace vědomě snaží vzpomenout. Explicitní učení bývá označováno jako to, při kterém si člověk uvědomuje, že se učí. Deklarativní paměť může být ještě rozdělena do dvou podtypů - sémantická a epizodická paměť. V případě **sémantické** paměti hovoříme o vybavování si obecných informací nebo obecně platných faktů. V případě vybavení si osobních zážitků nebo vzpomínek se jedná o paměť **epizodickou**. [6][7]

3.1.2 Pozornost

Pozornost je psychický stav jedince, který se projevuje zaměřeností a soustředěností vědomí. Souvisí s dalšími mentálními funkcemi, např. vnímáním,



Obrázek 3.2: Atkinson-Shiffrinův paměťový model [12]

paměti či jednáním. Je velmi důležitá pro zpracování aktuálních informací, jejich uchování a posléze i vybavení při určitém dění. Hlavní úlohou pozornosti je dostat vybrané informace do vědomí a napomáhat podprahovému zachycení některých podnětů. Pozornost může být upoutávána nejen podněty z okolí, ale může být vázána i na vlastní myšlenky, pocity a vzpomínky. Činnost pozornosti lze shrnout do tří hlavních funkcí – aktivace, selektivní zaměření na určitou oblast, regulace zaměření dle potřeby. [22][15][4]

Koncentrace

Koncentrace pozornosti je intenzita soustředění na daný podnět nebo aktivitu. Míra koncentrace pozornosti se liší s ohledem na právě vykonávanou činnost nebo aktuální situaci. Náročné jsou především aktivity nové a činnosti dosud nezvládnuté, protože se člověk musí na dílčí akce více soustředit ve větší míře. Při osvojení jednotlivých dovedností jim již nemusí věnovat tolik pozornosti. Zautomatizované činnosti probíhají většinou nevědomě a „přebytečná“ pozornost může být využita na důležitější aktivity. Například programátor znalý programovacího jazyka, se kterým zrovna pracuje, se nemusí soustředit na syntax a jazykové konstrukce daného jazyka, ale může se zaměřit na obsah a konkrétní algoritmy. Náročnost koncentrace pozornosti na různé podněty je rozdílná. Na vizuální informace se člověk může dívat neomezeně a může se lépe soustředit na různé aspekty, kdežto na zvukové podněty (např. výklad učitele) se musí soustředit v danou chvíli a v případě nezachycení některé z informací se k ní nemůže vrátit. [22]

Selektivita

Výběr zaměření na určité podněty a potlačení tendence vnímat i jiné informace je důležitým aspektem pozornosti. Je téměř nemožné vnímat vše, co se děje kolem nás, uvažovat o více problémech nebo zaobírat se několika činnostmi najednou. Schopnost selekce umožňuje lidem reagovat jen na určitý druh podnětů a jiné ignorovat. Obvykle jsou vybírány podněty, které

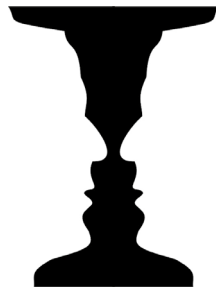
jsou nějakým způsobem důležité, nové a nebo zapadají do systému dříve nabytých zkušeností a znalostí. [22][7]

Distribuce

Schopnost rozdělovat pozornost na různé činnosti nebo na více zdrojů informací je rovněž velmi důležitá. Distribuci lze velmi dobře popsat na příkladu studenta, který poslouchá výklad svého vyučujícího a ještě zvládá zapisovat si poznámky v rozumném tvaru. Rozdělení pozornosti je snazší, když je každá z prováděných aktivit více zautomatizovaná a není nutné se na ni plně soustředit. Přenášení pozornosti z jednoho podnětu na druhý představuje schopnost a míru přizpůsobení se aktuálním podmínkám. [22][7]

Tenacita

Tenacita, tj. stabilita udržení pozornosti, reprezentuje míru změny koncentrace pozornosti nebo jejích výkyvů v průběhu nějakého časového intervalu. Soustředění se na jednu věc delší dobu může vést ke zhoršení koncentrace, snížení flexibility reakce, únavě, ztrátě motivace atd. Fluktuace je periodické kolísání pozornosti, které se projevuje střídáním zaměření na různé aspekty pozorovaného předmětu nebo jevu. Jde o fyziologický projev, který napomáhá udržení potřebné úrovně koncentrace tím, že snižuje jednostrannost jejího zatížení (viz obrázek 3.3). [22]



Obrázek 3.3: Rubinův obrazec, ve kterém jsou střídavě vidět lidské obličejové tváře a váza. [11]

3.1.3 Exekutivní funkce

Pod exekutivní funkce spadají vyšší psychické funkce, ke kterým se řadí především plánování, schopnost řešení problémů, rozhodování, úsudek atp. Tvořeny jsou řadou podřízených kognitivních operací, ze kterých je důležité zmínit hlavně pracovní paměť a pozornost. Kognitivní funkce mohou být

definovány jako schopnost člověka regulovat a ovládat své vlastní chování. [13]

Exekutivní funkce mohou být rozděleny do čtyř složek, které obsahují různé procesy: [9]

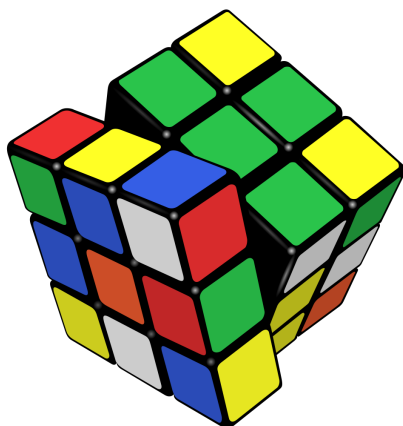
1. **Vůle** se popisuje jako schopnost záměrného jednání. Odpovídá procesům, které určují různé potřeby a přání jedince. Dále reprezentuje rozhodnutí, zda budou jednotlivé potřeby a přání uspokojeny. [9]
2. **Plánování** představuje schopnost získání, rozřídění a uspořádání vjemů a informací nutných k dosažení nějakého předem definovaného cíle. Tato složka se podílí na tvorbě plánů, zhodnocení jejich efektivity a případné změně plánu v závislosti na konkrétní situaci. [9]
3. **Jednání** je průběhem nějaké aktivity od jejího zahájení až po ukončení. Účelné jednání je nezbytné pro zvládnutí jakýchkoliv úkolů. [9]
4. **Výkon** je schopnost kontrolovat své jednání, regulovat intenzitu tempa vykovávání nějaké činnosti a v neposlední řadě i rozpoznání dokončení úkolu. [9]

3.1.4 Řeč

Základním prostředkem komunikace je jazyk. Pomocí něj člověk poznává svět, rozvíjí své myšlení, vyjadřuje své potřeby a komunikuje s ostatními lidmi. Konkrétní jazykovou dovedností je řeč. Řeč je úzce spojená s myšlením. Slouží ke zpracování informací na takové úrovni, jaké dosahuje myšlení určitého jedince. Myšlenkové operace mohou být pomocí řeči transformovány na slova a věty, které reprezentují jejich samotný význam. [23]

3.1.5 Zrak a prostor

Zrakově-prostorové schopnosti napomáhají člověku orientovat se v prostoru a chápat rozložení objektů ve svém okolí. Umožňují nejen manipulovat s dvou a třírozměrnými předměty na ploše (v prostoru), ale reprezentují i schopnost si tyto předměty představit a s touto představou hýbat či rotovat. Při tréninku prostorové orientace napomáhají různé hlavolamy, jako je například Rubikova kostka (obr. 3.4).



Obrázek 3.4: Rubikova kostka

3.2 ADHD

Porucha pozornosti s hyperaktivitou (Attention Deficit Hyperactivity Disorder) je psychologická porucha, která se projevuje již od raného dětství, nejvíce však ve školním věku. Děti s touto poruchou chronicky trpí nepozorností, nadměrnou aktivitou a impulzivitou. Největší potíže mají s udržením pozornosti při vykonávání nějaké činnosti nebo úkolu. Tyto potíže se mohou projevit už při hře. Děti trpící ADHD si hrají s jedním typem hračky kratší dobu a často přebíhají od jedné hračky k druhé. Nejvíce se ale obtíže projeví v situacích, kde jsou děti nuceny udržovat pozornost a soustředěnost na nudné a opakované úlohy (školní práce, domácí úkoly nebo jiné nepřiliš zajímavé úlohy). [14]

Jedinci trpící poruchou pozornosti s hyperaktivitou nejsou schopni domyslet důsledky svého chování a jejich projevy se mohou ostatním lidem zdát jako schválnosti. Z toho důvodu mohou tyto děti špatně vycházet se svými vrstevníky a stávají se oběťmi šikany. V konečném důsledku mohou mít snížené sebevědomí, deprese nebo pocity úzkosti, a ačkoliv za své chování nemohou, v dospělosti se nebudou moci začlenit do běžného života. [16]

3.2.1 Symptomy

Níže popsané symptomy jsou nejtypičtějším symptomy syndromu ADHD.

Hyperaktivita

Děti ve školním věku bývají aktivnější, ale jejich aktivita je spíše samoúčelná. Jsou nápadně pohyblivé, neklidné a živé. Projevy hyperaktivity jsou zřejmé hlavně v situacích, ve kterých je standardní chovat se ukázněně. [21]

Impulzivita

Impulzivní jedinci jednají nahodile a chaoticky, než aby postupovali systematicky, a mají problémy se sebeovládáním. Tito jedinci zbrkle vykonávají nebezpečné aktivity, aniž by si uvědomili jejich důsledky. [21]

Pozornost

Pozornost je s poruchou pozornosti s hyperaktivitou úzce spojena. Pro trpící ADHD je těžké dokončit zadané úkoly, mají problémy se soustředěním a dělají mnoho chyb. Častěji zapomínají, nevěnují pozornost svému zevnějšku a špatně se soustředí při rozhovoru s druhými lidmi. [21]

3.2.2 Příčiny vzniku

Příčiny vzniku ADHD nejsou zcela objasněné a nelze je všechny v současné době určit. Je nutné posuzovat příčiny v rámci celkového kontextu pacienta a jeho prostředí. Názory odborníků se však shodují v tom, že příčiny této poruchy jsou kombinací mnoha vlivů, především dědičnosti, poškození mozku nebo poškození činnosti neurotransmiterů¹. [16]

Genetika

ADHD může být z velké části dědičnou poruchou. U lidí s touto poruchou byl nalezen vysoký výskyt jedné konkrétní varianty genu, který ovlivňuje mozkovou aktivitu. Pokud má porucha genetický původ, nelze člověka trpícího touto poruchou zcela vyléčit, avšak je možné mu pomoci se s obtížemi vyrovnat. [21]

Poškození mozku

Řada odborníků se domnívá, že poruchu pozornosti s hyperaktivitou může zapříčinit i organické poškození mozku. Takové poškození by skutečně mohlo narušit chování nebo schopnost koncentrace, ale zatím neexistuje žádná studie, která by tuto příčinu plně prokázala. [21]

Mozková dysfunkce

Podle provedených výzkumů jsou u některých dětí trpících ADHD narušeny funkce neurotransmiterů (přenašečů signálů mezi neurony). Má-li člověk po-

¹Neurotransmitter je látka, která je uvolněna neuronem ke specifické cílové buňce a vyvolá v ní specifickou odezvu. [10]

škozené důležité mozkové pochody, může být výrazně a vážně narušeno jeho chování. [21]

3.3 Kognitivní trénink

Všechny výše zmíněné kognitivní funkce lze trénovat (procvičovat) a rehabilitovat (opětovně nabýt schopnost ztracenou úrazem nebo nemocí). Oblast tréninku a rehabilitace kognitivních schopností může být rozdělena na kognitivní trénink, kognitivní rehabilitaci a neurorehabilitaci: [7]

1. Kognitivním tréninkem se rozumí procvičování schopností u zdravých (nebo téměř zdravých) osob, které tímto tréninkem své schopnosti posilují. Posílené kognitivní funkce jsou v důsledku i odolnější proti případným kognitivním poruchám. [7]
2. Kognitivní rehabilitace je náprava poškozených kognitivních funkcí. [7]
3. Neurorehabilitace je komplexní péče o osoby s těžkým poraněním mozku. V dalších fázích léčby se postupně přechází na kognitivní rehabilitaci. [7]

Kognitivní trénink napomáhá léčbě poruch, které jsou způsobeny poškozením kognitivních funkcí (např. ADHD). Obsahem kognitivních tréninků je procvičování jednotlivých kognitivních funkcí pomocí cvičení, která jsou pro tento záměr speciálně navržena. Jedná se o verbální, písemné i kresebné úkoly. Tato cvičení se snaží napodobit důležité součásti běžného života a situace, v nichž se pacient musí umět řádně chovat. [7]

S rozšířením technologických zařízení (počítačů, tabletů, dotykových telefonů) jsou postupně tato cvičení převáděna do digitální podoby a jsou z nich vytvářeny počítačové hry.

3.3.1 Komerční systémy

Za účelem kognitivního tréninku bylo vytvořeno několik herních nástrojů a systémů.

HappyNeuron

HappyNeuron je systém, který poskytuje metody pro trénování mozkové aktivity a kognitivních funkcí. Zaměřuje se na všechny kognitivní funkce zmíněné v sekci 3.1. Tento systém je v provozu již více než 10 let, a proto je velmi vyspělým nástrojem nejen po technické, ale i odborné stránce. Společnost

vyvíjející tento systém si zakládá na konkrétním přizpůsobení jednotlivých úloh a cvičení pro každého pacienta.

Elevate

Elevate je mobilní aplikace vytvořená společností Elevate Labs. Mobilní aplikace byla vytvořena jako tréninkový program pro stimulaci a posilování mozkových funkcí. Je koncipována na trénink pozornosti, řečových schopností, paměti apod. Vzhledem k svému charakteru není aplikace zcela vhodná pro pacienty trpící poruchou kognitivních funkcí, ale spíše jako trénink mozkové aktivity zdravého člověka.

Lumosity

Lumosity je systém, který obsahuje velké množství her, jež jsou (nejen) po grafické stránce zpracovány velmi kvalitně. Systém je vytvořen společností Lumos Labs spolupracující s více než 100 výzkumnými pracovníky, kteří se problémem tréninku kognitivních funkcí zabývají. Systém Lumosity byl vytvořen jako jednoduchý online nástroj pro trénink kognitivních funkcí. Stejně jako u Elevate je aplikace koncipována spíše na trénink než na rehabilitaci.

Braingle

Braingle je webová aplikace vytvořená na principu sociální sítě. Lidé zde sdílejí své hry, hádanky, hlavolamy a kvízy, které mají za úkol trénovat mozkové funkce. Ostatní tyto hry hrají, hodnotí a sdílejí výsledky.

Výše popsané systémy a nástroje jsou velmi kvalitními reprezentanty kognitivního tréninku. Pracují na nich několikačlenné týmy konzultující své kroky s psychiatrickými odborníky. Většina z nich ale příliš nenapomáhá terapeutické činnosti. V tomto ohledu má největší zastoupení systém HappyNeuron, který přímo dovoluje interakci se systémem z pohledu terapeuta. Zmíněné komerční systémy, jak je již patrné z jejich klasifikování, jsou plně přístupné až po zaplacení nějaké částky a nelze do nich integrovat vlastní hry. Většina z nich poskytuje funkci bezplatného vyzkoušení, a proto bylo možné otestovat nějaké již vytvořené kognitivní hry a vytvořit si představu o průběhu her trénujících kognitivní funkce.

3.3.2 Kognitivní hry

Při analyzování systémů na kognitivní trénink bylo možné otestovat i některé z přístupných kognitivních her. Tento typ her má většinou několik společných vlastností:

- **Jednoduchost** - Všechny hry jsou po své grafické i logické stránce velmi jednoduché. Herní prvky jsou zřetelně a jednoduše rozděleny po herní scéně. Herní úkony nejsou příliš těžké na pochopení, cíl hry je intuitivní již od spuštění hry.
- **Úrovně** - Úrovně v těchto hrách tvoří velmi důležitou část. Téměř všechny hry jsou rozdělené na úrovně, které se liší například svou obtížností. Obtížnost se postupně zvyšuje a plnění dalšího úkolu je logicky těžší.
- **Odměny** - Po každém správně zahraném kole (tahu) je hráč odměněn body, pochvalou nebo přívětivým zvukovým efektem. Tato skutečnost hráče motivuje pokračovat a hru dohrát co nejlépe. Při chybném kroku je o body připraven, tento trest nutí hráče více se soustředit a získat své nahrané body zpět.
- **Pokyny** - Většina her je na začátku obohacena o nějaký tutoriál, který pomáhá hráči získat přehled o hře, o jejím cíli a předkládá užitečné rady, jež se mohou v průběhu hraní hodit. Během hry je možné si tyto pokyny připomenout stisknutím tlačítka označeného jako nápověda.

4 Návrh her

Primárně pro pacienty trpící ADHD je potřeba vytvořit 4 hry, které poslouží k rehabilitaci a tréninku některých kognitivních funkcí. Výsledné hry budou využívány v psychiatrické nemocnici v Dobřanech při léčbě ADHD. Při návrhu a vytváření těchto her je nutné brát zřetel na jejich integraci do systému BrainIn. Všechny hry by měly dodržovat ověřené postupy kognitivních her (blíže popsané v sekci 3.3.2) a měly by obsahovat funkcionalitu parametrizace.

4.1 Návrh parametrů

Průběh navrhovaných kognitivních her může být ovlivňován vstupními parametry, statistiky z odehrané hry jsou předávány parametry výstupními.

4.1.1 Vstupní parametry

Vstupní parametry budou obvykle záležet na charakteru úlohy. V matematických hrách by mohly být zadávány matematické operace (se kterými hráč bude pracovat), interval čísel (které se mohou v příkladech objevit) apod. V jiných hrách by tyto vstupní parametry mohly být zcela jiné. Některé vstupní parametry však mohou být u několika her stejné. Může se jednat o délku jedné hry, délku jednoho kola, odblokování možnosti nápovědy apod. Parametrizace nemusí zaručit pouze úpravu průběhu hry, ale také úpravu jejího vzhledu. Různé herní prvky mohou být upraveny tak, aby konkrétního pacienta hra bavila (např. při hře pexeso se pacientovi, který byl dříve automobilovým závodníkem, nahrají obrázky se závodními auty).

Z důvodu integrace her do systému BrainIn musí mít všechny hry podobné prostředí a stejné některé herní prvky. Proto budou i některé vstupní parametry stejné (trvání semaforu mezi koly, doba zobrazování hodnocení apod.).

4.1.2 Výstupní parametry

Stejně jako vstupní, tak i výstupní parametry se mohou v závislosti na hře měnit. Všechny hry by měly poskytovat informace o svém průběhu. Ideálním výstupním parametrem by mohla být procentuální úspěšnost, která by informovala o pacientově celkovém snažení, nebo o čase, za který pacient

celou úlohu dohrál. Dalšími parametry by pak v závislosti na charakteru hry mohly být: počet vygenerovaných herních prvků, počet správných a špatných odpovědí, počet pokusů, délka kola, čas spuštění apod.

Vzhledem k tomu, že výsledné hry budou začleněny do systému BrainIn, je nutné, aby výstupní parametry obsahovaly sérii parametrů datového typu *JSON*¹, které informují o konkrétních pozicích herních prvků, jejich změnách a o tom, zda s nimi hráč v průběhu hry nějak interagoval.

4.1.3 Psychiatrická nemocnice v Dobřanech

Některé vstupní a výstupní parametry byly vyžádány přímo PhDr. Karolínou Malou a dalšími terapeuty z PN v Dobřanech. Ve vstupních parametrech by podle PhDr. Karolíny Malé neměla chybět číselná hodnota, která rozdělí celou hru na části. Tyto části budou sloužit k měření pacientova výkonu v různém čase. Nutnou podmínkou je také vytvoření několika možností odměn, které se budou pacientovi zobrazovat při úspěšných herních krocích, konkrétní možnost bude zvolena právě vstupními parametry. Výstupní parametry se týkají hlavně počtu správných a špatných voleb a dříve zmíněného výkonu. Na základě sesbíraných dat budou moci terapeuti vytvořit tzv. pracovní křivku, která zobrazuje vývoj pacientova výkonu v průběhu celé hry.

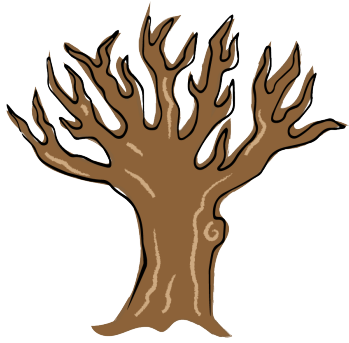
4.2 Systém odměn

Vzhledem k tomu, že pacienti trpící poruchou pozornosti s hyperaktivitou, kteří budou výsledné hry hrát, budou hlavně děti, je potřeba odměny vytvořit tak, aby zaujaly právě mladší jedince. Zároveň by měly být vytvořeny takové odměny, aby zaujaly jak chlapce, tak i děvčata. Tento systém odměn by měl nejen hráče motivovat k hraní, ale i zvyšovat rozvahu při plnění herních kroků. V případě správného herního tahu bude hráč odměněn a v opačném případě potrestán.

4.2.1 Strom

První odměnou by mohl být strom, na kterém by postupně rostly listy (květy) a při chybě by zase postupně opadávaly (viz obr. 4.1 a obr. 4.2). Tato odměna by se mohla zalíbit především dívkám.

¹JavaScript Object Notation je univerzální datový formát, kterým je možné přesouvat data. Dokáže přeměnit libovolnou datovou strukturu na formátovaný řetězec. [19]



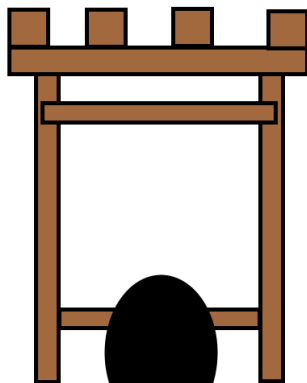
Obrázek 4.1: Stav stromu na začátku hry.



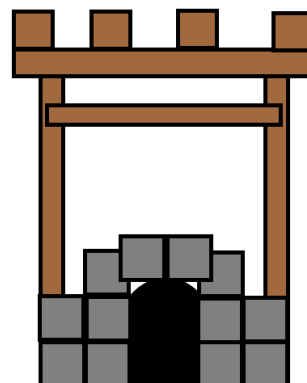
Obrázek 4.2: Stav stromu po několika úspěšných herních tazích.

4.2.2 Věž

Chlapce by mohla potěšit hradní věž, která bude postupně stavěna. Věž bude tvořena krychlovými kameny, jež se při špatném herním tahu sesypou z věže dolů (viz obr. 4.3 a obr. 4.4).



Obrázek 4.3: Stav hradní věže na začátku hry.



Obrázek 4.4: Stav věže po několika úspěšných herních tazích.

4.2.3 Měšec

Třetí odměnou by mohl být neutrální měšec, ve kterém by se počet mincí zvyšoval při korektním herním tahu a snižoval při tahu chybném (viz obr. 4.5 a obr. 4.6).



Obrázek 4.5: Vzhled měšce.



Obrázek 4.6: Vzhled mince.

4.3 Hry

Za odborné pomoci PhDr. Malé z PN v Dobřanech byly navrženy 4 hry, které by měly odpovídat základním požadavkům na kognitivní hry a měly by posloužit k kognitivnímu tréninku a léčbě pacientům s ADHD.

4.3.1 Harvest

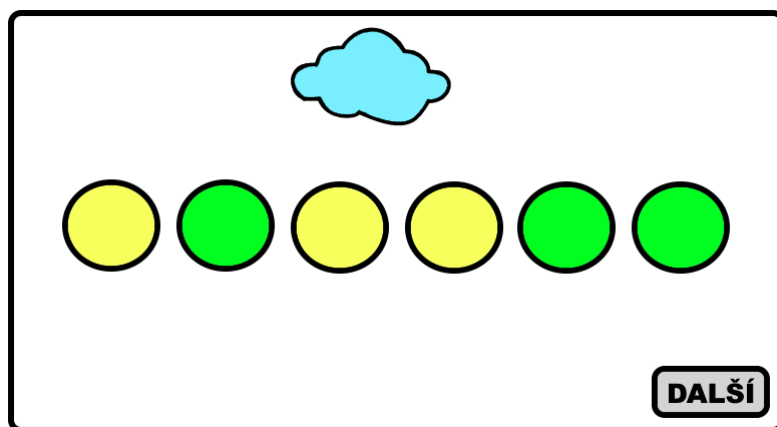
Tato hra je zaměřená především na poruchy pozornosti. Konkrétně trénuje tzv. přepínání pozornosti. Princip spočívá v plnění nějaké jednoduché úlohy, která se stane automatizovanou činností, začne hráče nudit a v tomto okamžiku přichází nějaký distraktor², po němž má hráč plnění úlohy přerušit. V tomto případě se bude jednat o sklizení hlávek salátu. Hráč musí rozlišit zdravé hlávky salátu (ty, co má sbírat) a zkažené hlávky (ty, co sbírat nesmí). Hlávky salátu budou rozděleny do záhonů a mezi záhony bude možné přecházet pomocí tlačítka (viz obr. 4.7). Po nějaké předem definované době se na obrazovce objeví mrak, bouřka nebo jen zazní hrom. V tuto chvíli bude jakákoliv sklizeň (zdravých i zkažených) hlávek salátu označena za chybnou a hráč by měl přejít na další záhon.

Vstupní parametry

Vstupní parametry této hry jsou vypsány v následujícím seznamu:

- **Délka hry** (přirozené číslo) udává v sekundách délku jedné hry. Od spuštění hry bude měřen čas a při dosažení této hodnoty bude hra automaticky ukončena.

²Distraktor je rys, vlastnost, objekt nebo jiný podnět způsobující jedinci problémy při výběrovém vnímání podnětu. [8]



Obrázek 4.7: Návrh herní scény hry Harvest.

- **Rozdělení herní doby** (přirozené číslo) rozděluje celý herní čas na intervaly. Hodnota reprezentuje velikost intervalů v sekundách.
- **Počet hlávek** (přirozené číslo) symbolizuje počet hlávek salátu v jednom záhonu.
- **Čas příchodu distraktoru** (přirozené číslo) udává v sekundách čas, po kterém by se měl zobrazit distraktor. Tento čas bude v průběhu hry různě měněn.
- **Koeficient distraktoru** (reálné číslo) reprezentuje hodnotu, kterou je původní (nebo již nově aktualizovaný) čas distraktoru vynásoben po úspěšném zaregistrování jeho příchodu a vykonání správných akcí.
- **Doba zobrazení distraktoru** (přirozené číslo) udává čas v sekundách, po který je distraktor zobrazován.
- **Pravděpodobnost příchodu náročnějšího distraktoru** (přirozené číslo) bude zadávána v procentech (tj. číslo od 0 do 100).
- **Poměr zdravých a zkažených hlávek salátu** (formátovaný text) je zadáván ve formátu klasické podoby poměru (1:1, 2:5 atp.).

Výstupní parametry

Výstupní parametry této hry jsou vypsány v následujícím seznamu:

- **Vygenerované hlávky salátu** (přirozené číslo) symbolizují počet, kolik bylo za celou hru vygenerováno těchto herních prvků.

- **Sebrané hlávky salátu** (přirozené číslo) udávají počet hlávek, které hráč sebral (špatné i správné).
- **Správně sebrané hlávky salátu** (přirozené číslo) jsou ty hlávky, které skutečně měly být sebrány.
- **Chybně sebrané hlávky salátu** (přirozené číslo) jsou hlávky, které hráč sebral při výskytu distraktoru a nebo byly zkažené.
- **Vynechané hlávky** (přirozené číslo) jsou hlávky salátu, které měly být sebrány, ale sebrány nebyly.

Taktéž pro každé kolo (časový interval) by měly být sbírány informace. Pomocí nichž je možné změřit hráčův výkon, sledovat míru pozornosti apod. Konkrétně jsou pro každý časový interval zaznamenávány informace o délce kola, vygenerovaných hlávkách, sebraných hlávkách, správně a nesprávně sebraných hlávkách a výkonu.

4.3.2 TestTubes

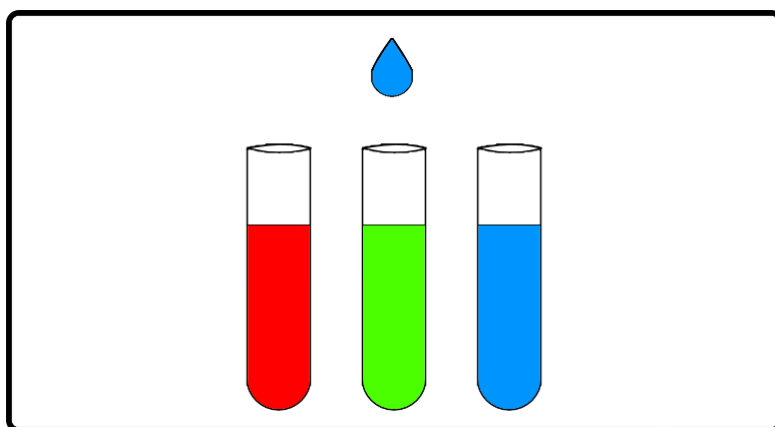
Účel této hry je téměř stejný jako u té předchozí, protože pomáhá léčit stejné neurologické obtíže.

Princip hry spočívá ve správném roztřídění barevných kapek do jedné ze tří zkumavek (viz obr. 4.8). Pacientovi jsou předkládány barevné kapky, každou z nich má za úkol přiřadit ke zkumavce, ve které se nachází tekutina se stejnou barvou. V průběhu se může objevit kapka černá (či jakkoliv jinak barevná), kterou musí hráč rozeznat a nikam ji nepřihazovat. Distraktor je v této hře reprezentován náhlým poškozením zkumavek. V tuto chvíli není možné do zkumavek nic přilévat a hráč bude muset počkat, dokud se zkumavky neopraví. Hra bude rovněž rozdělena na určité časové intervaly.

Vstupní parametry

Vstupní parametry této hry jsou vypsány v následujícím seznamu:

- **Délka hry** (přirozené číslo) udává v sekundách délku jedné hry. Od spuštění hry bude měřen čas a při dosažení této hodnoty bude hra automaticky ukončena.
- **Rozdělení herní doby** (přirozené číslo) rozděluje celý herní čas na intervaly. Hodnota reprezentuje velikost intervalů v sekundách.
- **Doba zobrazení kapky** (přirozené číslo) reprezentuje v sekundách dobu, po kterou bude zobrazena jedna kapka.



Obrázek 4.8: Návrh herní scény hry TestTubes.

- **Čas příchodu distraktoru** (přirozené číslo) udává v sekundách čas, po kterém by se měl zobrazit distraktor. Tento čas bude v průběhu hry různě měněn.
- **Koeficient distraktoru** (reálné číslo) reprezentuje hodnotu, kterou je původní (nebo již nově aktualizovaný) čas distraktoru vynásoben po úspěšném zaregistrování jeho příchodu a vykonání správných akcí.
- **Doba zobrazení distraktoru** (přirozené číslo) udává čas v sekundách, po který je distraktor zobrazován.
- **Poměr správně a nesprávně zbarvených kapek** (formátovaný text) je zadáván ve formátu klasické podoby poměru (1:1, 2:5 atp.).

Výstupní parametry

Výstupní parametry této hry jsou vypsány v následujícím seznamu:

- **Zobrazené kapky** (přirozené číslo) udávají počet, kolik bylo za celou hru vygenerováno těchto herních prvků.
- **Roztříděné kapky** (přirozené číslo) symbolizují počet kapek, které se hráč rozhodl roztřídit (špatně i dobře).
- **Správně roztříděné kapky** (přirozené číslo) jsou kapky, které byly správně roztříděny do jim náležejících zkumavek.
- **Chybně roztříděné kapky** (přirozené číslo) jsou ty kapky, které hráč roztřídil špatně nebo je vložil do poškozených zkumavek.

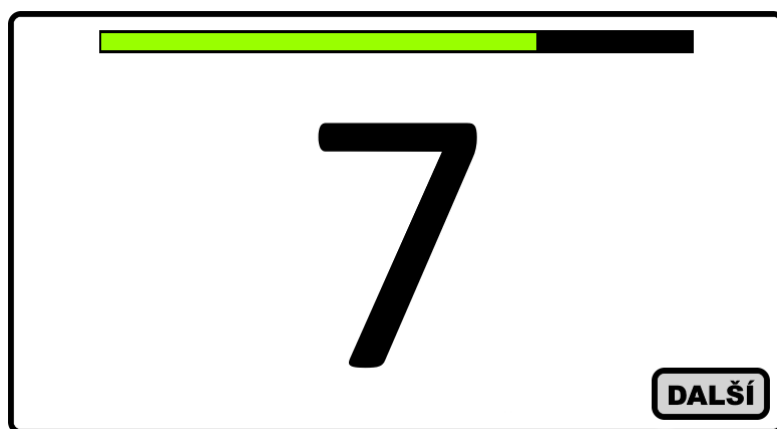
- **Vynechané kapky** (přirozené číslo) jsou kapky, které měly být roztrženy, ale nebyly.

Stejně jako u předchozí hry i zde je nutné sbírat konkrétní informace o jednotlivých časových intervalech. Měly by být zaznamenávány informace o délce kola, zobrazených kapkách, roztržných kapkách, správně a chybně roztržných kapkách a výkonu.

4.3.3 Calc

Hra Calc pomáhá trénovat exekutivní funkce a exekutivní pozornost. Proto je opět vhodná pro pacienty trpící poruchou pozornosti s hyperaktivitou, ale i pro zdravé jedince jako trénování obecného využití inteligence. Princip spočívá ve volbě nějaké matematické operace a jejího provedení. Hráči se budou postupně prezentovat jednociferná čísla, která bude muset postupně přičítat nebo odečítat (viz obr. 4.9). Hra by měla fungovat ve třech úrovních.

1. Hráč postupně přičítá prezentovaná čísla a výsledky zapisuje do textového vstupu.
2. Úkolem v této úrovni je prezentovaná čísla postupně přičítat a odečítat.
3. V této úrovni jsou zobrazovaná čísla označena barvou rozhodující o přičtení čísla k předchozímu výsledku či jeho odečtení.



Obrázek 4.9: Návrh herní scény hry Calc.

Vstupní parametry

Vstupní parametry této hry jsou vypsány v následujícím seznamu:

- **Maximální délka hry** (přirozené číslo) udává v sekundách maximální délku jedné hry. Čas bude od jejího spuštění započítáván a při dosažení této hodnoty bude hra automaticky ukončena. Ukončí hru tak, že nebyla korektně dohrána.
- **Počet čísel** (přirozené číslo), se kterými bude hráč v jednom kole počítat.
- **Doba zobrazení čísla** (přirozené číslo) udává v sekundách dobu, po níž se bude zobrazovat jeden příklad.
- **Limit** (přirozené číslo) je hodnota, kterou nesmí přesáhnout výsledek několika po sobě jdoucích příkladů.
- **Úroveň 1** (logická hodnota) rozhoduje o spuštění úrovně 1.
- **Úroveň 2** (logická hodnota) rozhoduje o spuštění úrovně 2.
- **Úroveň 3** (logická hodnota) rozhoduje o spuštění úrovně 3.

Výstupní parametry

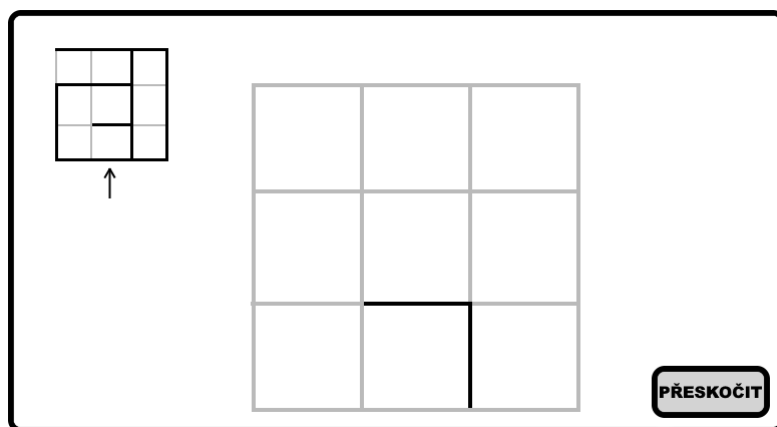
Výstupní parametry této hry jsou vypsány v následujícím seznamu:

- **Zpracované příklady** (přirozené číslo) symbolizují počet zobrazených čísel (příkladů) za celou hru.
- **Správně vypočtené příklady** (přirozené číslo) udávají počet správných výsledků.
- **Chybně vypočtené příklady** (přirozené číslo) udávají počet chybných výsledků.

Pro jednotlivé úrovně budou zaznamenávány stejné informace, tj. zpracované příklady, správné výsledky, chybné výsledky.

4.3.4 Grid

Tato hra by měla sloužit k trénování prostorové orientace a mentální rotace. Pacientovi je zobrazena čtvercová mřížka, jíž některé části chybějí (viz obr. 4.10). Cíl hry spočívá v zkonstruování mřížky totožné s předlohou. Požadavek na sestavení mřížky však může být i z jiného pohledu. V případě, že si hráč nebude vědět rady, měl by mít možnost konkrétní mřížku přeskočit.



Obrázek 4.10: Návrh herní scény hry Grid.

Vstupní parametry

Vstupní parametry této hry jsou vypsány v následujícím seznamu:

- **Maximální délka hry** (přirozené číslo) udává v sekundách maximální délku jedné hry. Čas bude od jejího spuštění započítáván a při dosažení této hodnoty bude hra automaticky ukončena. Ukončí hru tak, že nebyla korektně dohrána.
- **Počet kol** (přirozené číslo), které by měl hráč dohrát pro korektní ukončení hry.
- **Velikosti mřížek** (formátovaný text) reprezentují počty sloupců a řádků mřížek v jednotlivých kolech. Velikosti mřížek pro jednotlivá kola jsou oddělená čárkou.
- **Pohledy** (formátovaný text) udávají, z jakého pohledu je nutné sestavit mřížku v jednotlivých kolech. Jednotlivé pohledy jsou oddělené čárkou a jejich hodnoty jsou reprezentovány čísly od 0 do 3. Číslování postupuje vzestupně, po směru hodinových ručiček a hodnota 0 je pohled zdola.
- **Výplně vzorové mřížky** (formátovaný text) jsou odděleny pro jednotlivá kola čárkou a udávají, z kolika procent má být vzorová mřížka vyplněna.
- **Prázdná mřížka** (logická hodnota) rozhoduje o tom, zda se budou mřížky konstruovat od nuly.

Výstupní parametry

Výstupní parametry této hry jsou vypsány v následujícím seznamu:

- **Počet změn** (přirozené číslo) je počet, kolik částí mřížky bylo za celou hru změněno.
- **Počet správně umístěných** (přirozené číslo) udává počet správně umístěných částí mřížky.
- **Počet chybně umístěných** (přirozené číslo) udává počet chybně umístěných částí mřížky.

4.3.5 Společné parametry

Všechny hry mají některé vstupní a výstupní parametry společné.

Společné vstupní parametry jsou vypsány v následujícím seznamu:

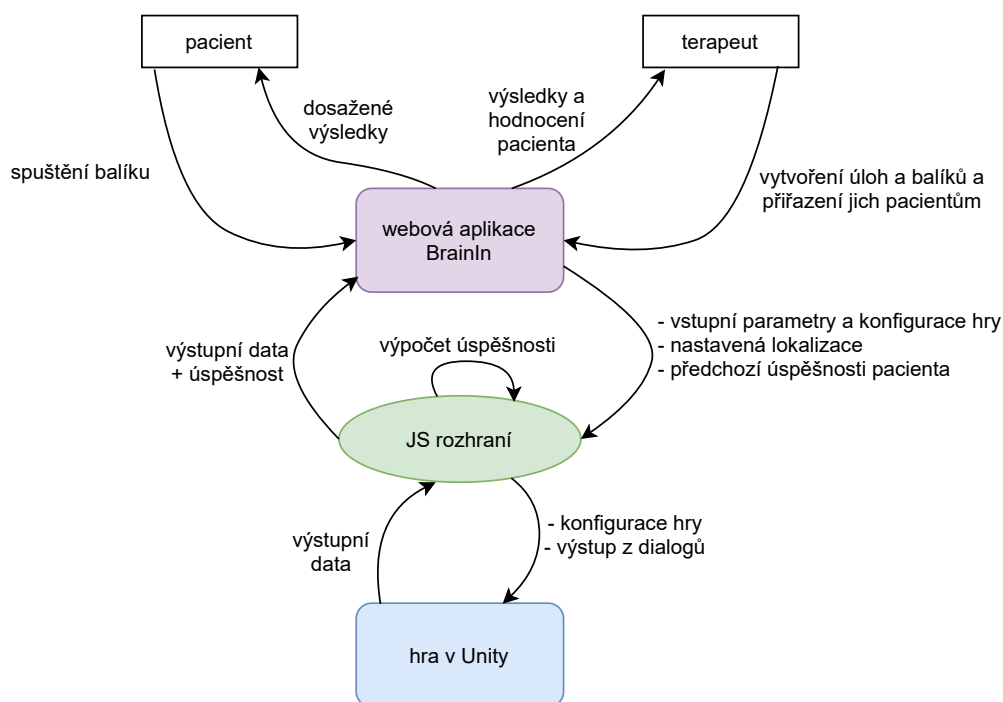
- **Tutoriál** (logická hodnota) symbolizuje skutečnost, zda má být před spuštěním samotné hry spuštěn průvodce hrou.
- **Typ odměny** (přirozené číslo) udává identifikační číslo odměny, která má být v dané hře zobrazována.
- **Trvání semaforu** (přirozené číslo) reprezentuje v sekundách dobu zobrazování semaforu před začátkem samotné hry.
- **Doba zobrazování hodnocení** (přirozené číslo) je čas, po který bude hráči zobrazováno hlášení o proběhlé hře.

Společné výstupní parametry jsou vypsány v následujícím seznamu:

- **Celkový čas** (přirozené číslo) úlohy od jejího spuštění do jejího ukončení.
- **Úspěšnost** (přirozené číslo) hráčova počínání vyjádřená procentuálně.
- **Datum a čas spuštění** (text) informuje o skutečném datu a času spuštění úlohy.
- **Lokalizace** (text) nabývá hodnot *cs*, *en* nebo *de*.

5 Specifika implementace

Webová aplikace BrainIn pracuje primárně s hrami, které jsou vyvíjeny pomocí herního engine Unity. Webová aplikace spouští a komunikuje s hrou pomocí rozhraní vytvořeného v jazyce JavaScript (viz obr. 5.1). Každé hře je proto přiděleno několik skriptů, které se starají o korektní spuštění, zobrazování správných dialogů, správnou volbu jazyka nebo výpočet úspěšnosti. Každou hru je tedy nutné vytvořit ve výše zmíněném herním engine a je nutné pro ni vytvořit (nebo upravit již existující) skripty pro komunikaci s webovou aplikací. Implementace každé hry musí zahrnovat nějaké rozdělení vstupních parametrů, vygenerování úrovní (případně herní scény) a průběh hry s průběžným sběrem statistik a informací. Pro každou hru je na základě požadavků PhDr. Malé nutné vytvořit vhodné grafické prvky. Pro správnou integraci her do webové aplikace BrainIn je nezbytné zahrnout do implementace také vytvoření šablony a lokalizovaných textů, které se budou v průběhu hry zobrazovat v závislosti na zvolené lokalizaci.



Obrázek 5.1: Diagram komunikace jednotlivých částí systému BrainIn.

5.1 Vytváření šablony

Při znalosti všech vstupních a výstupních parametrů, které by daná hra měla obsahovat, je možné vytvořit šablonu. Po spuštění vytvořené úlohy, v níž byly vyplněny všechny vstupní parametry, je vygenerován řetězec (např. řetězec níže), který obsahuje všechna tato data oddělená nějakým předem specifikovaným speciálním znakem (konkrétně několika znaky #). Dále je nutné si rozmyslet, z jakých výstupních dat bude počítána úspěšnost.

```
cs###01/01/2000 00:00:00###0###9###1###1###0###0###1###0###2###1###5
```

5.2 Herní engine Unity

Unity podporuje vývoj her v relativně graficky přívětivém vývojovém prostředí. V tomto prostředí se tvoří hlavně vzhled her nebo různé herní objekty. Logiku hry pak řídí skripty vytvářené v objektově orientovaném programovacím jazyce C#. [3]

5.2.1 Prvky vývojového prostředí

Scene

Herní objekty, grafika nebo tlačítka se umísťují do tzv. scény. Hra může obsahovat i několik scén (např. pro každou úroveň jednu). [3][2]

GameObject

Herní objekt je objekt, který je vytvářen v rámci jedné scény. Každý objekt má určenou svou polohu ve scéně a může obsahovat několik komponent, které ovlivňují jeho chování. Herní objekt může být složen i z několika dalších herních objektů (např. herní objekt auto je složeno z karoserie a 4 herních objektů reprezentujících kola). [3][2]

Prefab

Prefabrikát je herní objekt, který má již nastavené nějaké své vlastnosti a může být s těmito vlastnostmi vytvářen několikrát, třeba i v průběhu hry. Nejlépe lze funkci prefabrikátu popsat na příkladu. Vývojář vytvoří v jedné scéně žluté tlačítko, ze kterého vytvoří i prefabrikát. Nyní může napsat nějaký svůj skript, který z právě tohoto předaného prefabrikátu vytvoří takových tlačítek třeba 10. [3][2]

Component

Komponenta je část herního objektu. Lépe ji lze označit jako vlastnost daného objektu, protože v zásadě ovlivňuje hlavně jeho chování nebo vzhled. Komponentou může být už nějaká předem definovaná vlastnost (Unity nabízí například komponenty simulující nějaké fyzikální vlastnosti jako tíhové zrychlení), anebo vývojářem vytvořený skript, díky kterému je herní objekt obarvován, přemísťován apod. [3][2]

Asset

Asset je jakýkoliv soubor (obrázek, skript atp.), který se nachází v samotném projektu. Tento soubor může být přiřazován do různých komponent nebo může být vkládán na herní scénu. [3][2]

5.2.2 Průběh hry

Každá hra je spouštěna a vychází z hlavního skriptu **Main**. Tento skript (a další skripty, které dědí od **MonoBehaviour**) obsahuje tři funkce ovlivňující běh hry. Jsou to funkce **Awake()**, **Start()** a **Update()**. Funkce **Awake()** je volána při inicializaci herního objektu, který ten daný skript obsahuje ve svých komponentách. Skript **Main** je komponentou herního objektu *Camera*, ta je vytvářena ihned po vytvoření herní scény. Funkce **Start()** je volána na začátku prvního použití skriptu, poté je v nekonečné smyčce volána funkce **Update()**. Volání této funkce proběhne několikrát za sekundu. [5]

5.2.3 Šablona šablon

Programátoři a tvůrci webové aplikace **BrainIn** se současně s řadou českých i zahraničních terapeutů shodují, že by všechny hry v této aplikaci měly mít podobné herní prvky a měly by fungovat na základě nějakých stanovených postupů. Proto byla vytvořena kostra hry, která byla nazvána šablonou šablon. Tato kostra obsahuje herní scénu se všemi vzhledy tlačítek nebo herními objekty, které by každá hra měla obsahovat. Součástí implementace šablony šablon jsou také postupy pro ukládání různých informací o rozložení objektů po herní scéně, skripty řídící komunikaci s webovou aplikací, nebo třeba obsluha přehrávání zvuků.

6 Implementace

V této části je popsána implementace nejzajímavějších částí jednotlivých her. Vývoj proběhl v herním enginu Unity ve verzi 2018.4.7f1. Komunikace webové aplikace a her vytvořených v této verzi Unity je otestována a funguje. Dále bylo cíleno na nějakou jednotvárnost všech her projektu BrainIn. Z těchto důvodů byla tato verze vybrána i pro vývoj již zmíněných 4 her. Každá z her je vytvořena doplněním šablony šablon o několik dalších herních prvků, skriptů apod.

6.1 Vstupní parametry

Funkce `SetInputParameters()` ve třídě `InputParams` se stará o rozdělení vstupního řetězce podle předem stanovených oddělovačů. Samotný skript obsahuje všechny důležité vstupní parametry jako atributy třídy, do kterých se v průběhu funkce `SetInputParameters()` nastaví příslušné hodnoty. Správnost hodnot je kontrolována také v průběhu této funkce (např. zda je maximální doba číslo a zda je větší než 0). V případě jakékoliv chyby je hra ukončena a o této skutečnosti je pomocí příslušných dialogů informován i terapeut, který chybnou konfiguraci úlohy vytvořil.

6.2 Generování hry

Po rozdělení vstupních parametrů je typicky spuštěna generace dat do jednotlivých úrovní. Ve hrách `TestTubes` a `Harvest` je generace spouštěna v průběhu hry.

6.2.1 Seed

Konkrétní nastavení herních úrovní je generováno pomocí generátoru pseudonáhodných čísel. Seed generátoru je obvykle nastaven z aktuálního data a času při vytváření instance generátoru. Terapeut může do vstupních parametrů zahrnout i nějaký svůj vlastní seed, ale pokud žádný nezadá, bude seed vytvořen právě z aktuálního data a času. K tomuto přístupu se nabízí modelová situace, že pacient odehraje hru, ve které si ale nevedl vůbec dobře. Terapeut bude chtít zkontrolovat, co mu na té hře nešlo, a bude ho chtít nechat zahrát tu samou hru znovu. Při vytváření úlohy však nezadal

žádný svůj seed, takže se hra spustila s konfigurací data a času, a proto už nebude mít možnost totožnou hru opakovat. Tento problém vyřeší to, že se do výstupních parametrů přidá parametr reprezentující seed. Tato hodnota se však v jazyce C# z instance třídy `Random` nedá jednoduše získat. Řešením je vytvoření druhého pomocného generátoru. V případě, že terapeut do vstupních parametrů zadá nějaký svůj seed, bude pro generování herních dat využit. Pokud žádný nezadá, pomocný generátor vytvoří náhodné celé číslo, které bude použito jako seed pro skutečný generátor herních dat a na konci hry se společně se statistikami vytiskne do výstupu úlohy.

```
1 public static int SetRandomSeed(int seed){
2     int finalSeed = seed;
3     if(finalSeed == 0){
4         Random seedGenerator = new Random();
5         finalSeed = seedGenerator.Next();
6     }
7     rand = new Random(finalSeed);
8     return finalSeed;
9 }
```

6.2.2 Harvest

V této hře se generování týká jen zobrazovaných hlávek salátu. Generování salátového lůžka je voláno vždy, když má být zobrazeno nové salátové lůžko. Na začátku hry není možné všechna salátová lůžka vygenerovat, protože není nikdy dopředu známo, kolik jich zvládne hráč projít. Třída `LettuceGenerator` obsahuje statický konstruktor, který je zavolán před prvním voláním jakékoliv funkce této třídy. V konstruktoru se na základě vstupních parametrů `ButtonsRatio` a `ButtonsCount` zjistí hodnota jednoho dílu poměru. Do proměnných `OkLettuceCount` a `SpoiledLettuceCount` se uloží počty zdravých a shnilých hlávek salátu, jež jsou vypočteny vynásobením správné části poměru s hodnotou jednoho dílu poměru. Zbytek po dělení počtu hlávek v jednom lůžku součtem obou stran poměru tvoří hlávky, u kterých není možné pomocí poměru určit, zda budou shnilé nebo zdravé. Stav zbylých hlávek je tedy určen náhodně. Z výsledných počtů hlávek salátu je vytvořeno pole logických datových typů, které je naplněno příslušným počtem hodnot `true` (zdravé hlávky) a `false` (shnilé hlávky) a následně jsou tyto hodnoty náhodně zamíchány.

Z důvodu používání poměru tímto způsobem je nutné každý zadaný poměr převést do jeho základního tvaru. O zjednodušení do základního tvaru se stará funkce `SimplifyButtonsRatio()` z třídy `Tools`. Funkce zjistí pomocí Euklidova algoritmu největší společný dělitel dvou čísel (v tomto případě

levé a pravé složky poměru), tyto hodnoty jím vydělí a následně výsledky v podobě dvouprvkového pole vrátí.

6.2.3 TestTubes

V této hře se neukazuje více generovaných objektů najednou, ale vždy jen jeden (barevná kapka). Pokud by se před každým zobrazením kapky generovala kapka nová, sekvence zobrazených kapek by nebyla úplně náhodně přeházená. Kapky se totiž tvoří opět z nějakého poměru, a kdyby se vytvářely postupně, mohlo by se stát, že se správné a chybné kapky budou střídát, anebo budou zobrazeny vždy nejdříve ty správné a až potom ty chybné. Tento problém byl vyřešen vytvořením konstanty v třídě `Constants` nesoucí název `SIZE_OF_DROP_QUEUE`. Ta byla po nějakém pozorování výsledků generace nastavena na hodnotu 20. Ve třídě `DropGenerator` byla vytvořena instance datové struktury fronta, do které jsou po generaci přidány právě vygenerované kapky. Před každým zobrazením další kapky je zkontrolován obsah fronty. Pokud fronta není prázdná, je z ní vyňat jeden prvek, který reprezentuje právě zobrazovanou kapku. V případě, že je fronta prázdná, je znovu vygenerováno 20 kapek v zadaném poměru a v náhodně rozházeném pořadí. Jelikož je princip této hry velmi podobný principu hry Harvest, je samotné generování herních objektů v nějakém poměru vlastně totožný.

6.2.4 Calc

Generování dat do jednotlivých kol je v této hře bezpochyby nejzajímavější. Do každé úrovně se náhodně generují čísla, jejichž výsledek matematických operací nesmí přesáhnout limit, který byl zadán ve vstupních parametrech. Generovaná čísla musí být jednociferná, a nezahrnují hodnotu 0. O výše zmíněné generování se stará funkce `GenerateNumbers()` z třídy `LevelGenerator`, jejíž zdrojový kód je zobrazen níže. Nejprve byl vyzkoušen nejjednodušší přístup, který vygeneroval číslo, provedl matematickou operaci, ověřil podmínky a číslo přidal do pole. Tento přístup však nebyl úplně vhodný, protože po vygenerování všech čísel nemusela platit podmínka pro nepřekročení limitu a postup generování musel být spuštěn znovu. Pro vygenerování jedné úrovně dosahoval počet iterací cyklu i několika set tisíc. Z tohoto důvodu byl implementován lepší postup, který nastavuje přímo rozsah generovaných čísel. Konkrétní rozsah je určován pomocí vzorce 6.1.

$$\text{maximum} = \frac{\text{limit} - \text{mezivýsledek}}{\text{počet_čísel} - i} \quad (6.1)$$

Hodnota *maximum* udává největší možné číslo, které může generátor pseudonáhodných čísel vygenerovat. Tímto přístupem je počet iterací cyklu razantně snížen a samotná generace je mnohem rychlejší.

```
1 private static void GenerateNumbers(int start, int level) {
2     int number;
3     int range;
4     int tmpSum;
5     bool operation;
6
7     for (int i = start; i < numbers.Length; i++) {
8         range = (upperLimit - sum) / (numbers.Length - i);
9         range = range > 9 ? 9 : range;
10        do {
11            tmpSum = sum;
12            number = Tools.GetRandomInt(range) + 1;
13            operation = GetOperation(level, i);
14            tmpSum += operation ? number : -number;
15        } while (tmpSum < 1);
16
17        AddNumber(number, operation, i);
18    }
19 }
```

6.2.5 Grid

Při vytváření nových úrovní je nastavováno zabarvení hran u vzorové, ale i herní mřížky. Konkrétní vzhled mřížek pro jednotlivé úrovně generuje funkce `GenerateOneGrid()` z třídy `GridGenerator`. Tato funkce nejprve vytvoří dvě pole logických datových typů o velikosti počtu hran. Počet hran v mřížce je určen pomocí vzorce 6.2, kde e značí výsledný počet hran a n velikost mřížky.

$$e = 2n^2 + 2n \quad (6.2)$$

Pole pro vzorovou mřížku je před zamícháním naplněno hodnotami *true* podle procentuální výplně zadané ve vstupních parametrech úlohy. V poli, které reprezentuje hrany herní mřížky, je na hodnotu *true* upraven náhodný počet prvků. Zároveň se kontroluje, zda herní a vzorová mřížka nejsou totožné, a v případě, že totožné jsou, je proces vytváření herní mřížky zopakován.

6.3 Herní průběh

Herní průběh je řízen z hlavního skriptu `Main` pomocí funkcí, které mohou být pro různé hry rozdílné. Každá hra je doplněna o další skripty a třídy, jež pomáhají rozhodnout o správných či chybných herních krocích.

6.3.1 Harvest

Z pole logických datových typů (vzniklého v generující funkci) jsou vytvořeny konkrétní hlávky salátu. Každá hlávka je reprezentována jedním herním objektem. Herní objekt je možné v průběhu programu vytvořit pomocí Unity funkce `Instantiate()`. Pro vytvoření hlávky salátu vstupují do této funkce parametry `PrefabOptionButton` a `OptionArea.transform`. Parametr `PrefabOptionButton` je mnou vytvořený prefabrikát reprezentující hlávku salátu a parametr `OptionArea.transform` určuje rodiče vytvářeného objektu v stromové struktuře herních objektů.

Každé hlávce salátu (hernímu objektu) je přiřazena komponenta (skript) `OptionButton`. Tento skript udržuje o hlávce její stav (zda má být stisknuta nebo ne) a překrývá metodu `RespondToClickOnLeftButton()` (zděděnou od třídy `CursorHoverEffect`), která volá příslušné funkce po jejím stisknutí.

Jednou z funkcí, která je volána po stisknutí hlávky salátu, je funkce `OptionClick()` z hlavního skriptu `Main`. Uvnitř této funkce se na základě správnosti dané hlávky nastavuje její další vzhled, spouští se animace odměn, spouští se zvukový efekt a správnost kliknutí se uloží do výsledných statistik.

K dalšímu salátovému lůžku se přechází pomocí tlačítka „další“, které konkrétně volá funkci `NextClick()`. Cílem této funkce je ověřit, zda ještě není konec hry. V případě, že ještě konec není, zničí právě zobrazené salátové lůžko, zkontroluje potencionální výskyt distraktoru a vytvoří nové hlávky salátu. V opačném případě zaznamená konec hry do výstupních statistik a ukončí program.

Příchod distraktoru je ověřován ve funkci `Update()`, kde se porovnává čas nastavený vstupním parametrem hry s dobou, která uplynula od jeho posledního zobrazení. Všechny časové úseky, které je nutné nějakým způsobem měřit (např. doba zobrazení distraktoru, uplynulá herní doba), jsou spravovány třídou `RealTime`. Při inicializaci proměnné reprezentující nějaký čas (konkrétně instance třídy `Time`) jí je nastaven datum a čas inicializace. Ve chvíli přístupu k hodnotě této proměnné je okamžik inicializace odečten od aktuálního data a času, čímž je zjištěna uplynulá doba (tento přístup měření času je ve všech ostatních hrách stejný).

6.3.2 TestTubes

Na herní scéně jsou vytvořena tři tlačítka se vzhledem zkumavek a jeden herní objekt pro zobrazovanou kapku. Každá zkumavka má přiřazený skript `OptionButton` jako komponentu. Tento skript obsahuje referenci na jednu z prvních tří hodnot třídy výčtového typu `DropColors`. Herní objekt reprezentující kapku obsahuje komponentu `DropAttributes`. Tento skript uchovává informace o aktuálně zobrazené kapce (skutečný stav a barvu). V případě stisknutí kterékoliv ze tří zkumavek je porovnána barva aktuálně zobrazené kapky s barvou zkumavky a také hodnota stavu kapky. V případě, že je aktivní distraktor, je skutečný stav kapky nastaven na hodnotu *false*, a to i v případě, že má totožnou barvu jako jedna ze zkumavek. Herní průběh se jinak příliš neliší od hry Harvest (viz sekce 6.3.1).

6.3.3 Calc

Pro každou úroveň je vytvořena instance třídy `Level`, která obsahuje pořadí úrovně a dvě pole přirozených čísel. První pole je naplněno čísly, která se budou hráči postupně zobrazovat, a pole druhé obsahuje mezivýsledky po každém zobrazeném čísle. Každé číslo je reprezentováno instancí třídy `Number` obsahující nejen hodnotu daného čísla, ale také informaci, zda se to dané číslo bude přičítat nebo odečítat.

Funkce `RoundMainAreaControl()` (volána z funkce `Update()`) pomocí logické proměnné *numberDisplayPart* spravuje kontrolu doby zobrazení čísla na herní scéně. Pokud je hodnota nastavena na *true*, kontroluje, zda čas zobrazení jednoho čísla (měřeno v skriptu `RealTime`) nepřekročil dobu určenou ve vstupních parametrech, a volá funkci, která se stará o správné zobrazení časovače. V případě, že skončila doba zobrazení jednoho čísla, je proměnná *numberDisplayPart* nastavena na hodnotu *false* a je spuštěna fáze zadávání mezivýsledku.

Po zadání mezivýsledku je hodnota textového vstupu převedena na číselnou hodnotu a je zkontrolováno, zda je výsledek stejný jako prvek na daném indexu v poli výsledků. Na základě správnosti jsou přehrány zvukové efekty a je zobrazen správný výsledek s příslušným zabarvením. Pokud se jednalo o poslední číslo v daném kole, je spuštěno kolo nové (jinak je samozřejmě zobrazeno další číslo), a pokud se jednalo i o poslední kolo, je hra ukončena.

6.3.4 Grid

Na začátku každého kola je nejprve vytvořen objekt herní mřížky. Bylo implementováno 5 verzí herní mřížky (od velikosti 1 až do velikosti 5), které byly převedeny na prefabrikáty, ze nichž se nyní herní mřížky vytvářejí. Podle vygenerovaných dat se různé hrany herní mřížky obarví a na základě vstupního parametru je vzorová mřížka přetočena, čímž se docílí jiného pohledu.

Hrany herní mřížky tvoří tlačítka, která obsahují komponentu (skript) `OptionButton`. Tento skript ukládá o tlačítku informace o jeho správnosti a o jeho aktuálním stavu (zda je stisknuté, nebo ne). V případě stisknutí tlačítka je zavolána funkce `OptionClick()`, která rozhoduje, zda stisknutí tlačítka bylo správně či nikoliv. Na základě správnosti jsou přehrávány zvukové efekty, zaznamenávány statistiky atp. Po každém kliknutí na jednu hranu mřížky je ověřováno, zda byla herní mřížka převedena do své finální podoby. V takovém případě následuje další úroveň.

6.4 Grafické prvky

Pro některé hry a jejich herní objekty byly vytvořeny zcela originální grafické prvky.

6.4.1 Způsob grafické reprezentace

Před tvorbou grafiky do her bylo nutno pečlivě promyslet, zda bude pro účely her vhodná spíše rastrová, nebo vektorová grafika. Výsledné hry by měly být schopny běhu jak na počítačích (Full HD), tak i na jiných zařízeních, jako je např. tablet, kde může být od PC zcela rozdílné rozlišení. Z důvodu potencionálních změn velikostí herní grafiky (v závislosti na velikosti rozlišení a herní scény) byla vybrána grafika vektorová. Herní engine Unity však komponenty vektorové grafiky nepodporuje a existují jen experimentální verze balíčků podporujících vektorovou grafiku (tzv. preview verze), které by někdy v budoucnu měly být převedeny do verzí plných a měly by být do Unity doplněny. Instalované komponenty pro správu vektorové grafiky pracovaly s jednoduchou vektorovou grafikou velmi dobře. Avšak se složitější vektorovou grafikou, která byla vytvořena do těchto her v programech Adobe Illustrator a Adobe Photoshop, si tyto komponenty poradit nedokázaly. Podobnou náročnost zvládá tento balíček pro správu vektorové grafiky až od Unity verze 2019 a ještě ke všemu není zaručena funkčnost na všech zařízeních. Z těchto důvodů bylo od vektorové grafiky upuštěno a byl zvolen

rastrový přístup.

6.4.2 Harvest

Do hry Harvest byla vytvořena grafika pro tlačítka reprezentující hlávky salátu a pro distraktor.

Hlávky salátu

Grafiku reprezentující salát bylo nutné vytvořit ve dvou variantách. První variantou je zdravá hlávka salátu (obr. 6.1) a druhou variantou je salát shnilý (obr. 6.2).



Obrázek 6.1: Vytvořená grafika pro zdravou hlávku salátu.



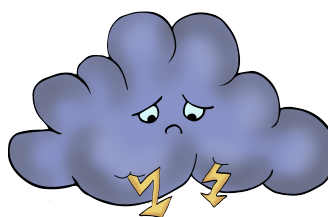
Obrázek 6.2: Vytvořená grafika pro shnilou hlávku salátu.

Distraktor

Distraktoru byl přiřazen vzhled mraku (obr. 6.3 a obr. 6.4).



Obrázek 6.3: Vytvořená grafika pro distraktor.



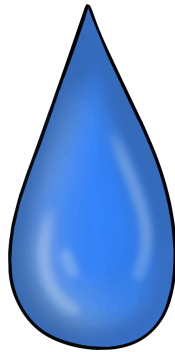
Obrázek 6.4: Vytvořená grafika pro náročnější distraktor.

6.4.3 TestTubes

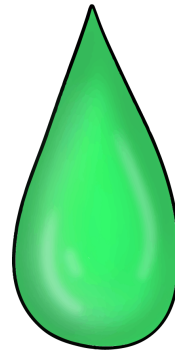
Do hry TestTubes byla vytvořena grafika pro zobrazované kapky a zkumavky.

Kapka

Zobrazovaná kapka byla vytvořena hned v sedmi barvách – červené, zelené (obr. 6.6), modré (obr. 6.5), šedé, oranžové, růžové a žluté.



Obrázek 6.5: Vytvořená grafika pro modrou kapku.



Obrázek 6.6: Vytvořená grafika pro zelenou kapku.

Zkumavka

Zkumavky byly vytvořeny v červené, zelené a modré variantě (obr. 6.7). Distraktor je reprezentován naprasklou zkumavkou (obr. 6.8).



Obrázek 6.7: Vytvořená grafika pro modrou zkumavku.



Obrázek 6.8: Vytvořená grafika pro prasklou modrou zkumavku.

6.4.4 Odměny

Ačkoliv byla grafika odměn vytvořena přesně podle návrhu, výsledek grafických prvků se mírně liší. Ukázka vzhledu některých stavů systému odměn je na obrázcích 6.9, 6.10, 6.11, 6.12, 6.13 a 6.14.



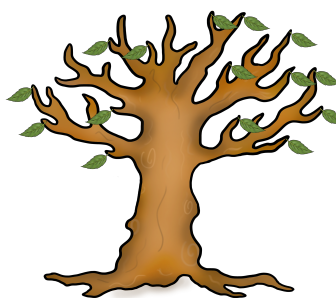
Obrázek 6.9: Vytvořená grafika pro měsíc.



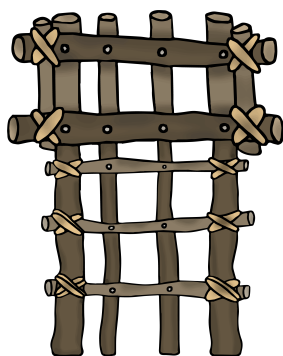
Obrázek 6.10: Vytvořená grafika pro minci.



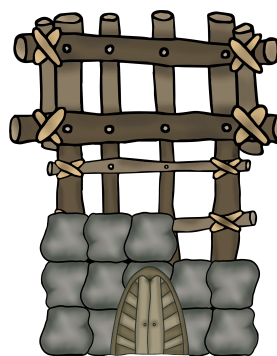
Obrázek 6.11: Vytvořená grafika pro počáteční stav odměny „strom“.



Obrázek 6.12: Vytvořená grafika pro jeden ze stavů odměny „strom“.



Obrázek 6.13: Vytvořená grafika pro počáteční stav odměny „věž“.



Obrázek 6.14: Vytvořená grafika pro jeden ze stavů odměny „věž“.

6.4.5 Přístup ke grafickým prvkům

Pro snadný přístup ke grafice byla vytvořena třída `SpritesContainer`, která obsahuje několik statických polí, do nichž se požadované assety přesunou přímo z vývojového prostředí Unity.

6.5 Systém odměn

Pro správné zobrazování odměn byla vytvořena funkce `PlayReward()`, která je použita ve všech vyvíjených hrách. Na začátku této funkce je nejprve ověřováno, která verze odměny byla nastavena vstupním parametrem. Podle zvolené varianty odměny je spuštěno další řízení zobrazování odměn.

Pokud se jedná o variantu měšce, je nejprve zjištěn dosud získaný počet mincí. Na základě parametru funkce `isPositive` je od dosud získaného počtu mincí jedna mince odečtena, nebo je k němu přičtena. V případě chybného herního kroku je spuštěna animace, při které se měšec na krátkou dobu zvětší a zčervená.

Pro varianty „strom“ a „věž“ pracuje algoritmus velmi podobně. Nejprve je na základě konkrétně zvolené varianty získána reference na konkrétní pole obsahující sprity všech stavů odměny. Parametr funkce `isPositive` ovlivňuje, zda se index v poli stavů herního objektu posune zpět anebo vpřed. Při tomto kroku je samozřejmě ošetřována situace, při které by se hodnota indexu dostala za hranice pole. V případě, že je parametr `isPositive` nastaven na negativní hodnotu, je spuštěna animace konkrétního herního objektu (u stromu opadávání listů, u věže borcení kamenných zdí). Na konci funkce je vzhled odměny nastaven na obrázek, jenž náleží vypočtenému indexu v poli obrázků.

```
1 private void PlayReward(bool isPositive){
2     ...
3     if(InputParams.Reward == 0){
4         Text moneybagText = ...;
5         int number = int.Parse(moneybagText.text.ToString());
6         number += isPositive ? 1 : (number > 0 ? -1 : 0);
7         moneybagText.text = number + "";
8         ...
9     } else {
10        Sprite[] sprites = InputParams.Reward == 1 ?
11            SpritesContainer.TreeRewardSprites :
12            SpritesContainer.TowerRewardSprites;
13
14        if(!isPositive)
15            rewardAnim.Play();
16
17        image = ...;
18        RewardIndex += isPositive ?
19            (RewardIndex == (sprites.Length - 1) ? 0 : 1) :
20            (RewardIndex == 0 ? 0 : -1);
21        image.sprite = sprites[RewardIndex];
22    }
23 }
```

6.6 Tutoriál

Pro každou hru byl vytvořen krátký návod jak hru hrát. Zobrazení tutoriálu ovlivňuje vstupní parametr úlohy.

Pro tutoriál byla vytvořena zcela nová scéna, která převzala některé funkce ze scény hlavní. Průběh tutoriálu je však neměnný a probíhá na základě jedné proměnné, jež určuje aktuální stav tutoriálu.

6.7 Lokalizace

Jedním z požadavků pro integraci her do systémů BrainIn je přeložení všech textů, které se ve hře objevují. Texty v tutoriálu, v nápovědě k ovládání a všechny popisky tlačítek apod. byly přeloženy do anglického a německého jazyka. Do stejných jazyků bylo nutné přeložit i celý obsah šablony každé hry.

6.8 JavaScriptové rozhraní

Skripty, které slouží především pro komunikaci mezi hrami a webovou aplikací BrainIn, nebylo nutné příliš upravovat. Nejmarkantnější změny se odehrály ve skriptech `input_data.js` (úprava rozdělení některých vstupních parametrů), `page_updater.js` (přidání cesty k mp3 souborům s nápovědou) a `unity_caller.js` (úprava cesty k sestavené hře).

6.9 Nápověda

Ačkoliv dříve vytvořené hry v systému BrainIn obsahují možnost nápovědy, po konzultaci s PhDr. Malou bylo rozhodnuto, že do těchto 4 her žádná nápověda přidána nebude. Jedním z důvodů byl fakt, že paní doktorka potřebuje vidět výsledky svých pacientů i v případě, že jim daná hra vůbec nepůjde. Použití nápovědy by v tomto případě postrádalo smysl, a pokud by byla hra pro pacienta opravdu příliš těžká a její obtížnost by znemožňovala její dohrání, má pacient možnost ji předčasně ukončit, nebo dané kolo přeskočit a informace o těchto krocích budou uloženy do výstupních statistik.

7 Testování

Testování vytvořených her bylo rozděleno do tří částí. V první části proběhlo ověření správné funkčnosti a korektnosti implementace pomocí programátorských nástrojů. V druhé části bylo realizováno beta testování průběhu her a kontrola správnosti výstupních statistik. V poslední části byly hry testovány na skutečných osobách trpících poruchou pozornosti s hyperaktivitou v psychiatrické nemocnici v Dobřanech.

7.1 Jednotkové testování

Pro kontrolu implementace byl použit testovací Unity framework *TestRunner* [20]. Bylo vytvořeno celkem 94 jednotkových testů, které ověřují hlavně kontrolu správnosti vstupních parametrů (funkce `SetInputParameters()`). Dále testují generaci herních dat a správnost některých pomocných funkcí z třídy `Tools`. Vytvořené testy rozhodně nestačí k úplnému pokrytí všech 4 her.

7.1.1 Odhalené chyby

Jednotkové testy odhalily několik chyb, které byly opraveny, některé z nich jsou vypsány v seznamu níže:

- U hry Harvest neprobíhala kontrola správnosti zadané hodnoty reprezentující pravděpodobnost výskytu náročnějšího distraktoru.
- U hry Grid nebyl kontrolován vstupní parametr, který určuje pohled mřížky.
- U hry Grid existovala konfigurace, při které by herní mřížka měla od začátku stejnou podobu jako mřížka vzorová.

7.2 Beta testování

Cílem tohoto testování bylo ověřit průběh hry a správnost výstupních statistik. Od každé hry byly vytvořeny tři úlohy – lehká, náročnější a obtížná varianta. Nakonfigurované hry byly odeslány celkem třem lidem ve věku od 20 do 22 let. Každý z nich si zahrál všechny konfigurace podle předem připraveného scénáře:

1. konfigurace (lehká) byla odehrána s maximálním soustředěním. Hráč se snažil dělat co nejméně chyb a soustředil se pouze na dohrání hry.
2. konfigurace (náročnější) byla přesným opakem první konfigurace. Hráč se snažil dělat chyby, působit roztěkaně, bezmyšlenkovitě používat některé funkce (vypnutí zvuků, přeskočení apod.).
3. konfigurace (obtížná) byla odehrána tak, že v polovině hry subjekt pochopil, že je pro něj hra příliš těžká a předčasně ji ukončil.

Každý z testerů zaznamenával své snažení pomocí programu OBS¹ a nahrávku ukládal do sdíleného online úložiště. K záznamu obrazovky testerů přiložili informaci o rychlosti jejich internetového připojení a dobu, po kterou se hra před spuštěním načítala.

7.2.1 Odhalené chyby

Po zhlédnutí všech nahrávek a výsledných statistik bylo odhaleno několik chyb, které byly vzápětí opraveny. Některé z nich jsou popsány v seznamu níže:

- Ve hrách, kde byl povolen tutoriál, se správně neukládal čas spuštění úlohy a chyběl ve výsledných statistikách.
- V tutoriálu u hry Harvest a TestTubes byla špatně nastavena oblast interakce kurzoru a bylo možné stisknout tlačítko při stisknutí jeho okolí.
- U některých her chybělo zaznamenávání odehraného času v jednotlivých kolech a do výstupních statistik byla zapisována výchozí hodnota.
- Ve výstupních statistikách chyběl výpis seedu.
- Animace odměny v případě chybného kroku na konci kola pokračovala i v kole následujícím.

¹Open Broadcaster Software je software používaný primárně pro nahrávání obrazovky počítače.

7.2.2 Rychlost načítání hry

V tabulce 7.1 jsou zobrazeny informace o rychlostech internetových připojení testerů a doba načítání jednotlivých her.

#	rychlost [Mbit/s]	Harvest [s]	TestTubes [s]	Calc [s]	Grid [s]
1	54.56	26	27	33	30
2	36.7	31	50	28	29
3	72.0	19	21	18	20

Tabulka 7.1: Rychlosti a doba načítání her u každého z testerů.

Velikost každé sestavené hry byla okolo 100MB a každá z her se spouštěla velmi dlouho. Doba spouštění závisí na rychlosti připojení, ne každý má však přístup k rychlému internetovému připojení. Z tohoto důvodu musela být velikost sestavených her snížena. Velikost těchto her tvoří hlavně grafické assety, u kterých byl zvolen jiný způsob komprimace (za cenu mírné, však nepozorovatelné změny kvality), velikost her byla zmenšena na hodnoty pohybující se okolo 25MB.

7.3 Testování v PN v Dobřanech

PhDr. Malá vytvořila dvě pracovní skupiny, jež měly za úkol vyzkoušet vytvořené hry na svých pacientech. Tyto pracovní skupiny si ve webové aplikaci BrainIn vytvořily několik konfigurací her, které si jejich pacienti zahráli a z nichž byly vyprodukovány výstupní statistiky. Výstupní statistiky byly terapeuti následně zanalyzovány.

Během dvou týdnů si implementované hry vyzkoušelo celkem 13 pacientů (chlapci i dívky ve věku od 5 do 13 let). Tento počet mohl být několikanásobně vyšší, ale z důvodu pandemie nemoci Covid-19 mohou terapie probíhat jen za přísných protiepidemických opatření. Proto se výrazně snižuje i počet pacientů přítomných na psychoterapii nebo počet psychoterapií jako takových. Terapeuti byli taktéž limitováni aktuálním počtem počítačů, které mohli dětem poskytnout.

Se všemi hrami, které byly v rámci této bakalářské práce vyvíjeny, byla PhDr. Malá velmi spokojená. Pozitivně ohodnotila nejen průběh hry, ale i grafické prvky. Analýza výstupních statistik, jež byly v průběhu tohoto testování vyprodukovány, odhalila, že výstupní hodnoty odpovídají zdravotním stavům testovaných pacientů.

8 Zhodnocení dosažených výsledků

V rámci této bakalářské práce byly vytvořeny celkem 4 parametrizované funkční hry, které pomáhají pacientům s ADHD při jejich léčbě. Průběh hry může být upraven vstupními parametry a každá hra produkuje výstupní statistiky, jež informují terapeuta o pacientově léčbě. Hry byly úspěšně integrovány do webové aplikace BrainIn.

Na rozdíl od kognitivních her popisovaných v sekci 3.3.2 disponují hry vytvořené v rámci této bakalářské práce několika žádoucími vlastnostmi:

- Integrací do systému BrainIn získaly hry schopnost parametrizace a je možné ovlivňovat jejich průběh.
- Každá hra je schopna na základě vstupních a výstupních parametrů kooperovat s terapeutem, který ji chce využívat.
- Využívat hry může kdokoliv, kdo má přístup do webové aplikace BrainIn, proto jsou vytvořené hry zcela nekomerční.

Dne 9. 4. 2021 proběhlo online školení s terapeuty z Fakultní nemocnice Plzeň a Centra pobytových a terénních sociálních služeb Zbůch. Při tomto školení jsem terapeutům prezentoval vytvořené hry a společně s několika dalšími kolegy vysvětloval základní orientaci ve webové aplikaci BrainIn. Na konci školení proběhl rychlý workshop, ve kterém si účastníci vyzkoušeli jednoduché úkoly (přiřazení pacienta, vytvoření úlohy atp.) a byly jim námi zodpovězeny všechny dotazy. Vzhledem k cíleným dotazům na mé hry a pozitivním reakcím si dovoluji usoudit, že budou vyvíjené hry používány nejen v psychiatrické nemocnici v Dobřanech, ale i na jiných pracovištích.

Vytvořená grafika působí velmi hezkým dojmem a navozuje společně s motivačními prvky ve hře příjemnou atmosféru. Více spokojen bych byl však s vektorovou reprezentací, a to nejen z důvodu menší paměťové náročnosti, ale i z důvodu minimální ztráty kvality při jakékoliv velikosti a rozlišení zařízení. Možná by bylo vhodné předělat hry do novější verze Unity a buď počkat na oficiální vydání balíčku pro správu vektorové grafiky, nebo zkusit výše zmíněnou experimentální verzi.

Hra Harvest byla v září 2020 prezentována veřejnosti na Dnech vědy a techniky v Plzni. V průběhu této interaktivní výstavy hra nasbírala několik

statistik a výstupních informací, které byly v rámci projektu BrainIn zaslány německým statistikům k dalšímu zkoumání a analýze.

V psychiatrické nemocnici v Dobřanech proběhl skutečný test těchto her na pacientech trpících ADHD. Výstupní hodnoty z tohoto testování odpovídají skutečným zdravotním stavům pacientů, a proto je velmi pravděpodobné, že hry budou moci sloužit pro terapeutické účely. Na základě dosažených výsledků testování a ohlasů pracovníků PN v Dobřanech budou vytvořené hry skutečně používány pro terapeutické sezení a domácí procvičování tamních pacientů.

9 Závěr

V rámci této bakalářské práce byla nejdříve prozkoumána teorie kognitivních funkcí a již existující kognitivní hry. Z těchto informací vznikl jakýsi základ pro návrh a implementaci vlastních her. Společně s PhDr. Karolínou Malou z PN v Dobřanech jsme probrali vstupní a výstupní parametry a na základě mých návrhů a jejích požadavků byl vytvořen návrh pro 4 hry sloužící k psychoterapii. Po seznámení se s webovou aplikací BrainIn a se všemi náležitostmi, které je nutné splnit pro integraci her do tohoto systému, byly implementovány 4 hry, jež trénují pozornost, exekutivní funkce, prostorovou orientaci, ale i jiné kognitivní funkce. V rámci implementace byly vytvořeny grafické prvky, které by měly pacienty motivovat při plnění herních cílů. Taktéž byl ke každé hře vytvořen krátký návod, který má pacientovi vysvětlit cíl a princip hry.

Hry byly otestovány nejen pomocí testovacích programátorských nástrojů, ale i několika testery v rámci beta testování. Jedna z her byla dokonce prezentována veřejnosti na interaktivní výstavě Dny vědy a techniky v Plzni.

Všechny požadavky PhDr. Malé byly splněny a hry byly otestovány i na skutečných pacientech a klientech psychiatrické nemocnice v Dobřanech. Taktéž byly splněny požadavky pro integraci her do systému BrainIn. Hry byly do BrainIn úspěšně začleněny a jsou aktivně využívány pro psychoterapii v psychiatrické nemocnici v Dobřanech.

V průběhu testování her na skutečných pacientech v PN v Dobřanech byly hry testovány i na klientech Centra pobytových a terénních sociálních služeb (CPTS) ve Zbůchu, a to zcela na základě iniciativy tamních terapeutů, kterým se hry velmi líbí a chtěli by je využívat při léčbě svých pacientů. Terapeuti z CPTS mi taktéž poskytli zpětnou vazbu a domluvili jsem se na budoucí pomoci při vytváření správných konfigurací úloh pro jejich pacienty.

Při realizaci této bakalářské práce jsem nabyl nové znalosti v oblasti vývoje kognitivních her a získal nové dovednosti v herním enginu Unity a programovacím jazyku C#.

Literatura

- [1] CHMELAŘOVÁ, D. Rehabilitace kognitivních funkcí. *Neurology for practice*. 2018, 17, 91, s. 62–69. ISSN 12131814. Dostupné z: <https://www.neurologiepropraxi.cz/artkey/neu-201691-0012.php>.
- [2] GEIG, M. *Sams Teach Yourself Unity Game Development in 24 Hours*. Pearson Education, 2014.
- [3] GOLDSTONE, W. *Unity game development essentials*. Packt Publishing Ltd, 2009.
- [4] HAVLÍČEK, O. *Pozornost v kognitivní psychologii* [online]. June 2014. Dostupné z: <http://www.ondrejhavlicek.com/2014/06/pozornost-v-kognitivni-psychologii/>.
- [5] HOCKING, J. *Unity in action: Multiplatform game development in C# with Unity 5*. Manning Publications Shelter Island, NY, 2015.
- [6] KLENEROVÁ, V. – HYNIE, S. Paměť a její poruchy. *Čs. fyziologie*. 2010, 59, s. 15–20.
- [7] KLUCKÁ, J. – VOLFOVÁ, P. *Kognitivní trénink v praxi*. Grada Publishing, a.s., 2009. ISBN 978-80-247-5580-9.
- [8] LENGSELDOVÁ, D. Syndrom hyperaktivity ADHD v předškolním a mladším školním věku. Bachelor's Thesis, Pedagogická fakulta, Univerzita Palackého v Olomouci, 2015.
- [9] LEZAK, M. D. et al. *Neuropsychological assessment*. Oxford University Press, USA, 2004.
- [10] MAĎA, P. – FONTANA, J. *Neurotransmisní systémy* [online]. n.d. Dostupné z: <http://fb.lt.cz/skripta/regulacni-mechanismy-2-nervova-regulace/5-neurotransmisni-systemy/>.
- [11] MATSUMOTO, D. – XU, H. – TAKENO, J. Simulation of the Cognitive Process in Looking at Rubin's Vase. *Procedia Computer Science*. 01 2018, 123, s. 265–270. doi: 10.1016/j.procs.2018.01.041.
- [12] MCLEOD, S. A. Multi store model of memory. *Simply Psychology*. February 2017. Dostupné z: <https://www.simplypsychology.org/multi-store.html>.

- [13] OBEREIGNERŮ, R. Exekutivní funkce. *Kulišťák Petr a kolektiv. Klinická neuropsychologie v praxi*. 2017.
- [14] PACLT, I. – PTÁČEK, R. – FLORIÁN, J. *Hyperaktivita*. Vzdělávací Institut ochrany dětí, 2006.
- [15] PLACEROVÁ, V. – PUŽEJOVÁ, Y. *Psychologie*. Střední zdravotnická škola a Vyšší odborná škola zdravotnická České Budějovice, 2016. ISBN 978-80-88058-88-5.
- [16] PLAČKOVÁ, M. – OTHERS. Poruchy pozornosti a poruchy pozornosti s hyperaktivitou. Bachelor's Thesis, Faculty of Science, Humanities and Education, Technical University of Liberec, 2015.
- [17] RABOCH, J. Kognitivní funkce, stárnutí a stravovací návyky. *Česká a Slovenská psychiatrie*, č. 2010, 2.
- [18] SKALA, P. Webová aplikace pro správu neurorehabilitačních programů. Master's thesis, Faculty of Applied Sciences, University of West Bohemia, 2019.
- [19] SMITH, B. *Beginning JSON*. Apress, 2015.
- [20] *Test Framework* [online]. Unity, 2019. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.test-framework@1.1/manual/index.html>.
- [21] VACKOVÁ, M. – OTHERS. Poruchy pozornosti s hyperaktivitou. Master's thesis, Faculty of Science, Humanities and Education, Technical University of Liberec, 2015.
- [22] VÁGNEROVÁ, M. *Obecná psychologie: důležitější aspekty lidské psychiky a jejich orgánový základ*. Charles University in Prague, Karolinum Press, 2017.
- [23] VÁGNEROVÁ, M. *Úvod do psychologie*. Charles University in Prague, Karolinum Press, 2002. ISBN 80-246-0015-3.
- [24] VÁLKOVÁ, L. *Rehabilitace kognitivních funkcí v ošetrovatelské praxi*. Grada Publishing, a.s., 2015. ISBN 978-80-247-5982-1.

Přehled zkratk

ADHD Attention Deficit Hyperactivity Disorder

CPTS Centrum pobytových a terénních sociálních služeb

JSON JavaScript Object Notation

KIV Katedra informatiky a výpočetní techniky

OBS Open Broadcast Software

PC personal computer

PN psychiatrická nemocnice

Seznam obrázků

2.1	BrainIn - šablona	11
2.2	BrainIn - úloha	12
2.3	BrainIn - balík úloh	13
2.4	BrainIn - překlady	13
3.1	Komponenty učení a paměti. [6]	14
3.2	Atkinson-Shiffrinův paměťový model	16
3.3	Rubinův obrazec	17
3.4	Rubikova kostka	19
4.1	Stav stromu na začátku hry.	26
4.2	Stav stromu po několika úspěšných herních tazích.	26
4.3	Stav hradní věže na začátku hry.	26
4.4	Stav věže po několika úspěšných herních tazích.	26
4.5	Vzhled měšce.	27
4.6	Vzhled mince.	27
4.7	Návrh herní scény hry Harvest.	28
4.8	Návrh herní scény hry TestTubes.	30
4.9	Návrh herní scény hry Calc.	31
4.10	Návrh herní scény hry Grid.	33
5.1	Diagram komunikace jednotlivých částí systému BrainIn.	35
6.1	Vytvořená grafika pro zdravou hlávku salátu.	45
6.2	Vytvořená grafika pro shnilou hlávku salátu.	45
6.3	Vytvořená grafika pro distraktor.	45
6.4	Vytvořená grafika pro náročnější distraktor.	45
6.5	Vytvořená grafika pro modrou kapku.	46
6.6	Vytvořená grafika pro zelenou kapku.	46
6.7	Vytvořená grafika pro modrou zkumavku.	46
6.8	Vytvořená grafika pro prasklou modrou zkumavku.	46
6.9	Vytvořená grafika pro měsíc.	47
6.10	Vytvořená grafika pro minci.	47
6.11	Vytvořená grafika pro počáteční stav odměny „strom“.	47
6.12	Vytvořená grafika pro jeden ze stavů odměny „strom“.	47
6.13	Vytvořená grafika pro počáteční stav odměny „věž“.	47
6.14	Vytvořená grafika pro jeden ze stavů odměny „věž“.	47

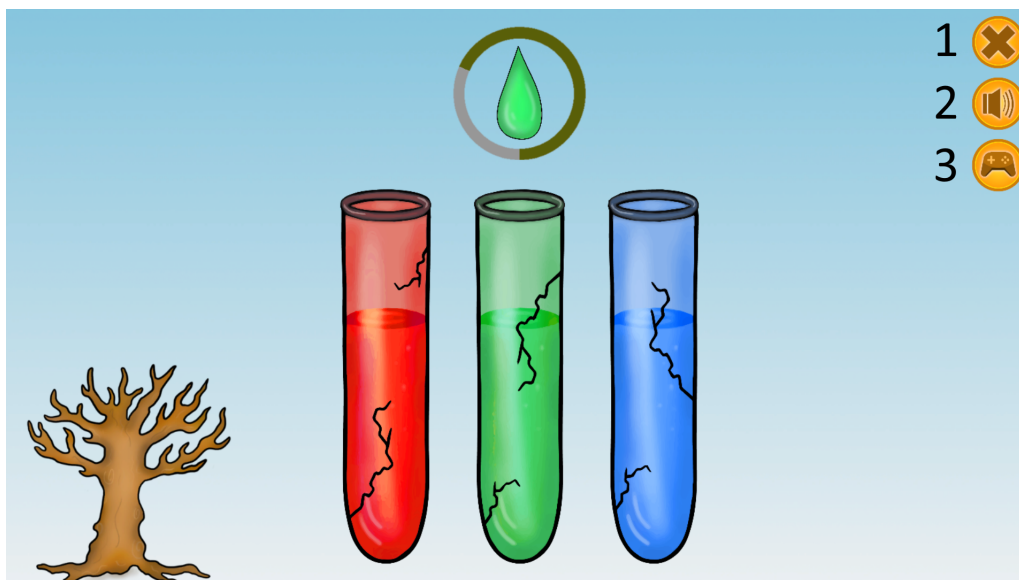
10.1	Funkce tlačítek	61
10.2	Ukázka hry Harvest.	62
10.3	Ukázka hry TestTubes.	62
10.4	Ukázka hry Calc.	63
10.5	Ukázka hry Grid.	63

10 Uživatelská dokumentace

10.1 Funkce tlačítek

Po spuštění hry je hráč přesunut buď do tutoriálu, anebo přímo do hry (záleží na konkrétní konfiguraci). Každá hra má několik stejných tlačítek. Jejich vzhled je na obrázku 10.1 a jejich funkce jsou následující:

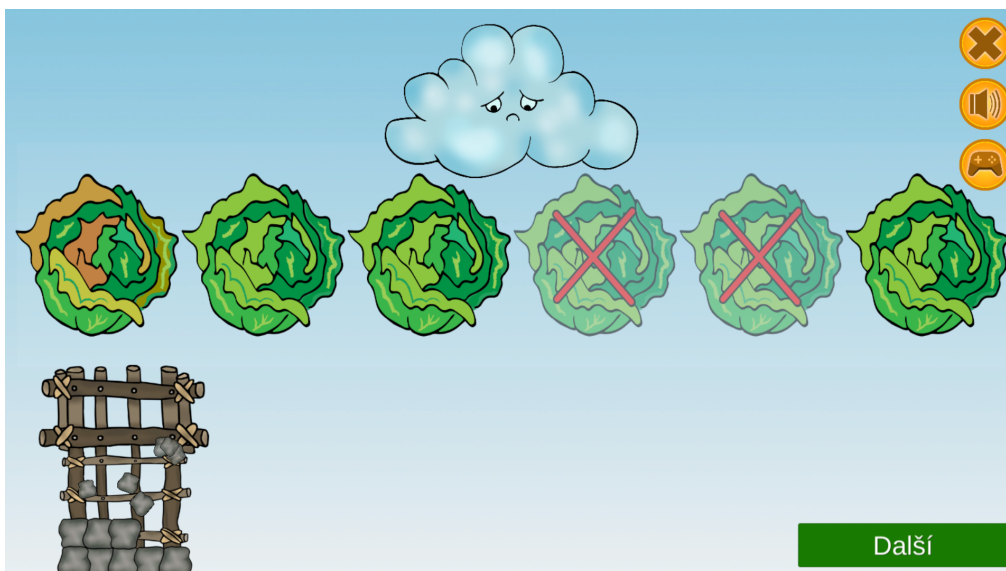
- Tlačítko 1 předčasně ukončí úlohu.
- Tlačítko 2 vypíná a zapíná zvukové efekty.
- Tlačítko 3 zobrazuje nápovědu k ovládání (a přehrává namluvenou nápovědu).



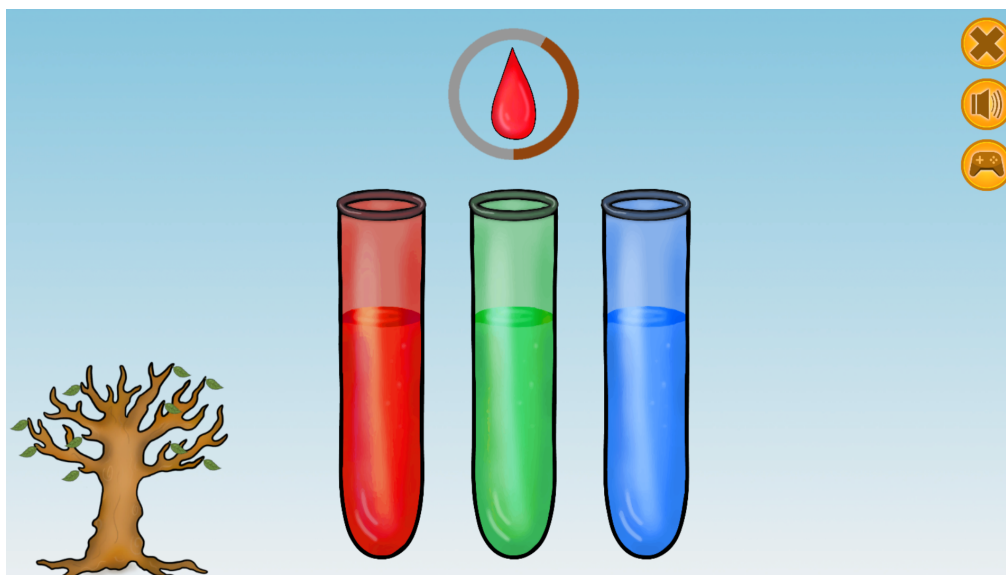
Obrázek 10.1: Obrazovka při hraní hry.

10.2 Ukázka her

Na obrázcích 10.2, 10.3, 10.4 a 10.5 jsou zobrazeny snímky obrazovky ze všech 4 her.



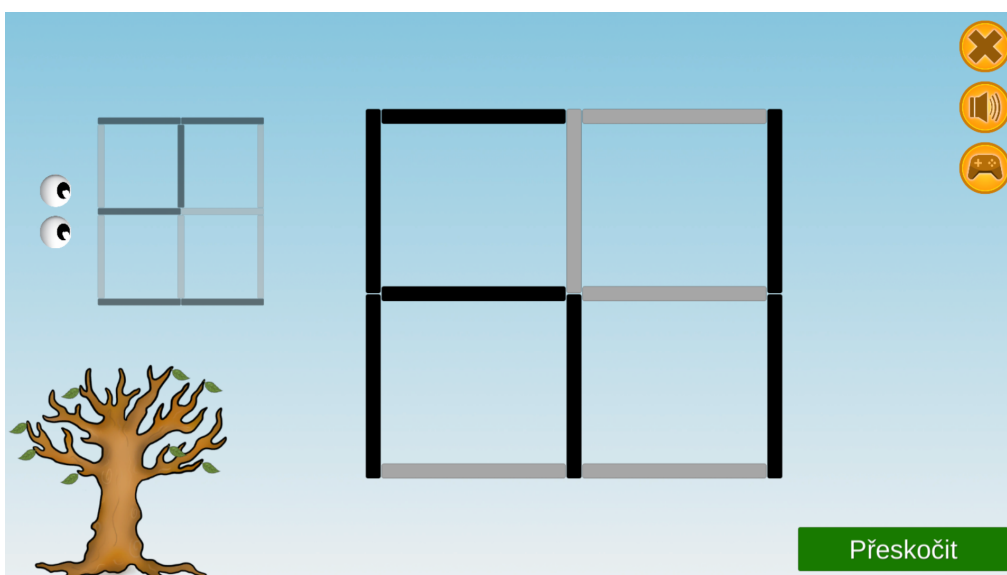
Obrázek 10.2: Ukázka hry Harvest.



Obrázek 10.3: Ukázka hry TestTubes.



Obrázek 10.4: Ukázka hry Calc.



Obrázek 10.5: Ukázka hry Grid.

11 Návod k sestavení

Návod na sestavení hry pro operační systém Windows.

1. Stáhněte a nainstalujte Unity ve verzi 2018.4.7f1. Konkrétní verze se instalují přes tzv. **Unity Hub**.
2. Po spuštění Unity klikněte v záložce **Projects** na tlačítko **Open** a zvolte cestu ke kořenové složce projektu. Projekt by se měl vzápětí otevřít v vývojovém prostředí Unity.
3. Dále zvolte **File** → **Build Settings**.
4. Ujistěte se, že volba **Platform** je nastavena na **WebGL**.
5. Do kolonky **Scenes In Build** přetáhněte všechny scény z adresáře **Scenes**.
6. Klikněte na tlačítko **Build** a vyberte cílovou složku.