

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Wingender** Jméno: **Antonín** Osobní číslo: **466328**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

3D Webová aplikace pro demonstraci různých úhlů pohledu na řešení problémů

Název bakalářské práce anglicky:

3D Web application for a demonstration of different problem-solving perspectives

Pokyny pro vypracování:

Vytvořte aplikaci, která bude sloužit pro účely demonstrace využívání různých úhlů pohledu na řešení problému. Cílem je především ukázat, že zjednodušený (omezený) pohled na problematiku ne vždy vede k nalezení řešení, případně správného (optimálního) řešení. Vytvořená aplikace bude sloužit pro výukové účely. Postupujte následujícím způsobem:

- 1) Analyzujte a definujte problematiku, nazývanou „Tunnel vision“ a na příkladech demonstруйте, jaké dopady omezené vnímání reality přináší.
- 2) Proveďte rešerši existujících úloh, které problematiku „Tunnel vision“ demonstруют a vyberte úlohy, které by bylo možné implementovat pomocí aplikace.
- 3) Navrhněte a vytvořte aplikaci, která prostřednictvím různých 3D pohledů bude implementovat vybrané úlohy. Aplikaci koncipujte jako výukový materiál ve formě otázek a odpovědí na ně, přičemž při hledání odpovědí bude uživatel hledat optimální úhel pohledu.
- 4) Vytvořenou aplikaci uživatelsky otestujte na vybrané skupině uživatelů a vyhodnoťte její přínosy.

Seznam doporučené literatury:

- [1] POKORNÝ, Pavel. Blender - naučte se 3D grafiku. BEN-Technická literatura. 2009. ISBN 80-7300-244-2.
- [2] ŽÁRA J., BENEŠ B., SOCHOR J., FELKEL P. Moderní počítačová grafika. 2. vyd. Brno: Computer Press, 2005. 628 s. ISBN 80-251-0454-0.
- [3] RÜSCHMEYER, Georg. Kniha optických klamů. Omega. 2017. ISBN: 978-80-739-0820-1.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Ph.D., katedra ekonomiky, manažerství a humanitních věd FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **04.02.2019**

Termín odevzdání bakalářské práce: **15.08.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Pavel Náplava, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

_____ Datum převzetí zadání

_____ Podpis studenta

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická

3D Webová aplikace pro demonstraci aplikace jiných úhlů pohledu

Antonín Wingender

Vedoucí: Ing. Pavel Náplava, PhD.

Studijní program: Softwarové inženýrství a technologie

Srpen 2019

Poděkování

Děkuji panu doktoru inženýru Náplavovi za poznatky a odezvu poskytované v průběhu vytváření bakalářské práce, a děkuji lidem co se zúčastnili uživatelského testování.

Prohlášení

Tuto práci jsem vytvořil sám, pouze za pomoci zdrojů uvedených v bibliografii, internetových zdrojů a znalostí získaných během svého studia. Dalším zdrojem informací mi byly poznatky získané od Ing. Pavla Náplavy PhD.

V Praze, dne 23. května 2019

Abstrakt

Tunnel vision a podobné úkazy jsou problémem u mnohých lidí. Největší dopad však mají na lidi v managementu a ty, jejichž rozhodnutí mají velký dopad na lidi kolem nich. Mít v takovém případě omezený pohled na věc, nebo neschopnost podívat se na věci z jiného úhlu, může mít dopad nejen na obraty největších světových firem, ale někdy bývájí i rozdílem mezi životem a smrtí lidí.

Cílem této bakalářské práce je upozornit na psychologický úkaz zvaný tunnel vision. Toho chci docílit pomocí tzv. „Aha” momentu, tedy náhlého nalezení řešení tam, kde předtím nemohlo být nalezeno. Pokud si při nějakém obtížném životním rozhodnutí lidé vzpomenou, že zkoušeli jednou jeden program, a v tom věci nebyly úplně jak se na první pohled zdály, tak mohou tuto práci považovat za úspěšnou.

Vytvořená aplikace bude sestávat ze šesti úloh, např. bludiště, jenž je možné vyřešit v trojrozměrném prostoru, nebo zdánlivě nesouvisejícími kostkami, které při správném úhlu pohledu vytvoří jasný nápis. Tyto úlohy jsou na první pohled často zdánlivě neřešitelné – ale při chytrém využívání 3D prostoru mohou uživatelé najít jejich jednoduché řešení. Náhlá jednoduchost tohoto řešení bude právě to, co onen „Aha” moment bude zprostředkovávat.

Klíčová slova: tunnel vision, úhel pohledu, 3D, webová aplikace

Vedoucí: Ing. Pavel Náplava, PhD.
Technická 2, Fakulta elektrotechnická,
Praha 6

Abstract

Tunnel vision and similar phenomena are a problem area for many people, but they have the greatest impact on high management and those whose decisions have a big impact on people around them. In such a case, having a limited view of things or not being able to look at things from a different angle can have an impact not only on the turnovers of the world's largest companies, but sometimes they are the difference between life and death.

The aim of this project is to inform readers about a psychological phenomenon called tunnel vision. I want to achieve that using what is called an „Aha” effect, or a sudden realization of solution where it could not be realized before. For example, if they think, during a difficult life decision that they once tried one app, where things were not quite as they seemed at first glance, I could consider the project successful.

The project will consist of a number of tasks, such as a maze that can be solved in a three-dimensional space, or seemingly unrelated cubes that create a clear inscription at the right angle. These tasks will often be pointless, insoluble - but users can find a simple solution when using 3D space smartly. The simplicity of this solution will be just what the „Aha” moment will convey.

Keywords: tunnel vision, angle of view, 3D, web application

Title translation: 3D Web app for demonstration of use of other angles of view

Obsah

1 Úvod	1	3.3 „Aha” efekt a tunnel vision	13
2 Tunnel vision a její důsledky	3	3.4 Závěr	14
2.1 Definice pojmu Tunnel vision	3	4 Definice problému a analýza existujících řešení	15
2.2 Kognitivní předsudky tvořící tunnel vision	4	4.1 Definice problému	15
2.2.1 Konfirmační zkreslení	4	4.2 Analýza existujících řešení	15
2.2.2 Vytrvalost víry	5	4.2.1 Výzkum tunnel vision	16
2.2.3 Zaujatost zpětného pohledu . .	5	4.3 Existující kurzy zaměřující se na tunnel vision	17
2.2.4 Zaujatost z výsledku	6	4.3.1 Mindtools.com	17
2.2.5 Kotvení	6	4.3.2 iMindMap	17
2.3 Příklady důsledků kognitivních předsudků	6	4.3.3 Coursera for business	18
2.4 Příklady negativních důsledků tunnel vision	7	4.3.4 Shrnutí	18
2.4.1 General Motors a nefunkční airbagy	7	4.4 Inspirace hlavolamy	18
2.4.2 Microsoft a jeho dlouhodobá vize	8	4.4.1 Amesova místnost	19
2.4.3 Kodak a neochota se přizpůsobit digitálnímu věku	9	4.4.2 Penroseovy schody	19
2.4.4 Samsung a neuvážená sázka na auta	9	4.5 Shrnutí kapitoly	20
2.5 Pozitivní důsledky tunnel vision	10	5 Návrh úloh	23
2.6 Shrnutí kapitoly	10	5.1 Popis a cíl jednotlivých úloh . . .	23
3 „Aha” moment	11	5.1.1 Bludiště	23
3.1 Definice „Aha” momentu	11	5.1.2 Nápis Think	25
3.2 „Aha” efekt v praxi	12	5.1.3 Rovná plocha	26
		5.1.4 Pro les nevidím stromy	27
		5.1.5 Zase o tom příliš nepřemýšlejme	29
		5.1.6 Postupka z karet	30

5.2 Pořadí úloh	32	8.2.4 Statistiky všech uživatelů ...	49
5.3 Shrnutí kapitoly	33	8.3 Zpracování jednotlivých úloh ...	50
6 Návrh řešení	35	8.3.1 Obecný postup při vytváření úloh	50
6.1 Principy fungování aplikace	35	8.3.2 Krátká zmínka o náhodné sekvenci úloh	51
6.2 Funkční a nefunkční požadavky aplikace	36	8.3.3 Bludiště	51
6.2.1 Funkční požadavky	36	8.3.4 Nápis Think	52
6.2.2 Nefunkční požadavky	37	8.3.5 Rovná plocha	52
6.3 Shrnutí kapitoly	37	8.3.6 Pro les nevidím stromy	53
7 Použité technologie a nástroje	39	8.3.7 Zase o tom příliš nepřemýšlejme	54
7.0.1 Výběr JavaScript Frameworku	39	8.3.8 Pokerové karty	54
7.0.2 Výběr nástroje pro vytvoření 3D prostředí	40	8.4 Analytická část aplikace	55
7.0.3 Výběr nástroje pro vytvoření 3D modelů	42	8.4.1 Propojení s databází Google Firebase	55
7.0.4 Výběr databáze pro ukládání analytických dat	42	8.4.2 Globální statistiky	56
7.0.5 Model nasazení	42	8.4.3 Získávání informace o „Aha” efektu	56
7.0.6 Proč jsem nepoužil herní engine?	43	8.5 Shrnutí kapitoly	57
7.1 Shrnutí kapitoly	44	9 Uživatelské testování	59
8 Implementace aplikace	45	9.1 Profil uživatele testujícího aplikaci	59
8.1 Adresářová struktura aplikace ..	45	9.2 Informace z uživatelského testování	59
8.2 Zpracování uživatelského rozhraní	46	9.2.1 Graf průměrné doby splnění úlohy	60
8.2.1 3D okno	47	9.2.2 Graf „podání se”	60
8.2.2 Popup okénka a tipy	47		
8.2.3 Uživatelské statistiky	48		

9.2.3 Grafy časů splnění úloh, závislých na pořadí	61
9.2.4 Vyvolání „Aha” momentu . . .	62
9.3 Co vyplývá z uživatelského testování.	62
10 Další možný vývoj aplikace	65
10.1 Úlohy	65
11 Závěr	67
Literatura	69
A Seznam zkratk	75
B Slovník pojmů	77

Obrázky

2.1 Znázornění tunnel vision	4	9.2 Graf zobrazující celkový počet „podání se”.	61
3.1 Králík nebo kachna?[58]	12	9.3 Graf zobrazující průměrný čas řešení úlohy 2 v závislosti na pořadí. 62	
4.1 Amesova místnost. [55]	19		
4.2 Amesova místnost - princip. [56]	20		
4.3 Penroseovy schody [57]	20		
5.1 Cvičení 1 - Bludiště	25		
5.2 Cvičení 2 - Nápis Think	26		
5.3 Cvičení 3 - Rovná plocha	27		
5.4 Cvičení 4 - Pro les nevidím stromy	29		
5.5 Cvičení 5 - Zase o tom příliš nepřemýšlejme	30		
5.6 Cvičení 6 - Postupka z karet	32		
7.1 Model nasazení	43		
8.1 Adresářová struktura aplikace.	46		
8.2 Hlavní okno aplikace	47		
8.3 Popup okno při spuštění aplikace	48		
8.4 Uživatelské statistiky	49		
8.5 Statistiky všech uživatelů - úvodní stránka.	50		
8.6 Vyskakovací okno po ukončení série úloh.	57		
9.1 Graf průměrné doby splnění úlohy.	60		



Kapitola 1

Úvod

Předmětem a cílem této bakalářské práce je vytvoření aplikace, která bude sloužit pro účely demonstrace využívání různých úhlů pohledu na řešení problému. Důležité je uživateli ukázat, že jeden úhel pohledu na problematiku často vede k negativním důsledkům pro jeho vlastní efektivitu. Aplikace bude postavená na principu vyvolání „Aha” momentu, tedy náhlého uvědomění si řešení problému, kde se původně jevil neřešitelný.

Bakalářská práce má následující strukturu:

V druhé kapitole (první kapitola je tento úvod) se věnuje „Tunnel vision”, tedy tendenci člověka zaměřit se na omezenou množinu informací a ignorovat širší kontext. Uvede také několik příkladů, až jak negativní mohou být důsledky tohoto každodenního psychologického úkazu.

Následně ve třetí kapitole definuje problém, který se aplikací snažím vyřešit. Popisuje vyhledávání jiných existujících řešení a důvody, proč jsem pro aplikaci zvolil tuto podobu.

Ve čtvrté kapitole definuje návrh aplikace, podobu jednotlivých úloh a shrne tento návrh do funkčních a nefunkčních požadavků.

V páté kapitole popíše průběh implementace aplikace, adresářovou strukturu aplikace, postup vytváření úloh a jejich propojení s webovým prostředím. Také zmíní analytickou část aplikace a jak byla implementována.

Výsledkem bakalářské práce je aplikace sestávající se z šesti úloh. Tyto úlohy využívají 3D prostor pro zprostředkování výše zmíněného „Aha” momentu. Ten je zprostředkován tak, že úlohy na první pohled vypadají neřešitelně, ale při jednoduché manipulaci s pohledem se uživateli okamžitě zjeví jejich skutečné řešení.

Kapitola 2

Tunnel vision a její důsledky

V této kapitole definuji pojem Tunnel vision a všechny psychologické úkazy, které tunnel vision vytvářejí. Definuji také obecné negativní důsledky tohoto úkazu a předkládám několik příkladů, jak se tunnel vision negativně projevil v historii několika známých světových firem.

2.1 Definice pojmu Tunnel vision

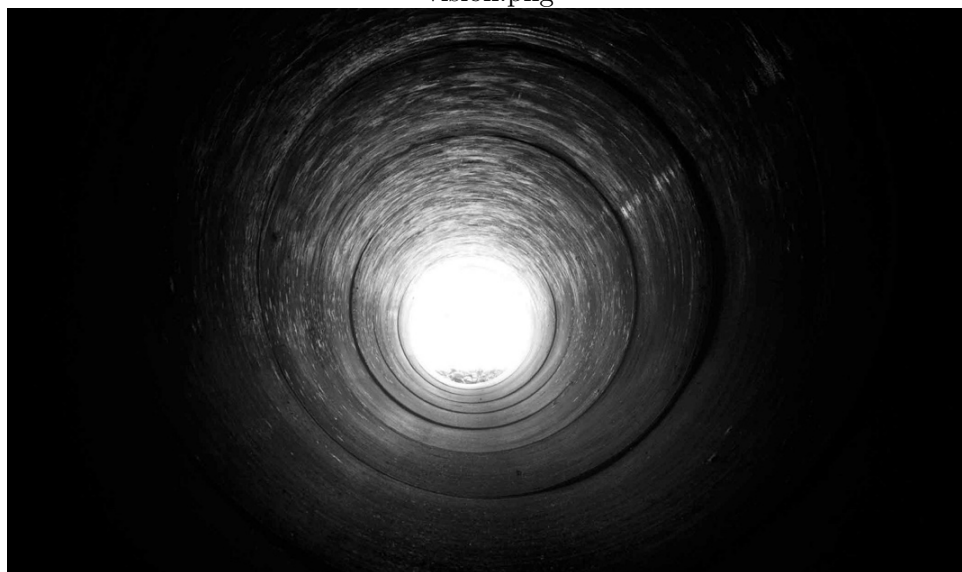
Tunnel vision bych rád přirovnal k něčemu, co se vyskytuje v umění již od starých řeckých tragédií. Podobný koncept se často v umění prezentuje jako „arogance“. Ačkoli diváci jasně vidí, že akce postavy povedou ke špatným důsledkům, samotná postava je slepá vůči všem náznakům i radám v ději a sama se často svede do záhuby.

Slovník Mirriam-Webster definuje Tunnel vision jako „extreme narrowness of viewpoint„[1], tedy extrémní úzkost úhlu pohledu, což je užitečná definice pro vytvoření si základní představy o tomto úkazu. Název tohoto pojmu je odvozený od neschopnosti člověka, který vyhlíží z tunelu, vidět celou situaci díky úzkosti tunelu. (Obrázek 2.1)

Dle jedné studie se tunnel vision častěji děje u přirozeně více nerozhodných lidí, kde se člověk spíše zaměří na jednu z jeho mnoha možností a nepřemýšlí o dalších možnostech, nebo o nich není ochotný získat více informací. V tomto případě tunnel vision může fungovat i jako protiváha k touze člověka získat co nejvíce možných informací. [35]

Tunnel vision je tedy definována vědeckou komunitou jako neschopnost zaznamenat či ignorovat důsledky rozhodnutí, které se neshodují s původní vizí člověka, jenž je vykonal. Tunnel vision postihuje lidi různých typů, od absolventů MBA, náhodných uživatelů internetu či manažery na vysokých

vision.png



Obrázek 2.1: Znárodnění tunnel vision

pozicích ve světových firmách.[2]

■ 2.2 Kognitivní předsudky tvořící tunnel vision

Lze říci, že tunnel vision je způsobena iracionálními, ale obvyklými předsudky lidské mysli. Díky těmto předsudkům je pro člověka obtížnější racionálně se rozhodovat, např. v případech, kdy musí pracovat s výrazně nedostatečnými informacemi. Tyto předsudky nazýváme kognitivními předsudky. V následující kapitole proberu hlavní kognitivní předsudky, které jsou často pod pojem tunnel vision zahrnovány.

■ 2.2.1 Konfirmační zkreslení

Konfirmační zkreslení je tendence vyhledávat nebo interpretovat důkazy způsoby, které upevňují již existující předpoklady, závěry či očekávání.[3]

Hlavní projev konfirmačního zkreslení je, že lidé mají díky tomuto kognitivnímu předsudku tendenci hledat informace, které podporují jejich hypotézy a vyhýbat se informacím, které je naopak nepodporují. [4] Jedna studie v publikaci Economics Letters [5] předkládá, že tento způsob ve většině případů nepodporuje hledání pravdy, a ani „nezíská tak přesvědčivé důkazy, než kdyby byla použita strategie vytváření situací které by explicitně ukázaly hypotézu jako špatnou, a nebýt úspěšný v jejich vytvoření“. Stejná studie také říká,

že „potvrzující důkazy“ vyhledávané pouze pro potvrzení předem určené hypotézy bývají „do určité míry falešné“.

Lidé také mívají tendenci interpretovat již existující vzpomínky skrze své vlastní předsudky a preferují situace, které potvrdí jejich současnou hypotézu, a vytěsňují takové situace, které by tuto hypotézu vyvrátily. Bylo dokázáno, že lidé nejsou schopni vyhodnotit kvalitu důkazů nezávisle od jejich vlastních názorů a mají obecnou tendenci dávat větší váhu informacím, které potvrzují jejich vlastní hypotézu, spíše než těm, které jí vyvracejí. [6] Je to jako kdyby se člověk ptal u potvrzujících důkazů „Mohu tomuto věřit?“ ale u důkazů vyvracejících „Musím tomuto věřit?“.

Dobrý příklad konfirmačního zkreslení můžeme pozorovat například u politických kampaní, kde lidé častěji vyhledávají informace, které podporují kandidáta, kterého si už vybrali předem. Více o tomto v podkapitole 2.3.

■ 2.2.2 Vytrvalost víry

Max Planck, světoznámý fyzik, kdysi řekl, že „nová pravda ve vědě nezvítězí přesvědčením jejích odpůrců, ale protože všichni její odpůrci časem zemřou, a nahradí je nová generace, která už je s ní seznámena.“ [7] Tato věta dobře ilustruje dopad „vytrvalosti víry“ (můžeme jí také někdy nazvat tvrdohlavost) na vědeckou a další komunitu. Například Albert Einstein, světoznámý fyzik, byl ten, kdo vyvrátil zažitý názor o vesmíru a vytvořil obecnou teorii relativity. V pozdějším věku však více a více pochyboval o nových objevech jako kvantová teorie. Pohlížel na ně ze zaujatého, negativního pohledu, což se odráží i na jeho slavném výroku „Bůh nehraje v kostky“. Kvantová teorie je nyní široce respektovaná v kruzích fyziky. [8]

Vytrvalost víry je přirozená tendence být odolný vůči změně názoru i přesto, že jsou člověku prezentovány jasné důkazy, které onen názor vyvrací. Lidé častěji zpochybňují původ nových důkazů, které vyvracejí jejich původní přesvědčení, než validitu jejich vlastního přesvědčení. [9] Bylo například dokázáno, že lidé si často vytvoří o novém člověku obrázek, který hrubě reprezentuje povahu člověka a vytrvají v tomto názoru, i když se povaha člověka ukáže být odlišná.

■ 2.2.3 Zaujatost zpětného pohledu

Studie (např. jedna z publikace Journal of Experimental Psychology [11]) ukázaly, že lidé zpětně považují výsledek za nevyhnutelný, nebo alespoň více předpovědatelný, než se jim původně jevil. Tento kognitivní předsudek se dá přirovnat k frázi „věděl jsem to celou dobu“. Lidé přeceňují svojí schopnost předpovědat události, protože se nakonec staly, ať už skutečně předpověděli událost na základě správných kritérií nebo ne.

Zaujatost zpětného pohledu může činit problémy v odsuzování nevinných lidí. Například pokud je obviněný odsouzen, může to ovlivnit rozhodnutí policie, která následně může špatné indikátory identifikovat jako správné a ovlivnit tak další vyšetřování. [12]

■ 2.2.4 Zaujatost z výsledku

Zaujatost z výsledku je velice podobná zaujatosti zpětného pohledu. Lidé projektují současné vědomosti - výsledky svých rozhodnutí - do minulosti, aniž by si byli vědomi, že výsledek ovlivnil jejich vnímání minulosti. [13]

Zaujatost z výsledku se liší od zaujatosti zpětného pohledu tím, že zaujatost z výsledku ovlivňuje vnímání, jestli určité rozhodnutí bylo nebo nebylo správné, ne jestli bylo jednoduše předpověditelné. Například vezměme příklad v kontextu lékařství. Pokud pacient zemřel při chirurgickém zákroku, tak doktoři častěji vnímají zákrok jako špatné rozhodnutí, ačkoli to mohlo být nejlepší rozhodnutí v danou chvíli. To může negativně ovlivnit další rozhodnutí v budoucnu.

■ 2.2.5 Kotvení

Kognitivní předsudek zaměřit se na první informace, které o dané problematice získáme, se v psychologii nazývá "**kotvení**" (v angličtině *anchoring*). [14] Díky tomuto efektu dáváme menší váhu případným dalším informacím, které by mohly situaci lépe objasnit. Ačkoli jedinci s vyššími kognitivními schopnostmi dokáží lépe odolat této tendenci, stále jí jsou ovlivňováni.[21]

■ 2.3 Příklady důsledků kognitivních předsudků

Předchozí kapitolu bych rád ilustroval příklady dopadů těchto kognitivních předsudků a tunnel vision jako celku. Nejprve popíšu obecné dopady tunnel vision. V následující kapitole pak popíšu skutečné příklady ze světa byznysu, u kterých se dá předpokládat, že tunnel vision měla velký negativní dopad.

Obvyklé obecné důsledky výše popsaných kognitivních předsudků bych popsal jako následující:

1. **Potřeba vyhrát:** Můžeme se například podívat do kasin a na hráče automatů pro dobrý příklad toho, jak se tunnel vision může dobře zneužít pro vyvolání potřeby vyhrát. Automaty často využívají fenomén „téměř

jsem vyhrál” aby hráči hráli dál, nebo **zaujatost zpětného pohledu** (viz kap. 2.2.3) pro uchopení jejich hráčů (když vyhrajeme, připadá nám, že to bylo vždycky pravděpodobné, že vyhrajeme, ačkoli nebylo).[15]

2. **Rozhodování se na základě špatných informací:** Nedostatečné množství informací může například způsobit odsouzení nevinného člověka, či neinformovaného voliče u voleb. Toto je často provázáno s tzv. **kotvením** (viz kapitola 2.2.5), tedy tendencí se zaměřit na první informaci. V tomto případě má vliv i **vytrvalost víry** (viz kap. 2.2.2), díky níž může obviněný člověk být nesprávně odsouzen díky tomu, že zpočátku vypadal jako vinný. [16]
3. **Náborové bubliny: Konfirmační zkreslení** (viz kapitola 2.2.1) nám může aktivně bránit pohlížet nezáujatě na názory, které sami nedržíme. Naopak, tato kognitivní zaujatost umožňuje vnímat situace tak, aby dávala smysl v kontextu našich zažitých přesvědčení. Toto může např. vést k vytvoření tzv. „náborových bublin”, kde člověk není vystavován opačným názorům a jeho vlastní jsou upevňovány. [17]
4. **Stagnace či rozpad společnosti:** V předchozí kapitole jsme si ujasnili, že minimálně např. **kotvení** postihuje vrcholové manažery společností stejně jako „obyčejné” lidi (viz kap. 2.2.5). Tyto kognitivní předsudky mohou vést k špatným rozhodnutím, nepřizpůsobení společnosti současným trendům a mnohému dalšímu. [18]

Společně tyto kognitivní předsudky utvářejí úkaz tunnel vision. Tunnel vision není tedy natolik pojem v psychologii, jako spíš obecný výraz zahrnující výše popisované předsudky do jednoho sousloví. Následně tedy budu o těchto předsudcích v následujících kapitolách mluvit již jen jako o tunnel vision, pokud nebude potřeba některý předsudek specifikovat.

2.4 Příklady negativních důsledků tunnel vision

Pro demonstraci dopadů tunnel vision v této podkapitole popíšu různé příklady toho, jak se tunnel vision negativně podepsala na výsledcích několika známých světových společností. U žádného z těchto případů neexistují vědecké studie, že tunnel vision byla hlavní příčina, ale dle mého názoru je více než pravděpodobné, že hrála významnou roli.

2.4.1 General Motors a nefunkční airbagy

V roce 2014 musela americká firma General Motors začít stahovat z oběhu přibližně 30 milionů aut kvůli problémům se zapalováním, které mohlo vypnout

motory při jízdě a zabránit airbagům v jejich funkci [19]. Díky tomuto problému muselo General Motors také zaplatit odškodné 124 rodinám obětí autonehod, kterým tato chyba vzala život.

Později se ukázalo, že tato úmrtí mohla vzniknout zčásti také neuváženými systémy společnosti General Motors. Zjistilo se, že General Motors vytvořila systémy a pravidla, jenž mohla k této tragédii přímo přispět.

Interní dokumenty General Motors v dané době ukazovaly, že ve firmě existovala pravidla, která zakazovala používání určitých frází nebo slov při popisování problémů managementu. Jako příklad mohu uvést, že zaměstnanci nesměli přirovnávat situaci k lodi Titanic, a nesměli používat až 69 slov, mezi nimi např. „inferno” (ohnivé peklo) nebo „deathtrap” (smrtelná past). Věří se, že toto mohlo přispět k podceňování vážnosti situace a vzniku těchto velkých problémů, kterým firma musela čelit.[20] Tato situace je dle mě dobrým příkladem vytvoření falešného dojmu bezpečí, probíraném v kapitole 2.3.

■ 2.4.2 Microsoft a jeho dlouhodobá vize

Společnost Microsoft čelila problému, kde si vytvořila příliš specifický cíl, tzv. „long-term vision”. Měnící se prostředí byznysu, trh, a podmínky následně zhoršily její efektivitu, protože její vize už nebyla tak dobře aplikovatelná na reálný svět.

Vize Microsoftu byla v devadesátých letech formulována jako „mít počítač v každé domácnosti”, a ačkoli to tehdy byla dobře aplikovatelná vize, v současné době internetu, mobilů a nižšího povědomí o tom, co počítač vlastně je, tato vize zaostává.

Microsoft, pravděpodobně právě díky těmto důvodům, pozdě zaznamenal výbušný růst internetu. Podle některých nereagovala společnost na jeho rychlý rozmach. Společnost zaznamenaně reagovala až po dnes již proslaveném proslovu Billa Gatese [22], který se konal v prosinci 1995. To však bylo až měsíce poté, co Netscape oznámil začátek rozmachu společnosti dotcom a léta poté, co počítačová experti rozpoznali potenciál transformace, jakou měl internet pro IT průmysl.

V roce 1999, po značné změně ve vnitřním vedení Microsoftu, firma opustila svoji původní vizi. V současné době už je formulována jako „umožnit každému jedinci a společnosti na planetě dosáhnout více” což je mnohem méně specifické a umožňuje to společnosti zkoušet nové věci a manévrovat v rámci své vize.[23] Myslím, že tento příběh je dobrým případem odvádění pozornosti od současnosti, probíraném v předchozí kapitole 2.3.

2.4.3 Kodak a neochota se přizpůsobit digitálnímu věku

Vzhledem k tomu, že hlavní činností společnosti Kodak byl prodej filmu, není těžké pochopit, proč se v posledních několika desetiletích ukázalo jako náročné udržet společnost profitabilní. Kamery se staly digitálními a pak zmizely v mobilních telefonech. Lidé šli z tisku obrázků ke sdílení on-line. Jistě, lidé z nostalgie stále tisknou fotoknihy a pohlednice, ale jejich objem bledne ve srovnání s tehdejší rozkvětem společnosti Kodak.

Společnost v roce 2012 podala návrh na ochranu před bankrotem, přestala prodávat film a prodala své patenty, než se v roce 2013 znovu objevila jako tentokrát už výrazně menší společnost. Ačkoli v předchozích dobách byla jednou z nejdělejších společností na světě, dnes má společnost tržní kapitalizaci menší než 1 miliardu dolarů.

Kodak byl tak zaslepen svým úspěchem a jeho pohled tak omezen na vlastní způsob činnosti, že zcela podcenil vzestup digitálních kamer. A to i přesto, že první prototyp digitálního fotoaparátu byl vytvořen v roce 1975 inženýrem Stevem Sassonem, který tehdy pro Kodak pracoval.[24] Kodak měl díky svému tunelovému vidění přílišné sebevědomí a nezaznamenal vývoj trhu. Oba tyto problémy byly popsány v kapitole 2.3.

2.4.4 Samsung a neuvážená sázka na auta

Tunnel vision může také vést manažery k tomu, aby vsadili příliš mnoho a příliš brzy na svou vizi. Rád bych dal za příklad Samsung, jednu z největších jihokorejských společností, zodpovědnou za přibližně 17 procent jihokorejského HDP. V roce 1992 si předseda představenstva společnosti Samsung Lee Kun Hee uvědomil, že jeho otec, zakladatel společnosti Samsung, oznámil vstup skupiny do automobilového sektoru, když vytvořil divizi Samsung Motors. Pan Lee ambiciózně prohlásil, že společnost Samsung Motors bude do roku 2010 patřit mezi 10 největších světových výrobců automobilů.

Společnost investovala velké prostředky do stavby nejmodernějšího výzkumného a konstrukčního zařízení a postavila továrnu na zelené louce se skvělou technologií výroby, čistými prostory a špičkovou robotikou[25]. Pan Lee přemístil několik nejzkušenějších manažerů společnosti Samsung z jiných divizí, aby tuto novou automobilovou iniciativu vedli.

Zisky v té době byly prchavé i pro etablované podnikatele v automobilovém průmyslu a pro začínající automobilové podniky bylo o to těžší se prosadit. Manažeři společnosti Samsung, korejské vládní úředníci a komentátoři průmyslu zpochybnili velkou sázkou společnosti na auta. Mnozí navrhli společný podnik nebo alianci, aby otestovali trh dříve, než učiní velký závazek se svými zdroji. Ale pan Lee se postavil za svou vizi.

Od doby, kdy první vůz odjel z montážní linky v roce 1998, utrpěla společnost Samsung Auto velké provozní ztráty a drtivé úrokové poplatky. Během několika let byla společnost Samsung nucena prodat svou automobilku za zlomek své počáteční investice automobilce Renault[26]. Pokud se podíváme do kapitoly 2.3, Samsung měl díky tunnel vision přílišné sebevědomí ve své vlastní schopnosti, a díky tomu ztratil velké množství peněz.

2.5 Pozitivní důsledky tunnel vision

Je ovšem také nutno zmínit, že tunnel vision nemá vždy jen negativní důsledky. Existují různé životní situace, kde se tunnel vision hodí. Popíšu zde dva stručné příklady, kdy může tunnel vision dle mého názoru mít pozitivní důsledky:

1. **Zabraňuje nerozhodnosti:** Bývá obvyklé, že se člověk hůře rozhoduje v situacích, kdy může řešit nějaký problém mnoha různými způsoby, než když má těchto možností méně. Tunnel vision umožní člověku zaměřit se pouze na úzkou množinu řešení a může tak zabránit nerozhodnosti. Samozřejmě, to může být na úkor kvality vybraného řešení.
2. **Umožňuje soustředění:** Dnešní svět je plný způsobů, jak rozptýlit pozornost člověka, od sociálních sítí po neustálou komunikaci s okolním světem. U kognitivně náročných profesí je však umění soustředit se na řešení problémů velice důležité. Tunnel vision může člověku umožnit soustředit se na jednu věc a vytěsnit okolní „šum“, což je v tomto případě prospěšné.

2.6 Shrnutí kapitoly

V této kapitole jsem nejprve definoval tunnel vision z obecného hlediska. Následně jsem formuloval různé kognitivní předsudky, které společně tunnel vision utváří, a promluvil jsem o nich z kontextu psychologické literatury (oproti tomu, jak se projevují v reálném světě).

Dále jsem poukázal na projevy tunnel vision v reálném světě, a to nejprve obecnými příklady jejich negativních důsledků. Následně jsem pro lepší ilustraci konceptu popsal několik příběhů ze světa byznysu, kde se dá předpokládat, že tunnel vision hrála důležitou roli.

V tomto okamžiku by čtenáři mělo být jasné, s jakými problémy se obecně budeme potýkat. Rád bych dále navázal kapitolou o „Aha“ momentu, tedy fenoménu, na kterém budu stavět své řešení.

Kapitola 3

„Aha” moment

V této kapitole shrnu, co je to „Aha” moment a definuji ho přesněji v kontextu poznatků z psychologie. Popíšu také, proč je tento úkaz dobrý způsob, jak si člověk může zapamatovat problematiku tunnel vision. Následně rozvedu, z jakého důvodu tento „Aha” efekt budu používat, a proč je důležitou částí mnou vyvíjené aplikace.

3.1 Definice „Aha” momentu

Einstein v roce 1929 řekl, „Někdy vím, že mám pravdu, aniž bych věděl proč”[38]. „Aha” moment, který v této kapitole budu nazývat i prozření, či „Heuréka!” moment, je definován jako náhlá změna ve vnímání stávajícího konceptu či vytvoření nového konceptu. Problematika, ke které se koncept vztahuje, často bývá před tímto pocitem vnímána subjektem jako obtížná a nepochopitelná. Prozření pak často vede ke změně kompetence subjektu v řešení daného problému, nebo i k celkovému řešení problému.[39] Tento pocit je často doprovázen čtyřmi hlavními charakteristikami, podle nichž ho můžeme odlišit od prostého pochopení[40]:

1. **Náhlost:** Řešení se objeví náhle, často bez předchozí realizace člověka, že se k prozření blíží. Bylo prokázáno, že pokud má subjekt pocit, že se „Aha” moment přibližuje, tak to často bývá naopak indikace, že žádný nenastane.[41]
2. **Jednoduchost/Prozření:** Problematika či problém, kterého se tento pocit týkal, je následně často vnímán jako jednoduchý, či alespoň výrazně jednodušší. Subjekt pocitu je náhle výrazně kompetentnější v dané problematice než byl předtím[16].
3. **Dobrý pocit:** Z prozření má člověk zaznamenanatelný pozitivní pocit. Není

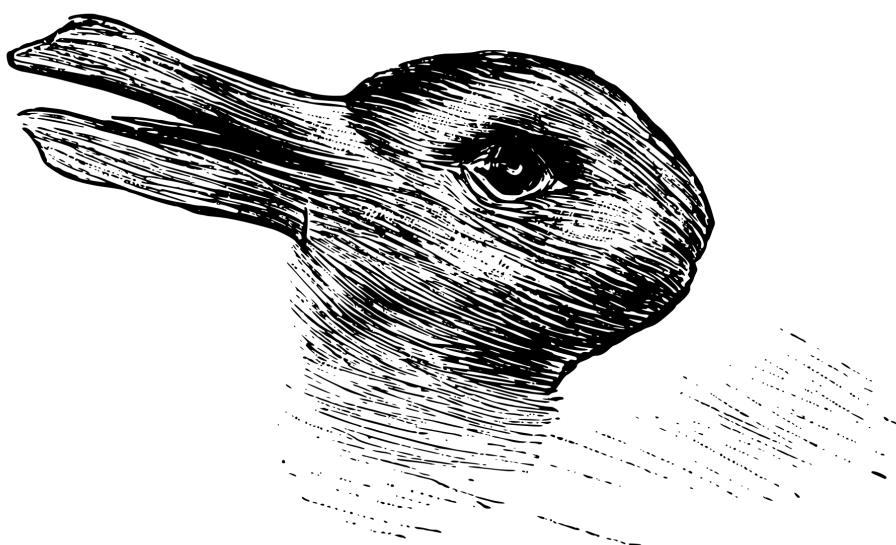
záhodno ho definovat jako pýcha, protože přichází už před případným ověřením svého vlastního závěru.

4. **Jistota:** Po „Aha” momentu si je člověk často velice jistý svým vlastním závěrem, i předtím, než je tento závěr ověřen skrze formální analýzu. Závěr mu připadá „intuitivní” a přirozeně vyplývající z vnímané zkušenosti.

3.2 „Aha” efekt v praxi

Typickým příkladem „Aha” momentu mohou být například známé obrázky typu „Mladá dáma nebo stařena” nebo „Králík nebo kachna”[42], kde prvotní zmatení a nepochopení situace často vyústí v rapidní změnu pohledu, která umožní člověku vidět obrázek oběma způsoby. Po tomto momentu prozření se tedy člověk stane „kompetentní” (tedy je schopný rozlišit oba „aspekty,”) ve vnímání tohoto hlavolamu (můžeme jej vidět na obrázku 3.1).

Welche Tiere gleichen ein- ander am meisten?



Kaninchen und Ente.

Obrázek 3.1: Králík nebo kachna?[58]

„Aha” momenty se často vyskytují v historii vědy či umění, od Mozartových zdánlivě jednoduše složených mistrovských děl[43] až po Archidemovo prozření

a formulaci jeho slavného zákona o chování kapalin[44], (dle jeho slavného výroku se tomuto momentu také říká „Heuréka” moment), vizi letu Leonarda da Vinci, Einsteinově vizi letu společně se slunečním paprskem a mnohé další. „Aha” momenty ovšem nejsou omezeny jen na případy matematiky, vědy či umění, se stejným psychologickým úkazem je možné se setkat například při pochopení vtípu nebo metafory.

Řešení skrze náhlé porozumění nebo „Aha” moment jsou častěji správnější než řešení nalezená skrze inkrementální a postupné zkoumání. Toto můžeme vysvětlit např. tím, že oproti analytickému řešení problému je „Aha” moment takzvaně „všechno nebo nic”. Analytické řešení tedy produkuje inkrementální výsledky, na jejichž základě se dá snadněji udělat chybný závěr, pokud nebyly zanalyzovány všechny informace.[45]. Stejně tak jistota subjektu v jeho vlastní kompetenci bývá vyšší u případů, kdy subjekt zažil „Aha” moment, oproti případům kdy k vyřešení problému postupoval pouze analytickým způsobem a žádný „Aha” moment neměl. [46]

Kognitivní procesy při „Aha” momentu také častěji vyústí v lepší dlouhodobou paměť vztahující se k dané tématice a většina subjektů poté nikdy nepřejde zpátky do „nekompetentního” stavu[47]. Je tedy velice nepravděpodobné, že by člověk mohl mít „Aha” moment o stejném problému více než jednou.

Je však možné mít prozření o něčem, čemu předtím nebylo věnováno dostatečné úsilí a snahy? Literatura o tomto problému nemluví zcela jasně. Bylo zjištěno, že vyřešení úkolu prozřením a vyřešení úkolu analytickou studií má fundamentálně jiný projev v mozkové aktivitě. Před vyřešením úkolu prozřením je vidět mozková aktivita směřující dovnitř, popsatelná jako vyhledávání potencionálních slabě spojených řešení, které již v mozku existují. Před vyřešením úkolu analytickým myšlením je vidět spíše intenzivní soustředění na vjemy před sebou - tedy na problém, který je řešen.[48] Je tedy pravděpodobné, že předchozí zkušenost či alespoň dostatečný repertoár souvisejících řešení je potřeba k vyvolání „Aha” momentu. Jak řekl známý vědec Louis Pasteur, „Náhoda se častěji stává připravené mysli”. [49]

■ 3.3 „Aha” efekt a tunnel vision

Vzhledem k výše uvedeným faktům si myslím, že „Aha” moment je skvělý způsob, jak upevnit povědomí člověka o tunnel vision. V této práci jsem ho použil k lepšímu zapamatování si problematiky tunnel vision, protože, jak jsem popsal výše, zkušenost vyvolaná „Aha” momentem častěji vyústí v zapamatování si dané problematiky než analytický přístup (analogický analytický přístup v tomto případě mohl být například vysvětlování konceptu tunnel vision a nutnosti pohlížet na věci několika způsoby). Laicky řečeno, silný pocit vyvolaný „Aha” efektem lépe upevní povědomí o problematice,

kterou se moje aplikace zabývá.

Zároveň můžu také využít dobrý pocit, který „Aha” efekt v člověku vyvolává. Lidé budou na aplikaci pozitivněji pamatovat, pokud v nich vyvolá dostatečný moment prozření. Tohoto jsem v aplikaci využil pro upevnění pochopení uživatele, že by se měl na věci dívat z různých úhlů.

■ 3.4 Závěr

V této kapitole jsem definoval „Aha” moment a jaké jsou jeho hlavní charakteristiky, tedy náhlost, jednoduchost, dobrý pocit a jistota. Následně jsem tento úkaz popsal do větší hloubky, abych čtenáři umožnil lepší přehled o problematice. Popsal jsem kde, bývají „Aha” momenty typické, že porozumění skrz „Aha” momenty bývají častěji správná než jiné formy porozumění a co je důležité pro jeho dosažení.

Ke konci kapitoly jsem popsal, proč je „Aha” moment dobrý psychologický úkaz pro upevnění povědomí o tunnel vision. Umožní lepší upevnění informace, že je nutné pohlížet na věci z více úhlů, stejně tak jako dobrý pocit, který si člověk po této zkušenosti odnese.

Kapitola 4

Definice problému a analýza existujících řešení

Tato kapitola nejprve s ohledem na zadání definuje problematiku, která bude řešena pomocí aplikace. Následně provedu analýzu způsobů výzkumu tunnel vision, včetně příkladů způsobu vedení studií, které se tímto tématem zabývají. Dále popíšu zajímavé kurzy a hlavolamy, které mě také inspirovaly při návrhu řešení.

4.1 Definice problému

Jak již bylo řečeno v kapitole Úvod, cílem této bakalářské práce je vytvoření aplikace, která bude sloužit pro účely demonstrace využívání různých úhlů pohledu na řešení problému a ukázkou, že rozšíření pohledu o další úhel může vyvolat „Aha” moment a takto vyřešit problém, který je zdánlivě neřešitelný.

4.2 Analýza existujících řešení

Cílem práce je vytvořit aplikaci, která s pomocí „Aha” efektu ukáže uživatelům dopady tunelového vidění a přínos více úhlů pohledu. Už v kapitolách 2 (Tunnel vision) a 3 („Aha” moment) jsem zmínil, že v této oblasti je již hodně výzkumu - proto než začnu vytvářet vlastní aplikaci, provedu analýzu metod výzkumu tunnel vision, která by mi mohla pomoci vhodně definovat úlohy. Dále popíšu zajímavé stránky, které se tunnel vision zabývají z pohledu byznysu a nakonec popíšu vybrané známé hlavolamy, které mě inspirovaly při vytváření úloh.

■ 4.2.1 Výzkum tunnel vision

V této podkapitole popíšu zajímavé studie tunnel vision, a jakým způsobem byly prováděny. Tyto studie mě svým provedením pomohly vhodně definovat typ úkolů, který uživatelé.

Zaměření se na první skutečnost při komplexním rozhodování

V již výše citované studii z publikace *Organizational Behavior and Human Decision Processes* [2] výzkumníci provedli čtyři experimenty, které měly za úkol prokázat, že člověk se při rozhodování často neproporčně soustředí na první jemu předložené rozhodnutí. Méně se pak soustředí na další předložená řešení. Také se snažili dokázat, že člověk přeceňuje kvalitu prvního jemu předloženého řešení, ačkoli mohou být ostatní řešení ekvivalentní.

Všechny čtyři experimenty měly podobnou strukturu - pro ilustraci problému se však zaměříme jen na ten první. V prvním experimentu bylo 82 účastníků studie instruováno, aby si představili že jsou zodpovědní za rozhodování imaginární počítačové společnosti, která chce vydat nový notebook. Měli vybrat jednu ze čtyř možností konfigurace notebooku. Každý notebook byl totožný, ale měl jeden atribut (pokaždé jiný) horší než ostatní. Každý z účastníků měl zhodnotit jeden náhodně vybraný laptop, ale měli informace i o jiných. Byli následně dotázáni, jaký design laptopu by měl být zvolen a jaká je pravděpodobnost, že jimi zhodnocený laptop bude ten, který zvítězí. Výsledek studie byl takový, že účastníci předpovědatelně nadsazovali kvalitu laptopu, který měli zhodnotit, a to nezávisle na tom, který hodnotili.

Konfirmační zkreslení

V jiné studii, tentokrát z publikace *William Mary Law Review*, [10] bylo 60 studentů hodnocených v jejich schopnosti identifikovat skutečné dopisy sebevrahů. Hodnotitelé jim dávali náhodně vybraná hodnocení jejich kompetence a později účastníkům řekli, že si ho vymysleli. Účastníci, kteří byli hodnoceni v této schopnosti kladně, se i přes jasný důkaz proti později sami identifikovali jako kompetentnější.

Ve stejné studii byl proveden ještě druhý experiment, který přidal do stejné situace ještě neustranné pozorovatele - ti, stejně jako účastníci, považovali za kompetentnější ty lidi, kteří byli dobře ohodnoceni, ačkoli se na konci výzkumu všichni dozvěděli, že tato hodnocení byla náhodná.

Kotvení

V mnohých studiích se zkoumá kotvení, či tzv. „anchoring effect“. Tento efekt jsem více probral v kapitole 2.2.5. Ale připomenu, že jeho základní projev je, že lidé se zaměří na první získanou informaci (o jakémkoli subjektu), a následně posuzují ostatní informace v kontextu té první, ačkoli může být

úplně chybná.

Podle těchto studií [27] se kotvení vyvolává tak, že položí účastníkovi studie nějakou otázku na kterou neví odpověď (Ve kterém roce se George Washington stal americkým prezidentem [28]) a „předsunout“ mu irelevantní „kotvu“. V tomto případě by to mohl být např. rok 1872 (skutečná odpověď je 1789). Účastníka se poté ptají, zda je skutečná hodnota vyšší nebo nižší než toto číslo. Nezávisle na to, co odpoví, se jejich odpověď přiblíží „kotvě“, ačkoli ta může být irelevantní či matoucí.

■ 4.3 Existující kurzy zaměřující se na tunnel vision

V této podkapitole jsem se zaměřil na kurzy pro svět byznysu, a to z důvodu, že tento svět se fenoménem tunnel vision obvykle rád zabývá, i kvůli důsledkům popsaným v podkapitole 2.3. Další důvod je, že svět byznysu je často výrazně zaměřen na osobní produktivitu a důležitost správného rozhodování, přičemž podobné zaměření dle mého názoru obvykle nevidáme v jiných sférách života.

■ 4.3.1 Mindtools.com

Mindtools.com je jeden z největších webových portálů pro management trénink na světě. Ročně obslouží asi 27 milionu uživatelů. Jejich portfolio kurzů a cvičení zahrnuje mnohé věci od schopností vést, řešit problémy, komunikační schopnosti, ale také lekce o způsobech myšlení, přemýšlení zeširoka a vyhýbání se problémům jako je např. Tunnel vision.

Jako kurz nejvíce související s tematikou bakalářské práce bych označil kurz jménem „Scenario Analysis - Exploring different futures“[30]. Tento kurz se zabývá hypotetickými scénáři, kde je úkolem uživatele definovat problém, posbírat důležitá data, oddělit určité od pravděpodobného, a s těmito daty sepsat nejlepší scénář řešení. Toto přímo souvisí s tunnel vision - kurz specificky odrazuje uživatele od zaměřování se pouze na určitou množinu dat na úkor jiných.

Mindtools také dostaly ocenění Queen's Award for Enterprise[29] – nejprestižnější cenu pro business ve Velké Británii.

■ 4.3.2 iMindMap

iMindMap je společnost původně se zabývající konceptem myšlenkových map, ale byla rozšířena o množinu různých dalších užitečných nástrojů pro

plánování firemních procesů a podnikání.

Jeden z jejich kurzů je podobný aplikaci, kterou se jako cíl tohoto projektu snažím vytvořit. Provede uživatele sérií textových úloh, které mají uživatele trochu vyvodit z míry – odpověď, která je zpočátku „jasná“, není ta správná, a člověk musí přemýšlet o jiných úhlech pohledu na danou problematiku. [31]

4.3.3 Coursera for business

Coursera je světoznámá stránka pro vzdělávání a dlouhé kurzy, které často trvá několik týdnů splnit. Nabízí stovky kurzů o desítkách témat od programování, fotografování či právě manažerskou efektivitu.

Coursera byla založena v roce 2012 profesory Stanfordské univerzity Andrew Ng a Daphne Koller. Princeton, Stanford, University of Michigan a University of Pennsylvania byly první univerzity, které nabídly obsah na této platformě. Nabídky se od té doby rozšířily tak, aby zahrnovaly specializace - sbírky kurzů, které budují dovednosti v konkrétním předmětu - stejně jako tituly a produkt pro rozvoj pracovních sil pro podniky a vládní organizace.

Jako zajímavý kurz na stránce Coursera for Business bych označil např. Business Strategy kurz[33]. Tento kurz, především v jeho pozdějších stupních, se zabývá formulací strategických rozhodnutí a zaměřuje se na obvyklé chyby manažerů a omezující psychologické úkazy, jako například tunnel vision.

4.3.4 Shrnutí

Kurzy mohou být účinný způsob, jak řešit povědomí o problému tunnel vision. Vytvořit podobný kurz je však náročné na přípravu i provedení. Naším cílem je také vytvořit aplikaci, ne kurz. Výše vyjmenovanými kurzy jsem se tedy pouze inspiroval - rád bych, podobně jako kurz, vytvořil souvislou sekvenci různých úloh. V následující kapitole popíšu různé hlavolamy, které mě inspirovaly v přípravě těchto úloh.

4.4 Inspirace hlavolamy

Jak jsem již psal v shrnutí předchozí kapitoly, snažil jsem se najít různé 3D hlavolamy a cvičení. Byť jejich cílem často bývá uživatele pouze nachytat, a nejsou propojeny s širším cílem vyvolat v uživateli povědomí o tunnel vision, jsou schopny vyvolat účinný „Aha“ moment. Vybral jsem jako příklad dvě úlohy, u kterých je důležitou součástí omezenost jednoho úhlu pohledu na

věc. Příklady dvou takových hlavolamů jsou popsána níže:

■ 4.4.1 Amesova místnost

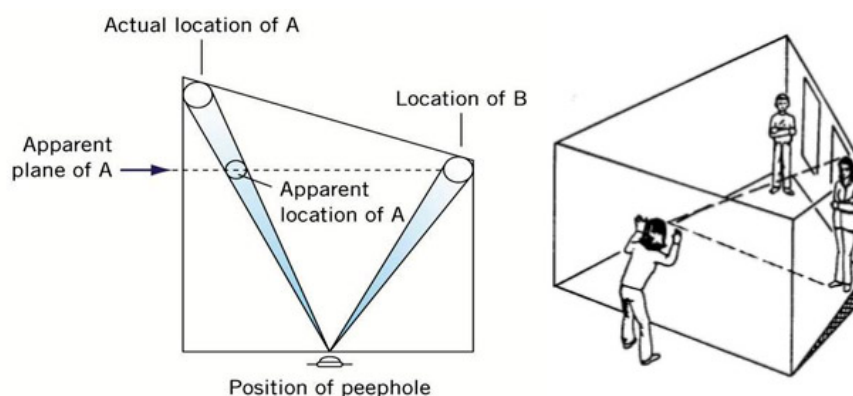
Amesova místnost je známý koncept, kde dva lidé stejné velikosti stojí na protější straně místnosti, podél zdánlivě kolmé zdi, a jeden vypadá o polovinu menší než ten druhý. Toto se děje díky chytrému optickému klamu, který „maskuje“ zakřivení zdi za nimi. Pro znázornění principu se můžeme podívat na obrázek 4.2. Člověk ze svojí pozice předpokládá, že místnost je obdélníková, a proto se zdá objekt ve vzdáleném rohu na jiné pozici než je. Opticky tak vypadá menší, než by měl být. Pro ilustraci se čtenář může podívat na tento obrázek (4.1):



Obrázek 4.1: Amesova místnost. [55]

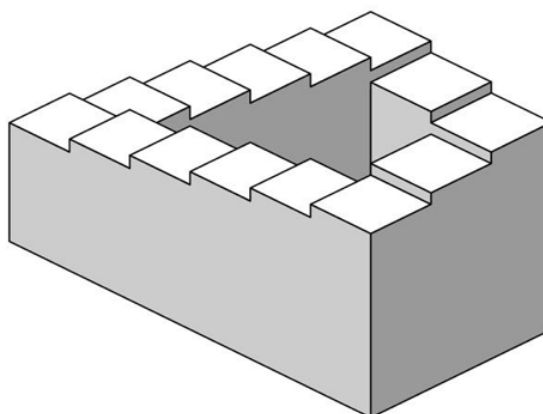
■ 4.4.2 Penroseovy schody

Penroseovy schody jsou jedna z nejznámějších optických iluzí. I přesto, že schody jsou vyobrazeny na obrázku s dvěma dimenzemi, pozorovatel má pocit, že schody nikdy nemají nejvyšší či nejnižší bod, a potenciálně člověk pohybující se na nich by neustále stoupal nahoru a zároveň by nikdy nahoru nestoupal. Tento paradox fascinuje generace již desetiletí. Penroseovy schody



Obrázek 4.2: Amesova místnost - princip. [56]

se staly inspirací pro mnoho umělců i filmařů, vyobrazené např. v nedávném snímku „Inception”. Penroseovy schody můžeme vidět na obrázku 4.3.



Obrázek 4.3: Penroseovy schody [57]

4.5 Shrnutí kapitoly

V této kapitole jsem definoval, na čem bude založeno řešení implementační části. Následně jsem vyhledal různé způsoby výzkumu tohoto problému ve vědeckých studiích, které jsem považoval za nejvíce autoritativní co se tohoto problému týče.

Dále jsem provedl analýzu existujících kurzů na internetu, které k řešení problému tunnel vision přistupují z více „analytického” hlediska. U těchto webů jsem stručně jejich popsal jejich historii, a u některých poukázal na kurz či lekci, kterou nabízí a která nejvíce souvisí s tématem této bakalářské práce. Nakonec jsem popsal několik známých 3D hlavolamů či paradoxů, které mě

inspirovaly při návrhu konceptů jednotlivých úloh (viz 5.1).

Po analýze existujících řešení jsem se rozhodl, že moje aplikace bude sestávat ze série úloh. Tyto úlohy budou v konceptu podobné výše zmíněným hlavolamům, jelikož důležitý princip spojující tyto úlohy bude poukázání na omezenost jediného úhlu pohledu.

V následující kapitole navážu na tuto kapitolu a navrhnou různé podoby úloh, které jsem se rozhodl vytvořit.

Kapitola 5

Návrh úloh

V této kapitole popíšu jednotlivé úlohy a jejich koncepty, které jsem pro aplikaci navrhl. U těchto návrhů jsem se pokoušel propojit způsoby řešení zmíněných v kapitole 4 - metody výzkumu tunnel vision, naučné kurzy a prostorové hlavolamy. Popíšu jejich koncept a jak jsem k němu přišel. Dále také popíšu cíl úlohy a její řešení.

5.1 Popis a cíl jednotlivých úloh

V této podkapitole popíšu jednotlivé úlohy, které jsem navrhl. Každá úloha bude popsána takto:

- **Popis úlohy:** Zmíním se, na jakém principu je úloha vytvořená, čím jsem se u vytvoření úlohy inspiroval, a proč jsem se tím inspiroval.
- **Úkol a řešení:** Popíšu hlavní úkol, který bude muset uživatel vyřešit, a jeho řešení.
- **Jak souvisí úloha s tunnel vision:** Zde poukážu na hlavní souvislosti mezi úlohou a tunnel vision.
- **Jak úloha vyvolá „Aha” efekt:** Zde popíšu, jakým způsobem úloha vyvolá „Aha” efekt (viz kapitola 3).
- **Vizuální podoba úlohy:** Nakonec představím náčrtek podoby úlohy. Tímto náčrtem se budu inspirovat při její implementaci.

5.1.1 Bludiště

Popis úlohy:

Tato úloha poukazuje na to, že lidé jsou často zvyklí řešit známou úlohu, kterou znají ze života, určitým způsobem. Mají pak tendenci ignorovat okolnosti, které by mohly situaci změnit.

V případě bludiště je toto tvrzení možno aplikovat takto - obvykle se bludiště řeší tak, že sledujeme cestu v daném bludišti, dokud úspěšně nenarazíme na cestu, která vede ke zdárnému konci. Případně další řešení bývá jít od konce a hledat cestu k začátku, kterýmžto způsobem si můžeme být jistí, že naše cesta skutečně vede ke konci.

V této úloze jsem tohoto zažitého způsobu řešení chtěl využít. V úloze neexistuje cesta z bludiště ven konvenčním způsobem, tedy jít z bludiště jeho vlastním východem. Jen když se podíváme na bludiště ze tří rozměrů, tak si uvědomíme, že máme ještě dvě možnosti, jak se z bludiště můžeme dostat - vrchem a spodem.

Koncept úlohy můžeme vidět na obrázku 5.1. (Cvičení 1 - Bludiště)

Úkol a řešení:

Úkolem je dostat kuličku, která je uprostřed bludiště na počátku úlohy, ven z bludiště. S kuličkou by mělo být možno hýbat, a zároveň by se měla hýbat na osách pohledu (tedy jakmile uživatel změnil pohled, kulička se stále pohybuje nahoru, i z jeho nového pohledu).

Řešení je dostat se za hranice bludiště, což je pouze možné vysunutím kuličky z bludiště vrchem či spodem.

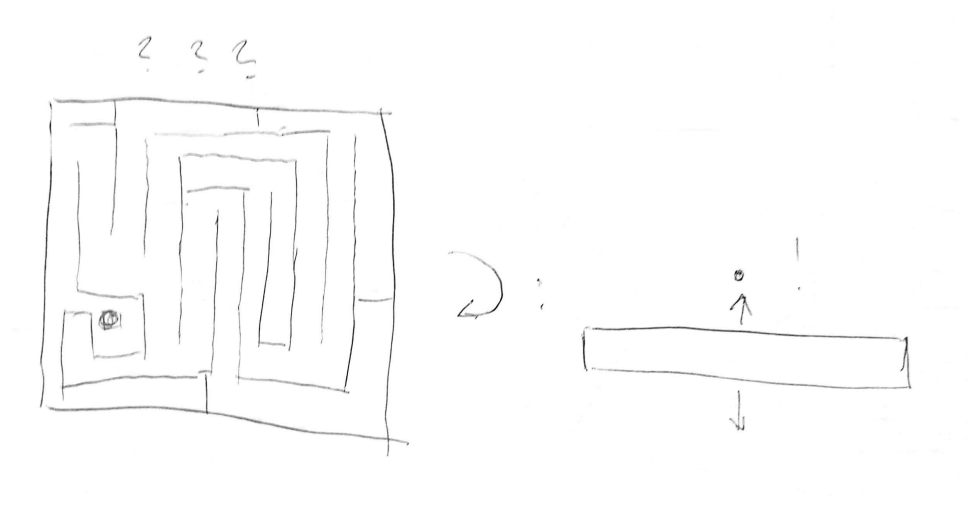
Jak souvisí úloha s tunnel vision?

Tato úloha poukazuje na to, jak předpoklad, že již známé informace mohou být stejně aplikovány v jakékoli podobné situaci, může zmást uživatele. Tunnel vision je v tomto případě způsobena **konfirmačním zkreslením** (viz kapitola 2.2.1). Uživatel má nějaký předpoklad, jak se bludiště řeší, a pouze změna pohledu mu umožní změnit názor na řešení. Zároveň také hraje roli vytrvalost víry (kap. 2.2.2) - uživatel přetrvává ve snaze řešit bludiště obvyklým způsobem i přes to, že to není možné.

Jak úloha vyvolá „Aha” efekt?

Úloha vyvolá „Aha” efekt ve chvíli, kdy si uživatel uvědomí, že změnou pohledu může pohybovat s kuličkou i směry nahoru a dolů.

Vizuální podoba úlohy:



Obrázek 5.1: Cvičení 1 - Bludiště

■ 5.1.2 Nápis Think

Popis úlohy:

U této úlohy jsem se inspiroval klasickými hádankami z filmů, kde se hlavní hrdina zaobírá složitou hádankou, zatímco jeho parťák stojí opodál. Protože však vidí celou situaci ze širšího pohledu, vyřeší nakonec hádanku.

Tato úloha začíná s kamerou uprostřed zdánlivě nepřehledné změti kostek, a ptá se: „Jaký nápis vidíte na obrazovce?“. Z určitého úhlu je však vidět, že kostky vytváří smysluplný nápis.

Koncept této úlohy můžeme vidět na obrázku 5.2 (Cvičení 2 - Nápis Think)

Úkol a řešení:

Úkolem je správně odpovědět na otázku „Jaký nápis vidíte na obrazovce?“. Správná odpověď je „Think“, tedy anglicky „mysli“. Toto slovo jsem zvolil vzhledem k souvislosti s povahou bakalářské práce. Správnou odpověď uživatel zjistí dostatečným oddálením kamery.

Jak souvisí úloha s tunnel vision?

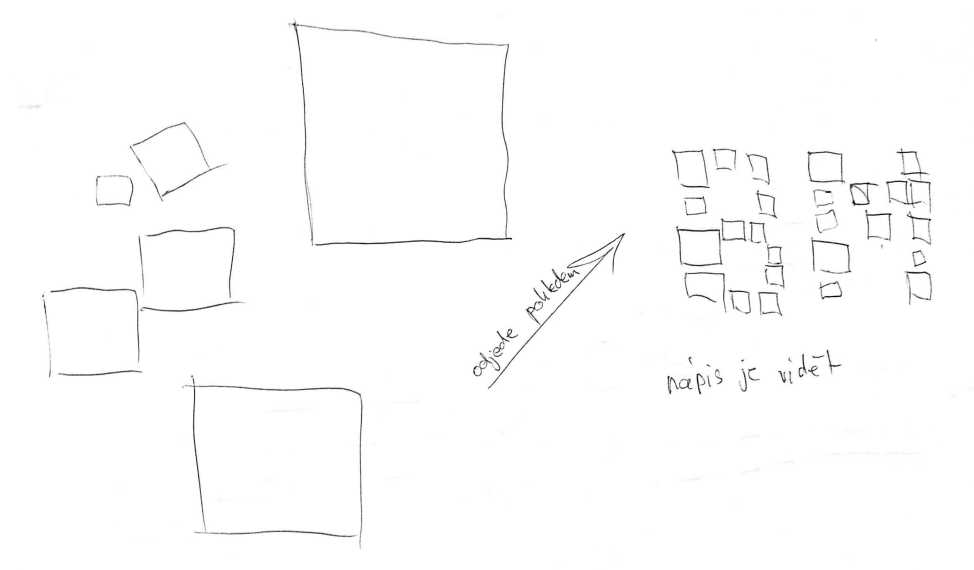
Omezený pohled, který poskytuje poloha kamery na počátku úlohy, zmate uživatele, který následně nechápe, jak ve změti kostek může existovat nějaký text či nápis. Jeho vidění je tedy nutně tunelové - vidí pouze malou část širší

situace. Tato úloha využívá principu **kotvení** (viz kapitola 2.2.5). Uživatel se často zaměří na informaci, že je ve zřeteli kostek, a to pak omezí jeho schopnost rozeznat, že z jiného úhlu pohledu to zřetel vůbec být nemusí.

Jak úloha vyvolá „Aha” efekt?

Úloha vyvolá „Aha” efekt, když si uživatel uvědomí, že z dostatečné dálky vypadá původně nepřehledná zřetel kostek jako písmena abecedy. Tímto si uživatel uvědomí, že řešení úlohy je dostatečně oddálit úhel pohledu.

Vizuální podoba úlohy:



Obrázek 5.2: Cvičení 2 - Nápis Think

5.1.3 Rovná plocha

Popis úlohy:

Na ploše máme rozmístěno mnoho náhodných objektů (jakékoli trojrozměrné objekty bránící ovládanému čtverci). Myší ovládáme pohybuující se čtverec o určité výšce a šířce. Tyto rozměry musí být dostatečně velké, aby v počátečním pohledu nebylo možné umístit čtverec.

K této úloze mě inspirovaly strategické hry. V těchto hrách musíme často umístit budovy na plochu, kde už mnohdy bývá mnoho jiných budov a objektů. Můžeme sice vyhledávat to pravé místo pro umístění budovy, ale často bývá lepší budovu umístit za hranice naší současné zástavby.

Tuto úlohu můžeme vidět na obrázku 5.3 (Cvičení 3 - Rovná plocha)

Úkol a řešení:

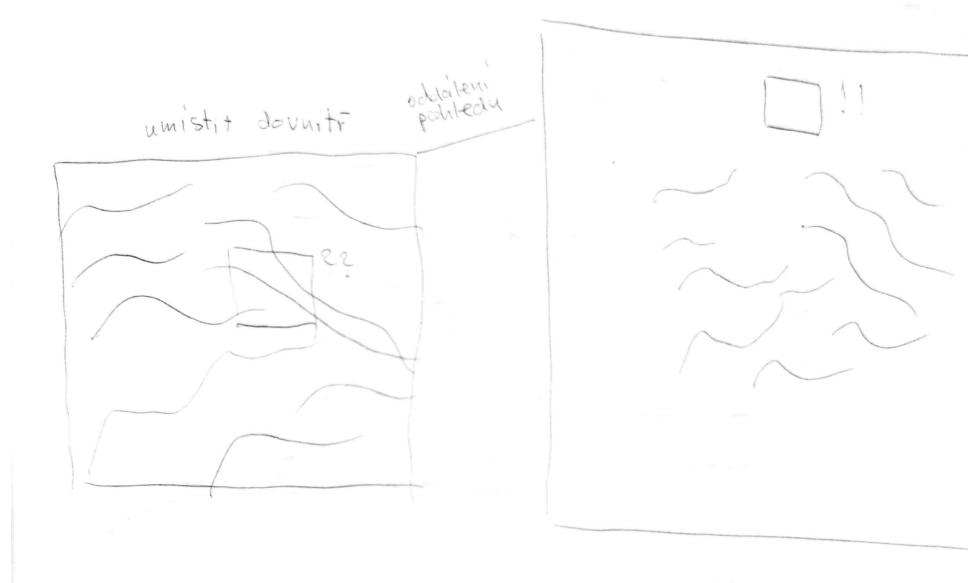
Úkolem v této úloze je označit úsek plochy na obrazovce, který neprotíná žádné objekty v herním světě. Úloha není vyřešitelná z výchozího pohledu kamery, proto musí uživatel kameru oddálit. Při oddálení kamery zjistí, že za hranicí pohledu již žádné objekty nejsou, pouze prázdná plocha. Na tuto plochu již může uživatel jednoduše kliknout a vyřešit tak úlohu.

Jak souvisí úloha s tunnel vision?

Tato úloha je klasický příklad tunnel vision. Místo toho, abychom se pokusili podívat se na věc z jiného, či širšího pohledu, uživatel se pravděpodobně pokusí umístit čtverec v rámci jeho výchozího pohledu kamery. Kdyby se však aktivně snažil vyhledávat jiný pohled, úloha by byla vyřešená mnohem snadněji.

Jak úloha vyvolá „Aha” efekt?

„Aha” efektu úloha docílí ve chvíli, když si uživatel uvědomí, že může umístit hrací čtverec i za hranice svého výchozího pohledu.

Vizuální podoba úlohy:

Obrázek 5.3: Cvičení 3 - Rovná plocha

■ 5.1.4 Pro les nevidím stromy

Popis úlohy:

U této úlohy jsem se inspiroval známým "zaměřením se na nepodstatný detail".

Úloha vytvoří náhodnou mřížku různobarevných objektů, kde jeden z těchto objektů je odlišný od všech ostatních. Objekty mohou mít různé vlastnosti (jako například otočení nebo barvu). Žádná z těchto vlastností však nebude unikátní pouze u jednoho objektu. Vlastnost, která je unikátní, by neměla být dobře vidět z počátečního pohledu.

Náčrtek této úlohy můžeme vidět na obrázku 5.4 (Cvičení 4 - Pro les nevidím stromy)

Úkol a řešení:

Označit správný objekt v množině ostatních objektů. U řešení této úlohy je důležité nenechat se zmást různou barvou či hledat pomocí jiných charakteristik, které nejsou unikátní jen pro jeden objekt. Řešení je označit správný objekt, který bude mít unikátní vlastnost (např. trochu jiný tvar) Tento objekt je také vyznačený při stisknutí tlačítka „nechat se podat“.

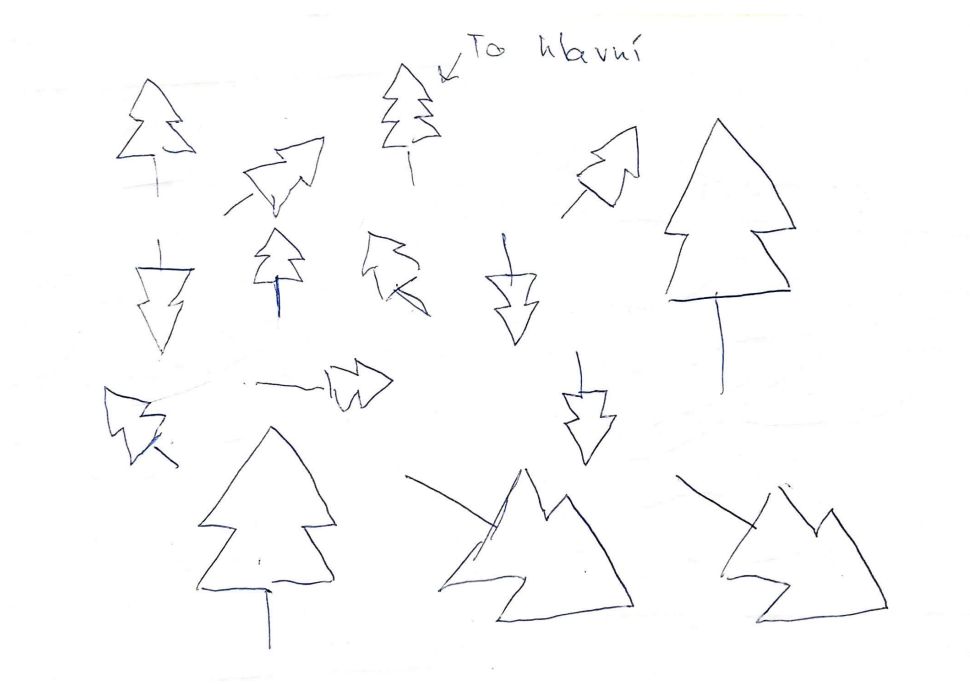
Jak tato úloha souvisí s tunnel vision?

Uživatel se často zaměří na jednu charakteristiku problému (v tomto případě barvu) a ignoruje ostatní proměnné, které by mohly lépe přispět k vyřešení problému (viz kapitola 2.2.5).

Jak úloha vyvolá „Aha” efekt

Při prozkoumání objektů zblízka si uživatel lépe uvědomí tu správnou unikátní charakteristiku objektu, na kterou se při svém hledání nezaměřil.

Vizuální podoba úlohy:



Obrázek 5.4: Cvičení 4 - Pro les nevidím stromy

■ 5.1.5 Zase o tom příliš nepřemýšlejme

Popis úlohy:

Úloha je o zaměření se na detaily a o přílišném přemýšlení o detailech problému, zatímco řešení je jednoduché a jasně viditelné přímo před námi.

Ve scéně je koule, která má na sobě texturu vytvářející různé tvary. Při otáčení koule se tyto tvary mění. Není tedy možné specifikovat jediný tvar na povrchu koule, a řešení musí být tedy koule sama.

Náčrtek tohoto konceptu můžeme vidět na obrázku 5.5 (Cvičení 5 - Zase o tom příliš nepřemýšlejme).

Úkol a řešení:

Zatímco u ostatních úloh souvisí řešení se změnou úhlu pohledu, v této úloze je změna pohledu naopak kontraproduktivní. Uživatel by se tedy neměl snažit hledat způsob, jak vyřešit úlohu změnou pohledu. U této úlohy je důležité, aby se nenechal zmást a odpověděl bez zbytečného přemýšlení. Správná odpověď je „koule“.

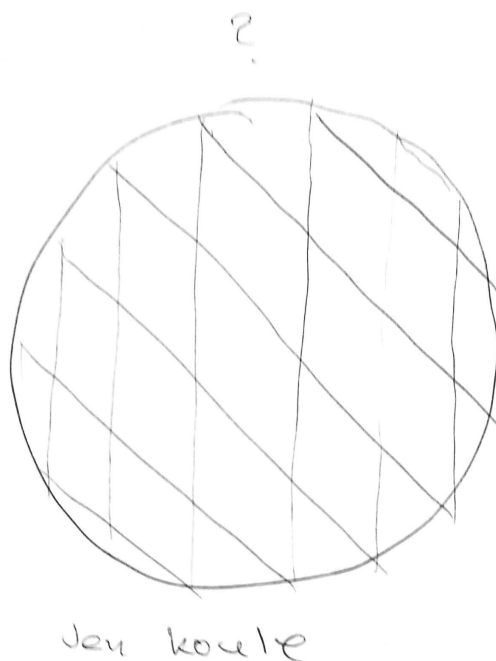
Jak tato úloha souvisí s tunnel vision?

Při řešení problémů se často stává, že nad nimi příliš přemýšlíme – z předchozích zkušeností očekáváme nějakou složitost či nějaký „chyták“, a snažíme se ho aktivně vyhledávat, i když tam žádný není. Toto opět naznačuje omezený pohled na problematiku, který ignoruje specifické okolnosti problému a soustředí se jen na jeden způsob jeho řešení. Toto je dobrý příklad **konfirmačního zkreslení** (více v kapitole 2.2.1).

Jak úloha vyvolá „Aha“ efekt

Uživatel by měl pohybovat s jeho pohledem tak, aby si uvědomil, že se tvary na kouli mění. Takto by si měl uvědomit, že správná odpověď je koule, nikoli tvary na jejím povrchu.

Vizuální podoba úlohy:



Obrázek 5.5: Cvičení 5 - Zase o tom příliš nepřemýšlejme

■ 5.1.6 Postupka z karet

Popis úlohy:

Úloha byla inspirována konceptem, kde předložíme uživateli určitou zavádějící informaci předtím, než začne řešit úlohu. Účastník se poté na tuto informaci

zafixuje, a tato fixace ho zpomalí v hledání skutečného řešení.

V této úloze předložím uživateli nejprve informaci o určité karetní postupce. Následně mu ukážu tři karty a zeptám se ho, jak by je složil do posloupnosti, která mu dává smysl. Předpoklad je, že uživatel se bude snažit složit karty do správné posloupnosti dle předtím zmíněné postupky, ale ve skutečnosti bude muset složit karty do sekvence „JAK”, tedy tím vytvoří slovo z českého jazyka.

Můžeme jej vidět na obrázku 5.6 (Cvičení 6 - Postupka z karet)

Úkol a řešení:

Seřadit karty do správné sekvence tak, aby zobrazovaly slovo „JAK”.

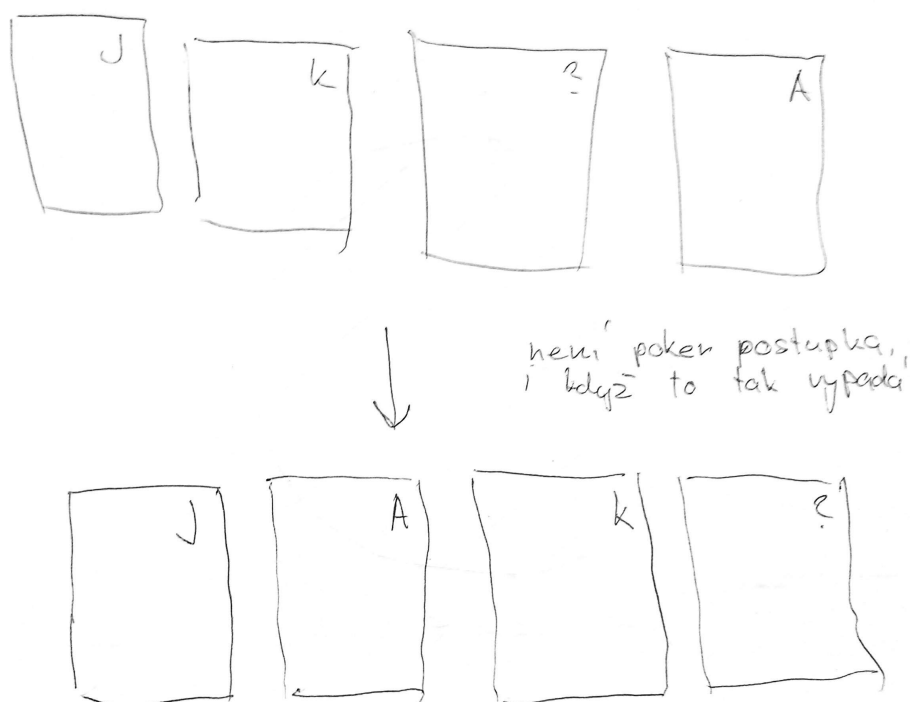
Jak tato úloha souvisí s tunnel vision?

V kapitole 4.2.1 jsem popsal studii, která jejím účastníkům nejprve představila zavádějící informaci, na kterou se uživatelé posléze zaměřili. Když před samotnou úlohou představím uživateli možnou postupku z karet, tak věřím, že se na tuto informaci zaměří dopředu.

Jak úloha vyvolá „Aha” efekt

Úloha vyvolá „Aha” efekt tehdy, když se uživatel přestane snažit seřadit karty do karetní postupky, a zamyslí se nad dalšími možnostmi. Posléze si uvědomí, že sekvence karet nemá souvislost s klasickou karetní postupkou, nýbrž s českým slovem.

Vizuální podoba úlohy:



Obrázek 5.6: Cvičení 6 - Postupka z karet

5.2 Pořadí úloh

V aplikaci samotné uživateli nabídnu sekvenci pěti úloh z šesti možných. Tento počet jsem zvolil, jelikož je dle mého názoru dostatečný pro cíl mé aplikace, ale zároveň nezabírá zbytečné množství času. Počet pěti také umožňuje další úlohy využít pro variaci, aby nebyly všechny průchody aplikací stejné. Těchto pět úloh bude buď v předem daném pořadí, nebo bude pro účely uživatelského testování náhodné.

Chci, aby předem dané pořadí úloh mělo dva pevné body, a to první úlohu (podkap. 5.1.1) na začátku a pátou úlohu (podkap. 5.1.5) na konci. Důvod, proč chci s první úlohou začít, je takový, že spojuje známý úkol (vyřešit bludiště) s nutností změny pohledu. Důvod, proč chci pátou úlohou skončit je, že uživatel po řešení série úloh s nutností změny pohledu bude mít snahu vyřešit úlohu stejným způsobem. Úloha 5 je tedy zamýšlena jako určitý „návrat do reality“, a má vyjádřit, že občas je řešení přímo před námi to správné.

Abych si tuto myšlenku otestoval, chci pro uživatelské testování učinit pořadí úloh náhodné. Toto mi umožní ověřit si, že sekvence, kterou jsem zamýšlel,

zmate uživatele jak na začátku, tak na konci série úloh.

■ 5.3 Shrnutí kapitoly

V této kapitole jsem popsal návrh šesti konceptů, ze kterých chci následně vytvořit sérii úloh v mojí aplikaci. U každého z těchto konceptů úloh jsem popsal jak jsem k němu došel, jeho cíle a řešení, a také jak souvisí s tunnel vision. Implementace těchto úloh je popsána v kapitole 8.3.

Kapitola 6

Návrh řešení

V této kapitole popíšu návrh aplikace, která bude řešit problém diskutovaný v předchozí kapitole, a jaké technologie a nástroje jsem použil pro její vývoj. Krátce zodpovím na otázku, proč jsem při vývoji nepoužil herní engine.

6.1 Principy fungování aplikace

Dle analýzy a návrhu úloh v předchozích kapitolách, aplikace bude mít podobu sekvence úloh pracujících se změnou pohledu. Takových úloh bude šest, ale uživateli bude předloženo pět úloh z nich vybraných (viz kapitola 5.2). Každá z těchto úloh by měla člověku nabídnout určitou úroveň interaktivity a zároveň ho přimět přemýšlet o aplikaci pohledem na věci z jiných úhlů v reálném životě.

Úlohy budou fungovat jako určitý kvíz nebo cvičení – po spuštění budou probíhat v pořadí definovaném v podkapitole 5.2. Předloží uživateli úkol, který bude možné vyřešit pohybem kamery v 3D světě, pohybem objektu ve scéně, nebo otáčením a přibližováním 3D kamery. Některé úlohy budou řešeny interaktivně, tedy přímým označováním objektů ve scéně, a některé budou řešeny zadáváním odpovědi do textového pole.

Klíčovým tlačítkem aplikace bude tlačítko „Nechám se podat“. Toto tlačítko vypomůže uživateli v případě, že úlohu nebude moct vyřešit, a pomůže uživateli zažít „Aha“ efekt (viz kapitola 3). U úloh, kde se uživatel nechá podat, toto tlačítko samo pohne kamerou či změní scénu tak, aby uživateli bylo okamžitě jasné, jaké je řešení.

Úlohy budou vytvořené tak, že na první pohled nebudou jednoduše vyřešitelné. Například u úlohy bludiště (viz podkapitola 5.1.1) nebude zřejmé, že můžeme pohled kolem něj otáčet, a např. u nápisu z kostek (viz podkapitola 5.1.2)

nebude zřejmé, že mají kostky zprostředkovávat jakoukoli jinou informaci, než že jsou kostky.

Člověk si bude moci každou úlohu přehrát znovu, čímž se posune zpátky v sekvenci či dopředu v sekvenci k úloze, kterou označil v uživatelském rozhraní. Může případně ze sekvence vystoupit, pokud zvolí úlohu, která v sekvenci není.

6.2 Funkční a nefunkční požadavky aplikace

Vzhledem k tomu, že jsem již v kapitole 5 popsal úlohy, které chci implementovat a v této kapitole navrhl obecnou podobu projektu, rád bych nyní zformalizoval požadavky aplikace. V této kapitole tedy popíšu hlavní funkční a nefunkční požadavky aplikace, tedy co potřebuji, aby výsledná aplikace splňovala.

6.2.1 Funkční požadavky

Jako hlavní funkční požadavky pro aplikaci jsem podle analýzy (kapitola 4.2) a návrhu úloh (kapitola 5) identifikoval tyto:

1. **Otáčení, přibližování a pohyb s kamerou:** Požaduji, aby se v jednotlivých úlohách aplikace dalo pohybovat, otáčet a přibližovat s kamerou.
2. **Tlačítko „nechat se podat“:** Požaduji, aby se v každé úloze dalo použít tlačítko „nechat se podat“, které umožní uživateli zjistit řešení dané úlohy.
3. **Ukazatel postupu:** Požaduji, aby aplikace obsahovala ukazatel postupu, který uživateli ukáže současný postup v sekvenci úloh a umožní mu vrátet se k již vyřešeným úlohám.
4. **Přepínání mezi úlohami:** Požaduji, aby se v aplikaci dalo jednoduše přepínat mezi jednotlivými úlohami, a to i těmi, které nejsou v sekvenci nebo nejsou již vyřešené.
5. **Uživatelské statistiky:** Požaduji, aby aplikace zapisovala všechny důležité informace o průchodu uživatele a umožnila uživateli vidět jeho statistiky. Těmito statistikami jsou doba průchodu úlohou, zda úlohu splnil, zda se u úlohy podal nebo ne, a v jakém pořadí byla úloha v sekvenci úloh.
6. **Statistiky všech uživatelů:** Požaduji, aby aplikace umožnila uživateli vidět souhrnné statistiky všech uživatelů v podobě grafů. Mezi těmito

statistikami bude průměrný čas splnění úloh, počet „podání se“ u jednotlivých úloh, a průměrná doba splnění úloh v závislosti na pořadí úlohy v sekvenci úloh.

■ 6.2.2 Nefunkční požadavky

Jako hlavní nefunkční požadavky pro aplikaci jsem identifikoval tyto:

1. **Vyvolání „Aha“ momentu:** Požaduji, aby jednotlivé úlohy aplikace vyvolaly v uživateli tzv. „Aha“ moment, který jsem blíže popsal v kapitole 3.
2. **Jednoduchost ovládní:** Požaduji, aby ovládní aplikace bylo pro uživatele intuitivní a nebylo překážkou při řešení jednotlivých úloh.
3. **Dostatečná komunikace s uživatelem:** Požaduji, aby aplikace uživateli dostatečně vyjasnila koncept aplikace pomocí např. nápovědy či vyskakovacích oken.

■ 6.3 Shrnutí kapitoly

V této kapitole jsem popsal návrh řešení jako 3D webovou aplikaci sestávající z šesti úloh, které se v uživateli budou snažit vyvolat tzv. „Aha“ moment (viz kapitola 3). Následně jsem definoval funkční a nefunkční požadavky aplikace.

V následující kapitole definuji použité technologie a nástroje, kterými budu toto řešení implementovat.

Kapitola 7

Použité technologie a nástroje

V této kapitole bych rád definoval technologie a nástroje, specifikoval jak a proč jsem jednotlivé nástroje vybral, a také zodpověděl na otázku, proč jsem nepoužil herní engine.

Aplikaci jsem vytvořil jako webovou aplikaci. Toto řešení jsem si zvolil z dvou hlavních důvodů:

- **Zkušenost s vývojem uživatelského rozhraní webových aplikací:** Jelikož mám zkušenosti v oboru vytváření frontendových webových aplikací, umožnilo mi to vyvinout uživatelské rozhraní rychleji.
- **Zájem o 3D webové aplikace:** Vzhledem k mému zájmu o 3D technologii ve webovém prostředí jsem se chtěl u této práce naučit s těmito technologiemi pracovat.

Pro vytvoření webové aplikace jsem použil obvyklou kombinaci HTML + CSS + JavaScript, což je standardní postup pro vytváření webových aplikací. Následně jsem potřeboval zvolit framework pro JavaScript, framework pro práci s 3D prostředím v JavaScriptu, a program pro modelování 3D modelů. Následuje popis postupu, jak a proč jsem tyto nástroje vybral.

7.0.1 Výběr JavaScript Frameworku

JavaScript framework je z mojí zkušenosti důležitá součást webového vývoje. Usnadňuje práci s obvyklým způsobem používání JavaScriptu. Při výběru jsem zvažoval tyto tři kandidáty: **Vue.js**, **Angular**, a **React**. Tyto tři frameworky jsou obecně uznávány jako nejrozšířenější na trhu.[50] Žádný z těchto frameworků nemá výhodu oproti ostatním, co se integrace 3D frameworku týče. Při výběru JavaScript frameworku jsem zvažoval následující kritéria:

Porovnání JavaScript frameworků			
Název	Familiérnost	Kvalitní dokumentace	Snadnost použití
React.js	Ne	Ano	Ne
Angular	Ne	Ne	Ano
Vue.js	Ano	Ano	Ano

Tabulka 7.1: Výběr JavaScript frameworku.

Familiérnost: Toto bylo hlavní kritérium při výběru frameworku. Je pro mne mnohem jednodušší pracovat s frameworkem, který obvykle používám při vývoji v práci nebo jinde. Zde u mě vyhrává **Vue.js**, protože jsem s tímto frameworkem dokončil několik projektů a je mi blízký.

Kvalitní dokumentace: Dobrá dokumentace je základem každého kvalitního vývojového nástroje a vždycky se snažím zjistit, jak je dokumentace provedená a jak snadno se v ní vyhledávají věci, které pomocí frameworku chceme učinit. Zde u mě vyhrává **Vue.js**, ale dokumentace **Reactu** je také kvalitní a přehledná. Dokumentace **Angularu** mi přišla poněkud komplikovaná a hůře čitelná.

Snadnost použití: U používání frameworku je pro mě důležitá nízká komplikovanost co se syntaxe a používání základních vlastností frameworků týče (např. podmíněné zobrazování prvků). V tomto ohledu mi **Vue.js** přijde jako zdaleka nejlepší, protože přímo pracuje s HTML atributy a je postaven tak, aby bylo intuitivní s ním vyvíjet.

Výsledek aplikace výše zmíněných kritérií na všechny tři frameworky jsem shrnul v následující tabulce:

Jako framework, který použiji pro vývoj aplikace jsem zvolil **Vue.js**. Nejen, že je mi známý a rád s ním pracuji, ale také má kvalitní dokumentaci a používá se snadno, takže mě baví s ním pracovat.

7.0.2 Výběr nástroje pro vytvoření 3D prostředí

Jako nástroj pro vývoj 3D prostředí jsem chtěl použít takový nástroj, který mi umožní vyvíjet 3D přímo na webu. Hlubší zdůvodnění rozhodnutí, proč jsem nechtěl použít dedikovaný nástroj na vytváření her jako např. herní engine, je v kapitole 7.0.6.

K vykreslování 3D objektů a manipulaci s nimi na webu můžu použít tzv. 3D framework, také nazývaný OpenGL framework. Existuje nemalé množství těchto frameworků, mezi kterými je možno zvolit. Tyto kritéria byla pro mě nejdůležitější při výběru 3D frameworku:

Porovnání frameworků				
Název	Existují již vytvořená propojení s Vue.js?	Má framework kvalitní dokumentaci?	Je používán v produkci u velkých společností?	Poznámky
Three.js	Ano	Ano	Ano	Nejvíce rozšířené na trhu
Babylon.js	Ano	Ano	Ano	
CopperLicht	Ne	Ne, pouze výčet tříd metod	Ne	
Phoria.js	Ne	Ne, pouze příklady	Ne	Málo vlastností a podpora
LightGL.js	Ne	Ano	Ne	Nemá dostatek vlastností

Tabulka 7.2: Výběr 3D frameworku.

Jednoduchost propojení s Vue.js: Vzhledem ke speciální povaze JavaScript frameworku, který jsem si zvolil, jsem hledal již existující propojení mezi Vue.js a Three.js, které bych případně mohl využít. Zde u mě není jasný kandidát na použití, myslím že **Three.js** a **Babylon.js** jsou v tomto ohledu srovnatelně kvalitní.

Existující dokumentace / příklady: Pro komplexní vývojový nástroj jako 3D JavaScript framework je potřeba kvalitní, přehledná dokumentace. Framework by měl také mít velké množství různých dem a příkladů použití. U tohoto kritéria u mě jasně vyhrává **Three.js**, protože má dle mého názoru nejlépe provedenou dokumentaci a nabízí desítky dem, představujících různé jeho vlastnosti či obvyklé případy použití.

Současné použití na webu: Ohlídím se také na to, jak je framework používán na webu. Je používán známými společnostmi či weby? Je dostatečně vyspělý, aby mohl být spolehlivě používán? V tomto ohledu u mě také vyhrává **Three.js**. Je používán u velkého množství webových stránek či experimentů, a stejně tak je používán velkými společnostmi, jako například firmou Facebook pro jejich JavaScript řešení pro virtuální realitu, React VR.

Výsledek aplikace těchto kritérií na nejznámější 3D frameworky jsem shrnul v následující tabulce:

Nakonec jsem se rozhodl použít knihovnu **Three.js**, protože je dle mého názoru nejrozšířenější řešení s nejvíce propracovanou dokumentací a příklady. Ačkoli **Babylon.js** je také kvalitní a rozšířené řešení, jejich dokumentace mi přišla nepřehledná a obtížně navigovatelná.

■ 7.0.3 Výběr nástroje pro vytvoření 3D modelů

Pro vytváření samotných 3D modelů jsem použil volně dostupný software **Blender**. Ačkoliv fakulta v rámci výuky poskytuje sbírku mnohem složitějších (a mnohem dražších) programů, jako je například 3DS Max či Maya od firmy Autodesk, **Blender** zůstává díky své jednoduchosti používání a díky tomu, že je zdarma, oblíbené řešení u většiny vývojářů pro jednoduché 3D modelovací projekty. [52] Rozhodl jsem se tedy pro tento software kvůli jednoduchosti používání a širokou komunitou na internetu, která tvoří mnoho naučných videí.

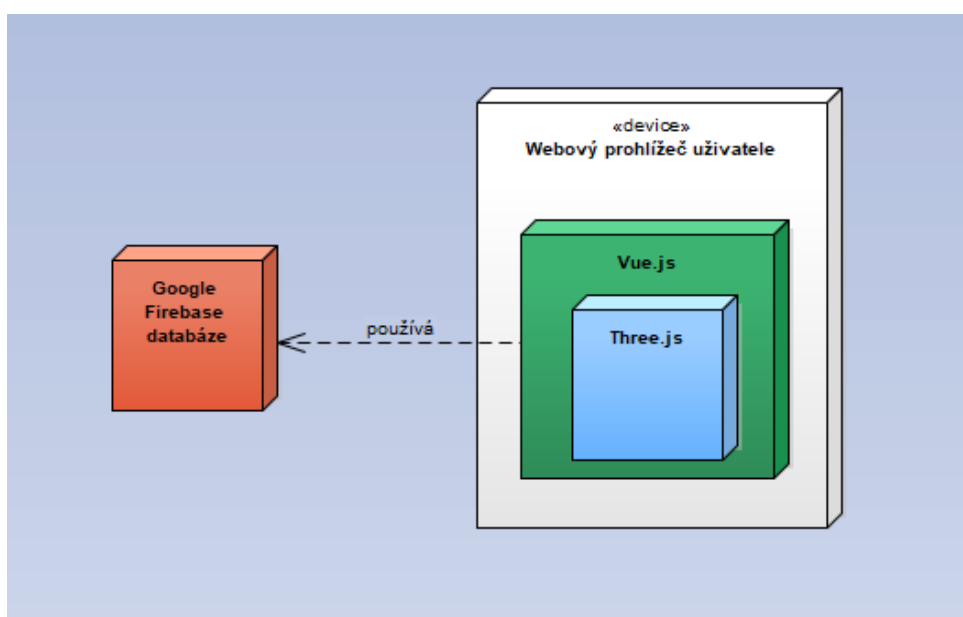
■ 7.0.4 Výběr databáze pro ukládání analytických dat

Pro ukládání dat jsem se rozhodl použít tzv. Google Firebase Firestore databázi. Tuto službu jsem zvolil z několika hlavních důvodů:

- **Je zadarmo:** Firebase poskytuje svoji databázi zadarmo pro méně než 50,000 přečtení za den. To je více než dostačující pro potřeby mé aplikace.
- **Jsem s ní dobře seznámen:** Firebase Firestore databázi jsem již používal na předchozích školních projektech. Bylo tedy pro mne jednodušší začít s ní pracovat a propojit jí s aplikací.
- **Je dobře propojitelná s Vue.js:** Pomocí knihoven, jako např. **vuefire**, mohu jednoduše využít tuto databázi s Vue, např. svázáním proměnné v aplikaci s entitami v databázi. Více o tomto později, v kapitole 8.4.

■ 7.0.5 Model nasazení

Pro lepší ilustraci používaných technologií jsem vytvořil model nasazení, vyobrazený na obrázku níže. (Obrázek 7.1)



Obrázek 7.1: Model nasazení

7.0.6 Proč jsem nepoužil herní engine?

Jelikož se všech šest úloh mé aplikace dá přirovnat k jakýmsi minihrám, herní engine jako např. Unity [54] (ale i mnohá další řešení) by mi mohl poskytnout jednoduché nástroje pro jejich vytváření.

Herní engine jako např. Unity nabízí určité pomůcky pro vývoj, například jednoduchou manipulaci s objekty ve scéně, manipulaci světla či knihovny a předpřipravené modely pro kolize či raycasting. Toto by umožnilo vytvořit úlohy mnohem snadněji, ale nepřineslo by mi to zajímavý náhled na to, jak 3D svět operuje na základní úrovni a jak s ním manipulovat, jelikož jsou tyto věci z programovacího hlediska pro uživatele vyřešené.

Naopak vytvoření 3D webové aplikace mi umožňuje např. snadněji vytvořit uživatelské rozhraní v JavaScriptu, a umožňuje mi využít mých předchozích znalostí z vytváření webových stránek. Tyto však nejsou nejdůležitější důvody, proč jsem se pro toto řešení rozhodl.

Od začátku jsem pro tuto bakalářskou práci chtěl vytvořit 3D aplikaci specificky ve webovém rozhraní za pomoci webových technologií. Tato technologie mě zaujala již předtím a chtěl jsem se s ní více seznámit. Je pravda, že Unity poskytuje možnost vytvářet hry, které fungují ve webovém browseru, ale chtěl jsem tento projekt pojmout i jako určitou technickou výzvu, spíš než jen výzvu ve vymýšlení zajímavých úloh.

■ 7.1 Shrnutí kapitoly

V této kapitole jsem shrnul, proč a jak jsem vybral různé nástroje pro implementaci aplikace. U každého nástroje jsem popsal hlavní důvod, proč jsem ho vybral, a u dvou nejdůležitějších (JavaScript framework a 3D framework) jsem vytvořil tabulky ilustrující, podle jakých kritérií jsem nástroj vybíral.

Kapitola 8

Implementace aplikace

V této kapitole se budu věnovat implementaci samotné aplikace. Nejprve popíšu adresářovou strukturu aplikace a jakým způsobem se kompiluje do jen několika souborů. Následně se budu věnovat uživatelskému rozhraní programovanému ve Vue.js a popíšu jeho hlavní prvky. Nakonec se rozepíšu o způsobu vytváření úloh, nejprve obecně a následně zmíním způsob, jakým jsem implementoval vybrané mechaniky některých úloh. Nakonec rozepíšu o analytické části, jakým způsobem jsem jí řešil a implementoval.

8.1 Adresářová struktura aplikace

Aplikace je vystavěna podobně, jak bývá zvykem u frontendových JavaScript aplikací. V kořenovém adresáři aplikace existuje soubor nazývaný se **index.html**, který je zodpovědný za vykreslení základní HTML struktury. Do něj jsou následně automaticky vloženy zkompileované **JavaScript a CSS soubory**.

Tyto soubory jsou vytvořeny automatickou kompilací souborů, které leží oddělené v různých adresářových složkách. Tato kompilace je uskutečněna skrze **Webpack**, což je technologie umožňující kompilace složitých adresářových struktur do základních souborů, jednoduše vložitelných do jakékoliv stránky.

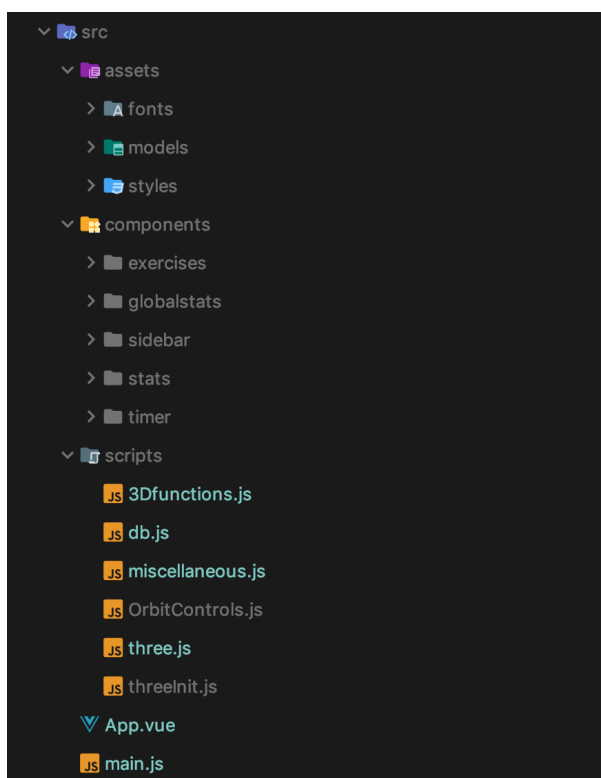
Adresářová struktura, se kterou pracuji, je zhruba rozdělená do složek **assets**, **components** a **scripts**. Adresářovou strukturu je možné najít na obrázku 8.1.

Ve složce **assets** jsou soubory potřebné pro běh aplikace, které ale nevykonávají JavaScriptový kód. Tato složka je rozdělena na tři podsložky, jmenovitě **fonts**, **models** and **styles**. V podsložce **fonts** jsou fonty, které jsem v aplikaci použil. Ve složce **models** jsou 3D modely, které jsem vytvořil pro aplikaci.

Třetí podsložka je **styles** - to jsou CSS styly, které jsem použil k změně vzhledu jednotlivých částí stránky.

Ve složce **components** jsou jednotlivé Vue komponenty, ze kterých se aplikace skládá. Více o specifických Vue komponentech se rozmluvím v kapitole o uživatelském rozhraní. Komponenty jsou například všechny úlohy (od 1. do 6.), hlavní obrazovka, postranní menu, či jednotlivé komponenty zobrazující statistiky v aplikaci. V komponentech jednotlivých úloh ukládám i pro ně specifické skripty, jako například funkcionalitu specifickou pro jednotlivé úlohy.

Ve složce **scripts** se pak nachází jednotlivé skripty, které nespádají pod jednotlivé komponenty. Například v souboru **3Dfunctions.js** jsou funkce, které se starají o inicializaci 3D prostředí. V souboru **db.js** jsou pak například funkce, které pracují s databází (o té později v kapitole 8.4).



Obrázek 8.1: Adresářová struktura aplikace.

8.2 Zpracování uživatelského rozhraní

Uživatelské rozhraní jsem chtěl jednoduché a minimalistické, s několika hlavními prvky. Rád bych v následujících podkapitolách popsal nejdůležitější

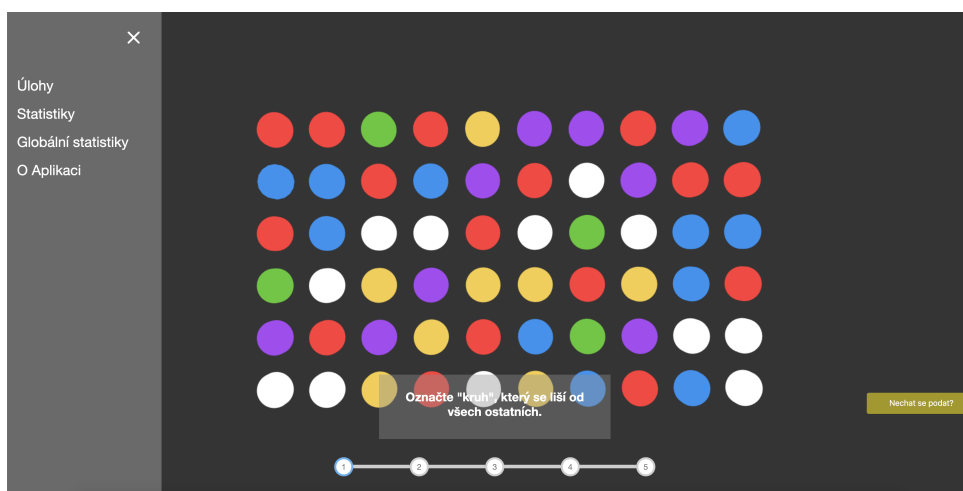
prvky uživatelského rozhraní a jakou funkci vykonávají.

8.2.1 3D okno

Nejdůležitějším prvkem je 3D okno, u něhož jsem chtěl, aby zabíralo celou plochu obrazovky. Všechny prvky tedy musely být umístěny „na“ toto okno. Všechny prvky, které okno překrývaly, tedy musely mít nastaveno, aby jejich pozice byla **absolutní**, tedy aby nerespektovaly pozice jiných prvků a byly na obrazovce tam, kde je umístím.

Důležitou částí 3D okna je tlačítko „Nechat se podat?“ které umožní uživateli změnit pohled na scénu tak, aby mu řešení přišlo okamžitě zřejmé. Tento pohled na věc by měl přinést tzv. „Aha“ moment, o kterém jsem mluvil v kapitole 3.

Nakonec bych rád vyzdvihнул tzv. „progress bar“, neboli ukazatel postupu, který uživateli ukazuje, na jakém místě v aplikaci je. Uživateli umožní také jít kamkoli zpátky, pokud se bude chtít zpětně podívat na úlohu, kterou už předtím dokončil.



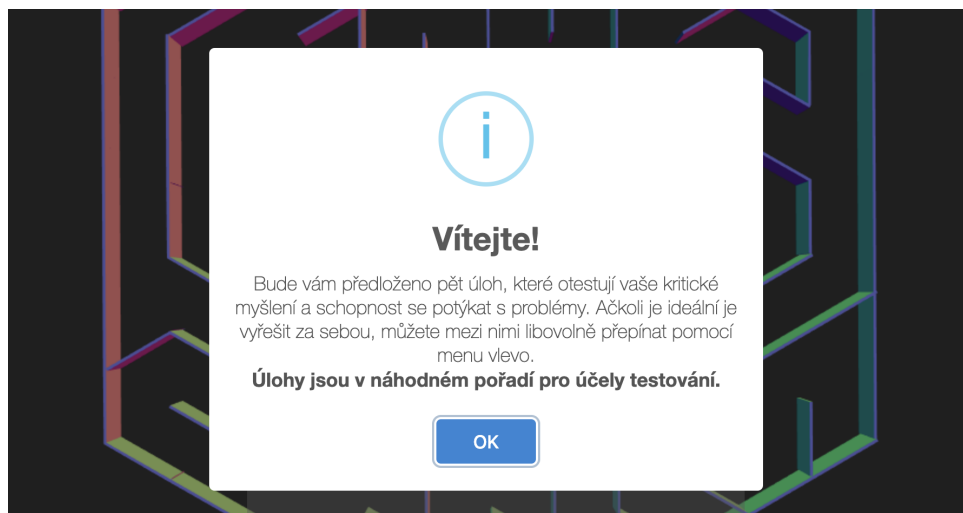
Obrázek 8.2: Hlavní okno aplikace

8.2.2 Popup okénka a tipy

Vzhledem k povaze problému a tím, že 3D prostor v dané úloze zabírá celou obrazovku, bylo ideální využít modálních oken pro reprezentaci komunikace s uživatelem (Modální okno je takové, které se objeví „přes“ stránku, a zakrývá jí nebo její část. Můžeme jej vidět níže na obrázku 8.3.

Tyto modální okna jsou využívána pro mnoho účelů, jako například navigování

uživatele skrze aplikaci, vytvoření nápovědy pro jednotlivé úlohy a další.

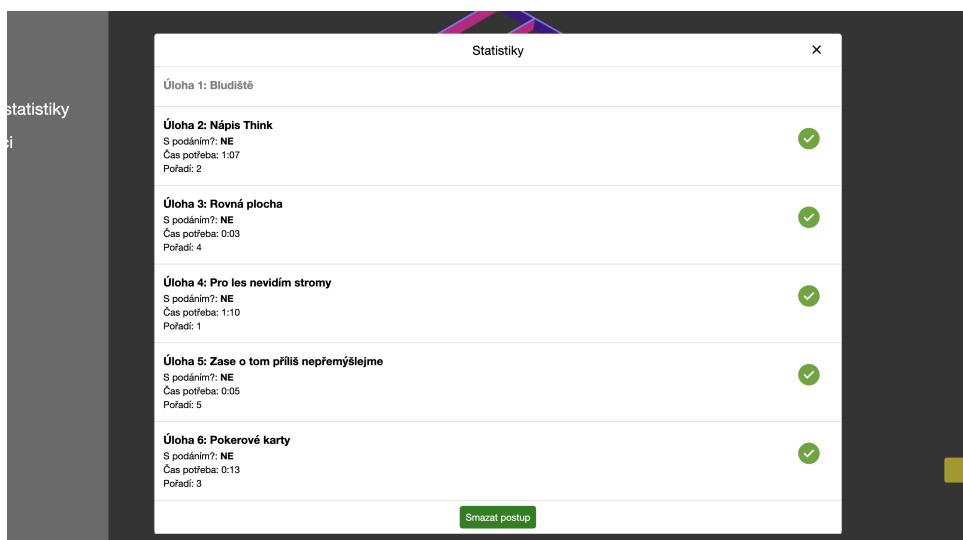


Obrázek 8.3: Popup okno při spuštění aplikace

8.2.3 Uživatelské statistiky

Vyskakovací okno s uživatelskými statistikami slouží k tomu, aby informovalo uživatele o jeho postupu v aplikaci. Okno nabízí pro každou úlohu tři hlavní informace - zda uživatel splnil danou úlohu nebo ne, za jak dlouho uživatel danou úlohu vyřešil a zda se u dané úlohy nechal podat nebo ne. Také, v případě náhodného průběhu (viz kapitola 8.3.2) umožní vidět, na jakém pořadovém místě ona úloha při řešení byla. Okno s uživatelskými statistikami můžeme vidět níže, na obrázku 8.4.

V uživatelských statistikách také nabízím uživateli možnost smazat svůj postup - v případě, že by chtěl zkusit sekvenci úloh znovu, tato funkce to umožňuje. Toto uživateli vymaže současné statistiky a nabídne znovu sekvenci pěti úloh.

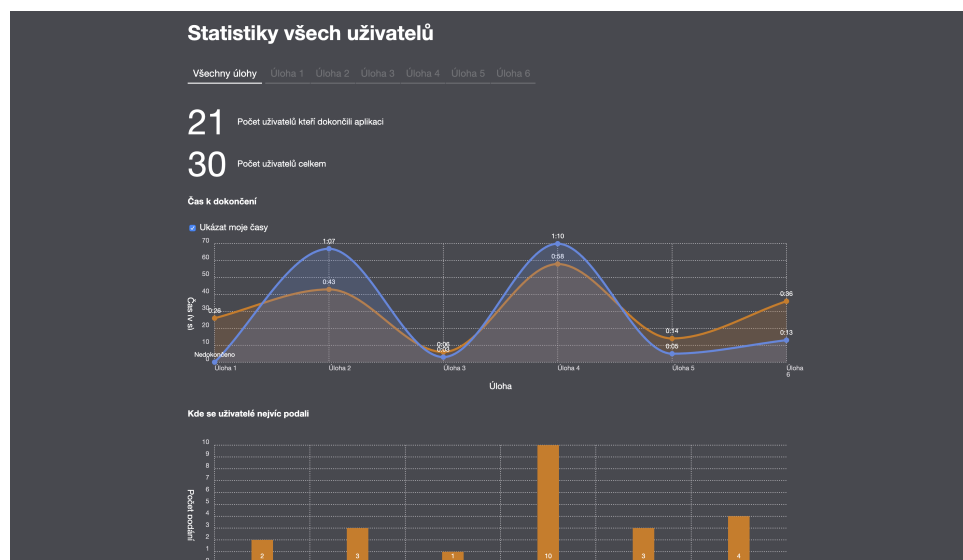


Obrázek 8.4: Uživatelské statistiky

8.2.4 Statistiky všech uživatelů

Statistiky všech uživatelů jsou statistiky, které dávají uživateli možnost se podívat na souhrnné statistiky všech uživatelů. V tomto okně je nejdůležitější prvek grafu - tyto grafy vyjadřují v grafické podobě (viz obrázek 8.5) nejdůležitější statistiky, jako například průměrnou dobu řešení úloh, u kterých úloh se uživatelé podávali nejčastěji, nebo podívat se, jaké má pořadí úloh vliv na jejich dobu řešení.

Tyto grafy jsou propojeny s databází, kterou používám pro ukládání analytických dat, tedy data jimi vykreslovaná jsou současná a aktualizovaná. Více o propojení s databází píšou v kapitole 8.4.1. Data se také aktualizují, kdykoli uživatel znovu otevře stránku. Uživatel si také u některých grafů může zvolit, zda chce zobrazit také své vlastní statistiky.



Obrázek 8.5: Statistiky všech uživatelů - úvodní stránka.

8.3 Zpracování jednotlivých úloh

V této kapitole popíšu způsob vytváření úloh pro tuto aplikaci. Nejprve popíšu obecnou strukturu každé úlohy a následně dám prostor jednotlivým úlohám. U několika úloh se také rozepíšu o tom, jak jsem naprogramoval jejich funkcionalitu.

8.3.1 Obecný postup při vytváření úloh

Každá úloha má hlavní soubor, který se nazývá **exerciscripts.js**. Tento soubor je umístěn u každé úlohy. Má na starosti všechny specifické funkce pro běh úlohy, jako například import a rozvržení 3D objektů ve scéně, či reakce 3D objektů na akce uživatele.

Reakce na tyto akce byly u několika úloh vytvořeny jako tzv. JavaScript listeners - tedy speciální událost ve webovém prohlížeči, na kterou moje aplikace reaguje. Pro ilustraci, tyto události jsou například **mousemove**, tedy kdykoli když uživatel pohne myší, nebo **click**, tedy když uživatel klikne myší do obrazovky.

Vytvoření scény obvykle probíhá takto - pokud neexistuje ještě žádná scéna, inicializuje se svět, kamera a ovládání kamery. Pokud scéna již předtím vytvořena byla (např. když přepínáme z jedné úlohy na druhou), tak pouze odstraníme všechny již existující 3D objekty. Následně načteme model z jeho příslušné složky (více o tomto v kapitole 8.1, případně importujeme i

jeho materiály. Oba tyto typy souborů vyžadují svůj vlastní „Loader“, tedy skript, který tyto soubory zpracuje. Následně se jednotlivé aplikace chovají dle vlastních pravidel. Více o těchto pravidlech v následující kapitole.

8.3.2 Krátká zmínka o náhodné sekvenci úloh

Jak jsem již popsal v kapitole 5.2, u uživatelského testování jsem úlohy nabídnul uživateli v náhodném pořadí.

Pro účely uživatelského testování jsem umožnil v kódu jednoduše přepnout (pomocí atributu **data.randomized** v hlavním komponentu **App.vue**), zda budou úlohy v náhodné sekvenci nebo v sekvenci, kterou jsem původně zamýšlel (tedy že ze začátku bude uživateli známá úloha jako bludiště a ke konci bude úloha, která nás vrátí „zpět do reality,“). Tato náhodná sekvence přetrvává, i když uživatel zavře a znovu otevře aplikaci, dokud například nepřepneme parametr **randomized** zpět na **false**.

8.3.3 Bludiště

Pro vytváření bludiště jsem použil generátor rozvržení bludiště na webu `maze-generator` [53].

Zvolil jsem hexagonální bludiště pro vyšší zajímavost úlohy. Importoval jsem ho do Blenderu a nechal jsem podle něj automaticky nakreslit křivku, kterou jsem následně extrudoval. Toto vytvořilo bludiště, které jsem následně uzavřel, aby se nedalo vyřešit.

U této úlohy jsem se také setkal s několika zajímavými výzvami. Jak jsem zmínil již v kapitole 7.0.6, jsou to typy úloh, kde by herní engine měl již předpřipravená řešení.

Pohyb v rámci os kamery, ne v rámci os světa

Pointa úlohy byla, že po otočení kamery můžeme z bludiště jednoduše vyjet kuličkou. Bylo tedy potřeba, aby po otočení kamery pohyb nahoru znamenal, že kulička vyjede nahoru, tedy osy jejího pohybu se přizpůsobovaly dle pozice kamery. Toho jsem následně dosáhl díky vytvoření nových vektorů kalkulovaných od vektoru mířícího ze středu kamery přímo dopředu, a díky tomu se koule vždy pohybuje po osách, které jsou intuitivní podle pohledu kamery.

Kolize s bludištěm

Bylo důležité, aby uživatel nebyl schopný vyjet z bludiště jednoduše stěnami.

„mousemove” DOM event, který se spustí vždy, když uživatel pohne s myší, více v příloze B), který má jako parametry přesnou polohu myši na ose X a Y. Posléze vyšleme paprsek ze středu kamery na polohu myši a bod, kde protne plochu světa, označíme zároveň jako střed označovací plochy. Tento algoritmus se odehraje vždy, když uživatel pohne myší v 3D okně.

Změna barvy po kolizi s objekty

Vzhledem k tomu, že chceme aby uživatel mohl zaznamenat moment, kdy plocha neprotíná žádné objekty, potřebujeme mu pomocí barev nějak indikovat, jestli daná plocha protíná nebo neprotíná jakýkoli objekt.

U tohoto řešení jsem si vypomohl skrze tzv. bounding box (více v příloze B), což je pomocná vlastnost 3D objektu, kterou Three.js nabízí. Pro každý objekt vytvoří Three.js jeho aproximaci jako kvádr. Dále nabízí funkci, která se nám pro řešení tohoto problému dobře hodí, a to možnost zjistit, zda se dva bounding boxy překrývají.

Tohoto jsem tedy využil v aplikaci. Vzhledem k nízkému počtu objektů jsem si dovolil „hrubou silou” (tedy pomocí for cyklu) zkontrolovat protnutí bounding boxu čtverce, který ovládá uživatel, se všemi objekty ve scéně. Toto má jedinou nevýhodu například u válcovitých nebo jehlanovitých objektů, a to že v samotném rohu jejich bounding boxů bude místo, kde se opticky čtverec a objekt neprotíná, ale kolize bude platit. Věřím ale, že možnost takto přesného umístění čtverce ovládaného uživatelem není důležitá pro vyvolání „Aha” momentu či řešitelnost úlohy.

■ 8.3.6 Pro les nevidím stromy

3D modelování této úlohy bylo poměrně jednoduché. Vytvořil jsem kruh, což je jeden ze základních tvarů v programu Blender. Tento kruh jsem extrudoval na požadovanou úroveň, což spíše než tvar kruhu vytvořilo tvar velice nízkého válce (dále v této podkapitole jej budu označovat jako „kruh,“). Pro low-poly kruh jsem využil filtr decimate, speciálně jeho funkci un-subdivide. Ten odebírá vertexy, které existují mezi jednotlivými „stěnami” kruhu, efektivně vytvářející mnohostranný n-úhelník. Ten má očividně daleko od tvaru kruhu, pokud se na něj nekoukáme z dostatečně velké dálky.

U implementace této úlohy jsem řešil převážně generaci náhodné matice kruhů, barev a schopnost kruhů označovat se po projetí myši.

Pro kruhy jsem použil matici o rozměrech 10x6, což mi přišlo jako nejlepší poměr mezi jednoduchostí a vyřešitelností úlohy. Zároveň, generování velkého množství (stovek až tisíce) kruhů je náročné na výkon počítače.

Generace náhodné matice kruhů

Při generaci náhodné matice kruhů jsem použil obvyklý algoritmus pro generaci čtvercové matice (viz obrázek 4.6). Algoritmus využívá dva for cykly a podle indexu vzdaluje počáteční pozici nově generovaného kruhu. Pro náhodnou selekci barvy jsem využil náhodný výběr z kolekce, využívajíc funkce z knihovny Math, které existují v JavaScriptu v základní podobě.

Po vygenerování kruhu algoritmus náhodně vymění jeden kruh za n-úhelník, který existuje v jiném modelu. Přidělím mu náhodnou barvu ze stejné kolekce jako ostatní kruhy.

Je možné aby jedna barva existovala na pouze jednom kruhu, nebo na všech, ale pravděpodobnost, že se něco takového stane, je extrémně nízká.

8.3.7 Zase o tom příliš nepřemýšlejme

V této úloze neexistovalo modelování – pouze jsem použil obyčejnou kouli, kterou jsem následně pokryl zajímavou texturou. Po experimentování jsem našel zajímavou texturu složenou pouze z trojúhelníků, a mapování textury na kouli vytvořilo zajímavé, soustředné kružnice, které napomohly unikátnosti a zajímavosti úlohy. Díky nim se uživatel soustředí na identifikaci tvaru na textuře koule, což přispívá k výslednému „Aha” efektu.

Tato úloha mi nezpůsobila žádné výraznější problémy s jejím vývojem, jelikož její řešení nevyžaduje žádnou komplikovanou interakci na straně uživatele. Největší technický problém v případě tohoto cvičení bylo importovat texturu na kouli.

8.3.8 Pokerové karty

Pro vytvoření modelů karet jsem vytvořil jednoduchou plochu, zakřivil její rohy a pokryl texturou klasických pokerových karet, kterou jsem našel na internetu. Tyto karty jsem vytvořil tři - J (tedy Jack), K (tedy King), a A (tedy Ace).

V této úloze jsem převážně musel vytvořit způsob, jak vyměňovat sekvenci karet tak, aby reagovala na uživatele. Pro toto jsem použil techniky z předchozích úloh - jmenovitě označování jednotlivých objektů při kliknutí myši. Karta se vždy po kliknutí myši vymění s kartou na pravé straně, případně přeskočí na začátek. Při vytváření této úlohy jsem budoval na zkušenostech z vytváření předchozích úloh.

U této úlohy jinak nebyly žádné neobvyklé programovací výzvy, se kterými bych se musel potýkat.

8.4 Analytická část aplikace

Analytická část aplikace zahrnuje především dva aspekty aplikace - a to okno s uživatelskými statistikami (podkapitola 8.2.3) a globálními statistikami (podkapitola 8.2.4). V této podkapitole popíšu nejprve způsob propojení s Google Firebase databází (viz kapitola 7.0.4), na kterou je aplikace napojená a následně popíšu, jakým způsobem jsem vytvořil jednotlivé grafy.

8.4.1 Propojení s databází Google Firebase

Jelikož databáze Google Firebase je postavena na technologii NoSQL, nemá předurčený datový model - v databázi vytváříme tzv. dokumenty, do kterých můžeme dát jakékoliv informace chceme. Toto poskytuje větší svobodu a možnost jednoduché změny databázového modelu, ale také méně rigidní způsob kontroly podoby dat.

V databázi jsem používal pouze jednu kolekci dokumentů, nazvanou **sessions**. Do každého dokumentu v kolekci sessions jsem vkládal informace o jednom průchodu uživatele aplikací, jako např. jestli uživatel dokončil průchod, čas a splnění jednotlivých úloh, a podobně.

Mezi jednotlivými uživateli rozlišuje databáze dle unikátního id, které se zapisuje v tzv. cookie. Cookie je informace, kterou si prohlížeč uloží v počítači, a přetrvává i při otevření a zavření prohlížeče. Pokud již neexistuje cookie s id zapsaný v paměti prohlížeče, aplikace vygeneruje nový cookie a inicializuje průchod aplikací.

Aplikace následně zapisuje informace o chování uživatele - například čas potřebný pro vyřešení úlohy, zda se nechal podat, a podobně. V případě náhodné sekvence úloh (podkapitola 8.3.2) také zapisuje pořadí aplikace v sekvenci.

Data vždy „svázu“ s proměnnou **session** v aplikaci. Tato proměnná následně reprezentuje dokument, který byl vytvořen pro tento průchod uživatele. Toto je mi umožněno pomocí knihovny **vue-fire**. Tato knihovna umožňuje aktualizace v reálném čase, což znamená, že se nemusím starat o to, jestli jsem opět získal data z databáze po tom, co jsem do ní něco vložil. Tento proces se děje automaticky.

8.4.2 Globální statistiky

V sekci Globální statistiky jsem využil pro vykreslování grafů knihovnu **chartist.js**. Tu jsem následně konfiguroval tak, aby byla propojena s mými daty a rozšířil jsem jí o různé animace pro vizuální kvalitu.

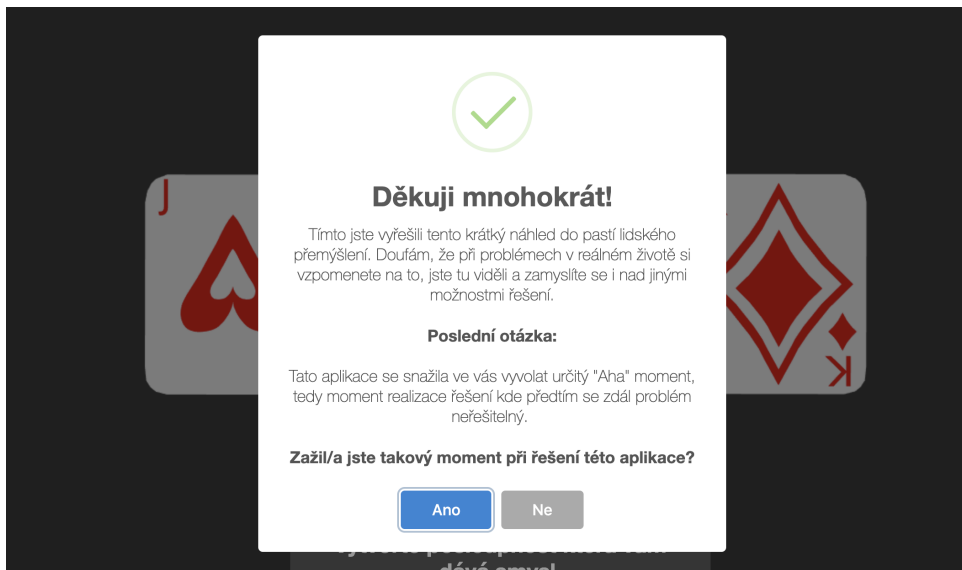
Pro globální statistiky jsou data získávány podobně jako pro uživatelské statistiky - pomocí knihovny **vue-fire** si svážu nějakou proměnnou (například **allSessions**) s databází, a používám ji jako reprezentaci všech dokumentů uložených v databázi. Pak již v samotné aplikaci pracuji s touto proměnnou jako se seznamem dokumentů a odfiltruji z něj potřebná data.

Data pro grafy v okně o globálních statistikách získávám z výše popsané databáze vždy, když uživatel otevře okno - jsou tedy automaticky aktualizované a uživatel se vždy může přesvědčit o tom, jak ovlivňuje průměrné statistiky všech uživatelů.

V případě grafu, který vyjadřuje průměrnou dobu splnění úlohy, se může uživatel podívat na grafické srovnání jeho vlastních časů oproti průměrným časům všech uživatelů aplikace.

8.4.3 Získávání informace o „Aha“ efektu

Pro získání informace, zda uživatel zažil „Aha“ efekt, jsem použil jednoduché vyskakovací okno. Po poslední úloze pokládám uživateli ve vyskakovacím okně otázku, zda při plnění úloh zažil moment „prozření“, aneb „Aha“ moment. Připadalo mi nejlogičtější umístit tuto otázku na konec. Pokud bych se na ní ptal po každé úloze, mohla by působit rušivě. Uživatel má možnost odpovědět buď „Ano“, nebo „Ne.“ Tuto informaci následně ukládám jako proměnnou do databáze. Na následujícím obrázku je možné vidět výše zmíněné vyskakovací okno:



Obrázek 8.6: Vyskakovací okno po ukončení série úloh.

8.5 Shrnutí kapitoly

V této kapitole jsem popsal postup vytvoření aplikace, která je předmětem této bakalářské práce. Nejprve jsem popsal adresářovou strukturu aplikace. Dále jsem mluvil o uživatelském rozhraní, popsal jeho hlavní prvky a k čemu slouží. Následně jsem popisoval vypracování jednotlivých úloh a vyzdvihl specifické implementace v závislosti na úloze. Nakonec jsem popsal uživatelské rozhraní, jak funguje propojení s databází a zajímavé vlastnosti okna s globálními statistikami.

Kapitola 9

Uživatelské testování

Uživatelské testování jsem díky tomu, že jsem vytvořil vlastní systém sbírání a vyobrazování dat, provedl pomocí aplikace samotné. V rámci toho, že uživatelé aplikaci používali, se automaticky zapisovala data o jejich průchodu, jmenovitě čas řešení úloh, jestli „se podali“ nebo ne, jestli úlohu splnili a v jakém pořadí úloha byla v sekvenci úloh. Více o sbírání dat jsem psal v kapitole 8.4.

9.1 Profil uživatele testujícího aplikaci

Testovací skupina pro uživatelské testování sestávala z uživatelů ve věku 18-30 let, kteří mají základní zkušenosti s ovládáním 3D aplikací. V této skupině byli převážně studenti programu Softwarové inženýrství a technologie, kolegové z práce a mně blízcí kamarádi, kteří všichni do těchto kritérií spadají. V době, kdy píšu tuto kapitolu, aplikaci otestovalo 21 uživatelů.

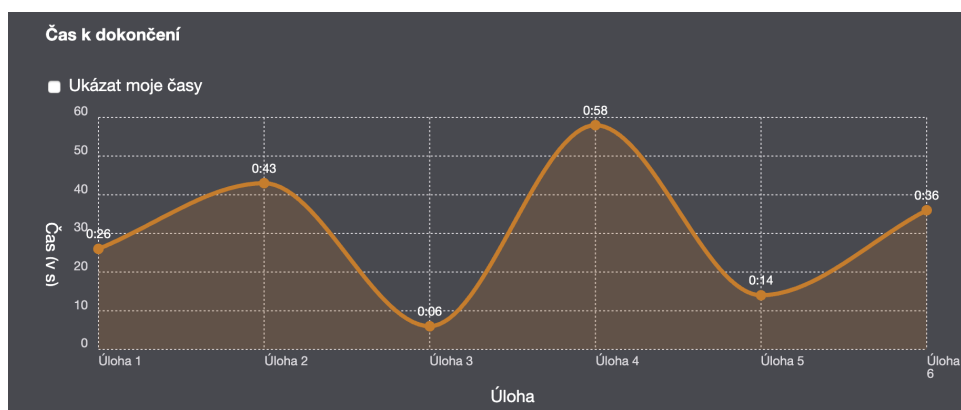
Každému člověku z testovací skupiny byl odeslán odkaz s aplikací a požadavkem, ať dokončí celou sérii úloh. Žádné další instrukce nebyly sděleny.

9.2 Informace z uživatelského testování

V této podkapitole postupně popíšu výsledky jednotlivých grafů, které je v aplikaci možno vidět v rámci okna s globálními statistikami. Jelikož se každý může kdykoliv podívat na aktualizovanou podobu dat, budu v této kapitole pracovat s daty přítomnými v době psaní této kapitoly, tedy přibližně týden před termínem odevzdání práce.

9.2.1 Graf průměrné doby splnění úlohy

První z grafů je graf, který ukazuje průměrnou dobu splnění úlohy. Tento graf agreguje data od všech uživatelů, kteří splnili jednotlivé úlohy, a zprůměruje jejich časy. U tohoto grafu je také možné tyto časy porovnat s časy uživatele. Graf je možné vidět na následujícím obrázku (9.1):



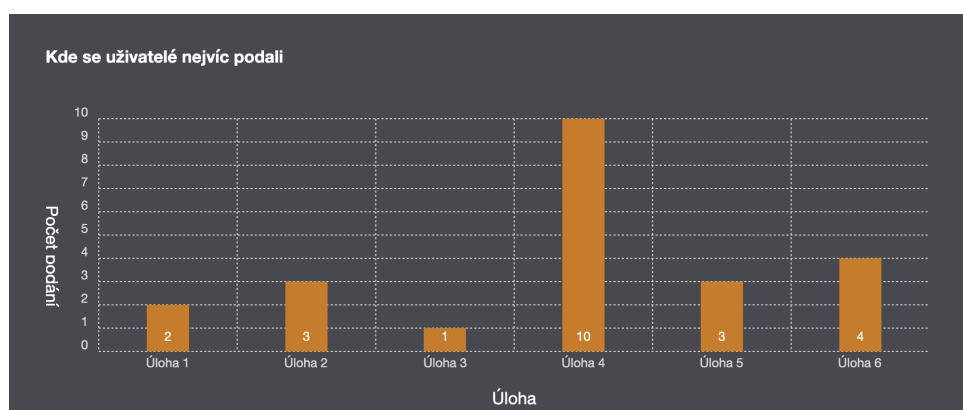
Obrázek 9.1: Graf průměrné doby splnění úlohy.

Jak vidíme, nejdéle uživatelům trvalo splnit Úlohu (kap. 5.1.2) a Úlohu 2 (kap. 5.1.4), zatímco Úloha 6 (kap. 5.1.6) také není daleko od těchto dvou úloh. Toto dle mého názoru může naznačovat, že úlohy 3 (kap. 5.1.3) a 5 (kap. 5.1.5) jsou pro uživatele příliš jednoduché, nebo vyžadují příliš málo interakce na to, aby byly splněny.

Zpětná vazba, kterou jsem od dvou uživatelů dostal potvrzuje, že alespoň úloha 3 (Rovná plocha, viz kap. 5.1.3) vyžaduje příliš malé množství interakce k vyřešení. Vzhledem k tomu, že uživatel ví, že se může pohybovat v 3D prostoru, tak téměř automaticky začne oddalovat obraz, což.

9.2.2 Graf „podání se„

Druhý graf, ze kterého jsem sbíral informace, je graf „podání se“. Tento graf zobrazuje množství „podání se“ tedy množství případů, kde uživatel stiskl tlačítko „podání se“. Toto naznačuje, že uživatel si nevěděl s danou úlohou rady. Graf můžeme vidět na obrázku zde (obrázek 9.2):



Obrázek 9.2: Graf zobrazující celkový počet „podání se”.

Na tomto můžeme vidět, že úloha 4 (kapitola 5.1.4) výrazně „vede” nad ostatními úlohami v počtu podání se. Společně s grafem 1 si tedy myslím, že můžeme říct, že úloha 4 je pro uživatele nejobtížnější. Dle mého názoru toto indikuje, že velké množství potenciálních řešení nejvíce koreluje s vysokou obtížností úlohy.

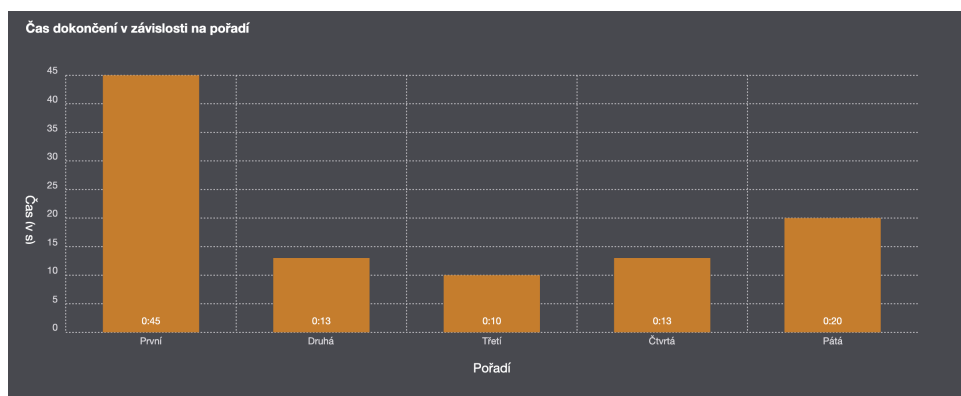
Zpětná vazba, kterou jsem získal od jednoho uživatele, mě také upozornila na to, že uživatel byl v případě čtvrté úlohy jednoduše „líný” se jí zabývat delší dobu, tak se nakonec nechal podat pro svoje vlastní pohodlí.

9.2.3 Grafy časů splnění úloh, závislých na pořadí

Dále, pro každou úlohu existuje graf, který zobrazuje průměrnou dobu splnění v závislosti na pořadí. Vzhledem k povaze grafu je pro něj třeba získat velké množství dat, abychom dostatečně pokryli všechny kombinace úloha/pořadí.

Je tedy obtížné získat kvalitní informace v době psaní této kapitoly. U některých úloh je například pouze jedna instance, kde daná úloha byla v tomto pořadí, a u některých kombinací úloha/pořadí nemám zatím vůbec žádná data. Jelikož spolu pořadí úloh u těchto grafů nemůže souviset (jednotlivé průchody spolu nesouvisí) zvolil jsem histogram pro vyobrazení této informace.

Obecně se však zdá, že pokud je úloha na začátku, tak je častěji řešená déle než pokud je řešená na konci. Zdá se, že toto platí u úlohy 2 (kap. 5.1.2) a dokonce i 5 (kap. 5.1.5). Jako příklad bych rád ukázal graf pro úlohu 2 (obrázek 9.3):



Obrázek 9.3: Graf zobrazující průměrný čas řešení úlohy 2 v závislosti na pořadí.

V případě získání většího množství uživatelů bude možné učinit lepší závěry z těchto dat.

9.2.4 Vyvolání „Aha“ momentu

Na konci sekvence úloh pokládám uživateli otázku, zda při plnění úloh zažil moment „prozření“, aneb „Aha“ moment. V současnosti více než **80% uživatelů** odeslalo pozitivní odezvu, tedy že skutečně „Aha“ moment zažili. Samozřejmě, lze vznést kritiku, že když uživatelé sami hlásí svoje subjektivní pocity, může to ovlivnit výsledná data. Dle mého názoru však není lepšího způsobu získání této informace v rámci jedné webové aplikace.

Tato skutečnost znamená, že jeden z nejdůležitějších cílů této aplikace se zdá být splněn, alespoň v současné testovací skupině - aplikace úspěšně vyvolávala v uživatelích „Aha“ moment.

9.3 Co vyplývá z uživatelského testování

Co se týče obtížnosti úloh, byla přibližně třetina relativně jednoduchá oproti ostatním, což může vést k otázce, zda nejsou tyto úlohy příliš jednoduché na to, aby zprostředkovaly dostatečný „Aha“ moment. U úloh jako např. Rovná plocha či Zase o tom příliš nepřemýšlejme má určitě cenu nad něčím takovým přemýšlet a zjistit, zda by se nedaly nějak zkomplikovat či přímo jinak implementovat.

Co se týče vlivu pořadí úloh na obtížnost, bohužel mám v době psaní této kapitoly nedostatečné množství dat, abych mohl provést kvalitní závěry. U současného malého množství dat se však zdá, že úlohy se řeší tím déle, čím

více na začátku jsou. Toto není překvapivý závěr, ale je třeba jej otestovat s větším množstvím dat.

Uživatelé se také ve většině shodli, že aplikace v nich vyvolává „Aha” moment (dle mého dotazu ukázaném na obrázku 8.3), tudíž dle mého názoru mohou považovat aplikaci za úspěšnou.

Kapitola 10

Další možný vývoj aplikace

Vzhledem k tomu, že mě tato aplikace a téma zajímá, rád bych pokračoval v úpravě a vývoji aplikace do budoucnosti. V této kapitole krátce popíšu různé nápady a způsoby, kterými bych úlohy v aplikaci dál rozšířil.

10.1 Úlohy

V této podkapitole krátce zmíním návrhy na rozšíření jednotlivých úloh.

Úloha 1 - Bludiště

U úlohy 1, bludiště (viz kap. 5.1.1), by bylo možno vybírat bludiště z většího množství přednastavených bludišť, nebo dokonce umožnit uživateli přepínat mezi nimi. Uživateli by se přitom nemuselo říci, že přepínání mezi bludišti ničemu nepomůže.

Úloha 2 - Nápis Think

U úlohy Nápis Think (viz kap. 5.1.2) by bylo možné vytvořit různá jednotlivá písmena nezávisle na nápisu a z těch složit různá slova, která by uživatel musel najít. Také je u této úlohy možnost rozmístit náhodná písmena vzdálená od skutečného nápisu tak, aby mátlá uživatele.

Úloha 3 - Rovná plocha

U třetí úlohy (viz kap. 5.1.3) bych rád učinil úlohu těžší. Toho by se dalo dosáhnout tím, že objekty budou pokrývat širší plochu, nebo že budou vytvářet matoucí obrazec, jako např. spirálu.

Úloha 4 - Pro les nevidíme stromy

U úlohy 4 (viz kap. 5.1.4) by šlo použít jiné tvary místo nízkých válců. Také bych mohl na této úloze otestovat použití animací - např. objekty otáčející se různými rychlostmi.

Úloha 5 - Zase o tom příliš nepřemýšlejme

U úlohy 5 (viz kap. 5.1.5) mě nenapadá žádná možnost rozšíření úlohy. Myslím si ale, že by u ní bylo dobré otestovat různé tvary a pořadí v sekvenci úloh.

Úloha 6 - Postupka z karet

U úlohy 6 (viz kap. 5.1.6) by bylo zajímavé hledat jiné sekvence karet, které by se daly použít. Také by u této úlohy bylo dobré využít 3D prostor více - například otáčením karet.

Kapitola 11

Závěr

Zadáním bakalářské práce bylo vytvořit aplikaci, která bude sloužit k demonstraci využívání různých úhlů pohledu na řešení problému. Tohoto se mělo docílit pomocí vyvolání „Aha” momentu v uživateli (viz kapitola 3).

Dle mého názoru jsem zadání práce splnil. Vytvořil jsem sérii úloh, které demonstrují výhody změny úhlu pohledu a velká část vybrané skupiny uživatelů potvrdila, že v nich aplikace „Aha” moment skutečně vyvolala.

V bakalářské práci jsem popsal tunnel vision, a jaké psychologické úkazy mohou zhoršovat naši schopnost dobře se rozhodovat. Popsal jsem také výše zmíněný „Aha” efekt, který jsem chtěl využít k tomu, aby člověku upevnil povědomí o důležitosti jiného úhlu pohledu na věc.

Pro zlepšení povědomí o tunnel vision jsem se rozhodl vytvořit aplikaci, která sestává ze šesti úloh. Každá z těchto úloh je navržena tak, aby její řešení nějakým způsobem vyžadovalo změnu úhlu pohledu. Může být například založená na tom, že nevíme, že jsme v 3D světě, jsme na špatném místě, či se soustředíme na špatné detaily při řešení. Součástí úloh je vždy tlačítko „nechat se podat”, které má uživateli umožnit „Aha” efekt i v případě, pokud nemůže úlohu vyřešit sám o sobě.

Tento projekt může čtenáři posloužit jako zajímavá „minihra”, či jako ilustrace konceptu tunnel vision zábavným způsobem. Mě osobně tato úloha umožnila lépe se zamyslet nad zajímavým psychologickým úkazem a zlepšit svoje schopnosti ve vývoji webu a práci s 3D prostředím.

Myslím si, že v současnosti aplikace představuje dobře zpracovaný základ, který můžu případně rozšířit za rámec této bakalářské práce. Věřím také, že v současné podobě se dá aplikace použít na hodinách ISP pana doktora inženýra Náplavy. Nejnovější verze vytvořené aplikace se nachází na tomto url: <https://wingender-bakalarska-prace.eu-gb.mybluemix.net/>



Literatura

- [1] Merriam-Webster, Dostupné zde:
<https://www.merriam-webster.com/dictionary/tunnelvision>, Navštíveno 20.5.2019.
- [2] Focus induced tunnel vision in managerial judgment and decision making: The peril and the antidote, Steven S. Posavac , Frank R. Kardes, J. Joško Brakus, Organizational Behavior and Human Decision Processes, listopad 2007, str. 102-111, Dostupné zde:
<https://www.sciencedirect.com/science/article/pii/S0749597810000579> Navštíveno: 28. 7. 2019
- [3] The Psychology of Judgment and Decision Making, nakladatelství McGraw-Hill, 1993, Plous, Scott, str. 233
- [4] How we know what isn't so: The fallibility of human reason in everyday life., nakladatelství The Free Press, Gilovich, 1991.
- [5] Confirmation bias: A ubiquitous phenomenon in many guises, Nickerson, Economics Letters, červen 1998, s. 179, Dostupné zde:
https://www.researchgate.net/publication/280685490_Confirmation_Bias_A_Ubiquitous_Phenomenon_in_Many_Guises
Navštíveno 14.7.2019
- [6] Improving prosecutorial decision making: Some lessons of cognitive science, Burke, A., William Mary Law Review, duben 2005, s. 10, Dostupné zde:
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=707138
Navštíveno 12.7.2019
- [7] Scientific Autobiography and Other Papers, Planck, Max, 1950, nakladatelství Philosophical Library. Dostupné zde:
https://www.astrofilibresciani.it/Biblioteca_UAB/

Biblioteca/A%20Scientific%20Autobiography%20and%20Other%20Papers.pdf Navštíveno 14.7.2019

- [8] Einstein: Jeho život a vesmír (2007), Walter Isaacson, str. 299, nakladatelství Paseka.
- [9] Understanding the limits of limiting instructions: Social psychological explanations for the failures of instructions to disregard pretrial publicity and other inadmissible evidence, Lieberman, J. D. Arndt, J., *Psychology Public Policy and Law*, září 2000, Dostupné zde:
https://www.researchgate.net/publication/232574707_Understanding_the_limits_of_limiting_instructions_Social_Psychological_Explanations_for_the_Failures_of_Instructions_to_Disregard_Pretrial_Publicity_and_Other_Inadmissible_Evidence Navštíveno 13.7.2019
- [10] Perseverance in self-perception and social perception: Biased attributional processes in the debriefing paradigm, Ross, L., Lepper, M. R., Hubbard, M., *William Mary Law Review*, duben 1975, Dostupné zde:
https://www.researchgate.net/publication/21969171_Perseverance_in_Self-Perception_and_Social_Perception_Biased_Attributional_Processes_in_the_Debriefing_Paradigm Navštíveno 15.7.2019
- [11] The „saw-it-all-along” effect: Demonstrations of visual hindsight bias., (Harley, E. M., Carlsen, K. A., Loftus, G. R., *Journal of Experimental Psychology*., 2004, Dostupné zde:
<http://faculty.washington.edu/gloftus/Downloads/HarleyCarlsenLoftus.pdf> Navštíveno 15.7.2019
- [12] The multiple dimensions of tunnel vision in criminal cases, Findley, K. A., Scott, M. S., *Wisconsin Law Review*, 2006, Dostupné zde:
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=911240 Navštíveno 15.7.2019
- [13] Outcome bias in decision evaluation, Baron, J. Hershey, J. C., *Journal of Personality and Social Psychology*, 1988, Dostupné zde:
<https://www.sas.upenn.edu/~baron/papers.htm/judg.html> Navštíveno 15.7.2019
- [14] Overcoming the Inevitable Anchoring Effect: Considering the Opposite Compensates for Selective Accessibility, Mussweiler, T., Strack, F., Pfeiffer, T., *Personality and Social Psychology Bulletin*, listopad 2000, Dostupné zde:
<https://journals.sagepub.com/doi/10.1177/01461672002611010> Navštíveno 15.7.2019
- [15] The cognitive psychology of gambling, Mark D. Griffiths, *Journal of Gambling Studies*, březen 1990, Dostupné zde:

- <https://link.springer.com/article/10.1007/BF01015747> Navštíveno 28.7.2019
- [16] Wrongful Convictions: The Effect of Tunnel Vision and predisposing Circumstances in the Criminal Justice System, Bruce A. MacFarlane, září 2008, Dostupné zde: https://www.attorneygeneral.jus.gov.on.ca/inquiries/goudge/policy_research/pdf/Macfarlane_Wrongful-Convictions.pdf Navštíveno 28.7.2019
- [17] Echo chamber and trench warfare dynamics in online debates, Karlsen, R., Steen-Johnsen, K., Wollebæk, D., Enjolras, B., European Journal of Communication, 32(3), s. 257–273., Dostupné zde: <https://journals.sagepub.com/doi/full/10.1177/0267323117695734> Navštíveno 28.7.2019
- [18] Diversification: best practices of the leading companies, Graham Kenny, Journal of Business Strategy, 33(1), s. 12-20, Dostupné zde: <https://docplayer.net/6559177-Diversification-best-practices-of-the-leading-companies.html> Navštíveno 28.7.2019
- [19] CNN Money, GM Recalls 30 Million Vehicles, Dostupné zde: <https://money.cnn.com/2014/10/04/news/companies/gm-recalls-30-million-vehicles/index.html> Heather Long, 4.10.2014. Navštíveno 1.4.2019.
- [20] Time.com, These Are The 69 Words GM Employees Were Forbidden from Using, Dostupné zde: <http://time.com/103737/gm-words-nhtsa-general-motors> Sam Frizell, 17.5.2014. Navštíveno 1.4.2019.
- [21] Anchoring and cognitive ability, Oscar Bergman and co., Economics Letters, vydání 107, duben 2010, s. 98, Dostupné zde: <https://www.sciencedirect.com/science/article/pii/S0165176509004327> Navštíveno 16.5.2019.
- [22] The Road Ahead - Bill Gates Comdex Keynote, Dostupné zde: <https://www.youtube.com/watch?v=o000Xjpjvfc> 1995. Navštíveno 10.5.2019.
- [23] BusinessInsider.com, Microsoft CEO Satya Nadella says Bill Gates' original mission 'always bothered me', Dostupné zde: <https://www.businessinsider.com/microsoft-ceo-satya-nadella-bothered-by-bill-gates-mission-2017-2> Matt Weinberger, 21.2.2017. Navštíveno 3.4.2019.
- [24] The New York Times, Kodak's First Digital Moment, James Estrin, 2015, Dostupné zde: <https://lens.blogs.nytimes.com/2015/08/12/kodaks-first-digital-moment/> Navštíveno 20.5.2019.

- [25] Groupe Renault, Busan Plant (RSM), Dostupné zde: <https://group.renault.com/en/our-company/locations/our-industrial-locations/busan-plant/> Navštíveno 20.5.2019.
- [26] BBC.com, Renault takes over Samsung, Dostupné zde: <http://news.bbc.co.uk/2/hi/business/725235.stm>, 25.4.2000. Navštíveno 20.5.2019.
- [27] A literature review of the anchoring effect, The Journal of Socio-Economics únor 2011, str. 35-42, Dostupné zde: <https://www.sciencedirect.com/science/article/pii/S1053535710001411> Navštíveno 30.5.2019.
- [28] Putting adjustment back into the anchoring and adjustment heuristic: differential processing of self-generated and experimenter-provided anchors, Psychological Science, prosinec 2001, str. 391-396, Dostupné zde: <https://journals.sagepub.com/doi/10.1111/1467-9280.00372> Navštíveno 30.5.2019.
- [29] Mindtools.com, Awards page, Dostupné zde: <https://www.mindtools.com/awards.html> Navštíveno 3.4.2019.
- [30] Mindtools.com, Scenario Analysis: Exploring Different Futures, Dostupné zde: https://www.mindtools.com/pages/article/newSTR_98.htm Navštíveno 19.5.2019.
- [31] iMindMap.com, Kurz, Certified Training in Mind Mapping and Applied Innovation, Dostupné zde: <https://imindmap.com/training/online-courses/brainstorming-that-works/> Navštíveno 29.3.2019.
- [32] Businessballs.com, O firmě, Dostupné zde: <https://www.businessballs.com/about/> Navštíveno 11.5.2019.
- [33] Coursera.com, Business Strategy Specialization Course, Dostupné zde: <https://www.coursera.org/specializations/business-strategy> Navštíveno 21.5.2019.
- [34] ManageMagazine.com, Tunnel Vision – Its Drawbacks and How to Stay Clear of it, Dostupné zde: <https://managemagazine.com/article-bank/self-handicapping-leadership/tunnel-vision-its-drawbacks-and-how-to-stay-clear-of-it>, Decker and Mitchell, 15.12.2016. Navštíveno 3.1.2019.
- [35] Indecisiveness and informational tunnel vision, Eric Rassin and co., Personality and Individual Differences vydání červenec 2008, s. 96, Dostupné zde: <https://www.sciencedirect.com/science/article/pii/S0191886908000962> Navštíveno 15.5.2019.

- [36] Indecisiveness and informational tunnel vision, Eric Rassin and co., Personality and Individual Differences vydání červenec 2008, s. 97, Dostupné zde:
<https://www.sciencedirect.com/science/article/pii/S0191886908000962> Navštíveno 15.5.2019.
- [37] Health information on the internet, Tabitha Tonsaker, Gillian Bartlett and Cvetan Trpkov, Canadian Family Physician May 2014, Dostupné zde:
<http://www.cfp.ca/content/60/5/407.short> Navštíveno 15.5.2019.
- [38] Albert Einstein, The Saturday Evening Post, 26. říjen 1929, Dostupné zde:
https://www.saturdayeveningpost.com/wp-content/uploads/satevepost/what_life_means_to_einstein.pdf Navštíveno 5.7.2019.
- [39] John Kounios a Mark Beeman, The Cognitive Neuroscience of Insight, Annual Review of Psychology, leden 2014, s. 73, Dostupné zde:
<https://www.annualreviews.org/doi/full/10.1146/annurev-psych-010213-115154> Navštíveno 4.7.2019.
- [40] Sascha Topolinski a Rolf Reber, Gaining Insight Into the „Aha” Experience, Current Directions in Psychological Science, prosinec 2010, s. 402, Dostupné zde:
<https://journals.sagepub.com/doi/abs/10.1177/0963721410388803?journalCode=cdpa> Navštíveno 3.7.2019.
- [41] Metcalfe J., Premonitions of insight predict impending error, Journal of Experimental Psychology: Learning, Memory, and Cognition, prosinec 1986, 623-634. Dostupné zde:
<https://psycnet.apa.org/record/1987-03081-001> Navštíveno 5.7.2019
- [42] Rabbit-duck illusion, Wikipedia, Dostupné zde:
https://en.wikipedia.org/wiki/Rabbit%E2%80%93duck_illusion
- [43] Hadamard, The Psychology of Invention in the mathematical field, 1945, s. 16. Dostupné zde:
<http://worrydream.com/refs/Hadamard%20-%20The%20psychology%20of%20invention%20in%20the%20mathematical%20field.pdf>
 Navštíveno 5.7.2019
- [44] David Coghlan, Understanding insight in the context of Q, Action Learning: Research and Practice, 2012, s. 249. Dostupné zde:
<https://www.tandfonline.com/doi/abs/10.1080/14767333.2012.711238?needAccess=true> Navštíveno 5.7.2019
- [45] Carola Salvi, Emanuela Bricolo, John Kounios, Edward Bowden Mark Beeman, Insight solutions are correct more often than analytic solutions, Thinking Reasoning, 2016, s. 443-460, Dostupné zde:

<https://www.tandfonline.com/doi/abs/10.1080/13546783.2016.1141798> Navštíveno 4.7.2019

- [46] C. Danek, A. H. Fraps, Working wonders? Investigating insight with magic tricks, *Cognition*, únor 2014, s. 174-185, Dostupné zde: <https://www.sciencedirect.com/science/article/pii/S0010027713002138> Navštíveno 5.7.2019
- [47] Danek, A. H., Fraps, T., von Müller, It's a kind of magic—what self-reports can reveal about the phenomenology of insight problem solving, *Frontiers in Psychology*, prosinec 2014, Dostupné zde: <https://www.frontiersin.org/articles/10.3389/fpsyg.2014.01408/full> Navštíveno 3.7.2019
- [48] John Kounios, Mark Beeman, The Aha! Moment: The Cognitive Neuroscience of Insight, *Current Directions in Psychological Science*, srpen 2009, s. 212, Dostupné zde: <https://journals.sagepub.com/doi/abs/10.1111/j.1467-8721.2009.01638.x> Navštíveno 3.7.2019
- [49] Louis Pasteur, přednáška, University of Lille (7. prosinec 1854).
- [50] Front end frameworks popularity (React, Vue and Angular), 16. 12. 2018, Dostupné zde: <https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190> Navštíveno 15.5.2019.
- [51] Threejs.org, Dostupné zde: <https://threejs.org> Navštíveno 15.5.2019.
- [52] Official Blender stats page, Dostupné zde: <https://www.blender.org/about/website/statistics/> Navštíveno 15.5.2019.
- [53] Maze Generator, Dostupné zde: <http://www.mazegenerator.net/> Navštíveno 13.3.2019.
- [54] Unity.com, Dostupné zde: <https://unity.com> Navštíveno 13.5.2019.
- Obrázky:**
- [55] <http://www.perceptionsense.com/2010/06/ames-room.html> Navštíveno 12.8.2019
- [56] http://www.amagiciansmind.com/wp-content/uploads/2017/03/ames_room_construction.jpg Navštíveno 12.8.2019
- [57] https://en.wikipedia.org/wiki/Penrose_stairs Navštíveno 12.8.2019
- [58] https://en.wikipedia.org/wiki/Rabbit%E2%80%93duck_illusion Navštíveno 11.8.2019



Příloha A

Seznam zkratk

HTML - Hyper Text Markup Language, jazyk složený z tagů používaný pro rozvržení webových stránek.

JS - JavaScript, skriptovací jazyk používaný převážně na webu

CSS - Cascading Style Sheet, jazyk používaný pro stylování HTML elementů

DOM - Document Object Model, prostředí vytvářející stromovou strukturu z HTML a XML tagů.

Příloha B

Slovník pojmů

Tunnel vision – fenomén, kde se člověk soustředí jen na úzkou množinu reality, omezujíc se často na nízký počet případných řešení.

Vertex, Face – „bod“ 3D modelu. Dalo by se porovnat s uzlem grafu. Minimálně tři vertexy mohou tvořit tzv. face. Například krychle má 6 faces a 8 vertexů. Složitější modely mohou mít tisíce až statisíce vertexů.

Blender – Program pro vytváření 3D scén a modelů. Současně velice populární, z nemalé části díky tomu, že je kompletně zadarmo.

Bounding box – Souřadnice pomocného kvádrů, který je dostatečně velký, aby v sobě obsahoval celý 3D objekt. Více informací: https://computersciencewiki.org/index.php/Bounding_boxes

DOM Event - Událost ve stromové struktuře webové stránky. Například kliknutí myši. Více informací: <https://developer.mozilla.org/en-US/docs/Web/Events>

Ray, či Raycasting – Ray je anglické slovo pro paprsek, a Raycasting je proces vytvoření paprsku. V 3D prostoru se často paprsky využívají pro zjištění vzdálenosti mezi dvěma objekty, či jestli na téhle trajektorii daný objekt může protnout jiný.

Herní engine - Herní engine je program pro zjednodušení a umožnění vytváření videoher. Sestává z mnoha základních kamenů, od jádra pro vykreslování po fyziku a skriptovací knihovny.

Extrudování - V 3D modelování extrudování vyjadřuje „vytlačení“ 2D objektu do 3D podoby, podle hranice a určité osy objektu. Například z plochy můžeme jednoduše extrudovat kostku, či z kruhu válec.

Rendering - Vykreslení 3D scény do 2D obrázku. Podle složitosti scény může

trvat milisekundy až několik týdnů. Např. tato aplikace většinou vykresluje 60 snímků za sekundu.

Framework - Soubor knihoven a kódovacích postupů, které mají za cíl učinit vytváření webových stránek v JavaScriptu pohodlnější. Jeden z takových frameworků je např. Vue.js

webpack - Často používaná knihovna pro kompilaci a zjednodušení výsledných souborů JavaScript projektu. Často může ze složité systémové struktury s mnoha různými typy souborů vygenerovat pouze .js a .css soubory.