



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Název:** SAGE2 - Vstupní zařízení a ovládání platformy  
**Student:** Kristýna Böhmová  
**Vedoucí:** Ing. Jiří Chludil  
**Studijní program:** Informatika  
**Studijní obor:** Webové a softwarové inženýrství  
**Katedra:** Katedra softwarového inženýrství  
**Platnost zadání:** Do konce zimního semestru 2020/21

### Pokyny pro vypracování

1. Analyzujte různé druhy vstupních zařízení a jejich možného využití na platformě SAGE2 se zaměřením na herní ovladače (pro Xbox, PlayStation a další).
2. Na základě předchozí analýzy vybraných vstupních zařízení navrhnete a implementujete ovládání aplikací na SAGE2.
3. Navrhnete prototyp WebGL herní aplikace v multiplatformním herním enginu Unity3D.
4. Rozšířte práci o možnost nastavení speciálního ovládání a přiřazení jednotlivých tlačítek a senzorů ovladače.
5. Prototyp a GUI podrobte uživatelskému testování.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 19. února 2019





**FAKULTA  
INFORMAČNÍCH  
TECHNOLÓGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **SAGE2 - Vstupní zařízení a ovládání platformy**

*Kristýna Böhmová*

Katedra softwarového inženýrství

Vedoucí práce: Ing. Jiří Chludil

15. května 2019



---

## Poděkování

Tímto chci poděkovat mému vedoucímu práce Ing. Jiřímu Chludilovi za čas, rady a odborné vedení po celou dobu mé práce. Chci také poděkovat své rodině a přátelům za neustálou podporu během celé doby mého studia.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. května 2019

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2019 Kristýna Böhmová. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Böhmová, Kristýna. *SAGE2 - Vstupní zařízení a ovládání platformy*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.



---

# Abstrakt

Tato práce se zabývá možnostmi propojení herních vstupních zařízení se SAGE systémem. Rozebírá základní informace o vstupních zařízeních a architektuře SAGE. Na základě podrobné analýzy, testování a implementace několika způsobů propojení, jsem přišla na způsob komunikace gamepadu a SAGE serveru. Dále se zabývám některými částmi prototypu videohry, vyvíjené v Unity engine a exportované jako WebGL aplikace pro SAGE. Představuji několik základních možností implementace umělé inteligence. Prototyp byl vytvořen v průběhu minulého roku v mém volném čase a v rámci předmětu Programování grafických aplikací.

**Klíčová slova** SAGE2, SAGElab, gamepad, vstupní zařízení, Unity, hra, xbox.

---

# Abstract

This bachelor thesis addresses the possibility of connecting game input devices with SAGE system. It analyzes basic information about input devices and SAGE architecture. On the basis of the detailed analysis, testing and implementation of a few methods of connecting, I have come up with a way for gamepad to communicate with SAGE server. In addition, I address some parts of the video game prototype developed in the Unity engine and exported as a WebGL application for SAGE. I present a few simple methods how to implement artificial intelligence. The prototype was created during last year and I have worked on it in my free time and also during school subject Programming of graphical applications.

**Keywords** SAGE2, SAGElab, gamepad, input devices, Unity, videogame, xbox.

---

# Obsah

Úvod	1
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Analýza</b>	<b>5</b>
2.1 SAGE . . . . .	5
2.2 Vstupní zařízení . . . . .	7
2.3 Analýza použitých technologií . . . . .	17
2.4 Existující řešení . . . . .	17
<b>3 Návrh</b>	<b>21</b>
3.1 Hlavní idea . . . . .	21
3.2 Analýza požadavků . . . . .	22
3.3 Modely a diagramy . . . . .	23
3.4 Prototyp Hry . . . . .	24
<b>4 Realizace</b>	<b>31</b>
4.1 Průběh realizace . . . . .	31
4.2 Input Manager . . . . .	32
4.3 Instalační a uživatelská příručka . . . . .	32
<b>5 Testování</b>	<b>37</b>
<b>Závěr</b>	<b>39</b>
<b>Seznam použité literatury</b>	<b>41</b>
<b>A Seznam použitých zkratk</b>	<b>43</b>
<b>B Obsah přiložené flash disku</b>	<b>45</b>



---

## Seznam obrázků

2.1	Architektura Sage2. Získáno z odkaz . . . . .	7
2.2	České Qwertz rozložení. Image Copyright © 2006 Petr Sládek alias slady odkaz . . . . .	9
2.3	Ukázka klasické myši. Získáno z odkaz . . . . .	11
2.4	Fotka DDR Podložky. Získáno z odkaz . . . . .	13
2.5	Ilustrace Xbox 360 Contolleru. Získáno z odkaz . . . . .	14
2.6	Ilustrace DualShocku od Playstation. Získáno z odkaz . . . . .	14
2.7	GameCube ovladač. Získáno z odkaz . . . . .	15
3.1	Diagram tříd zobrazující dědičnost entit pro prototyp hry. . . . .	23
3.2	Diagram nasazení zobrazující propojení SAGE a gamepadů. . . . .	23
3.3	Stavový automat Dřevorubce v editoru Unity. . . . .	28
4.1	Ukázka z prototypu hry Innkeeper. . . . .	33



---

# Úvod

Tématem této bakalářské práce je propojení zařízení se SAGE2 systémem a prototypem videohry. Dále se zabývá možností provedení analýzy vstupních zařízení se zaměřením na gamepady a joysticky. Vývoj her se rozvíjí každým rokem víc a víc a vývojáři se snaží tvořit videohry pro zařízení všeho druhu. Videohry se nyní dají hrát na zařízeních od počítačů a konzolí po telefon, chytré hodinky nebo ledničky, nebo třeba i na světelných panelech na budově ČVUT FEL. Díky SAGE laboratoři na ČVUT FIT, máme možnost testovat videohry i na velikém rozlišení. Jediná nevýhoda je, že videohry musí být dělané pro vstup z klávesnice nebo myši, která je v SAGE systému nativně implementovaná. Proto se pokusím propojit různá vstupní zařízení se systémem SAGE a poté k tomu vytvořit jednoduchý prototyp videohry, na kterém by se daly různé gamepady a joysticky testovat. Nejdříve si rozeberu SAGE a jeho architekturu, poté se zaměřím na několik vybraných vstupních zařízení a krátké informace k nim. Dále se pokusím popsat aktuální řešení, prototyp hry a zajímavé postupy řešení umělé inteligence v tomto prototypu.





---

## Cíl práce

Hlavním cílem bakalářské práce je analýza SAGE systémů a jejich architektury, vstupních zařízení a implementace propojení s SAGE2 systémem. Budu se snažit prozkoumat existující řešení a možnost jejich využití v mé bakalářské práci. Prozkoumám oficiální dokumentaci pro vývojáře od SAGE, abych pochopila jak by se dal můj problém vyřešit.

Díličními cíli jsou prozkoumání implementace videohry v Unity Enginu a vytvoření prototypu hry, na které by se propojení gamepadu a SAGE2 mohlo testovat. Budu se snažit využít prototyp hry, na kterém jsem pracovala během posledního roku ve volném čase. Práce by měla pomoci budoucím studentům PGA a jiných předmětů k vytvoření vlastních prototypů videoher s možností ovládání přes herní vstupní zařízení.



---

# Analýza

## 2.1 SAGE

Předtím než se pustím do rozebírání vstupních zařízení, vysvětlím co přesně znamená SAGE.

### 2.1.1 Co to je SAGE?

Zkratka SAGE zastupuje delší název: Scalable Adaptive Graphics Environment. V překladu bychom to mohli nazvat jako Škálovatelně přizpůsobivé grafické prostředí. Vývoj SAGE započal v roce 2009, kdy byl Univerzitě v Illinois, Chicago (UIC) udělen grant v hodnotě 1.9 milionů dolarů na vývoj perzistentních, kolaborativních služeb pro globální cyber infrastrukturu. SAGE je operační systém umožňující zobrazení a sdílení data-intenzivních informací ve všech možných rozlišení a formátu z mnoha zdrojů, na stěně tvořené z mnoha monitorů fungujících jako jeden velký. SAGE je cross-platformní, komunitně založený a má otevřený zdroj (open source). Avšak, my se budeme zabývat pouze SAGE2 rozhraním (Scalable Amplified Group Environment, uvedená na konci roku 2014). V překladu: Škálovatelně zesílené skupinové prostředí. SAGE2 je stavěné na prohlížečích a vše je děláno před Chrome webový prohlížeč. Oproti SAGE bylo zcela předěláno uživatelské rozhraní, aby bylo možné pracovat rychleji a efektivněji na velkých stěnách tvořených monitory. Není potřeba žádný speciální software. Stačí jen vzít a přesunout obsah z jednoho počítače na vizuální reprezentaci stěny ve webovém prohlížeči. Pro SAGE2 můžeme snadno vyvíjet aplikace, vše je napsané v Javascriptu (oproti SAGE, kde se využívalo C++). [1]

### 2.1.2 SAGElab - telestěna

SAGElab je laboratoř CESNETu spadající pod Fakultu Informačních Technologií a Fakultu Elektrotechnickou na Českém Vysokém Učení v Praze. Cílem laboratoře je podporovat vzdělání a výzkum v síťových a vizuálních technologiích, vývoj nových aplikací a experimentální práci v těchto oblastech. [2] V laboratoři se nachází 20 full HD monitorů NEC s tenkým rámečkem. Celkově se tedy jedná o 10k rozlišení distribuované na řídicím serveru a pěti klientech. Vše je propojeno pomocí optického switchu: Linky 10GB/s pro každý klientský server a 40 GB/s pro řídicí server. SAGE2 je postaveno na technologiích HTML5, Javascript a Node.js. [3]

### 2.1.3 Architektura

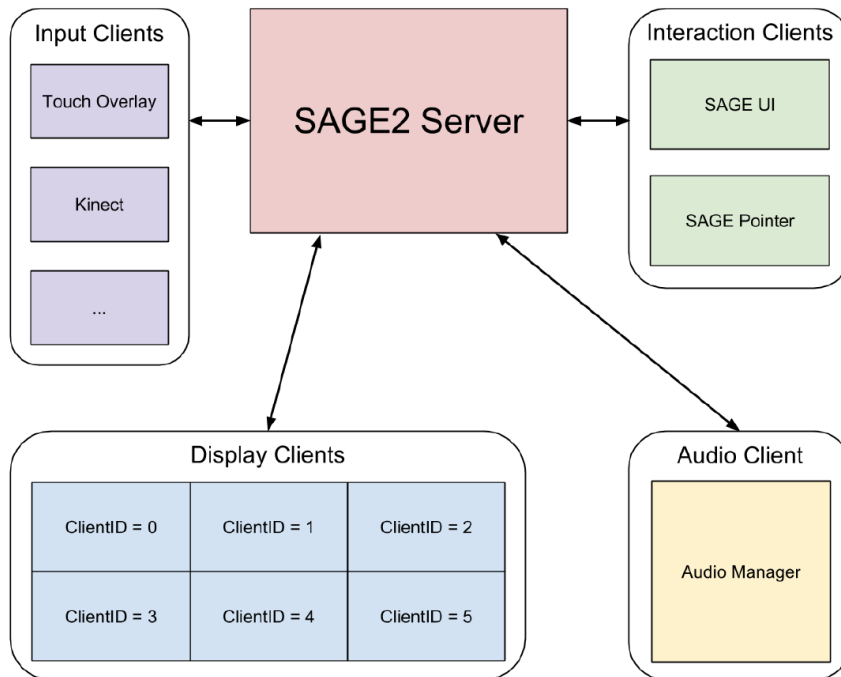
Hlavní komponenta architektury SAGE2 je Server na kterém běží systém. Server ovládá pomocí RPC (Remote procedure call - vzdálené volání procedur) Display Clienty, kteří zobrazují určitou část větší obrazovky. Například pro 4 řady a 5 sloupců je to 1/4 výšky a 1/5 šířky obrazu. Každý z těchto klientů má ClientID, kterým se dá určit. Pod těmito klienty běží jeden objekt aplikace (prohlížeč). Server pošle zprávu 'create application' všem klientům, klient vytvoří novou instanci z třídy aplikace. Dále pak server posílá příkazy 'draw' do každé této instance aplikace (určené podle frame rate). Klienti poté vracejí zprávu serveru před každým novým snímkem o provedení vykreslení. Pro konzistenci mezi jednotlivými klienty a klienty, kteří se chtějí připojit ukládáme si takzvaný 'State' (stav), který určuje které aplikace poběží a popřípadě jejich změny, co se provedly během jejich běhu. Stav se ukládá automaticky na serveru. Pomocí 'Interaction Clients' (interaktivních klientů) mohou uživatelé ovládat aplikace běžící na SAGE. K ovládání využívají SAGE UI a SAGE Pointer. Dále můžeme také připojit vstupní zařízení přímo na server, kterým tak můžeme ovládat (Input Clients). Poslední komponenta je pak Audio Client, díky kterému může server přehrávat zvukové soubory. [4]

#### **Příklady aplikací běžících na SAGE2:**

Sketchfab Viewer - Prohlížeč 3D modelů uložených ve službě Sketchfab.

Deep Viewer - Prohlížeč gigapixelových obrázků

Oba dva vytvořil Jiří Kubišta.



Obrázek 2.1: Architektura Sage2. Získáno z odkaz

## 2.2 Vstupní zařízení

### 2.2.1 Co je to vstupní zařízení

Vstupní zařízení je hardware, který odesílá data do počítače a díky tomu s ním umožňuje interakci a jeho ovládání. Mezi nejznámější vstupní zařízení patří myš a klávesnice. Dále pak mezi ně řadíme například i skenery, digitální kamery nebo gamepady. Dnes se také dají využít audio zařízení jako jsou například mikrofony pro ovládání počítače pomocí hlasového rozpoznávání. Některá vstupní data mohou jít do počítače přímo pro zpracování. V této kategorii jsou například čárové kódy, řeč přes mikrofon a zařízení, které umožňují ovládat aplikace pohybem. Avšak některá data se musejí zpracovat předtím, než se objeví na monitoru (například magnetický disk).[5]

### 2.2.2 Jak budeme rozebírat vstupní zařízení

První si řekneme něco o historii vývoje zařízení a něco málo o firmách, které je vyvíjí. Dále si rozebereme jejich využití. Rozebereme si na kterých operačních systémech je můžeme používat a jak se dají připojit a nainstalovat. Popíšeme si možné využití v našem projektu a na SAGE nebo jak je využít v herním

enginu Unity. A v neposlední řadě si popíšeme jejich rozložení či mapování.

### 2.2.3 Klávesnice

Klávesnice je základní vstupní zařízení vycházející z vynálezu amerického inženýra-tiskaře Christophera L. Sholese (některé zdroje uvádějí příjmení Scholes) z 19. století - psacího stroje. Oproti psacímu stroji klávesnice může i obsahovat i numerickou klávesnici a několik funkčních kláves. Klávesnice může buď být samostatná deska s klávesami připojená pomocí USB k počítači nebo je jeho součástí (Notebooky). Jednotlivé klávesy vydávají různé elektrické signály do počítače, který zpracuje kódovou kombinaci odpovídající této klávese.

Dle Ing. Jiřího Kačera a Ing. Miroslava Šnoreka, Csc. v díle Přídavná zařízení [6] lze klávesnice rozdělit do tří skupin podle funkce:

- **Abecedněčíselná** - základní klávesnice vycházející z psacího stroje
- **Pouze číselná** klávesnice - využívaná především u pokladen nebo součástí větších klávesnic
- **Specializovaná** klávesnice - klávesy vyvolávají speciální funkce v počítači (Enter, Delete, F1)

Dále autor dělí klávesnice podle elektronické části:

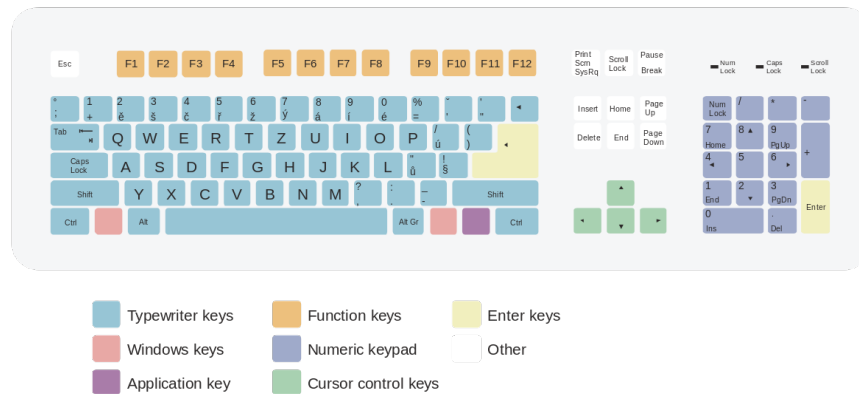
- **Encoded** - zakódovaná
- **Nonencoded, Decoded** - nezakódovaná

[6, str. 165-169] [7]

Klávesnice se dají dále dělit podle jejich rozložení kláves. Ať už za důvodem efektivnějšího programování nebo kvůli různým jazykům. Mezi nejznámější patří: QWERTY - nepoužívanější rozložení (vytvořené Christopherem L. Sholesem), QWERTZ - česká verze, která má například prohozené Z a Y, Dvorak - zjednodušená klávesnice se znaky, které se nejvíce používají blíže u sebe a mezi cizojazyčně například patří Arabská, Ruská a Čínská.

Klávesnice je pole spínačů uspořádané do mřížky, nazývané matice kláves (key matrix). Elektrické signály jsou posílány do mikrořadiče klávesnice (přímo umístěného v klávesnici), který signál zpracuje pomocí zabudovaného programu a pošle jako data nebo příkaz do počítače jako snímací kód (scan code). Klávesy můžeme rozdělit podle jejich provedení:

- Mechanické
- Mechanické s pěnovým prvkem



Obrázek 2.2: České Qwertz rozložení. Image Copyright © 2006 Petr Sládek alias slady odkaz

- Mechanické s gumovou membránou
- Klávesy s kapacitními spínači
- Fóliové klávesnice
- Laserová virtuální klávesnice

V průběhu existence klávesnice lze připojit k počítači jednou z následujících přípojek: DIN-5 (AT klávesnice), Mini-DIN (PS/2), USB a popřípadě bezdrátové připojení pomocí Bluetooth nebo IrDA.

Základní typ klávesnice má 102 kláves (včetně českých znaků). Rozložení kláves můžeme rozdělit do několika skupin[8]:

- Základní pole alfanumerických kláves - a-z, CAPS LOCK, SHIFT, CTRL, ALT, TAB, BACKSPACE, ENTER
- Pole funkčních kláves - F1-F12, ESC, PRINTSCREEN, PAUSE, SCROLL LOCK, Fn
- Pole numerických kláves - 0 - 9, +, -, \*, /, ENTER, NUM LOCK
- Pole řídicích kláves - šipky (←, ↑, ↓, →), INSERT, HOME, END, DELETE, Page Up, Page Down
- Klávesy, které lze také nalézt u současných klávesnic: WIN - Windows START

Klávesnice lze používat na nejznámějších operačních systémech (Windows, Linux, MacOS). Pro připojení k počítači se využívá převážně USB kabel a pro

bezdrátovou komunikaci bluetooth. Avšak dříve se připojovala hlavně pomocí DIN-5 konektoru a později Mini-DIN (PS/2). Instalace softwaru většinou není třeba, nebo se ovladače nainstalují automaticky.

Využití v projektu: Jelikož přijímání vstupů z klávesnice je implementované v SAGE2, tak se budeme soustředit na jiné zařízení.

### 2.2.4 Počítačová Myš

Myš je malé hardwarové zařízení určené pro ovládání jednou rukou. Na vrchní straně se může nacházet jedno až tři tlačítka a kolečko pro scrollování. Avšak myš může být vybavena i více tlačítky například ze stran (hlavně u herních myši). Avšak hlavním významem myši je možnost pohybu kurzoru ve 2D prostoru kombinací vertikálního a horizontálního pohybu po osách X a Y. Pohyb umožňuje snímač na spodní části myši.

Podle různých snímačů pak myši rozdělujeme do tří základních skupin:

**Elektromechanická myš** - vybavena pogumovanou kovovou kuličkou na spodní části, která se při pohybu otáčí a tím díky elektronickým kontaktů vysílá elektrické impulsy do počítače pro zaznamenání pohybu.

**Optomechanická myš** - zde se kovová kulička dotýká dvou válečků, avšak při pohybu světelná emisní dioda vysílá paprsky přes otvory v kodérech na světelné detektory.

**Optická myš** - vybavena dvěma světelnými emisními diodami (červenou a infračervenou), čočkami a fotodetektory, které detekují pohyb podle toho jestli přijmou odražené světlo nebo ne. [9]

V roce 1963 přišel na svět s vynálezem počítačové myši Dr. Douglas C. Engelbar, který ve středisku Stanfordského výzkumného institutu Augmentation Research Center. Myš byla pojmenovaná jako "X-Y ukazatel". Jednalo se o malou dřevěnou krabičku se dvěma kolečky s kolnými osami a tlačítkem, propojená kabelem s počítačem. Kolečka byla napojena na potenciometry, které přenášeli proměnnou napětí do počítače a díky tomu umožnila myš snadnou a rychlejší práci. [10] [11]

Vstupní zařízení myš můžeme dělit podle druhu připojení k počítači[12]:

- 1. **Sériové** - 9pinový konektor (RS-232) a u Apple zařízení pomocí ADB.
- 2. **PS/2** - podobně jako u klávesnic, dnes se nachází hlavně u starších počítačů.
- 3. **USB** - nejvíce používaný způsob kabelového připojení v dnešní době.





Obrázek 2.3: Ukázka klasické myši. Získáno z odkaz

- 4. **Bezdrátové infračervené** - do počítače je zapojené externí přijímací zařízení, které komunikuje s myší. Nevýhodou je nutná přímá viditelnost přijímače a myši.
- 5. **Bezdrátové rádiové** - funguje na stejném principu jako infračervené avšak není potřeba přímé viditelnosti. Dnes známé hlavně pod názvem Bluetooth.

Mezi nejznámější výrobce myši patří: Xerox, Microsoft, Logitech, Mouse Systems (s myší ProAgio), Agilent a Apple. [13]

#### 2.2.4.1 Trackball

Mezi myši bychom také měli zahrnout zařízení TrackBall (kulový ovladač), které funguje jako obrácená myš. Ovládání funguje tak, že uživatel otáčí přímo kuličkou a díky tomu může dosahovat vyšší přesnosti. Další výhodou může být ušetření pracovní plochy, jelikož nám stačí jen prostor na umístění ovladače. To

je jeden z důvodů, proč bylo určeno na vestavění do těla přenosných zařízení. [9] [12]

### 2.2.4.2 Touchpad

V neposlední řadě musíme zmínit zástupce myši u přenosných zařízení, které najdeme dnes na všem noteboocích - Touchpad. Touchpad je sestaven ze série vodičů ve dvou vrstvách oddělených vrstvou dielektrika (izolantu). Skrz tyto vodiče prochází proud, který sepne kolmé vodiče, pokud je na vstupu potenciál země (prst). Krom pohybu kurzoru, vyhodnocuje touchpad i poklepání nebo rolování.

Roku 1988 se podařilo George E. Gerpheideovi sestavit první touchpad, který pak hlavní využití získal u Applu, která si jej nechal licensovat pro své produkty. [14] [15]

## 2.2.5 Gamepady

### 2.2.5.1 Taneční podložka

X-PAD, Extreme Dance Pad se používá hlavně k tanečním hrám známým jako DDR (Dance Dance Revolution od firem jako jsou Konami, Nintendo a nebo Disney), kde má hráč za úkol mačkat nohama šipky na podložce podle pořadí v kterém se mu objevují na obrazovce. Podložka se hlavně využívá u konzolí, jako jsou například PlayStation, Xbox nebo Nintendo.

Podložka na které budu vyvíjet má 10 vstupním tlačítek: 8 šipek do různých směrů, Select a Start.

Pokud chceme začít používat podložku u počítače, stačí ji pomocí USB zapojit a počítač automaticky reaguje na vstup, jako kdybychom zapojili jiný gamepad. Nepotřebujeme instalovat žádný driver. Není potřeba žádné napájení, vše je zprostředkováno pomocí USB kabelu. Podložka funguje na Windows, Mac i Linux systému.

Využití v projektu: Taneční podložku budeme využít na pohyb postavy ve 3D prostoru. Hráč bude stát na podložce a nohama ovládat směr a rychlost. Při držení šipky dopředu se bude pohybovat dopředu. Otáčení však bude řešeno na ovladači.

### 2.2.5.2 Xbox 360 Controller

Xbox 360 controller je vstupní zařízení určené pro konzole Xbox 360 od Microsoftu. Kontrolér byl poprvé představen v roce 2005. Pro testování budu využívat verzi s USB připojením, existují však i bezdrátové verze, napájený pomocí AA baterií. Pro připojení k PC využívám adaptér, jelikož Xbox používá jiné USB porty.

Ovladač nabízí 2 analogové joysticky spolu s možností oba zmáčknout jako trigger, 11 tlačítek: X, Y, B, A, Back, Start, LT, LB, RT, RB, Xbox Guide a



Obrázek 2.4: Fotka DDR Podložky. Získáno z odkaz

jednu směrovou podložku (Directional pad).

Využití v projektu: Ovladač bude využíván na rozhlížení pomocí sticků, tlačítka půjdou aktivovat různé akce a druhým stickem se bude ovládat myš/menu.

### 2.2.5.3 Playstation 4

Playstation 4 obsahuje hned několik ovladačů: DualShock 4, Playstation Move, Playstation Camera, Playstation VR a další.

#### DualShock 4

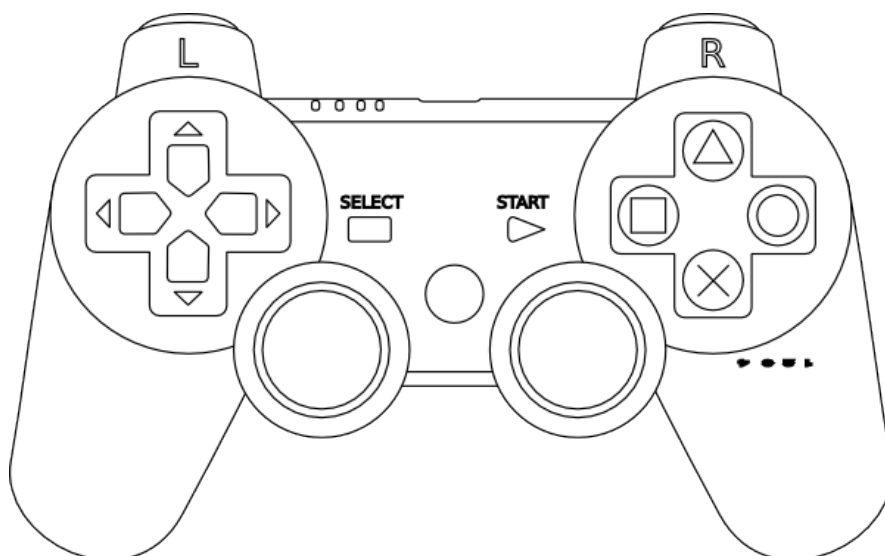
Hlavní ovladač vychází z ovladače předchozí generace PS3. Připojuje se ke konzoli pomocí Bluetooth 2.1 a EDR. Oproti Xbox ovladači, můžeme uprostřed ovladače najít touchpad. Dále obsahuje tlačítko Options, které nahrazuje původní tlačítka Select a Start, nebo tlačítko Share pro sdílení fotek ze hry. Klasicky jako na podobných ovladačích nalezneme směrová tlačítka na levé straně, dva sticky, PlayStation tlačítka a klasicky tlačítka se značkami trojúhelníku, kolečka, čtverečku a křížku. Zepředu se pak nachází 2 trigger a 2 bumpery. DualShock4 obsahuje technologii PlayStation Move, která se využívá pro roz-



Obrázek 2.5: Ilustrace Xbox 360 Contolleru. Získáno z odkaz

poznání pohybu a vzdálenosti ovladače. Je to první ovladač od společnosti Sony, který oficiálně podporuje kromě Playstation i PC. [16]

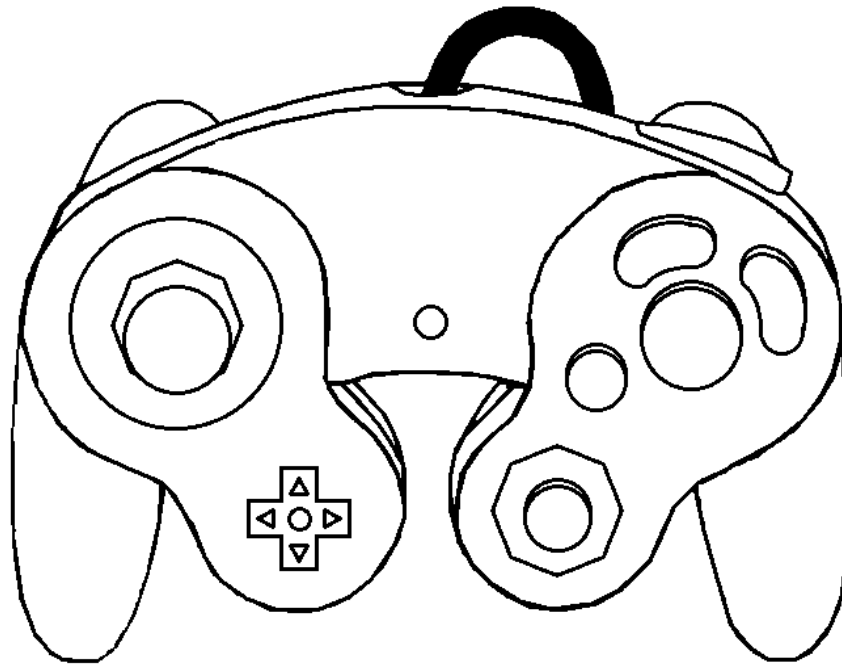
Využití v projektu: DualShock4 může nahradit Xbox 360 Controller, jelikož obsahuje podobné rozložení tlačítek. Můžeme s jeho pomocí ovládat kameru a nahradit si myš a některé vstupy z klávesnice.



Obrázek 2.6: Ilustrace DualShocku od Playstation. Získáno z odkaz

#### 2.2.5.4 GameCube

GameCube je konzole od firmy Nintendo z roku 2001, kdy byla představena v Japonsku.[17] Ovladač nabízí 4 přední tlačítka, 3 tlačítka na rameni, Start, joystick, c-stick a d-pad. Složením tlačítek i rozložením se podobá dnešním ovladačům. [18]



Obrázek 2.7: GameCube ovladač. Získáno z odkaz

#### 2.2.6 Joysticks

Pod pojmem Joystick si každý představí nějaký knipl na ovládání, které najdeme například v letadlech. Není se pak čemu divit, když i ve hrách se využívají pro letecké a vesmírné simulátory. Klasický joystick se skládá ze stíčku, který lze naklánět do 360 stupňů, doprava, doleva, od sebe i k sobě. K rozeznání polohy využívá dva potenciometry. Potenciometr měří napětí, které odpovídá poloze na ose X a Y. V obchodech lze také nalézt, trojrozměrný páky, kde se třetí směr nastavuje otáčením vrcholu páky ovladače kolem své osy. Joysticky se využívají i v grafické práci. Například oproti světelnému peru se vůbec nechvějí. Další výhodou je, že se k využití stačí pouze jedna ruka a tím umožňuje hratelnost i pro hendikepované hráče. Počet tlačítek se může velmi lišit. Levnější joysticky jich obsahují méně a dražší pak větší počet. Do počítače se připojují dnes pomocí USB portu, kdežto dříve k tomu sloužil hlavně gameport. Na trhu jsou dostupné i bezdrátové verze, kde se akorát

zapojí přijímač do USB. Některé kniply mohou mít i zabudovaný systém pro zpětnou vazbu. Mezi další přídatné funkce můžeme započítat klobouček, plynový pedál nebo kniply vyráběné jako přesné kopie z opravdových letadel.

### 2.2.7 Virtuální realita

#### 2.2.7.1 Virtuální brýle

Hlavní komponentou virtuální reality je headset, který nám umožňuje vizuální simulaci reálného světa. Brýle se mohou skládat buď z jedné obrazovky (využití telefonů) nebo ze dvou menších pro každé oko a jsou doplněny o uchycení na hlavu a sluchátka. Brýle většinou obsahují čočky, pro automatické upravení pro pohyb a pozici oka. Obraz se do brýlí posílá z počítače pomocí HDMI či USB, nebo pokud se jedná o mobilní zařízení, který obraz samostatně renderuje. Pro získání perfektního obrazu je potřeba minimální snímková frekvence o 60FPS a 100 stupňové zorné pole (lépe pak 180 stupňové). [19] Headset umí snímat pohyby hlavy pomocí akcelerometrů, takže uživateli umožňuje se rozhlížet jak horizontálně tak i vertikálně. Obraz by měl mít alespoň 300 pixelů na palce pro nemožnost rozlišit jednotlivé pixely. Některé headsety bývají vybaveny i sluchátky pro 3D prostorový zvuk. Ovladače jsou pak připojeny k headsetu bezdrátově. [20]

Další vstupní zařízení stojící za zmínku jsou Myo Armband, Leap Motion Controller, Joy-Con, Paddle nebo Jog dial.

## 2.3 Analýza použitých technologií

### 2.3.1 Unity

Unity je meziplatformní herní engine, vyvíjený společností Unity Technologies. Unity je ve vývoji již čtrnáct let (oznámeno roku 2005) a můžeme ho používat na Windows, macOS i Linux systému. Na Unity můžeme vyvíjet 2D i 3D hry. Hlavní skriptovací jazyk je C (dříve i Javascript, ten se však již přestal od roku 2017 podporovat). Hry však můžeme sestavit na 27 různých platformech, od mobilních po herní konzole nebo i chytré televize. Unity je odjakživa zdarma a proto se stal jednou z nejpobulárnějších možností mezi enginy. Pro moji bakalářskou práci budu využívat možnosti sestavené hry jako WebGL aplikaci na web, kterou SAGE2 dokáže spustit. [21]

#### 2.3.1.1 Unity Input Manager

Input Manager je třída, ve které je možnost definovat všechny různé vstupní osy a herní akce pro projekt. Nalezneme ji v záložce Edit->Project Settings->Input. Osy využíváme ze dvou hlavních důvodů: Umožňují volání a referencování jednotlivých os pomocí jejich jmen ve skriptech a umožňují hráčům vaší hry přenastavení ovládání podle jejich potřeby. Mezi osy patří: horizontální a vertikální pohyb, 3 tlačítka na střelbu (Fire), skok, myš X a Y, kolečko myši a další. Jednotlivé nastavení os se zobrazí při spuštění hry na PC. Zde si mohou nastavit jednotlivé tlačítka a u nich mají, co ve hře ovládají. Proto se doporučuje, psát vstupy pomocí Input Manageru, místo zachytávání jednotlivých tlačítek, pokud chcete toto nastavování využít. Avšak takovýto manager si můžeme napsat i vlastní, kde si uděláme jednu třídu pro nastavování proměnných, další pro tlačítka, které jsou na ně napojená a které pak budeme odchyťovat pomocí předchozích proměnných.[22]

## 2.4 Existující řešení

### 2.4.1 Sage Pointer

Oficiální zdrojový kód k SAGE2 systému. Po připojení jako klient na adresu SAGE2 serveru se nám automaticky otevře webová stránka index.html. Samotné zpracování vstupů probíhá v některém z javascriptů, takže této stránce nebudeme věnovat moc pozornosti a zaměříme se na skripty, které se spolu s html stránkou načítají. Jedním z nich se nazývá SAGE2\_interaction.js. Tento skript se zabývá pointrem, nahráváním souborů a sdílením obrazovek. Pokud potřebujeme přijímat vstupy z klávesnice a myši, vše se bude posílat přes pointer.

## 2. ANALÝZA

---

Pro registraci stisknutí klávesy obsahuje tuto metodu:

```
/**
 * Handler for key down
 *
 * @method pointerKeyDownMethod
 * @param event {Object} key event
 */
this.pointerKeyDownMethod = function(event){
    // Get the code of the event
    var code = parseInt(event.keyCode, 10);
    // exit if 'esc' key
    if (code === 27) {
        this.stopSAGE2Pointer();
        if (event.preventDefault) {
            event.preventDefault();
        }
    }
    else {
        this.wsio.emit('keyDown', {code: code});
        if (code === 9) {
            // tab is a special case - must emulate keyPress event
            this.wsio.emit('keyPress', {code: code, character:
                String.fromCharCode(code)});
        }
        // if a special key - prevent default (otherwise let continue to
        keyPress)
        if (code === 8 || code === 9 || (code >= 16 && code <= 46 && code !==
        32) || (code >= 91 && code <= 93) || (code >= 112 && code <= 145)) {
            if (event.preventDefault) {
                event.preventDefault();
            }
        }
    }
};
```



## 2.4.2 Sage a zachycení vstupu

Vstupy se v SAGE2 zachytávají pomocí javascript eventů a SagePointeru (v překladu událost, využívají je například HTML stránky).

Funkce pro zachytávání eventů vypadá zhruba takto:

```
event: function(type, position, user, data, date) { }
```

Výpis kódu 2: Parametry Eventu

Typy eventů:

- **pointerPress** (button down) - data.button udává jestli jde o “levé” nebo “pravé” tlačítko, které způsobilo event
- **pointerRelease** (button up) - stejné jako down jen se objeví, při uvolnění tlačítka
- **pointerMove** - zaznamenává pohyb pointeru
- **pointerDoubleClick** - dvojklik levým tlačítkem
- **pointerScroll** (pohyb kolečkem) - data.wheelDelta obsahuje hodnotu, o kterou se otočilo kolečko. Pokud se díváte na stránku textu, pozitivní hodnota by měla způsobit posun textu nahoru. A naopak negativní posun textu dolů.
- **klávesnice** (abeceda a diakritika, interpunkce a čísla) - data.code obsahuje kód znaku, data.character je znak ve formátu string - text
- **specialKey** (speciální znaky na klávesnici zahrnující například Backspace, Delete, Shift, etc. . .) - data.code obsahuje kód znaku, data.state udává jestli je tlačítko “nahore” nebo “dole”
- **widgetEvent** (stisk tlačítka, interakce s posunovátkem, vstup textu) - data.identifier udává id UI položky, která vyvolání eventu způsobila, data.action popisuje jaká druh události byl vygenerován a data.text obsahuje text, pokud se jednalo o vstup textové události.

Pozice má x a y hodnoty. Uživatel má přidělené ID, jméno a barvu. Všechny tyto informace mohou být použity pro synchronizaci eventů a dat objektů. [23]



---

## Návrh

### 3.1 Hlavní idea

Hráč se ve hře stane starostou fantasy vesničky. Za úkol ji mají vybudovat od základů a dohlížet na její fungování a prosperitu, ať už se jedná o výstavbu nových budov, získávání potravy pro vesničany a nebo obchod s hrdiny procházející místním krajem. Uprostřed luk, lesů a řek hráč z ničeho se snaží vybudovat velké království. Na začátku začíná pouze se jednou budov, skladištěm, několika surovinami a možností nabídnout život ve vesnici 4 návštěvníkům. Jako první je třeba zabezpečit přísun základních surovin na výstavbu budov. Pomocí dřevorubce získá klády a po jejich doručení na pilu, jednotlivé prkénka. V dole zase můžete těžit kámen, železo, zlato a další drahocenné minerály. Avšak zanedlouho mají vesničané hlad a hráč musí získat i zdroj potravy, jinak budou nešťastný a nebo i můžou stávkovat v práci. Hlavním cílem je získat co největší bohatství a přitom přežít vesničanů až do konečné výstavby hradu.

## 3.2 Analýza požadavků

### 3.2.1 Funkční požadavky

Funkční požadavky této práce jsou:

- **F1** hráč bude moci pokládat nové budovy rozšiřovat tím svojí vesnici,
- **F2** najímat nové vesničany pro práci v nových budovách a možnost sběru, přenosu nebo výroby surovin pro rozvoj vesnice,
- **F3** prohlížení inventáře budov a tím zjišťovat, kolik věcí se kde nachází a nebo kolik zásob vesničané mají na skladu,
- **F4** prohlížení potřeb vesničanů k možnému zjištění problémů a jejich řešení, jako je například postavení hospody pro doplnění hladu,
- **F5** prohlížení akcí vesničanů a možné kontroly odkud kam se zrovna doručuje zboží, či suroviny,
- **F6** pozastavení hry a opětovnému puštění hry, pokud hráč potřebuje na chvíli odložit ovladač.

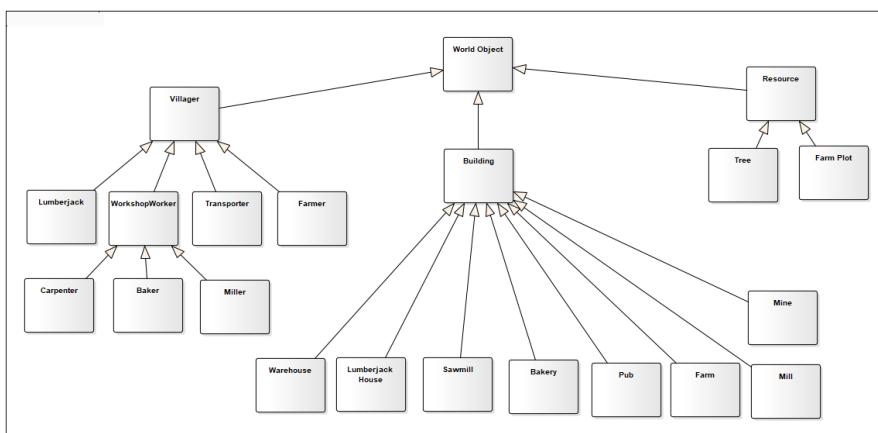
### 3.2.2 Nefunkční požadavky

Nefunkční požadavky této práce jsou:

- **NF1** Unity aplikace na verzi 2018.3,
- **NF2** aplikace spustitelná v prohlížeči jako WebGL a tím i spustitelná na systému SAGE,
- **NF3** možnost ovládání přes herní ovladač a mapování kláves na jednotlivé tlačítka,
- **NF4** jednoduchost používání i pro příležitostní hráče.

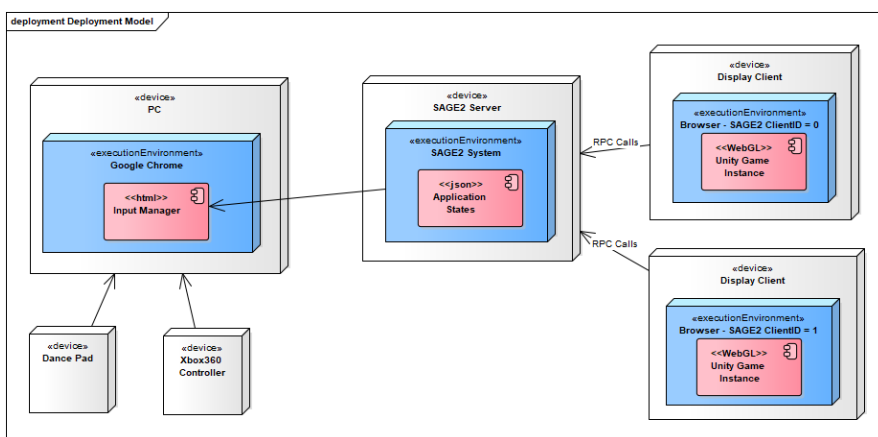
### 3.3 Modely a diagramy

První diagram zobrazuje dědičnost objektů v prototypu hry. Společná vlastnost všech objektů ve hře je, že někde ve světě existují, tedy mají pozici, rotaci a nějakou velikost. Dále se WorldObject dělí na Vesničany, Budovy a Zdroje. Do vesničanů počítám všechny NPC. Do budov pak všechny objekty, které vyrábí nebo obsahují některý z itemů. A nakonec do resources zase objekty, které fungují pouze jako zdroje surovin.



Obrázek 3.1: Diagram tříd zobrazující dědičnost entit pro prototyp hry.

Diagram nasazení zobrazuje propojení vstupního zařízení s klientem a cestu, přes kterou komunikuje se serverem a s jednotlivými instancemi hry na SAGE Displayích. Gamepad se připojí přes USB do počítače (klienta) a zde pomocí Input Manageru komunikuje se SAGE serverem. SAGE server si pak organizuje jednotlivé Displaye a posílá jim instrukce.



Obrázek 3.2: Diagram nasazení zobrazující propojení SAGE a gamepadů.

### 3.4 Prototyp Hry

#### 3.4.1 Jednotlivé objekty (NPC, budovy, suroviny)

##### 3.4.1.1 Nehratelné postavy

###### Návštěvník

Je druh postavy, které se objeví na kraji mapy a postupem času cestuje do centra vesnice. Ve vesnici se snaží doplnit své potřeby a čeká jestli nebude zaměstnaný hráčem a nestane se z něj vesničan. Oproti hrdinům si nekupuje žádné vybavení jen navštěvuje veřejné budovy.

###### Vesničan

Vesničanem se stane postava, pokud se hráč rozhodne, že ji přijme mezi své pracovníky. Vesničan se poté snaží získat práci a čeká jestli nějaká nová budova nenabídne možné pracovní místo. Mezitím se bude snažit doplňovat potřeby a bude se potulovat po městě. V mezechase se bude vesničan snažit nalézt i místo pro bydlení. Pokud nalezne volnou obytnou budovu, tak ji obsadí a bude se do ní vždy v noci vracet.

###### Pracovník

Po získání práce se z vesničana stane pracovník. Pracovník je mnoho druhů. Mezi základní patří dřevorubec, horník, farmář. Každý z nich mají speciální denní režim, kdy plní přidělené úkoly. Dřevorubec se například snaží pokácet dospělý strom v určitém dosahu a pokud se mu tak nedaří, rozhodne se zasadit nový strom nedaleko místa, kde je zaměstnaný. A takhle stále dokola. Během jakékoliv aktivity navíc kontroluje, že má splněné základní potřeby, pokud tomu tak není, pozastaví svoji práci a snaží se ji doplnit.

###### Nosič

Nosič je speciální druh pracovníka, který přenáší věci z jedné budovy do druhé. Pokud nosič dostane za úkol odnést například prkno z pily, prioritně se podívá jestli není ve vesnici nějaké stavba, která toto prkno nevyužije na výstavbu. Dále se snaží doručit prkénko do kterékoliv jiné budovy, který ho může zpracovat a na konci se pokusí prkénko doručit do skladiště, pokud ho nikde jinde nikdo nepotřebuje. Pokud žádná budova nic nepotřebuje odnést, nosič se kouká na jednotlivých skladů a přemýšlí, jestli některou ze surovin nemůže někam donést.

###### Hrdina

Hrdina je nehratelná osoba, která jednou za čas prochází vesnicí. Při její návštěvě se pokouší doplnit své potřeby a koupit nové vybavení, zbraně a oblečení. Pokud uvidí v některém obchodě věc, která je lepší než některý

z těch kterou již vlastní a zároveň má dostatek financí, aby si ji koupil, zajde do onoho podniku a provede nákup. Starou věc zahodí a novou si nechá. A takhle stále dokola, dokud už nic nemůže najít a nebo dokud mu nedojdou finance. Poté se vypraví na dobrodružství. Pokud hrdinu najmete do své guildy, bude pro vás moct nosit speciální vzácné věci z jeho dobrodružství. Do guildy můžete přijmout jen omezené množství hrdinů. Poté budete moct sledovat jejich statistiky, levely, potřeby, myšlenky a shrnutí jejich nebezpečných misí.

### 3.4.1.2 Budovy

#### **Skladiště**

Hlavní budova hry s kterou jako jedinou hráč začíná na počátku nové hry. Ve skladišti je umožněno skladovat veškeré suroviny a přenosné objekty ve hře. Nosiči jsou jediný NPC, kteří jsou schopni interagovat se skladištěm. Pokud si hráč skladiště rozklikne uvidí malou tabulku se všemi surovinami a jejich počtem v konkrétním skladišti.

#### **Hospoda**

Hospoda je budova umožňující doplnit potřebu hladu. Ať už se jedná o pracovníky, nosiče nebo hrdiny, všichni se mohou chodit najíst do hospody. Pokud se v pekárně objeví chleba je sem dopraven nosiči.

#### **Dřevorubecký srub**

Dřevorubecký srub je budova určena jako výchozí místo pro dřevorubce. Pokud se dřevorubec rozhodne jít zasadit nový strom nebo pokácet již vzrostlý, vždy tomu tak bude z této budovy. Po pokácení stromu kládu umístí do vnitřního inventáře budovy, kterou pak může vyzvednout některý z nosičů.

#### **Pila**

Pila je budova určená pro opracování kmenů na prkna. Samotná budova si kontroluje jestli se v ní nachází pracovník a pokud tomu tak je, automaticky zpracovává dřevo.

#### **Farma na obilí**

Podobně jako Dřevorubecký srub, farma je výchozí bod pro farmáře, který buď chodí zasívat nové obilí na určené pole, nebo jej může chodit sklídit.

#### **Mlýn**

Mlýn funguje na stejném principu jako Pila. Dostane surovinu, v tomto případě obilí a to zpracuje za přítomnosti pracovníka do mouky.

### 3. NÁVRH

---

#### **Pekárna**

V pekárně se vyrábí chleba z mouky. Podobně jako funguje Mlýn a Pila.

#### **Důl**

Je specifická budovat tím, že za přítomnosti pracovníka generuje náhodně obnos železné a zlaté rudy.

#### **Výrobce zbraní**

Výrobce zbraní umožňuje hráči vyrábět dřevěné helmy a meče. Opět vyžaduje přítomnost pracovníka

#### **Obchod**

Obchod je budova, která funguje bez nutné přítomnosti některého z vesničanů. Stačí pokud nosiči donesou od Výrobce zbraní helmy a meče a hrdinové procházející vesnicí se mohou nakoupit vybavení, pokud na něj mají dostatek financí a pokud by nošení tohoto vybavení hrdinům nějak přispělo.

#### **3.4.1.3 Itemy/Objekty ve světě**

##### **Strom**

Strom je objekt vytvořený díky NPC a slouží jako zdroj surového dřeva.

##### **Pole**

Pole jsou umisťovány hráčem a jsou využívána pro pěstování obilí.

##### **Dekorace**

Mezi dekorace můžeme zahrnout trávu, kamení nebo lampy. Nemají žádný využití ve hře. Slouží pouze jako dekorace a jsou to herní objekty bez skriptů. Lampa jako jediná může být umístěna hráčem.



## 3.4.2 Umělá inteligence

### 3.4.2.1 Behaviour Tree

Strom chování si můžeme představit jako orientovaný graf, kde jednotlivé vrcholy představují příkazy a přechody (hrany) představují podmínky pro přechod do další fáze. Tři základní stavy, kterých můžou vrcholy nabývat jsou: Úspěch, Neúspěch a Běžící. Dále můžeme dělit vrcholy na tři druhy: Composite, Decorator a Leaf. Jednotlivé vrcholy můžou tvořit i struktury: Sekvence, Selector. [24]

#### Využití

Hrdina na začátku hry dojde do centra mapy a snaží se najít hospodu, kde by se mohl občerstvit. Snaží se dále doplnit všechny ostatní potřeby. Poté se vnoří do cyklu, který se provede třikrát, kde může navštívit obchod, jít trénovat své dovednosti, odpočívat nebo se socializovat. Po doběhnutí těchto aktivit se hrdina vypraví na dobrodružství, což znamená, že dojde na kraj mapy a despawnuje se. Další možnost je, že hrdinu akorát deaktivujeme na pár minut a pak ho necháme znova navštívit vesnici, kde se bude opakovat celý strom. Pro každou aktivitu v cyklu, napíše vlastní pod-strom, který se pokusí najít budovu, dojít k ní a poté s ní interagovat.

### 3.4.2.2 State Machine

State Machine neboli Konečný Automat se využívá v herním průmyslu pro jednoduchou umělou inteligenci. Pokud potřebujeme od objektu, aby zvládl opakovat několik akcí a nechceme vytvářet vlastní systémy, automat je ideální volba.

Unity samo o sobě neobsahuje předlohu pro tvorbu stavového automatu, avšak pokud se podíváme jak fungují animace, můžeme tvrdit opak.

#### Animator Controller

Animator Controller dovoluje uspořádání animačních klipů a jejich přechodů přímo v editoru Unity. Pro jednotlivé přechody můžeme nastavit podmínky (jako například přechod z chůze do běhu nastane při převýšení určité rychlosti a naopak) a možnost prolnutí jednotlivých animací. Jako výchozí zde najdeme stavy 'Entry', 'Exit' a 'Any State'. Dále si zde můžeme nastavit některou z animací jako výchozí (nejčastěji 'Idle' animace).

Jednou z dalších možností, je přidávání skriptů k jednotlivým stavům. Takže pokud bychom vzali pryč vizuální stránku (animace) a jen si vytvořili stavy se skripty, získáváme tak možnost tvorby jednoduchého konečného automatu přímo v editoru. Jako příklad si rozebereme Dřevorubce.

### 3. NÁVRH

---

#### Dřevorubec

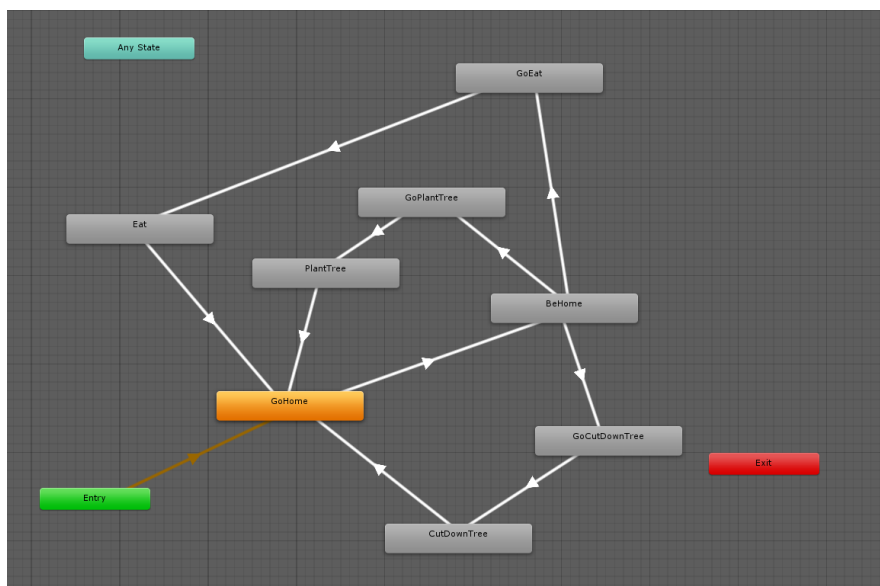
Seznam jednotlivých stavů:

Entry, GoHome, BeHome, GoPlantTree, PlantTree, GoCutDownTree, CutDownTree, GoEat, Eat.

Seznam podmínek pro přepínání mezi stavy:

NearDestination, TreeIsDown, FoundTreeToCut, FoundPlaceToPlant, TreeIsPlanted, IsHungry, DoneEating.

Automat začíná ve stavu s názvem Entry, kde se po spuštění hned dostane do prvního “default” stavu. V tomto případě se jedná o Go Home. Go Home znamená, že se vesničan snaží najít cestu a dojít k jeho zaměstnání (v budoucnu bude nahrazeno vlastní budovou pro domov). Poté, co je již blízko svého cíle se přepne do stavu “Be Home”, kdy je deaktivovaný v blízkosti budovy “Lumberjack House” a rozhoduje se, co bude dál dělat. Pokud nalezne strom, dojde k němu a pokácí ho. Pokud ne, vezme si novou sazenici a dojde ji zasadit. Třetí možností je, že zjistí, že má hlad a bude se snažit nalézt nejbližší hospodu a najíst se.



Obrázek 3.3: Stavový automat Dřevorubce v editoru Unity.

### 3.4.2.3 Unity Navigation a Navmesh

Unity nabízí možnost využití vnitřního systému pro navigaci - NavMesh (navigation mesh), což je vlastně síť bodů, která tvoří plošinky představující možná místa, kde se hráč nebo AI může pohybovat. Pro hledání cesty využívá A\* algoritmus (dnes nejobvyklejší pro AI hledání cest). Spolu se samotnou navmeshí používáme komponenty pro Agenty, Překážky a Linky - Spojnice, které mohou spojovat dvě navmesh v místech, kde se nedá pohybovat. NavMeshAgent jsou jednotlivé AI objekty, které můžeme dále nastavovat: velikost, rychlost, náročnost hledání cest, priority a povolené oblasti pro pohyb. Agenti již v sobě mají zabudovaný systém pro vyhýbání se objektům a sobě navzájem. Pokud má objekt ve světě udělat "díru" do NavMesh, stačí když řekneme, že je to překážka a jestli má vyřezávat prostor do této sítě.



---

# Realizace

## 4.1 Průběh realizace

Jako první jsem zkusila napsat skript na rozpoznání vstupu gamepadu přímo v Unity. Unity je naštěstí už přizpůsobený herním ovladačům a vstup lze zachytit buď pomocí funkce `Input.GetButton` a nebo již v před připraveném `InputManageru`. [25]

Unity projekt jsem pak zkompilevala a nahrála na simulátor SAGE2 rozhraní na počítači. Pokud byl gamepad přímo v zařízení, na kterém běží SAGE2, tak vstupy přijímal. Zde jsem narazila na první problém. Zjistila jsem, že pokud používám vstup z jiného klienta, SAGE2 pointer bohužel podporuje jen nějaký omezený počet vstupů.

SAGEPointer podporované vstupy: Pozice myši, levé, pravé a prostřední tlačítko myši, kolečko myši, velká písmena, některé speciální znaky. Dále pak interakce na UI canvasu: Tlačítka, posuvníky, toggle, textové pole. Podporovány jsou pak i některé vstupy z `InputManageru` - horizontální, vertikální, střelba, 1, 2, 3, skok a Submit. [23]

Jako další krok jsem rozhodla napsat si skript, který bude překládat vstupy gamepadu do Unity aplikace. Jelikož jsem myslela, že když je Unity webová aplikace, stačí napsat v javascriptu jednoduché posílání vstupů do Unity z webu. Bohužel zde nastal stejný problém jako při prvním testování. Dále mě napadlo ještě spustit Unity pomocí jiného skriptu - ne přímo v SAGE2. Bohužel javascript nelze spustit pomocí jiného javascriptu.

Závěr: Unity se pouze tváří jako samostatná aplikace na SAGE2, bohužel je to ale jen webview - SAGE2 otevře hru jako webovou stránku.

Jako další krok jsem se rozhodla využít semestrální práce Jiřího Kubišty, který na předmětu MI-MAI na Fakultě informačních technologií, pracoval na

propojení SAGE a 6-ti osé myši SpaceMouse. Použiji základ práce, kde se pomocí javascriptu a html propojí zařízení přes websockety. Klient se připojí na SAGE server a posílá vstupy z gamepadu, podle toho jak odchyťává eventy. Ve své práci potřebuji odchyťávat více tlačítek a pouze dvě osy, takže tomu zdrojový kód přizpůsobím. Dále přidám možnost na přenastavení tlačítek, aby posílali na SAGE simulaci jiné klávesy.

### 4.2 Input Manager

Input Manager se skládá ze dvou souborů. Jedním z nich je html stránka, kde si uživatel může nastavit vlastní mapování vstupů pro Unity a druhým je javascript, ve kterém běží zpracování vstupu a komunikace se serverem.

#### 4.2.1 Mozilla Gamepad API

Jako první příklad je kus kódu z MozillaGamepadAPI, kde pomocí javascriptu, analyzujeme všechny zařízení (controllery) připojené k počítači a poté zpracujeme jejich tlačítka a osy. Poté se už jen kontroluje změna hodnot tlačítek a os v Update funkci.

#### 4.2.2 Zachycení tlačítek

Hlavní funkce pro zachycení a přemapování tlačítek se jmenuje `buttonActions`. Postupně prochází tlačítka a kontroluje, jestli nějaké nebylo zmáčknuté. Pokud ano, tak se odešle jeho namapovaná hodnota přes socket, jako kdyby uživatel stiskl klávesu na klávesnici.

### 4.3 Instalační a uživatelská příručka

#### 4.3.1 Klient

Na klientský počítač není potřeba nic instalovat, stačí jen připojit vstupní zařízení a pak pomocí internetového prohlížeče se připojíme na SAGE server a za základní adresu doplníme `inputManager.html` a otevře se nám tabulka pro nastavení mapování.

#### 4.3.2 SAGE

Na SAGE je potřeba nainstalovat Input Manager. Instalace probíhá vložením souborů `inputManager.js` do `public/src` složky a `inputManager.html` do `public`. Dále pak instalace Unity hry probíhá stejně jako instalace jiných aplikací na SAGE. Stačí přetáhnout na klientovi do okna zip složku s hrou a instrukcemi. SAGE pak automaticky aplikaci rozbalí a spustí.

### 4.3.3 Ovládání

Ovládání vysvětlím na přímém vstupu z klávesnice a myši. Stačí si pak jen tyto klávesy namapovat, tak jak nám vyhovují v inputManageru.

#### Kamera

- WSAD - pohyb do stran - popřípadě přiblížení kurzoru myši ke straně okna hry
- QE - rotace kamery o 45 stupňů
- Kolečko myši (popřípadě + a - pokud hra běží mimo SAGE2) - přiblížení a oddálení
- H - změna pohledu kamery mezi Isometrickým a Reálným pohledem
- J - synchronizace pohledů pro monitory na SAGE2

#### Budovy a vesničani

- Kliknutím levým tlačítkem - pokládání budov/lidí a možnost prohlížení informací
- R - rotace před položením
- X (popřípadě ESC mimo SAGE2) - zrušení nákupu nebo zobrazení informací
- P - pauza



Obrázek 4.1: Ukázka z prototypu hry Innkeeper.

#### 4. REALIZACE

---

```
function updateStatus() {
  if (!haveEvents) {
    scangamepads();
  }

  var i = 0;
  var j;

  for (j in controllers) {
    var controller = controllers[j];
    var d = document.getElementById("controller" + j);
    var buttons = d.getElementsByClassName("button");

    for (i = 0; i < controller.buttons.length; i++) {
      var b = buttons[i];
      var val = controller.buttons[i];
      var pressed = val == 1.0;
      if (typeof(val) == "object") {
        pressed = val.pressed;
        val = val.value;
      }

      var pct = Math.round(val * 100) + "%";
      b.style.backgroundColor = pct + " " + pct;

      if (pressed) {
        b.className = "button pressed";
      } else {
        b.className = "button";
      }
    }

    var axes = d.getElementsByClassName("axis");
    for (i = 0; i < controller.axes.length; i++) {
      var a = axes[i];
      a.innerHTML = i + ": " + controller.axes[i].toFixed(4);
      a.setAttribute("value", controller.axes[i] + 1);
    }
  }

  requestAnimationFrame(updateStatus);
}
```

Výpis kódu 3: Smyčka pro zpracování vstupu z Gamepadu - Javascriptu [26]



```
function buttonActions(buttons) {
  for (var i = 0; i < buttons.length; i++) {
    if (buttonsPressed[i] !== buttons[i].pressed) {

      var data = {};
      data.code = buttonsBindings[i].charCodeAt(0);
      data.character = buttonsBindings[i];
      wsio.emit('keyDown', data);
      console.log("Pressed " + buttonsBindings[i]);
      wsio.emit('keyPress', data);
      buttonsPressed[i] = !buttonsPressed[i];
    }
  }
}
```

Výpis kódu 4: Zachycení gamepad tlačítka



---

## Testování

Hlavní testování prototypu proběhlo díky předmětu Programování grafických aplikací. Testováno bylo hlavně uživatelské rozhraní, díky čemuž jsem zjistila, co hráčům není jasné při první hře. Většina nepochopila jak přesně se ovládá kamera a jak vlastně začít s výstavbou a přepravou surovin. Další uživatelé byli informováni před začátkem hry o věcech, který nebyly jasné a nebo byly věci předpracovány. Dále je v plánu krátký tutoriál, který by hráči vysvětlil jednoduše ovládání a jak začít. Další možností by byly různé úkoly, které by se postupně otvíraly, jak by se člověk dostal dál ve hře.

Další testování proběhne po odevzdání práce, kdy budu ukazovat práci malým dětem a ptát se na jednoduché dotazy a vysvětlovat základní funkce. Jelikož se bude jednat o děti předškolního věku, nebo o děti chodící na základní školu, nemám v plánu chystat náročné scénáře a dotazníky. Vše proběhne pouze ústní komunikací.



---

## Závěr

Díky vlastnímu skriptu a webové stránce jsme schopni propojit vstupní zařízení se SAGE2 systémem. V budoucnu by bylo dobré tuto možnost integrovat přímo do stávajícího řešení, které podporuje myš a klávesnici. Další problém, který je potřeba prozkoumat je synchronizace jednotlivých Unity aplikací mezi sebou, jelikož se aktuálně pouští na každém klientovi samostatně. Avšak pro nás se nyní otvírá možnost testovat prototypy videoher z Unity v velkém rozlišení. Dalším možným výzkum je samozřejmě rozšíření pro podporu multiplayeru jak lokálního, tak po síti.

mybibliographyfile



---

## Seznam použité literatury

1. *SAGE2: Scalable Amplified Group Environment* [online]. 2014 [cit. 2019-03-21]. Dostupné z: <http://sage.sagecommons.org>.
2. *About lab* [online]. [Cit. 2019-03-15]. Dostupné z: <https://sagelab.cesnet.cz/en/about-lab>.
3. KUBIŠTA, Jirka; BRÉM, Ondra. *Vizualizace dat - systémy SAGE2 a SAGE* [online]. 2019 [cit. 2019-04-05]. Dostupné z: [https://docs.google.com/presentation/d/13psQRuytN75\\_7JM0bn7LT1fqM\\_b0iv9TzovHz5u7dQY/](https://docs.google.com/presentation/d/13psQRuytN75_7JM0bn7LT1fqM_b0iv9TzovHz5u7dQY/).
4. *SAGE2 developer documentation* [online]. [Cit. 2019-03-18]. Dostupné z: <http://sage2.sagecommons.org/download/1309/>.
5. *How Computers Work: Input and Output* [online]. [Cit. 2019-03-20]. Dostupné z: <https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading06.htm>.
6. KAČER, Jiří; ŠNOREK, Miroslav. *Přídavná zařízení*. 1. vydání. Praha: Ediční středisko Českého vysokého učení technického, 1986.
7. VÍTEK, Jan. *Klávesnice a jejich historie* [online]. 2006 [cit. 2019-03-20]. Dostupné z: <https://www.svethardware.cz/klavesnice-a-jejich-historie/14552>.
8. BOUCHALA, Petr. *Klávesnice* [online]. [Cit. 2019-03-20]. Dostupné z: <http://boucpe.wz.cz/me4a/klavesnice.pdf>.
9. FRANTALOVÁ, Zora; MAREK, František. *Vstupní zařízení počítačů*. 1. vydání. Praha: Senzo, 1993. ISBN 978-80-90-12452-3.
10. RAKOWSKI, David. *Historie a vývoj počítačové myši* [online]. 2003 [cit. 2019-03-19]. Dostupné z: <https://www.fi.muni.cz/usr/jkucera/pv109/2003p/xrakowsk.htm>.
11. RAUNER, Karel. *Jak funguje počítačová myš?* [online]. [Cit. 2019-03-19]. Dostupné z: <https://dSPACE5.zcu.cz/bitstream/11025/21659/1/Rauner.pdf>.

12. RAMEŠ, Jiří. *Vstupní zařízení PC - Myš* [online]. [Cit. 2019-03-19]. Dostupné z: <http://vstupnizarizeni.rames.info/2mouse.html>.
13. FRANZ HACKENBURG, Dr. techn h.c. *10 nejznámějších počítačových myší všech dob* [online]. 2014 [cit. 2019-03-19]. Dostupné z: <https://g.cz/10-nejznamejsich-pocitacovych-mysi-vsech-dob/>.
14. POMAZAL, Jiří. *Ovládání dotekem - jak funguje?* [online]. 2007 [cit. 2019-03-19]. Dostupné z: <https://notebook.cz/clanky/technologie/2007/ovladani-dotekem-jak-funguje>.
15. RAMEŠ, Jiří. *Vstupní zařízení PC - Touchpad* [online]. [Cit. 2019-03-19]. Dostupné z: <http://vstupnizarizeni.rames.info/3touch.html>.
16. MICHALA. *Konzole, které vládou světu* [online]. 2018 [cit. 2019-05-10]. Dostupné z: <https://gamingready.cz/2018/06/13/konzole-ktere-vladnou-svetu/>.
17. *Nintendo GameCube Console Information* [online]. [Cit. 2019-05-12]. Dostupné z: <https://www.consoledatabase.com/consoleinfo/nintendogamecube/>.
18. *Nintendo GameCube controller* [online]. [Cit. 2019-05-12]. Dostupné z: [https://nintendo.fandom.com/wiki/Nintendo\\_GameCube\\_controller](https://nintendo.fandom.com/wiki/Nintendo_GameCube_controller).
19. *Technology behind Virtual Reality*. [online]. 2018 [cit. 2019-04-20]. Dostupné z: <https://www.newgenapps.com/blog/how-vr-works-technology-behind-virtual-reality>.
20. PÁNEK, Jan. *Přehled headsetů pro virtuální a rozšířenou realitu*. [online]. 2016 [cit. 2019-04-20]. Dostupné z: <https://www.svethardware.cz/prehled-headsetu-pro-virtualni-a-rozsirenou-realitu/40915>.
21. *Unity User Manual* [online]. [Cit. 2019-03-13]. Dostupné z: <https://docs.unity3d.com/Manual/index.html>.
22. *Unity Input Settings* [online]. [Cit. 2019-03-13]. Dostupné z: <https://docs.unity3d.com/Manual/class-InputManager.html>.
23. NISHIMOTO, Arthur. *Unity Sage2 Application* [online]. 2017 [cit. 2019-04-12]. Dostupné z: <https://bitbucket.org/sage2/sage2/wiki/SAGE2%20Unity%20Applications>.
24. SIMPSON, Chris. *Behavior trees for AI: How they work* [online]. 2014 [cit. 2019-03-13]. Dostupné z: [http://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior\\_trees\\_for\\_AI\\_How\\_they\\_work.php](http://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php).
25. *Unity Conventional Game Input* [online]. [Cit. 2019-03-13]. Dostupné z: <https://docs.unity3d.com/Manual/ConventionalGameInput.html>.
26. *Gamepad API from Mozilla*. 2019. Dostupné také z: [https://developer.mozilla.org/en-US/docs/Web/API/Gamepad\\_API/Using\\_the\\_Gamepad\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Gamepad_API/Using_the_Gamepad_API).



## Seznam použitých zkratk

**GUI** Graphical user interface

**SAGE** Scalable Amplified Group Environment

**AI** Artificial intelligence



---

## Obsah přiložené flash disku

readme.txt.....	stručný popis obsahu flash disku
exe .....	adresář se spustitelnou formou implementace
├─ Innkeeper.....	adresář se spustitelnou verzí hry na SAGE
├─ InputManager .....	adresář se soubory webové aplikace
src	
├─ impl.....	zdrojové kódy implementace
├─ thesis .....	zdrojová forma práce ve formátu $\text{\LaTeX}$
text .....	text práce
├─ thesis.pdf .....	text práce ve formátu PDF