



Zadání bakalářské práce

Název:	VoxelMaze I. - animace a optimalizace UX
Student:	Mariia Korshunova
Vedoucí:	Ing. Radek Richtř, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Počítačová grafika
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

VoxelMaze je kooperativní hra ve voxelově stylizovaném světě pro až 4 hráče. Jedná se o část skupinového projektu.

- 1) Důsledně otestujte stávající verzi hry.
- 2) Proveďte rešerši UI obdobných kooperativních her (po síti i lokální multiplayer).
- 3) Proveďte rešerši možností animace voxelově zobrazených modelů.
- 4) Analyzujte možnosti animace.
- 5) Navrhněte vylepšené rozhraní, UX. Zlepšete dojem z prototypu pomocí animací.
- 6) Navržené změny implementujte.
- 7) Otestujte hru po provedených změnách a porovnejte výsledky.

Bakalářská práce

VOXELMAZE I. - ANIMACE A OPTIMALIZACE UX

Mariia Korshunova

Fakulta informačních technologií ČVUT v Praze
Katedra teoretické informatiky
Vedoucí: Ing. Radek Richtř, Ph.D.
27. června 2021

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2020 Mariia Korshunova. Všechna práva vyhrazena.

Tato práce vznikla jako školní díla na Českém vysokém učení technické 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 bez uplatněných zákonných licencí nad rámec oprávnění uvedených v Prohlášení je nezbytný souhlas autora.

Odkaz na tuto práci: Mariia Korshunova. *VoxelMaze I. - animace a optimalizace UX*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Shrnutí	x
Seznam zkratk	xi
1 Analýza a návrh	1
1.1 O projektu VoxelMaze	1
1.1.1 Historie projektu	1
1.1.2 Popis hry	2
1.2 Uživatelské rozhraní	3
1.2.1 Základní principy tvorby uživatelského rozhraní	3
1.2.2 Rešerše obdobných kooperativních her	3
1.2.3 Testování aktuální verze	11
1.2.4 Návrh vylepšeného uživatelského rozhraní	11
1.2.5 Shrnutí	19
1.3 Animace	20
1.3.1 Reprezentace 3D modelu	21
1.3.2 Přehled existujících programů pro vývoj 3D grafiky a animace	24
1.3.3 Rešerše možností animace	26
1.3.4 Rešerše možností animace voxelově zobrazených modelů	29
1.3.5 Návrh animace postav	30
2 Realizace návrhů	33
2.1 Uživatelské rozhraní	33
2.1.1 Odejít	35
2.1.2 Otevřít chat	35
2.1.3 Zvolit jinou herní postavu	36
2.1.4 Připojit se ke hře	36
2.2 Animace	41
2.2.1 Vytvoření kostry	41
2.2.2 Nastavení závislosti kostí	42
2.2.3 Inverzní kinematika	43
2.2.4 Nastavení váhy	43
2.2.5 Klíčové snímky	43
2.3 Testování	44
2.3.1 Dotazník	45
2.3.2 Výsledky	47

3 Závěr	51
3.1 Doporučení pro další práce	51
Obsah přiloženého média	55

Seznam obrázků

1.1	VoxelMaze, biom Egypt	2
1.2	VoxelMaze, biom vodní	3
1.3	VoxelMaze, diagram aktivit	4
1.4	Counter Strike, založení nové hry	5
1.5	Counter Strike, rozhraní hry	6
1.6	Among US, založení hry	7
1.7	Minecraft, nová hra	8
1.8	Minecraft, rozhraní hry	9
1.9	Quake III Arena, nová hra	9
1.10	Quake III Arena, rozhraní hry	10
1.11	VoxelMaze, stávající verze, menu připojení	12
1.12	VoxelMaze, stávající verze, menu nastavení	12
1.13	VoxelMaze, stávající verze, hlavní menu	13
1.14	VoxelMaze, stávající verze, UI hry	13
1.15	Hierarchie uživatelského rozhraní VoxelMaze	14
1.16	VoxelMaze, wireframe hlavního menu	15
1.17	VoxelMaze, wireframe seznamu místností	16
1.18	VoxelMaze, wireframe výběru postavy	17
1.19	VoxelMaze, wireframe hry	18
1.20	VoxelMaze, wireframe mapy	19
1.21	VoxelMaze, herního menu	20
1.22	VoxelMaze, konce hry	21
1.23	Vrchol, hrana a polygon	21
1.24	Topologie 3D modelu	22
1.25	Polygonální modelování	22
1.26	NURBS	23
1.27	3D sculpting	23
1.28	CAD modelování	24
1.29	2D animace	26
1.30	2D vektorová animace	27
1.31	Stop motion animace	27
1.32	Motion capture animace	28
1.33	Keyframing	29
1.34	Případy užití animace	31
1.35	Prototyp animace chůze	31
1.36	Prototyp animace běhu	32
2.1	VoxelMaze, nastavení tlačítka	33
2.2	Reprezentace grafických prvků v Unity	35
2.3	VoxelMaze, realizace UI hlavního menu	37
2.4	VoxelMaze, realizace UI vytvoření místnosti	37
2.5	VoxelMaze, realizace UI výběru postavy	38
2.6	VoxelMaze, realizace UI hry	38

2.7	VoxelMaze, realizace UI mapy	39
2.8	VoxelMaze, realizace UI výhry	39
2.9	VoxelMaze, realizace UI prohry	40
2.10	Pipeline vytvoření kostry	41
2.11	Vytvoření kostry	42
2.12	Nastavení rodičovství kosti	43
2.13	Nastavení vah	44
2.14	Časová osa animace	44
2.15	Klíčové snímky animace skoku	45
2.16	Výsledek testování, 3. otázka	48
2.17	Výsledek testování, 6. otázka	48
2.18	Výsledek testování, 9. otázka	48
2.19	Výsledek testování, 10. otázka	49
2.20	Výsledek testování, 11. otázka	49
2.21	Výsledek testování, 12. otázka	50
2.22	Výsledek testování, 13. otázka	50

Seznam tabulek

1.1	Porovnání přivětivosti uživatelského rozhraní	11
-----	---	----

Seznam výpisů kódu

2.1	Opuštění aktuální místnosti	35
2.2	Chat	35
2.3	Změna postavy	36
2.4	Příprava ke hře	36

Chtěla bych poděkovat především svému vedoucímu pánu Ing. Radku Richtrovi, Ph.D. za pozitivní přístup k vedení mé práce a velkou morální podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při 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, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 2. března 2021

.....

Abstrakt

Hlavní cílem této bakalářské práce je navrhnout a implementovat uživatelské rozhraní pro aplikaci VoxelMaze, která vznikla v týmu studentů oboru Počítačová grafika v rámci předmětu BI-SP1.2. Rozhraní hry VoxelMaze, původně navržené na míru laboratoři SAGElab, je v rámci této práce upraveno pro širší použití pomocí analýzy existujících úspěšných her, kde byly vyjeveny klíčové vlastnosti doporučené k implementaci. Poté byly vytvořeny náčrtý řešení pro VoxelMaze, které byly i následně provedeny. Druhým cílem práce je návrh a realizace animací hlavních herních postav aplikace. Prozkoumala jsem existující možnosti animace a vybrala nejvhodnější metodu pro grafiku aplikace VoxelMaze. Po vytvoření prototypů pohybů byla implementována vylepšená animace. Výsledkem mé práce je implementované uživatelské rozhraní a realizované animace herních postav, čímž se zlepšil celkový dojem uživatele z aplikace.

Klíčová slova VoxelMaze, animace, uživatelské rozhraní, multiplayer, SAGE2, testování

Abstract

One of the goals of this bachelor thesis is to design and implement a user interface for the VoxelMaze application, which was developed by a team of Computer Graphics students in the BI-SP1.2 course. Initially, I analyzed several existing successful games to highlight the key features recommended for implementation, then sketched solutions for VoxelMaze, which were then implemented and tested by selected candidates from the same specialization. The second goal of the work is to design and implement animations of the main game characters of the application. I explored existing animation options and chose the most suitable method for the graphics of the VoxelMaze application. After prototyping the movements, animation was implemented. As a result of my work, the user interface and animations of the game characters were implemented, improving the overall user experience of the application.

Keywords VoxelMaze, animation, user interface, multiplayer, SAGE2, testing

Motivace

Málokdo si umí představit 21. století bez počítačových her. S velkým pokrokem počítačových technologií rostou i možnosti našich strojů. Každý rok jsme schopni vytvářet více realistickou grafiku, nastavovat její laťku výš a výš. Proto pro vývojáře grafického obsahu je důležité neustále sledovat vývoj a používat nové technologie, aby se uživatelovi chtělo vracet k produktu. Výstup této práce na příkladu hry VoxelMaze ukáže, jak lze přistupovat k této problematice z hlediska uživatelského rozhraní a grafické reprezentace obsahu aplikace.

Aplikace VoxelMaze je kooperativní RPG hra vytvořena v rámci předmětu BI-SP1.2 a BI-VHS, která pokračuje ve vývoji i dnes. Aktuálně je hra implementovaná pouze pro SAGE2 laboratoř, kde běží na 5 serverech synchronizujících obraz do jednoho celku. Projekt je rozsáhlý a pracuje na něm více studentů. Rozhodli jsme se, že VoxelMaze přeneseme i na Windows - nejvíc používanou platformu ve světě počítačových her, aby téměř kdokoliv, kdo má počítač, měl možnost si zahrát naši hru. Žádné postavy v aktuální verzi hry nemají realizovanou animaci a při interakci s jinými postavami/objekty uživateli není úplně jasné, jak se vyvíjí okolnosti a jak má reagovat dále. Tato bakalářská práce se zabývá návrhem a implementací uživatelského rozhraní pro Windows na základě již existujícího rozhraní pro platformu SAGE2 a realizací animací hlavních postav VoxelMaze.

Cíl práce

Cílem práce je na základě všeobecně uznávaných principů zrealizovat návrh uživatelského rozhraní aplikace VoxelMaze pro operační systém Windows, dále po rešerši existujících metod vytvo-

ření animace 3D voxelových modelů navrhnout prototypy stavů všech herních postav a implementovat je.

Postup

Tato práce je rozdělena na 2 části: teoretickou část, která je věnována analýze stávající hry a návrhu lepší verze, a praktickou, ve které probíhá implementace návrhů.

Teoretická část se bude zakládat na rešerši UI obdobných kooperativních her. Následně bude provedena rešerše možností animace voxelově zobrazených modelů a obecných možností animace.

Na základě výsledků analýzy a návrhu z teoretické části bude implementováno v praktické části uživatelské rozhraní pro Windows a realizována animace postav. Následně budou otestovány provedené změny uživatelského rozhraní a porovnány výsledky stávající a upravené verze aplikace.

Výsledky práce

Výsledkem mé práce jsou splněné cíle stanovené výše: implementované přehledné uživatelské rozhraní pro počítačovou aplikaci VoxelMaze odpovídající požadavkům uživatele a realizované animace všech hlavních herních postav.

Závěr

Pomocí této práce se aplikace VoxelMaze posunula výš na schodech vývoje a jsem si jistá, že v brzké době ji čekají i další velké změny. V aplikaci samozřejmě vždy bude místo pro vylepšení implementace.

Seznam zkratk

OS	Operační systém
NURBS	Non-uniform rational basis spline
SAGE2	Scalable Amplified Group Environment
UI	User Interface - uživatelské rozhraní
VFX	Vizuální efekty

Analýza a návrh

Teoretickou část bakalářské práce jsem rozdělila na 3 logické části: popis projektu, uživatelské rozhraní a animace. Ve dvou posledních částech bude provedená analýza současného stavu, rešerše obdobných řešení, vyjevení slabých míst a návrh vylepšeného řešení.

1.1 O projektu VoxelMaze

Voxel (*volumetric, pixel*) analogie 2D pixelů ve 3D prostoru [4].

RPG (*Role-playing game*) žánr počítačové hry, kde jeden hráč ovládá postavu s určitou sadou charakteristik.

Multiplayer režim počítačové hry, během kterého hraje více než jedna osoba.

Herní engine základní software pro vývoj počítačové hry.

SAGE2 (*Scalable Amplified Group Environment*) webový server, aplikace v Node.js [5].

UI (*User Interface*) uživatelské rozhraní.

Pipeline sada propojených prvků definujících postupné zpracování dat.

VoxelMaze je voxelová RPG pro 2 až 4 hráče (*multiplayer*), která je realizovaná pomocí multiplatformního herního engine Unity [12]. Původně aplikace je implementovaná pouze pro webové rozhraní SAGE2, nyní je ve stavu implementace pro platformy Windows a Android. Podporovatelná vstupní zařízení: ovládače (*gamepad*), klávesnice a myš.

1.1.1 Historie projektu

První funkční verze hry vznikla v rámci práce nad semestrální úlohou v předmětu Softwarový týmový projekt 1 (BI-SP1) v letním semestru v roce 2020. Seznam členů a jejich odvedené práce:

- **Oleksandr Khokhych** — implementace jádra programu,
- **Pavel Plotnikau** — implementace klient-serverové části,
- **Mariia Korshunova** — tvorba grafiky,

- **Nina Grechushkina** — tvorba grafiky,
- **Denis Dubin** — hudební doprovod.

Dále práce nad aplikací pokračovala v předmětech Softwarový týmový projekt 2 (BI-SP2.1) a Virtuální herní světy (BI-VHS) v zimním semestru v roce 2020, kde v týmu přibyli další dva vývojáři:

- **Artem Moskalenko** — tvorba grafiky,
- **Michal Rudolf** — tvorba grafiky.

1.1.2 Popis hry

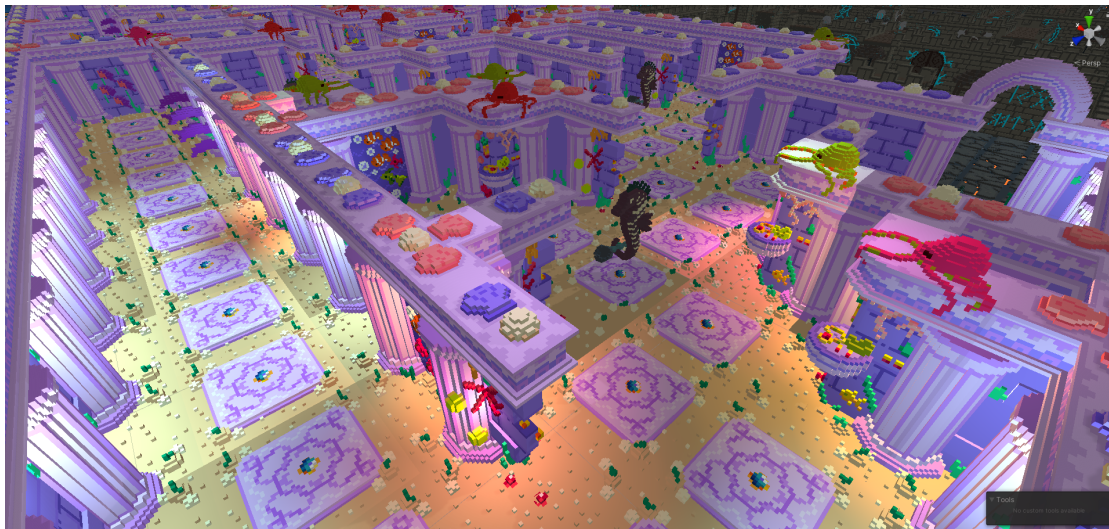
Každá postava má za cíl přežít v náhodně generovaném bludišti, v němž se odehrává příběh, a dobýt co nejvíce věží, kde jsou uloženy cenné znalosti o světě postav, a získat tím výkonnou technologii, která zachrání svět. Postavy se cestou potkávají s nepřáteli a obcházejí nebezpečné pasti. V aktuální verzi hry existují 4 postavy reprezentující 4 různé biomy (viz příklady biomů na Obrázcích 1.1 a 1.2) jedné staré civilizace. Pátý biom je biom zahynulých mudrců, kteří zakleli a schovali nejdůležitější část technologie ve své věži.

1. **Elly** reprezentuje sladký biom (*Sweet*),
2. **Alice** reprezentuje biom vodní (*Water*),
3. **Robot Daniel** reprezentuje neonový biom (*Neon*),
4. **Bob** reprezentuje písčiny Egypt.



■ **Obrázek 1.1** VoxelMaze, biom Egypt

Na základě dokumentace udělané mým kolegou Pavlem Plotnikau v rámci jedné z iterací předmětu BI-SP1 jsem vytvořila diagram popisující průběh jednoho herního kola (viz Obrázek 1.3) pro lepší pochopení hry.



■ Obrázek 1.2 VoxelMaze, biom vodní

1.2 Uživatelské rozhraní

Analýza uživatelského rozhraní by se měla zakládat na všeobecně uznávaných konceptech, které by byly aplikovatelné pro návrh jakékoli aplikace. Tak máme možnost objektivně provést analýzu přívětivosti a použitelnosti rozhraní.

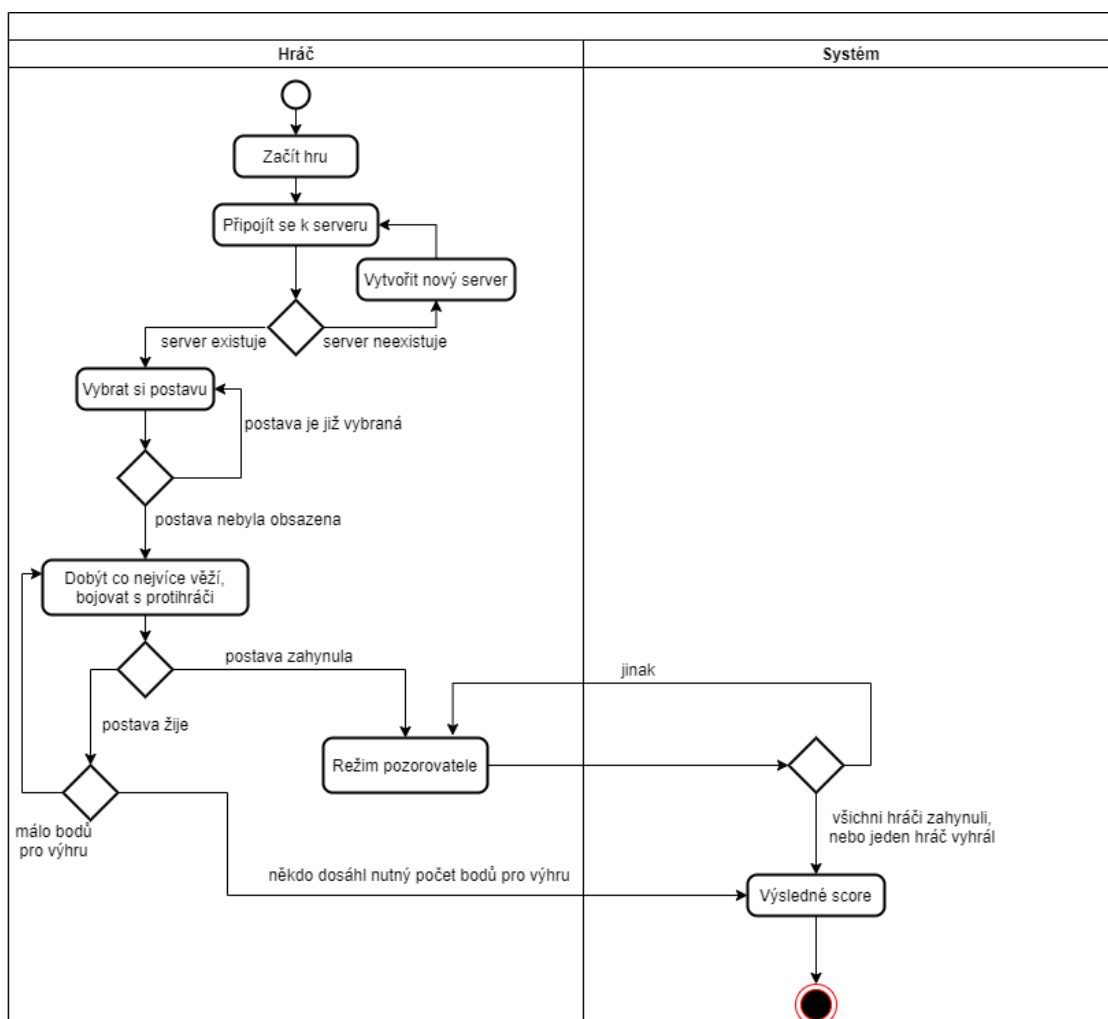
1.2.1 Základní principy tvorby uživatelského rozhraní

Dle [2] existují následující základní principy vývoje uživatelského rozhraní:

1. konzistence — jednou z výhod kompatibility je přenesení znalostí a zvyků uživatele z jiných programů,
2. pohodlnost — nesmíme zatížit paměť uživatele tím, že si bude muset zapamatovat funkcionalitou každého prvku, rozhraní aplikace musí být intuitivní,
3. vyšší kvalita — často dobrou zkušeností je nechat uživatele rozhodnout se samostatně, co pro něj bude nejlepším řešením. Tohle pravidlo je aplikovatelné nejen v informatice: v případě stavby většího komplexu budov architekti musí vymyslet, jak budou proloženy asfaltové cesty mezi domy. Neefektivnějším řešením je ale nechat na trávníku ceduli s nadpisem povolujícím vstup na trávník. A již za pár dní na trávníku budou vidět vyšlapané místními obyvateli cestičky; zbývá už jenom vyplnit cesty asfaltem. V našem případě rozhraní také musí odpovídat potřebám uživatele a nejlepším řešením bude poskytnout uživateli možnost ovlivnit návrh.

1.2.2 Rešerše obdobných kooperativních her

Na základě výše popsaných principů jsem určila jednotlivé body splňující kritéria přívětivosti UI. Následně mnou bude provedená analýza uživatelského rozhraní již existujících úspěšných řešení. Pro účely inspirace a rešerše byla vybrána podmnožina čtyř typických zástupců počítačových her, odpovídajících popisu aplikace VoxelMaze: multiplayer, režimy lokálního a online připojení, soutěž dvou a více hráčů.



■ **Obrázek 1.3** VoxelMaze, diagram aktivit

■ Konzistence

- referenční body — uživatel každou chvíli musí mít přehled o aktuálním stavu aplikace: navigační mapy, záhlaví karet, stromové struktury,
- uložení výsledků interakce s rozhraním — musíme oznámit uživatele o neuložené změně konfigurace,

■ Pohodlnost

- flexibilní rozhraní — možnost použití více vstupních zařízení: myš, klávesnice atd.,
- srozumitelné zprávy bez použití odborné terminologie při komunikaci s uživatelem,
- intuice — zaměřením se na rozpoznání prvků rozhraní, ne na jejich zapamatování,
- zrušení poslední akce — implementace funkce zrušení poslední akce a zopakování zrušené akce (UNDO/REDO),
- rychlé cesty — defaultní parametry umožňující přeskočení zdlouhavého nastavení,
- vizuální pomocníky — tvar kurzoru, změna barvy pozadí atd.

■ Vyšší kvalita — provést testování UI.

Counter Strike

Counter-Strike [13] je online hra, jejíž podstatou je soutěž dvou týmů — teroristů a protiteroristů / speciální jednotky. Oba týmy v každém kole můžou získat body plněním cílů nebo zničením nepřátelského týmu. Nejpopulárnějším scénářem je situace, kdy teroristé potřebují v určitém okamžiku umístit na konkrétním místě bombu a speciální jednotky ji musí zneškodnit. Poté, co tým zvítězí v určitém počtu kol, vyhraje i celý zápas.

Uživatelské rozhraní aplikace je maximálně jednoduché. Na úvodní obrazovce v menu jsou pouze 4 možnosti:

1. Nová hra (*New game*),
2. Najít servery (*Find Servers*),
3. Nastavení (*Options*),
4. Ukončit (*Quit*).



■ Obrázek 1.4 Counter Strike [13], založení nové hry

Při vytvoření nové hry (viz Obrázek 1.4) v menu je hodně parametrů, nad kterými by vidící poprvé aplikaci uživatel musel přemýšlet, ale jsou předvyplněné defaultními hodnotami pro ušetření času. Vytvoření hry lze uskutečnit pomocí jenom jednoho okénka, referenční body pro orientaci v nastavení proto nejsou třeba. V případě, že se uživatel po ukončení nastavení parametrů rozhodne odejít a hru nezačínat, aplikace se nezeptá, zda si uživatel je jistý tím, že chce odejít, a změněnou konfiguraci neuloží.



■ **Obrázek 1.5** Counter Strike [13], rozhraní hry

V režimu hry (viz Obrázek 1.5) jsou vidět pouze zbývající do konce kola čas a hodnoty aktuálního stavu hráče: zdraví, peníze, náboje atd. Vizuelní nápovědy jako statistika kola, zbývající čas a zprávy s událostmi informují hráče o aktuálním stavu hry. Základní funkce — chůze a střelení — se řídí klávesy **W**, **A**, **S**, **D**, mezera (*Space*) a myší. Najít další funkce je možné pomocí stisknutí speciálních kláves, což není tak intuitivní záležitost a může to být nepříjemné pro nového uživatele neseznámeného s rozhraním:

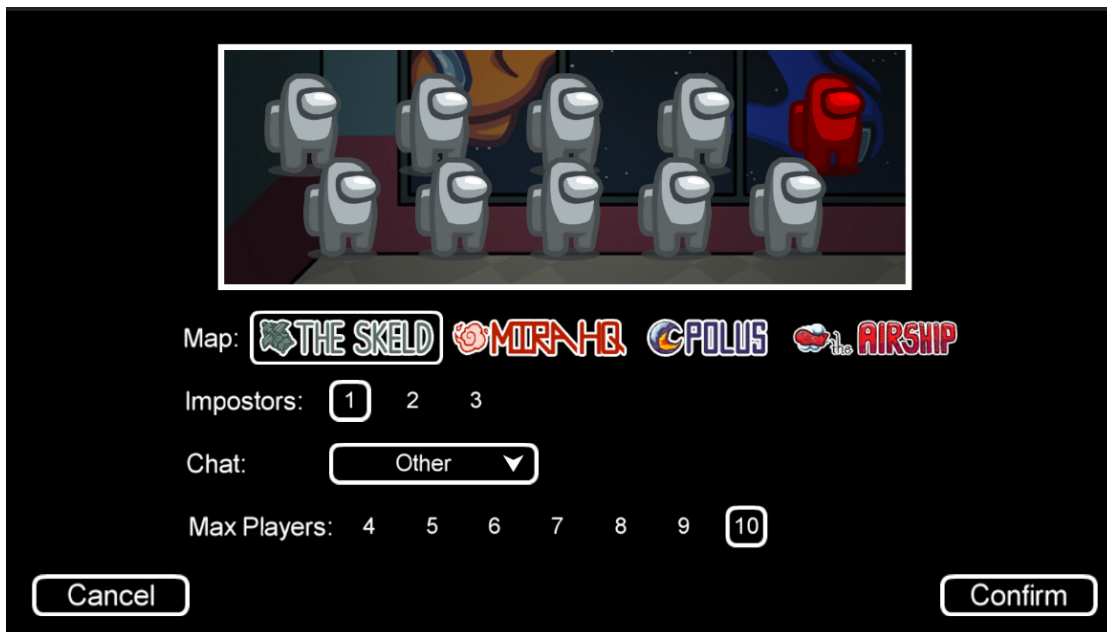
- klávesa **B** — otevřít obchod se zbraněmi,
- klávesa **E** — změnit používanou zbraň,
- klávesa **Z** — vyslat zprávu pomocí rádia atd.

Among Us

Among Us [8] je online hra, ve které hráči musí najít zrádce (*Impostor*) svého týmu na vesmírné lodi. Po celé lodi je rozmístěno spousta úkolů, jež splnění je potřebné k přežití celé posádky. Cílem hry je plnění úkolů a odtajení zrádce, který postupně zabíjí spoluhráče. Among Us je obdobou populární hry Mafie, kde pro úspěšné dokončení hry je potřeba uvést své soupeře do rozpaků a udělat vše pro to, aby se neprozradil.

Uživatelské rozhraní menu je velmi intuitivní i pro uživatele pouze se základní angličtinou. Hned na začátku hra nabízí 2 možná připojení: lokální a online. Ostatní možnosti nastavení jsou reprezentované pomocí napovídajících ikoněk. Pro založení nové hry uživatel potřebuje jenom základní povinné nastavení (viz Obrázek 1.6), všechno má defaultní hodnoty:

- výběr mapy,
- určení maximálního počtu hráčů,
- určení počtu negativních postav (*Impostorů*).



■ **Obrázek 1.6** Among US [8], založení hry

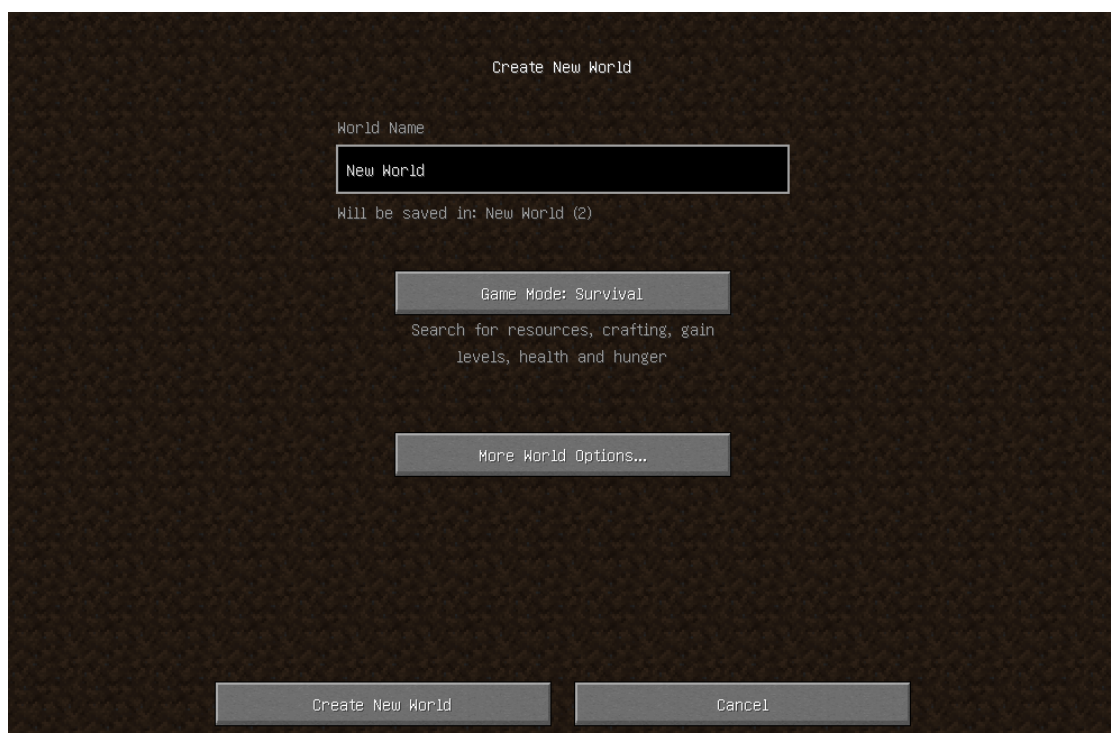
Při vytvoření nové hry uživatel se vystačí pouze s jedním okénkem nastavení, referenční body zde také nejsou třeba. Rozhraní také nabízí mapu hracího prostoru, která pomáhá zorientovat se na vesmírné lodi a rozhodnout se, jakým se vydat směrem. Ovládání je realizováno pomocí klávesnice (klávesy **W**, **A**, **S**, **D**) a myši.

Běžnému uživateli, co vidí aplikaci poprvé a nerad čte zdlouhavé popisy pravidel, by nebylo jasné, co je cílem hry a co se od něj očekává, kdyby ve hře chyběla napovídající animace šipek ukazujících směr k jednomu z úkolů na lodi pro členy posádky. Pro *Impostera* se zase v okamžiku, kdy je možné zabít jednoho ze spoluhráčů, rozsvítí tlačítko „Zabít“ (*Kill*).

Minecraft

Minecraft [1] je hra s otevřeným světem (*open-world game*). Stylově je herní svět zcela tvořen voxely (bloky): terén, objekty a hráči. Pro texturování se používají textury s nízkým rozlišením (16x16 texelů). Cíle hry jsou jednoduché: hráči získávají prostředky na stavení budov, hradů a celých měst. V Minecraftu neexistují prakticky žádná omezení, kromě výšky staveb — jinak si hráči mohou vytvářet cokoli.

Úvodní menu nabízí vhodnou kombinaci barev, bílé písmo na tmavě šedém pozadí je dobře čitelné. Pro nového uživatele, co nechce měnit žádná nastavení a zatím není seznámen s aplikací, je snadné najít cestu k založení nové hry. Žádné referenční body rozhraní v menu nenabízí, aplikace se také neptá na případnou nepotvrzenou změnu nastavení. Při vytvoření nové hry (viz Obrázek 1.7) podle mne není vhodně zvoleno pořadí rozmístění tlačítek „Vytvořit nový svět“ (*Create New World*) a „Zrušit“ (*Cancel*), intuitivně člověk hledá potvrzovací tlačítko vpravo a tlačítko zrušení akce vlevo.



■ **Obrázek 1.7** Minecraft [1], vytvoření nové hry

Hra má minimálně zatížené rozhraní (viz Obrázek 1.8), na obrazovce uživatel vidí pouze aktuální inventář, současný stav postavy a zmenšenou mapu. Ovládat postavu je možné klasicky pomocí **W**, **A**, **S**, **D**, mezery a myši. Pomocí speciálních kláves se zase otevře přístup k dalším funkcím:

- klávesa **M** dává přístup k mapě,
- klávesa **E** otevře pytel se sadou věcí, co vlastní postava,
- klávesa **F** prohodí levou ruku s pravou atd.

Uživatel v každém kroku má možnost zanechat založení nové hry a vrátit se na původní obrazovku. Cíle hry jsou jasné, proto ovládání a zorientování se v herním prostoru musí být snadné pro pochopení. Animované zprávy při jakékoliv události informují hráče o stavu hry.

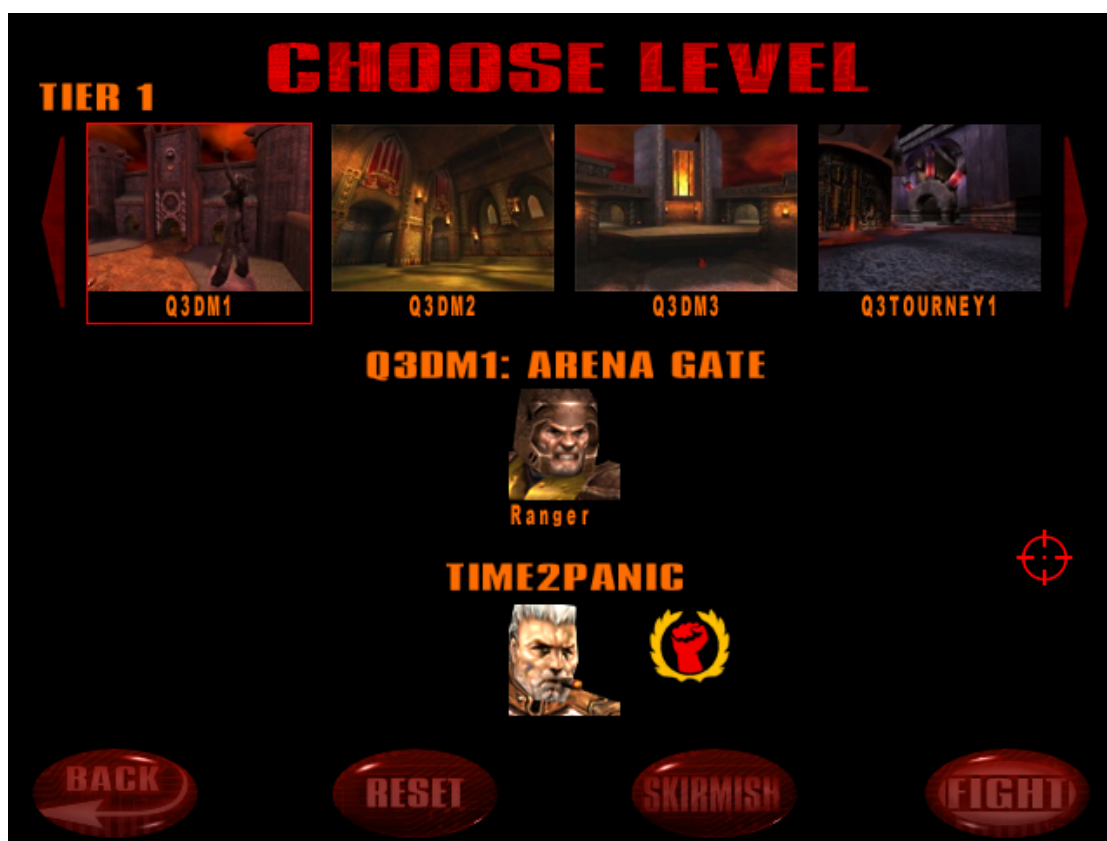
Quake III Arena

Quake III Arena [7] je další online hra, ve které cílem hlavní postavy (gladiátora) je zničit nepřátelé. Hra má více režimů, pro jednoho hráče a pro vícero:

- single player,
- multiplayer
 - Free For All — každý bojuje za sebe,
 - Team Deathmatch — týmový souboj,
 - Tournament — jeden proti jednomu,
 - Capture The Flag (CTF) — souboj o vlajku.



■ Obrázek 1.8 Minecraft [1], rozhraní hry



■ Obrázek 1.9 Quake III Arena [7], založení nové hry

Uživatelské rozhraní aplikace nabízí v prvních 2 řádcích nejpopulárnější volby uživatele: single



■ Obrázek 1.10 Quake III Arena [7], rozhraní hry

player a multiplayer. Pro vytvoření nové hry je potřeba pouze zvolit mapu a vybrat postavu, je možné využít defaultního nastavení a kliknout rovnou na „Začít souboj“ (*Fight*) (viz Obrázek 1.9), problém s orientací v menu uživatel mít nebude. Během hry uživateli je přístupné klasické ovládání postavy pomocí kláves **W**, **A**, **S**, **D**, **Shift**, mezera a myši. Je také možné zobrazit pomocí klávesy **Tab** seznam všech postav — živých a mrtvých. O zabití protihráče / smrti naší postavy nás informuje zpráva uprostřed obrazovky (viz Obrázek 1.10).

Porovnání rozhraní

Na závěr shrneme přívětivost uživatelského rozhraní jednotlivých aplikací pomocí tabulky se dříve definovanými kritérii 1.1. Na základě tabulky můžeme udělat závěr, že v případě minimálního rozhraní není potřeba se zabývat referenčními body pro orientaci, ujistěním se, zda uživatel opravdu chce zahodit provedené změny a také zrušením/zopakováním posledně provedené akce.

Naopak, jak je vidět z tabulky, má smysl se zaměřit na implementaci uživatelského rozhraní s intuitivními prvky a vizuální interakci.

	Counter-Strike	Among Us	Minecraft	Quake III Arena
referenční body	ne	ne	ne	ne
neuložená konfigurace	ne	ne	ne	ne
více vstupních zařízení	ano	ano	ano	ano
srozumitelné zprávy	ano	ano	ano	ano
zrušení poslední akce	ne	ne	ne	ne
defaultní parametry	ano	ano	ano	ano
rozpoznání rozhraní	ano	ano	ne	ano
vizuální pomocníky	ano	ano	ano	ano

■ **Tabulka 1.1** Porovnání přívětivosti uživatelského rozhraní

1.2.3 Testování aktuální verze

V této podkapitole bude otestována stávající verze aplikace VoxelMaze. Jelikož UI aplikace pro OS Windows zatím není implementováno, ukázána bude pouze verze realizována pro platformu SAGE2. Připomenutí: aktuální verze hry se stále nachází ve stavu vývoje a proto se může hodně lišit od konečné verze, která bude popsána na konci této práce: programátoři našeho týmu se nestále zabývají úpravami logiky hry a přidáním nových herních mechanik.

Připojení

Po spuštění aplikace je k vidění synchronizační menu (viz Obrázek 1.11), které hned nabízí uživateli zadání jména místnosti pro připojení. Pro připojení k jedné místnosti je třeba zadat stejný název, jenž slouží jako unikátní identifikátor pro každou místnost.

Menu nastavení a hlavní menu

Po připojení k místnosti se uživatel dostává k menu nastavení, kde si může vybrat část obrazovky pro hru zmáčknutím libovolného tlačítka na ovladači/gamepadu (viz Obrázek 1.12) a přejít do hlavního menu (viz Obrázek 1.13).

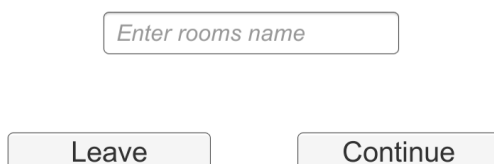
Hra

Po tom, jak všichni uživatelé zmáčknou tlačítko „Game“, hra začíná. Aktuálně na obrazovce je vidět 4 herní části určené pro 4 hráče, které nejsou žádným způsobem graficky odděleny. V horní části každé ze 4 obrazovek je uveden aktuální počet bodů (*Score*) každého hráče. Chybí možnost zobrazení mapy pro alespoň minimální orientaci v bludišti. Žádná z postav nemá realizovanou animaci — při chůzi se postava pouze pohybuje dopředu. Spojit se s jinými uživateli není možné. Na konci hry všichni účastníci budou přeneseni zpátky do hlavního menu.

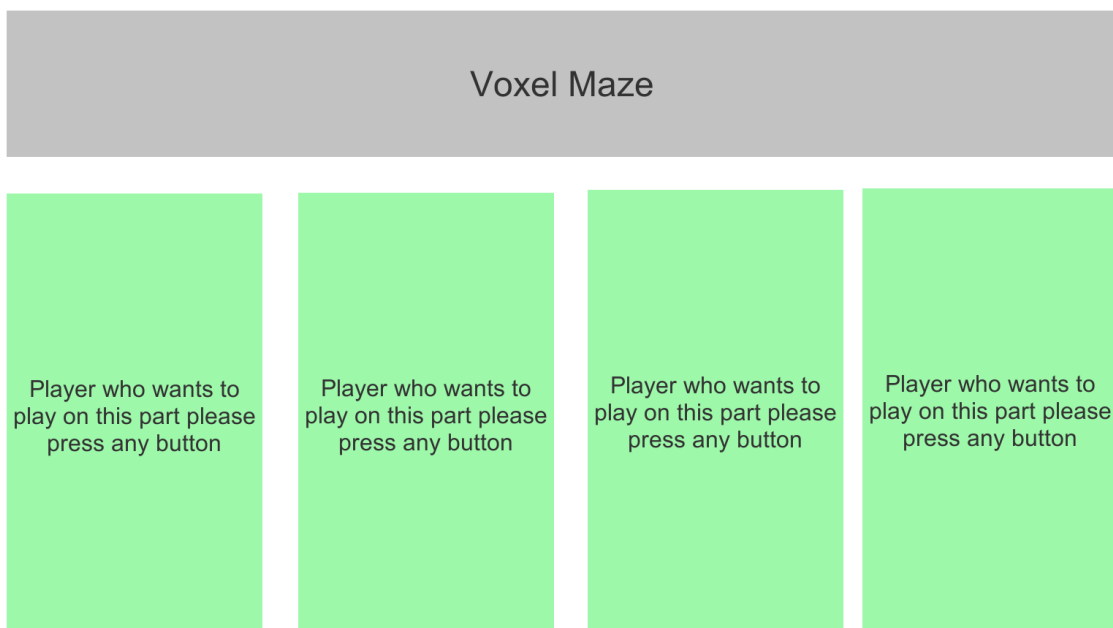
1.2.4 Návrh vylepšeného uživatelského rozhraní

Na rozdíl od platformy SAGE2, kde byly současně spouštěny a synchronizovány 4 verze hry, ovládání hry ve verzi aplikace pro operační systém Windows bude upraveno pro použití pouze jedním hráčem. V nové verzi bude odstraněné menu synchronizace display-klientů, místo něhož bude implementováno menu místností, kde bude možné buď vytvořit nové připojení, nebo se připojit k již existujícímu. Dále bude implementována možnost výběru herní postavy. Některá část rozhraní, která je již implementována na SAGE2 (realizované funkce), bude převzata a upravená pro počítačovou verzi.

Jak víme z předmětu BI-TUR (Tvorba uživatelského rozhraní), návrh uživatelského rozhraní by správně vždy měl procházet určitou pipeline:

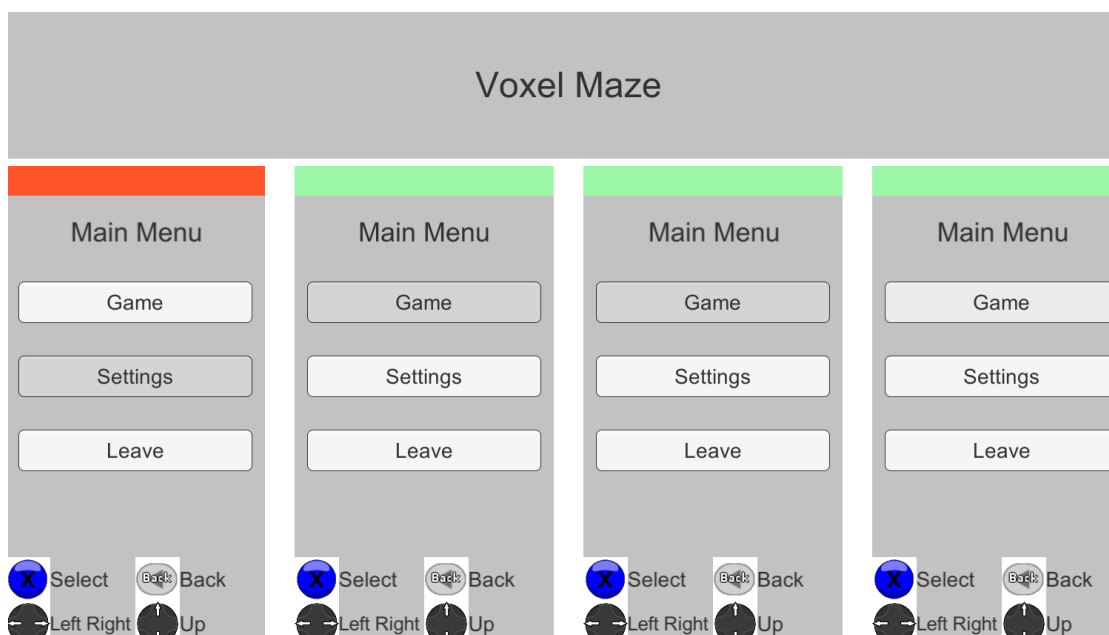


■ **Obrázek 1.11** VoxelMaze, stávající verze, UI menu připojení

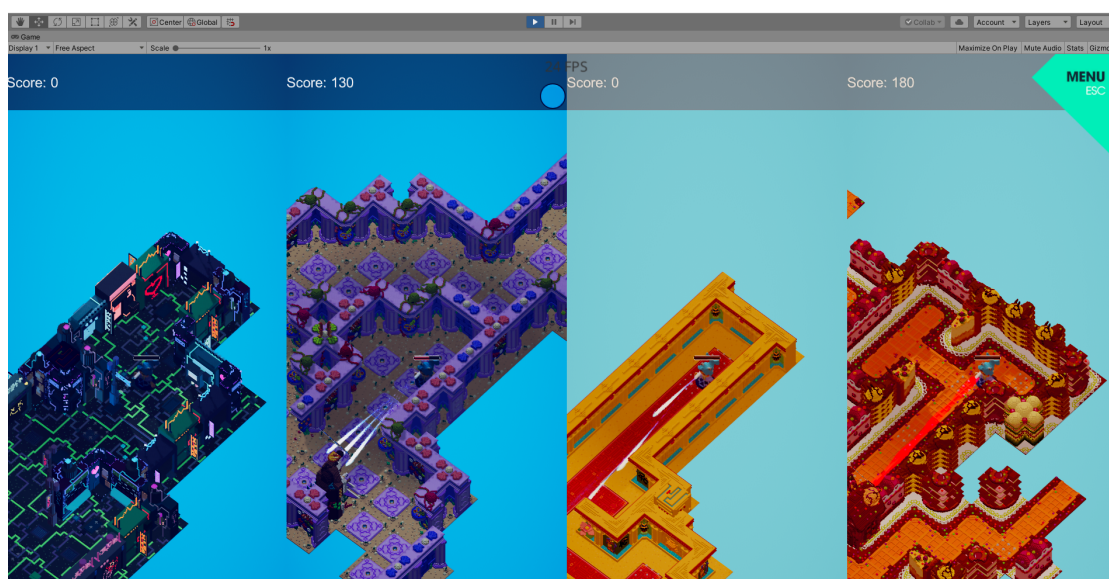


■ **Obrázek 1.12** VoxelMaze, stávající verze, UI menu nastavení

1. náčrty celku — informační architektura,
2. náčrty stránek — logický návrh stránek,
3. maketa — vizuální návrh stránek,
4. prototyp — funkčnost.



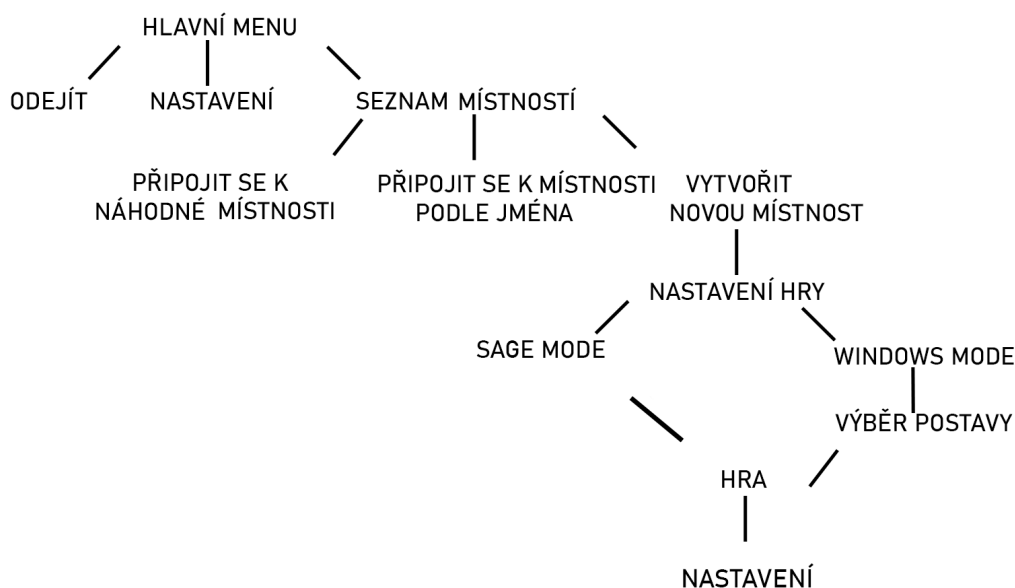
■ Obrázek 1.13 VoxelMaze, stávající verze, UI hlavního menu



■ Obrázek 1.14 VoxelMaze, stávající verze, UI hry

Proto nejdříve bude vytvořeno schéma popisující hierarchii jednotlivých součástí uživatelského rozhraní, poté budou nakresleny náčrty (tzv. wireframy), tyto náčrty se předají našemu kolegovi zabývajícímu se jejich barevnou realizací. Barevné prototypy mnou budou implementovány a dále otestovány vybranými zájemci z cílové skupiny.

Na Obrázku 1.15 je představeno schéma s hierarchií uživatelského rozhraní aplikace.



■ Obrázek 1.15 Schéma UI VoxelMaze

Hlavní menu

Hlavní menu (viz Obrázek 1.16) je první, co uvidí uživatel v aplikaci VoxelMaze. Pozadí hlavního menu musí uvést hráče do kontextu a prezentovat mu hlavní charakteristiky hry: na snímku budou ukázány herní biomy (sladký, neon, Egypt a vodní), jeden ze kterých uživatel bude během hry zastupovat. Zde je potřeba dát pouze podstatné věci bez zbytečného zatížení obsahu:

- Seznam místností (*List of rooms*)
- Nastavení (*Settings*)
- Odejít (*Exit*)

Seznam místností

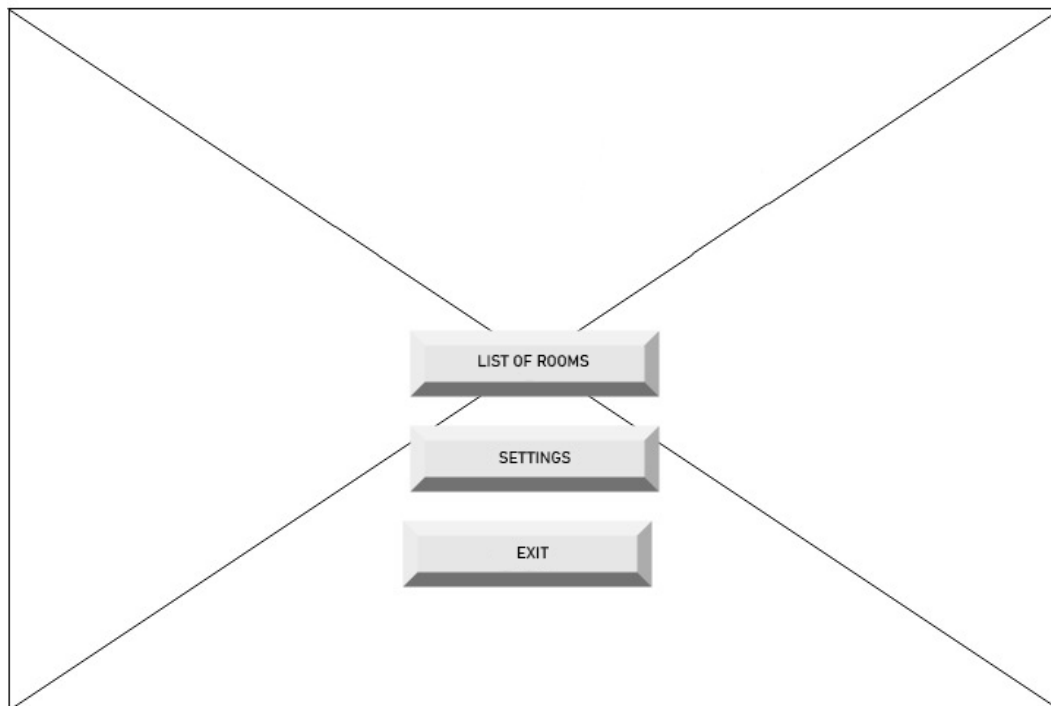
Další menu se seznamem místností, nabízí uživateli několik možností:

- Vytvořit novou místnosti (*Create room*)
- Připojit se k již existující místnosti pomocí názvu (*Join room by name*)
- Připojit se k náhodné místnosti (*Join random room*)
- Vrátit se do hlavní místnosti (*Back to menu*)

Uživatel zde také může zadat své jméno. Pole je ale nepovinné, hráč případně dostane vygenerované jméno, které bude vypadat následovně: **Playerxxxx**, kde **xxxx** je náhodné číslo od 1000 do 9999.

V levém okénku se zobrazí všechny vytvořené místnosti, u kterých je uvedena informace o počtu připojených hráčů, zda místnost dovoluje připojení a jméno místnosti (viz Obrázek 1.17).

HLAVNÍ MENU



■ **Obrázek 1.16** VoxelMaze, wireframe hlavního menu

Nastavení hry

Pro vytvoření nové místnosti uživatel musí určit povinné parametry:

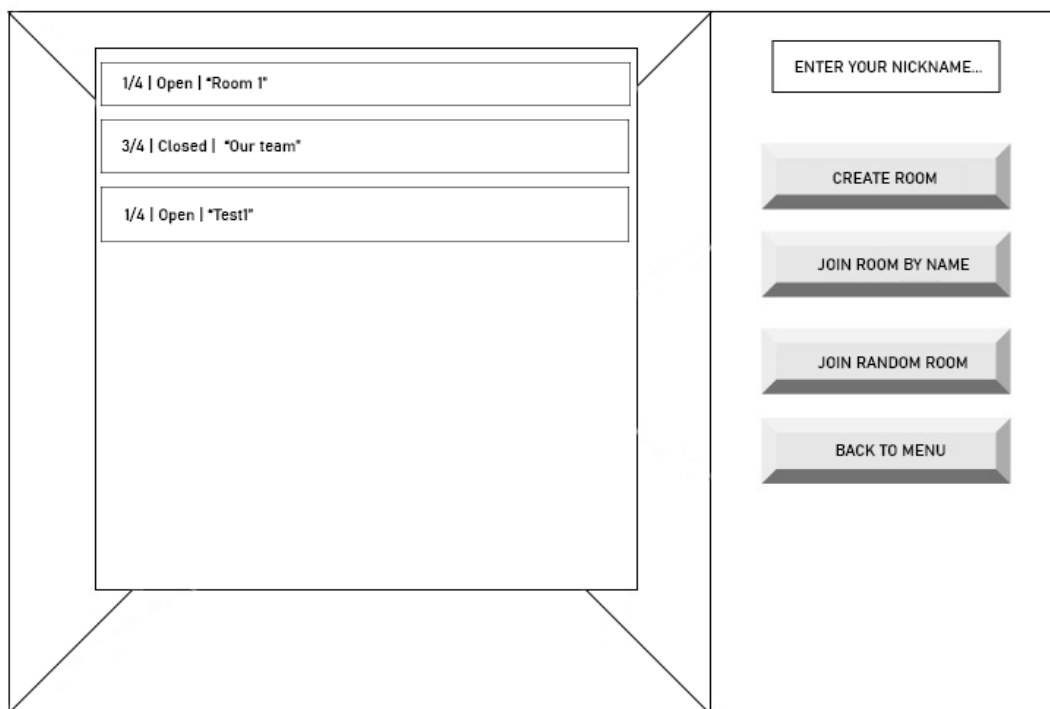
- Jméno místnosti (*Rooms's name*)
- Otevřenost místnosti (*Room is open*)
- Viditelnost místnosti (*Room is visible*)
- Počet hráčů (*Number of players*)
- Imitace Sage modu (*Sage dev mode*)

Jméno místnosti slouží jako identifikátor připojení, proto jméno musí být unikátní. *Otevřenost místnosti* definuje, zda se ostatní hráči mohou připojit k dané místnosti. *Viditelnost místnosti* určuje, zda informace o nově vytvořeném připojení se ukáže i ostatním hráčům v levém okénku. *Počet hráčů* je možné definovat pouze v rozsahu 1–4. *Sage mode* umožňuje spustit hru v režimu Sage, který umožňuje řízení více postav z jednoho počítače. Daný režim se zatím nachází ve stavu vývoje a bude do budoucna implementován.

Výběr postavy

V případě, že uživatel nezaškrtně *Sage mode*, bude mu k dispozici také menu výběru hlavní postavy. Každý hráč, je-li jich více, si může vybrat právě jednu hlavní postavu zastupující jeden ze 4 biomů: robot Daniel, Alice, Elly a Bob.

SEZNAM MÍSTNOSTÍ



■ **Obrázek 1.17** VoxelMaze, wireframe seznamu místností

Každého hráče reprezentuje 1 obdélník s ikonou jeho postavy uvnitř a jménem nahoře (viz Obrázek 1.18). Změna postavy je možná pomocí šípek „vlevo“ a „vpravo“ na obě strany od ikony postavy. Každý hráč má ve svém obdélníku tlačítko:

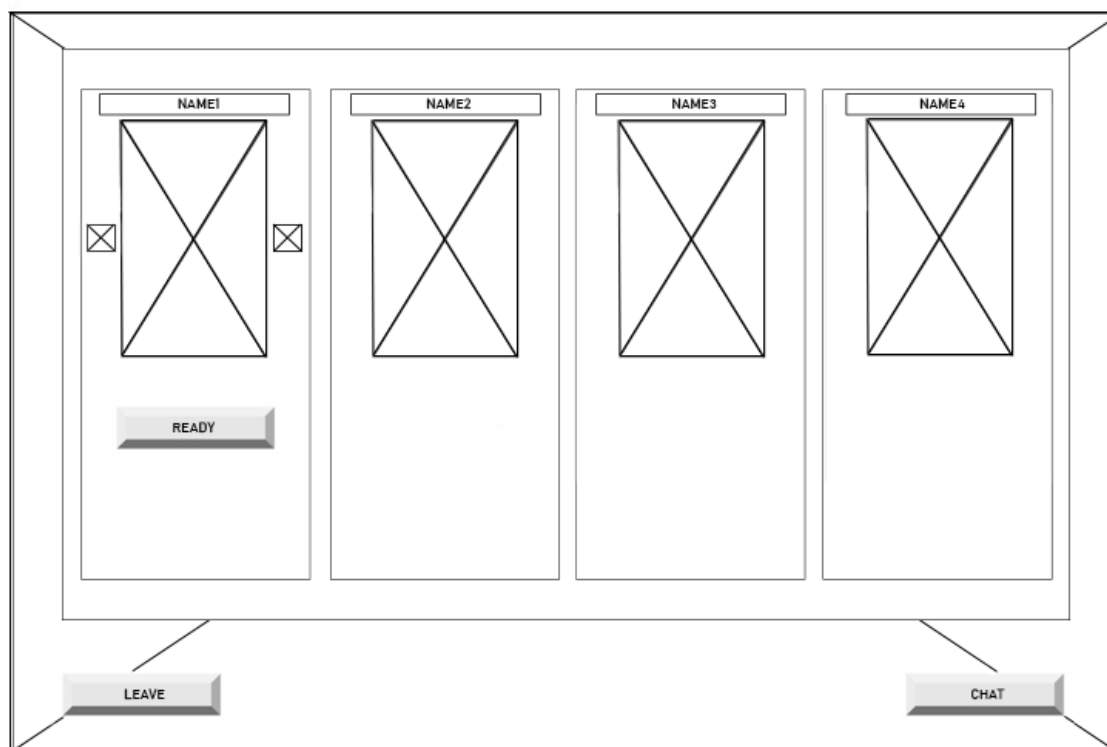
- „Ready“ v případě, že se hráč připojil k již existující místnosti. Stisknutí tlačítka signalizuje připravenosti začít hru. Pokud 2 nebo více hráčů si zvolí stejnou postavu, jejich tlačítko „Ready“ nebude aktivní.
- „Start“ má uživatel, který je hostem místnosti. Tohle tlačítko umožňuje zahájit hru. Tlačítko „Start“ nebude aktivní, dokud požadovaný počet hráčů (určuje se při založení nové místnosti) nebude ve stavu „Ready“.

Tlačítko „Chat“ v dolním pravém rohu obrazovky zobrazí okénko s chatem, pomocí kterého uživatelé mohou mezi sebou komunikovat. Zprávy v okénku mají 3 různé barvy:

- žlutá — systémová zpráva — informuje o událostech uživatelů, např. o připojení hráče k místnosti,
- modrá — vlastní zpráva hráče,
- bílá — zpráva ostatních hráčů.

Pro odeslání zprávy je potřeba zadat text do vstupního pole a stisknout šípku „vpravo“ nebo tlačítko „Enter“. Chat bude schován po kliknutí na jakékoliv jiné místo, než chatové okénko.

VÝBĚR POSTAVY



■ Obrázek 1.18 VoxelMaze, wireframe výběru postavy

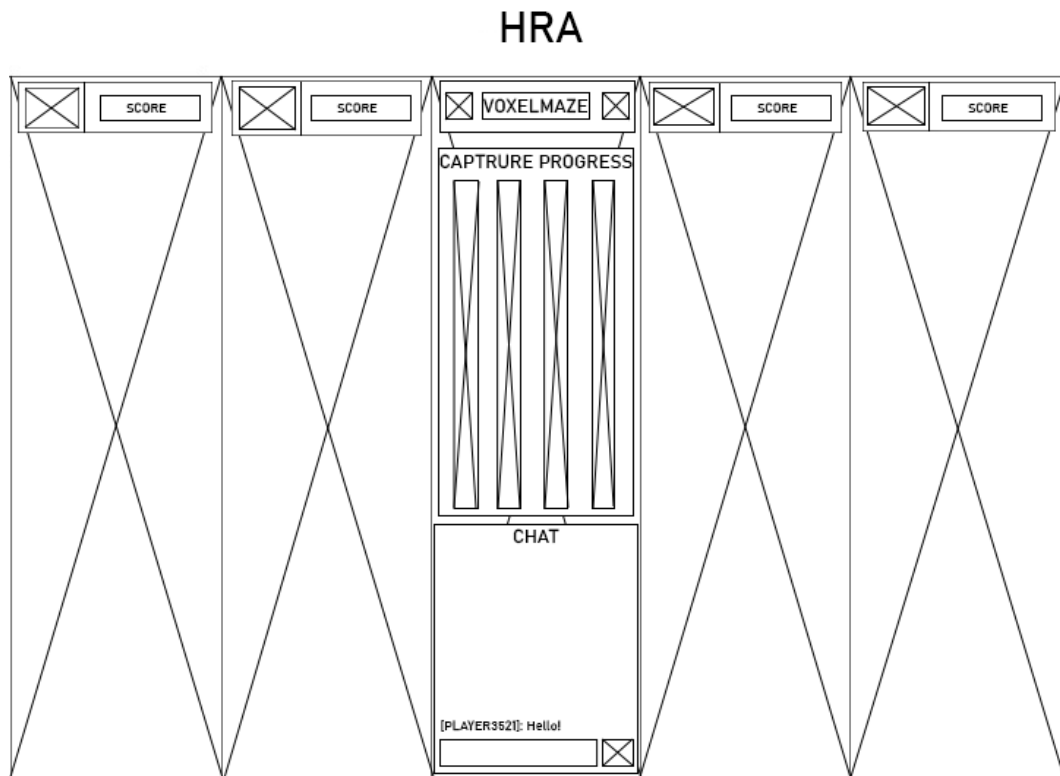
Hra

Po výběru postav následuje obrazovka se hrou, která je rozdělená na více svislých částí: 4 obdélníky jsou určeny pro 4 hráče ve svých biomech a pátá část je prostřední obdélník se statistikou a chatovým okénkem. (viz Obrázek 1.19).

Sloupce v horní části obdélníku ze statistikou informují hráče o tom, jak moc jsou vyvinuté jejich věže v biomech a jak moc jsou hráči blízko k výhře. Vyvinutí svých věží přináší hráčům také určité body, které lze znovu utrácet za další vývoj věží nebo nákup spojenců. Množství těchto bodů je možné vidět nahoře každého ze 4 obdélníků. S věžemi je možné interagovat pomocí integrované mapy (viz Obrázek 1.20), již lze otevřít stisknutím klávesy **M**.

Mapa

Mapa slouží pro lepší orientaci hráče v bludišti. Cestičky na mapě se vykreslují ne na začátku, ale během prozkoumání bludiště. V pravé části okénka uživateli je k dispozici přehled všech 9 věží hry (2 z každého biomu a 1 hlavní). Pokud uživatel rozklikne jednu z věží, pozice na mapě se změní z pozice hráče na lokaci vybrané věže. Pokud vybraná věž je vedlejší, v levé části obrazovky se také otevře interaktivní okénko se 3 možnými funkcemi:

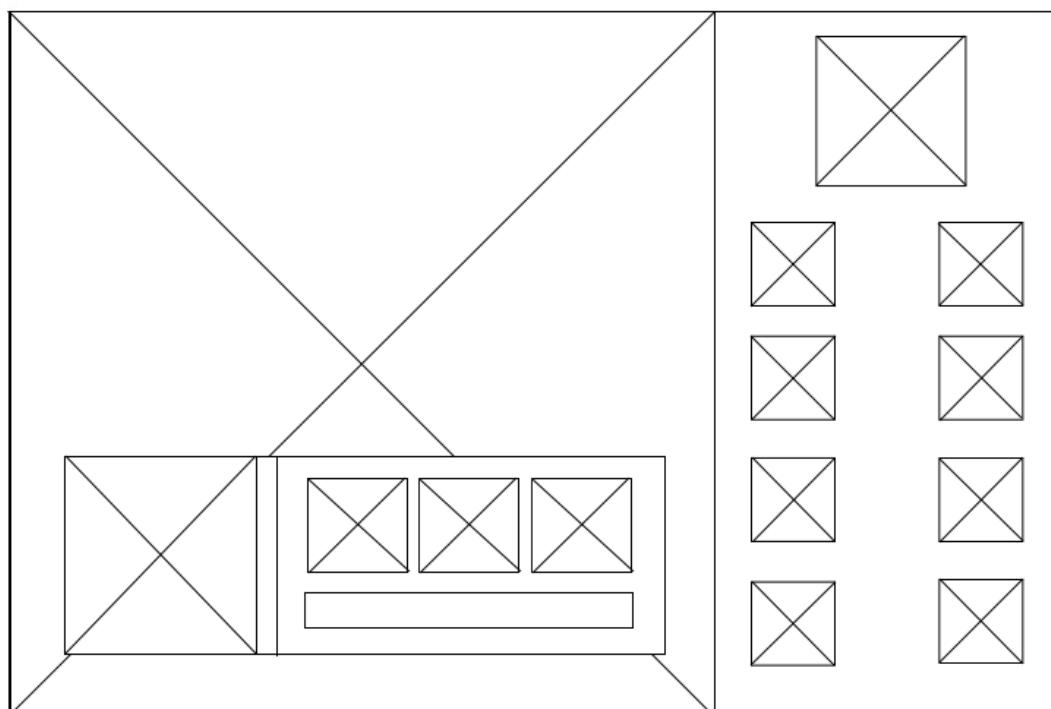


■ **Obrázek 1.19** VoxelMaze, wireframe hry

1. posílit obranu (*Defence*) — zabrání dobytí věže nepřítelem,
2. přejít na další úroveň (*Level up*) — zvýší rychlost dobytí hlavní věže a růstu bodů, maximální úroveň je 5,
3. generovat spojence (*Spawn Mob*) — koupit tzv. moba za body, maximální počet spojenců je také omezený a je roven 15.

Odejít z mapy je možné opět pomocí tlačítka **M**.

MAPA



■ Obrázek 1.20 Voxelmaze, wireframe mapy

Herní menu

Stisknutím tlačítka **Esc** se otevře herní menu (viz Obrázek 1.21), kde uživatel najde 3 možnosti:

1. Zpátky do hry (*Continue*) — uzavře menu a vrátí hráče do hry,
2. Nastavení (*Settings*) — ukáže přehled nastavení,
3. Odejít (*Leave*) — ukončí hru a přesměruje hráče na hlavní menu.

Opětovným stisknutím tlačítka **Esc** lze z herního menu odejít a vrátit se ke hře.

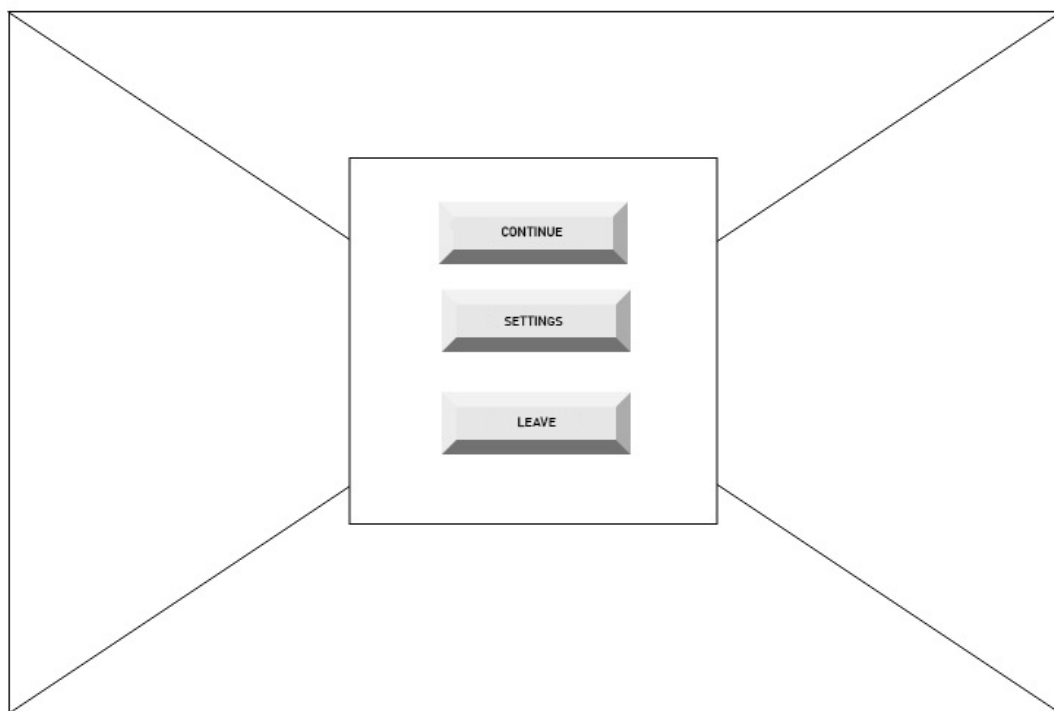
Konec hry

Po dobytí hlavní věže jedním ze hráčů nebo zabití všech protihráčů hra končí a vítězovi se zobrazuje obrázek s notifikací o vítězství a tlačítkem pro návrat do hlavního menu (viz Obrázek 1.22).

1.2.5 Shrnutí

V podkapitolách výše mnou byly představené náčrty stránek uživatelského rozhraní, které budou dále předány našemu grafiku Michalu Rudolfovi pro uskutečnění vizuálního návrhu. Po grafické realizaci se bude pokračovat v práci nad rozhraním v kapitole Realizace návrhů a pro každý wireframe budou implementovány své funkce.

HERNÍ MENU



■ Obrázek 1.21 VoxelMaze, wireframe herního menu

1.3 Animace

„Animovat“ doslova znamená dávat život. Animace se zabývá přidáním pohybu / vytvořením iluze pro těleso, které se nemůže hýbat samo.

Jedním z cílů této bakalářské práce je realizace animace herních postav aplikace VoxelMaze. Na základě toho je potřeba definovat základní pojmy potřebné k návrhu a realizaci.

3D grafika (trojrozměrná grafika) počítačová grafika pracující s trojrozměrnými objekty.

Vrchol (*vertex*) elementární částice objektu / bod, který má své vlastní souřadnice (x, y, z) v trojrozměrném prostoru (viz Obrázek 1.23).

Hrana (*edge*) přímka spojující 2 vrcholy. V grafice se také jmenuje jako rozdělovač polygonů (viz Obrázek 1.23).

Polygon (*face*) obvykle trojúhelník, který se skládá ze 3 hran. Složitější polygon lze vytvořit pomocí připojení dalších vrcholů (viz Obrázek 1.23).

Polygonální síť (*mesh*) tvar trojrozměrného objektu tvořený souborem vrcholů, hran a polygonů [3].

Topologie polygonální sítě způsob rozmístění polygonů definujících 3D objekt (viz Obrázek 1.24) Správná topologie polygonální sítě je důležitá pro korektní deformaci modelu při animaci a dodržování minimálního počtu polygonů.

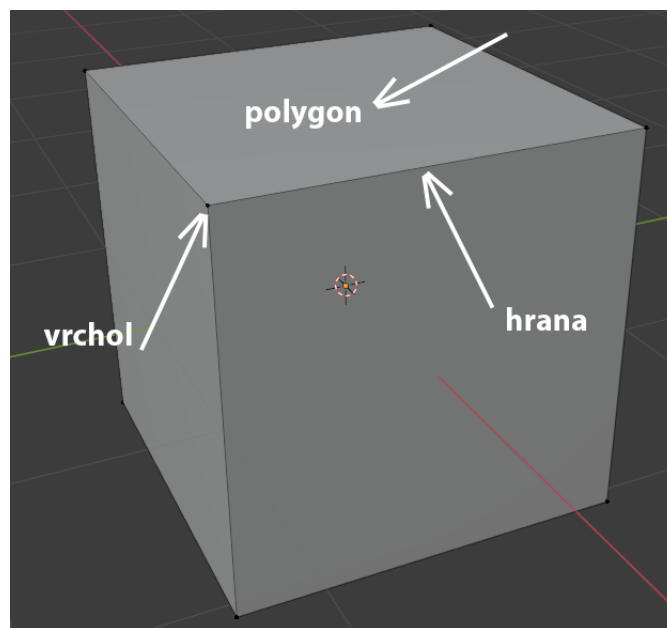
Plug-in rozšíření aplikace.

KONEC HRY

YOU WON!



■ Obrázek 1.22 VoxelMaze, wireframe konce hry



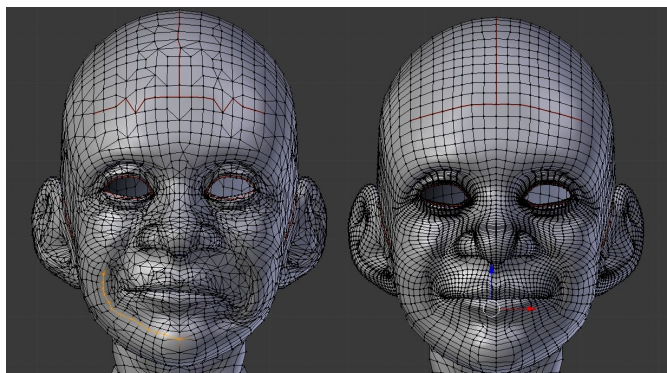
■ Obrázek 1.23 Vrchol, hrana a polygon

1.3.1 Reprezentace 3D modelu

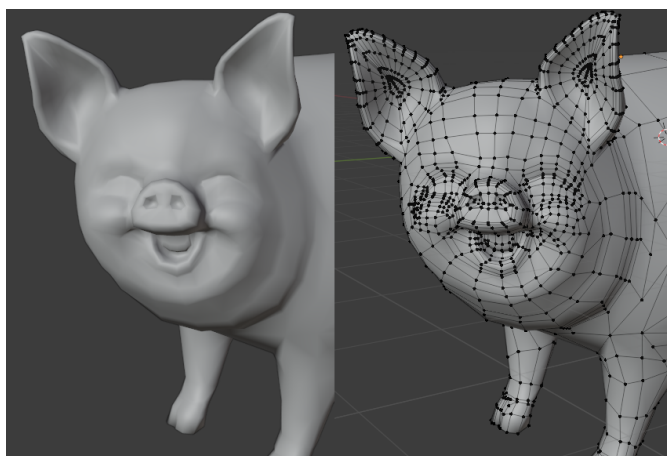
Nejprve je třeba definovat, jak je reprezentován objekt ve 3D. Níže je popsáno 5 nejvíce používaných základních principů 3D modelingu, každý z nich má svou oblast použití:

1. Polygonální modelování

Umožňuje manipulaci se základními stavebními prvky modelu: vrcholy, hrany a polygony. Pomocí polygonálního modelování [10] je možné vytvořit objekt jakékoli složitosti spojením více skupin polygonů, což umožňuje lépe popsat prostorové vlastnosti objektu: topologie, tvar povrchu (viz Obrázek 1.25). Polygonální modelování je nejvíce používaným principem.



■ **Obrázek 1.24** Různé topologie jednoho 3D modelu, zdroj: <http://thilakanathanstudios.com/2016/09/why-do-we-need-topology-in-3d-modeling/>



■ **Obrázek 1.25** Polygonální modelování, model vytvořený pro společnost Victoria VR, a.s.

2. Voxelové modelování

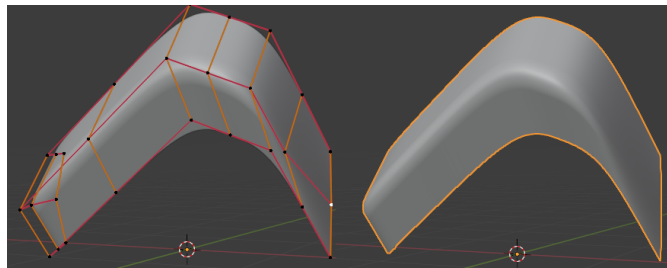
Voxel (z angličtiny: **vo**lumatic, **pix**el) je analogií pixelu v trojrozměrném prostoru. Voxelové objekty jsou reprezentovány prostorovou mřížkou sestavenou z množiny malých krychlí — voxelů. V paměti jsou uloženy jako trojrozměrné pole [9]. Voxely se používají nejen v počítačových hrách a počítačové grafice (*Voxel art*), ale také jako reprezentace trojrozměrných dat v medicíně.

3. Modelování pomocí NURBS (Non-uniform rational basis spline)

Matematická forma představující křivky a různé povrchy [11]. Křivky se definují „řídícími body“ v prostoru, jež tvoří kostru modelu. Na základě kostry vzniká trojrozměrná geometrická plocha (viz Obrázek 1.26). Modely mají plynulé přechody, proto tenhle druh modelování je hodně využíván pro modelování organických objektů: rostliny, lidé, zvířata atd.

4. 3D sculpting

Také je známé jako „digitální sculpting“ (*binary sculpting*) [14]. Typ 3D modelování, při kterém uživatel interaguje s digitálním modelem stejným způsobem, jakým by modeloval hlinu. Deformace polygonální sítě vzniká pomocí speciálních nástrojů - štětců. Při používání sculptingu lze docílit velmi přesné detailizace modelu (viz Obrázek 1.27), proto se hodně využívá v modelování obličejů, anatomických forem, svalů.



■ Obrázek 1.26 Modelování pomocí NURBS



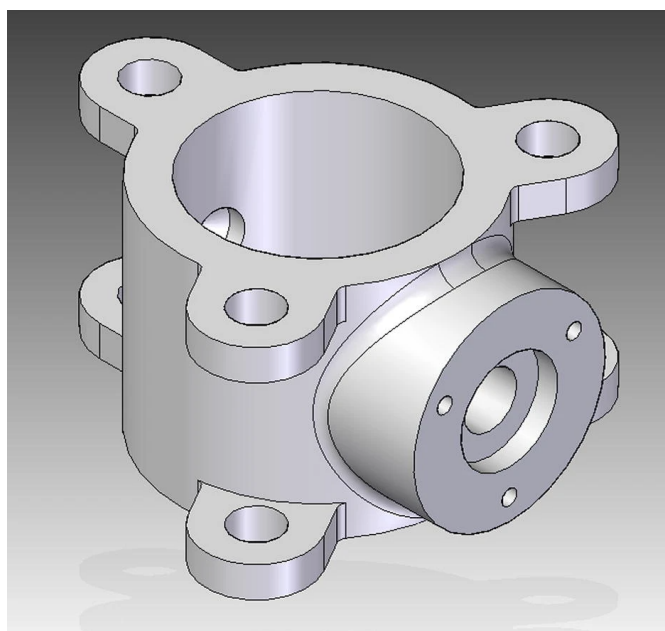
■ Obrázek 1.27 Modelování pomocí 3D sculptingu, hlava přítele

5. Průmyslové modelování (*Industrial modelling*)

Systémy CAD (*Computer Aided Design*) [6] se používají k vytváření přesných kopií skutečných objektů. U tohoto typu modelování jsou brány v úvahu také vlastnosti materiálu modelovaného objektu. Při modelování se jako základní stavební prvky používají ne polygony, ale souvislé tvary (viz Obrázek 1.28). Průmyslové modelování je používáno ve strojírenství, lze ho rozdělit na více podtypů:

- parametrické,
- objemové,
- povrchové.

Pro tvorbu grafiky aplikace VoxelMaze bylo zvoleno polygonální modelování a tato práce bude dále zaměřena na vytváření animace 3D modelů představující sebou soubory polygonů.



■ **Obrázek 1.28** Průmyslové modelování pomocí systému CAD

1.3.2 Přehled existujících programů pro vývoj 3D grafiky a animace

V současné době existuje velká variace nabízených programů pro tvorbu animace 3D modelů. Uživatel má k dispozici celé sady hotových knihoven ulehčujících práci. Zpravidla výběr nástroje záleží na problematice uživatele: tvorba videoher, reklam, konceptů, prezentací atd. Každý software má své slabé a silné stránky: různé složité uživatelské rozhraní, nabídka funkcí, jiná peněžní hodnota a požadavky na hardware uživatele.

Pro účel výběru nejvhodnějšího programu pro animaci budou porovnány několik nejpoužívanějších grafických redaktorů určených pro práci s 3D grafikou.

■ Maya

Jeden z největších zástupců výkonných nástrojů pro vytváření animace, je velmi populární ve filmovém průmyslu v odvětví vizuálních efektů (*VFX*). Maya poskytuje velkou nabídku nástrojů a příležitostí, díky kterým uživatel může dosáhnout opravdu kvalitních výsledků. Editor je používán největšími producenty animace, jedním ze kterých je například Pixar.

Známé animované postavy: Davy Jones — Piráti z Karibiku (*Pirates of the Caribbean*), Hulk, Glum (*Gollum*).

Známé filmy: Myšák Stuart Little (*Stuart Little*), Máša a medvěd.

Vizuální efekty: Matrix, Pán prstenů (*The Lord of the Rings*).

Výrobce: Autodesk.

Podpora OS: Windows, macOS.

Náklady: zdarma pro studenty.

■ 3ds Max

Nástroj 3D Studio Max od stejné společnosti Autodesk je více zaměřen na návrh designu a vytváření inženýrských prototypů, dobře interaguje s AutoCADem. Na rozdíl od editoru Maya 3ds Max poskytuje více nástrojů pro umělce. Má například výkonný nástroj „*Particle Systems*“ umožňující simulaci ohně, kouře, deště atd. Technologie „*Hair and Fur*“ umožňuje vytvářet realistické vlasy, srst, trávu aj.

Znamé filmy: Jurský park (*Jurassic Park*), Alenka v říši divů (*Alice in Wonderland*), Avatar.

Vizuální efekty: 2012, Lara Croft: Tomb Raider, Iron Man.

Výrobce: Autodesk.

Podpora OS: Windows.

Náklady: zdarma pro studenty.

■ Blender

Blender je bezplatný software pro 3D modelování. Vývojářům se podařilo v jednom grafickém editoru zkombinovat velké množství nástrojů pro jeden objekt: modelování, 3D sculpting, práce s texturou, rigging, animace, práce s VFX.

Nevýhodou programu je poměrně složité rozhraní, které pro začátečníka není snadné k pochopení. Program však má velkou podporu v sociálních sítích a na kanálech youtube.com, kde se podrobně popisuje, jak v editoru pracovat.

Blender v současné době není populární nástroj ve filmovém průmyslu. V době založení prvních filmových studií již byly k dispozici výkonné grafické nástroje (3D Studio Max, Maya a Cinema 4D) a Blender neměl dostatečnou technickou podporu, co mohlo při nějaké chybě v programu způsobit studiím škodu ve velké peněžní hodnotě.

Znamé filmy: Big Buck Bunny, Elephants Dream.

Vizuální efekty: Spider-Man 2, Pí a jeho život (*Life of Pi*). **Výrobce:** Blender Foundation.

Podpora OS: Windows, macOS, Linux.

Náklady: zdarma.

■ Cinema 4D

Další velký hráč na aréně výkonných 3D editorů. Program je zaměřen na práci s video efekty a animací. Na výstupu má velmi kvalitní videa a realistickou animaci. Cinema 4D má oproti předchozím softwarům jednodušší rozhraní a také poskytuje velkou nabídku plug-inů.

Znamé filmy: V tom domě straší (*Monster House*), Muži, kteří nenávidí ženy (*The Girl with the Dragon Tattoo*).

Vizuální efekty: Spider-Man: Far From Home – Perception, Lví král (*The Lion King*) — KTM Productions.

Výrobce: Maxon.

Podpora OS: Windows, macOS, Linux.

Náklady: 76,43 Kč za prvních půlroku pro studenty.

■ Houdini

Houdini je populární výkonná softwarová aplikace pro práci s filmovými a vizuálními efekty. Používá se pro simulaci destrukce, kouře, výbuchů, kapalin, aj. Na rozdíl od 3ds Max, Maya a Cinema 4D Houdini je procedurální 3D aplikace, která má zcela jiný přístup k generaci a přenosu dat. Takové procedurální programy vyžadují velký výpočetní výkon.

Znamé filmy: Odvážná Vaiana: Legenda o konci světa (*Moana*), **Vizuální efekty:** Hra o trůny (*Game of thrones*),

Výrobce: Side Effects Software.

Podpora OS: Windows, macOS, Linux.

Náklady: omezená verze zdarma.

■ LightWave 3D

Plnohodnotný 3D grafický editor, který se nejčastěji používá ve filmovém průmyslu. Podporuje pokročilý systém kostního rámování, inverze a animace. Má také nástroj pro práci s částicemi - Hyper Voxels a editor povrchu je užitečný nástroj pro nastavení vlastností materiálu. K programu lze připojit celou řadu modulů, což rozšiřuje jeho funkčnost. Lze tedy získat specializovanější software.

Vizuální efekty: Alice in Wonderland, Avatar, Iron Man, The Walking Dead.

Výrobce: NewTek.

Podpora OS: AmigaOS, Windows, macOS.
Náklady: 4193,38 Kč jednorázový příspěvek.

Pro práci nad projektem VoxelMaze byl zvolen grafický editor Blender, který nabízí zcela bezplatnou distribuci. Blender je také „otevřeným“ softwarem — software s otevřeným zdrojovým kódem — poskytujícím možnost každému uživateli vnést svoje úpravy, přidat vlastní plug-iny. Editor nevyžaduje příliš výkonný hardware. Minimální požadavky:

- 64-bit dual core 2Ghz CPU s SSE2 podporou,
- 4 GB RAM,
- 1280×768 display,
- Grafická karta 1 GB RAM, OpenGL 3.3.

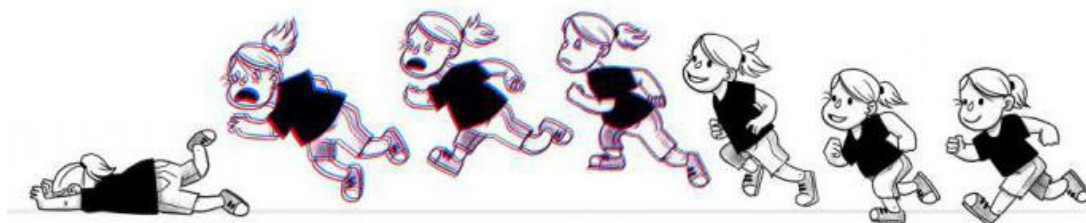
V Blenderu je možné využít hned více funkcí pro práci s jedním objektem, což pro naši aplikaci je optimální: po importu hotového modelu a jeho optimalizaci se uživatel i hned může zabývat animací, což usnadňuje práci při složitější pipeline.

1.3.3 Rešerše možností animace

V dnešní době je obtížné si představit zaměření, pro které by nebyla aplikovatelná počítačová grafika a nebyla užitečná animace. Využívá se jako pomocný nástroj ve vzdělávání, zdravotnictví, architektuře i reklamě. A interaktivní počítačová grafika se stává více dostupným a populárním prostředkem komunikace mezi člověkem a počítačem. Od doby vzniku první jednoduché animace vyvinulo se více druhů animací sloužících různým účelům. Níže jsou popsány nejvíce využívané principy animace.

2D animace

2D animace je nejstarším druhem animace v grafice. Animátor vytváří klíčové snímky na průhledném základu (viz Obrázek 1.29), který je umístěn na barevném pozadí. Po zjištění potřebného počtu mezilehlých snímků, grafici se zabývají jejich realizací a výsledné snímky spojené do jedné série vytváří iluzi pohybu.



■ **Obrázek 1.29** 2D animace,

zdroj: <https://autogear.ru/article/235/833/cto-takoe-multiplikatsiya-tehnologii-sozdaniya-multfilmov/>

2D vektorová animace

2D vektorová animace je analogem 2D animace a využívá stejné principy tvorby klíčových snímků. Vektorová animace je však odlišná v tom, že vytvářené objekty je možné zvětšovat, otáčet nebo deformovat beze ztráty kvality obrázku (viz Obrázek 1.30). Předějit degradaci obrázku je možné

díky definování kreslených objektů křivkami, které jsou popsány matematickými formulemi, např. animace tekoucí vody, vlajících vlasů atd.



■ **Obrázek 1.30** 2D vektorová animace,
zdroj: <https://www.freepik.com/free-vector>

Stop motion

Princip fázové animace (*Stop motion*) animace funguje na základě změny pozice a pózy natáčecího objektu pro každý další snímek (viz Obrázek 1.31). Princip je podobný procesu tvorby 2D animace, ale místo kreslení 2D obrázků animátor používá skutečné (nedigitální) objekty. Tahle technologie je oproti 2D animaci časově náročnější: je potřeba nastavit kvalitní stabilní osvětlení, vytvořit dekorace, připravit lokaci pro natáčení.



■ **Obrázek 1.31** Stop motion animace,
zdroj: <https://www.thisiscolossal.com/2018/10/stop-motion-video-game-vokabulantis/>

Motion capture

Snímání pohybu (*Motion capture*) je způsob animace dovolující předat na obrazovku plynulé pohyby člověka/zvířete v reálném čase. Na určitých místech aktéra jsou umístěny snímače, který zaznamenávají jakékoliv pohyby (viz Obrázek 1.32). Speciální software dále porovnává výsledky záznamu a řídicí body modelu připraveného k animaci a digitalizuje pohyby aktéra.



■ **Obrázek 1.32** Motion capture animace,
zdroj: <https://www.gamasutra.com/blogs/MitchellClifford/20131017/202611/>

3D animace

3D animace je výsledkem umístění a manipulace s objektem v trojrozměrném prostoru. Místo kreslení postavy nebo vyrábění skutečného modelu se objekt definuje v digitální formě. Na základě principu vytváření animace rozlišujeme více metod:

1. Animace klíčových snímků (*Keyframing*)

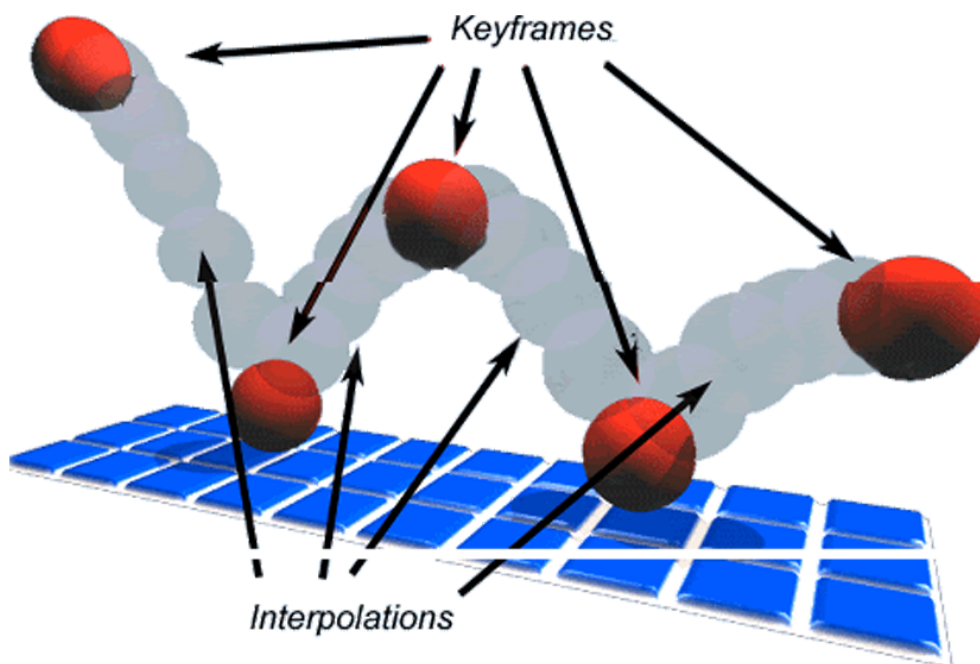
Rozmístění klíčových snímků (*Keyframing*) je jedním z nejpobulárnějších způsobů přidání pohybu 3D objektu. Na časové ose se v každém klíčovém bodě definují poloha, natočení a rozměr animovaného objektu. Na základě těchto klíčových bodů software následně interpoluje (vypočítává) mezilehlé hodnoty (viz Obrázek 1.33) a vytváří animaci dávající pocit plynulého pohybu. Tento způsob animace je obdobný procesu vytváření 2D animace ve filmových studiích: nejdříve se kreslí klíčové snímky, následně práci přebírají umělci dokreslující mezilehlé snímky.

2. Zachycení pohybu (*Motion Capture*)

Jak bylo zmíněno výše (viz sekci), na aktérovi jsou rozmístěny snímače / speciální markery, jež se zachycují desítkami kamer a následně se analyzují speciálním softwarem. Na základě nasbíraných dat se vytváří 3D model pohybů, který se poté zpracovává animátory. Animovaná postava po zpracování animaci má stejné pohyby jako natáčený aktér. Odstraňování markerů v editorech je časově mnohem efektivnější oproti kompletnímu vytváření animace postavy od úplného začátku.

3. Procedurální animace (*Procedural animation*)

Procedurální animace se používá pro vytvoření iluze pohybu v případě, že objekt nelze animovat pomocí klíčových snímků, např. animace částicových systémů (kouř, oheň, voda), tkanin, srsti aj. Také není možné předem definovat chování animované postavy pro interakci s určitým předmětem ve hře, jehož tvar/rozměr není znám. Proto tato metoda provádí výpočty



■ **Obrázek 1.33** Keyframing,
zdroj: <http://www.erimez.com/misc/Softimage/tutorials>

animace objektu v reálném čase a již nejsou potřeba externí soubory uchovávající pozice vrcholů v čase.

Pro účely aplikace VoxelMaze byla zvolená metoda animace realizována pomocí klíčových snímků, jelikož herních předmětů, s nimiž interagují hlavní postavy, není mnoho a všechny možné interakce jsou předem známé.

1.3.4 Rešerše možností animace voxelově zobrazených modelů

Jak bylo zmíněno výše, existuje mnoho způsobů reprezentace trojrozměrných objektů. Polygonální modely jsou dnes nejběžnějším formátem díky použití v počítačové grafice a počítačových hrách. Voxely mají hodně výhod oproti polygonálnímu modelování, např. neomezená detailizace. Jednou z nevýhod ale je obtížná animace voxelových modelů, jež v počítačových hrách hraje důležitou roli: interakce s virtuálním prostředím musí poskytovat uživateli dojem interakce s reálným světem pro lepší zapojení do hry.

Podle možností animace voxelových objektů rozlišujeme více způsobů:

- interpolace mezi větším množstvím 3D modelů — pro jednu animaci se vytváří několik podobných 3D modelů lišících se pouze v umístění některých voxelů, jež vytváří dojem pohyblivých částí,
- elastické animace — 3D objekt se animuje po částech (např. pomocí keyframingu), u kterých se respektuje vlastnost tvrdého povrchu, proto nemůže dojít ke kolizi a následnému překrývání několika částí jednoho modelů,
- neelastické animace — opak animace elastické — není detekce kolize objektů na sebe (např. animace postav v Minecraftu) a dochází k překrývání.

Pro aplikaci VoxelMaze byl zvolen princip elastické animace, která dovoluje přidat pohybům 3D modelu dojem plynulosti.

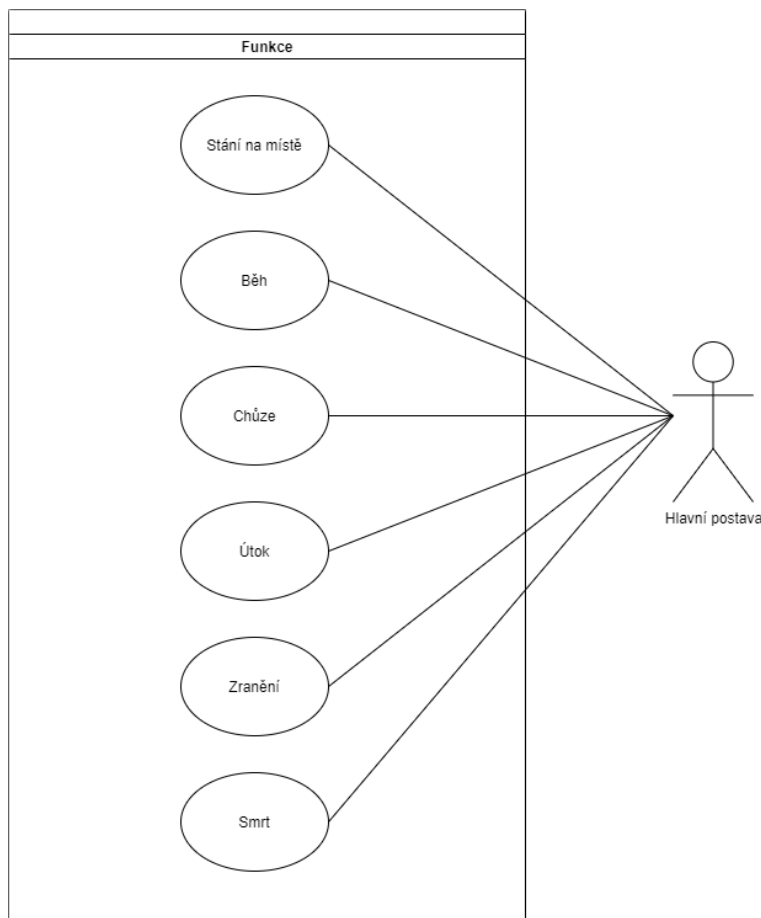
1.3.5 Návrh animace postav

Pro realizaci animace nejdříve je potřeba zanalyzovat chování všech hlavních postav aplikace VoxelMaze a definovat klíčové momenty. Každá postava během hry může mít několik stavů:

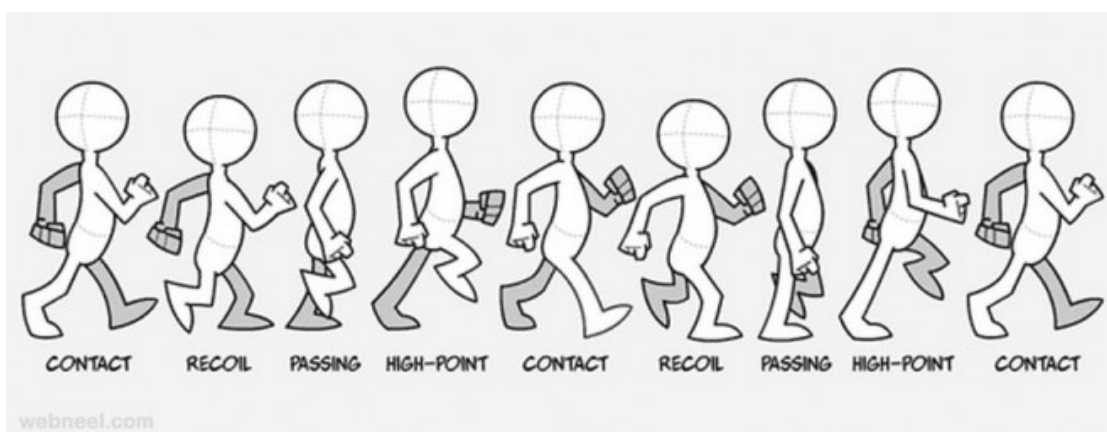
- pro případ, že postava stojí na jednom místě a žádným způsobem neinteraguje s herním světem, je potřeba vytvořit zajímavou animaci **nudící se postavy** pro výzvu hráče pokračovat ve hře.
- Základní způsob pohybu po hernímu poli je **chůze**.
- Použitím kombinace speciálních kláves postava může zrychlit tempo pohybu — je potřeba vytvořit **animaci běhu**.
- Pro přeskocení případných překážek postava také potřebuje animaci **skoku**.
- Když se postava setkává s nepřítelem, musí na něj **zaútočit**, aby měla šanci k přežití.
- Při útoku nepřítele postava dostává **zranění**.
- V případě, kdy se uživateli nepodaří uniknout útoku, postava **umírá**.

Na základě výše zmíněných funkcí bylo vytvořené schéma případů užití (*Use cases*) (viz Obrázek 1.34).

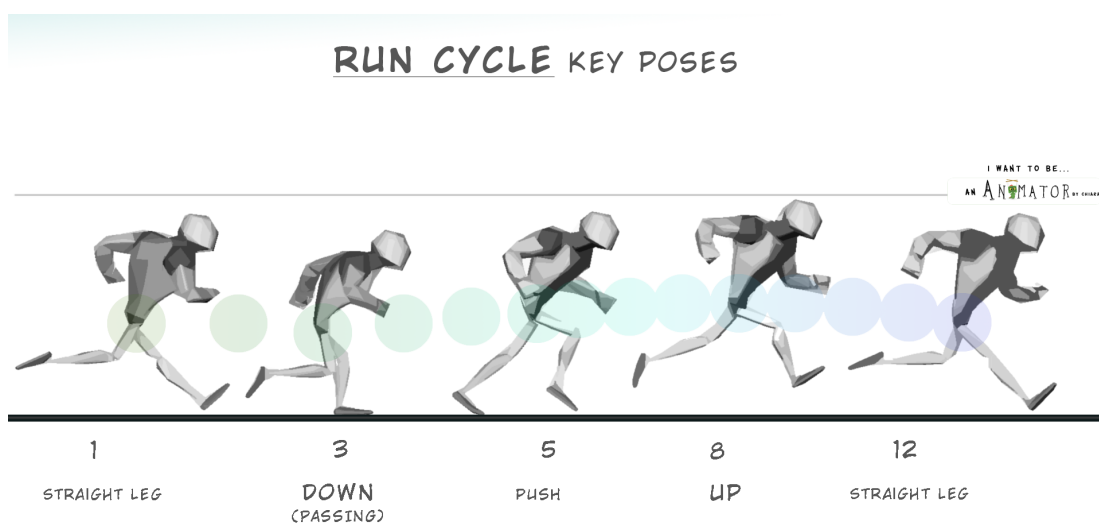
Jelikož pro animaci postav byla zvolena metoda klíčových snímků (*Keyframing*), pro vytvoření plynulých pohybů stačí definovat hlavní pózy, jež program Blender následovně interpoluje a vytvoří plnohodnotnou animaci. Pro takové animace jako jsou chůze a běh jsem se inspirovala již existujícími prototypy animace rozloženými na jednotlivé klíčové pozice (viz Obrázek 1.35 a Obrázek 1.36). Pro návrh animací skoku, útoku, stání na místě, zranění a smrti byla natočena vlastní videa s předpokládanými pohyby, na základě kterých budou následovně realizovány pohyby herních postav.



■ Obrázek 1.34 případy užití animace



■ Obrázek 1.35 prototyp animace chůze
zdroj: <https://webneel.com/walk-cycle-animation>



■ Obrázek 1.36 prototyp animace běhu ¹

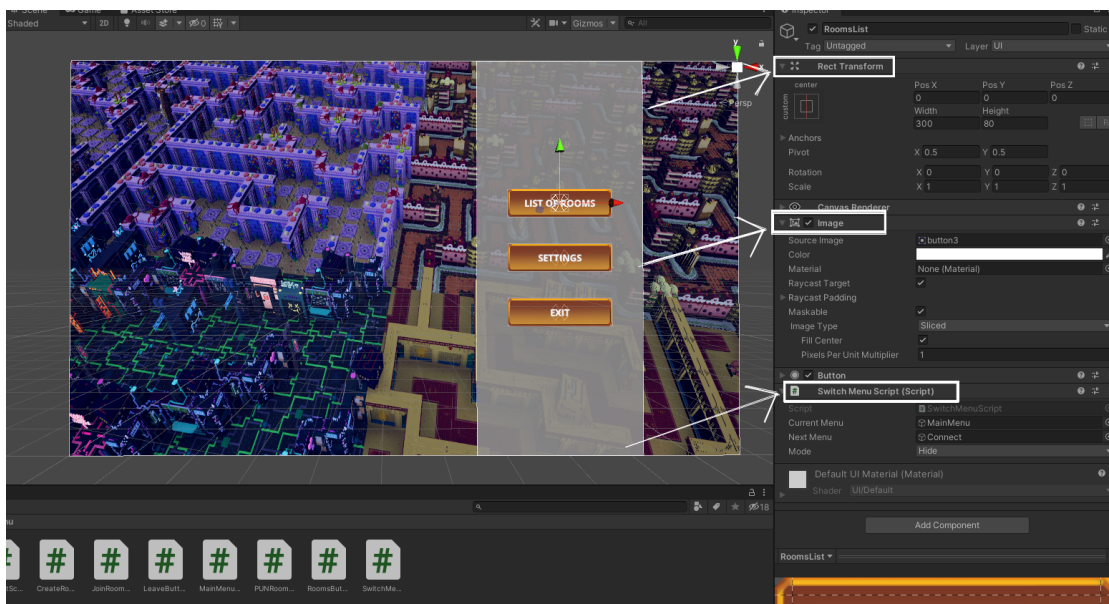
Kapitola 2

Realizace návrhů

Tato kapitola je věnována implementaci uživatelského rozhraní aplikace VoxelMaze pro operační systém Windows a vytvoření animací hlavních herních postav pomocí klíčových snímků. V každé podkapitole budou popsány použité nástroje a technologie, vysvětlen postup práce a popsána samotná implementace. Na konci této kapitoly bude provedeno testování s porovnáváním výsledku předchozí a aktuální práci.

2.1 Uživatelské rozhraní

Pro vývoj v multiplatformním herním engine Unity se používá objektově orientovaný programovací jazyk C#. Implementace uživatelského rozhraní je realizovaná pomocí integrovaných prvků uživatelského rozhraní v Unity (obrazky tlačítka slidery a td.) (a také pomocí naspaných naámi skriptů). Rozhraní je založeno Každá scéna v herním engine Unity je realizována pomocí souboru grafických prvků a objektů třídy Tlačítko (*Button*) nesoucími informace hlavně o události spouštějící odpovídající skript, velikosti, lokaci a přiděleném obrázku (viz Obrázek 2.1).



■ Obrázek 2.1 VoxelMaze, nastavení tlačítka

Ve zmíněných skriptech většina implementovaných tříd dědí atributy tříd *MonoBehaviourPun* a *MonoBehaviourPunCallbacks* poskytovaných plug-inem PUN (*Photon Unity Networking*):

- třída *MonoBehaviourPun* poskytuje atribut *photonView* pro identifikaci objektu v síti, také definuje způsob pro klienta, jak aktualizovat vzdálené instance,
- třída *MonoBehaviourPunCallbacks* také poskytuje *photonView*, ale také i všechny zpětné události (*callbacks*), které může PUN volat.

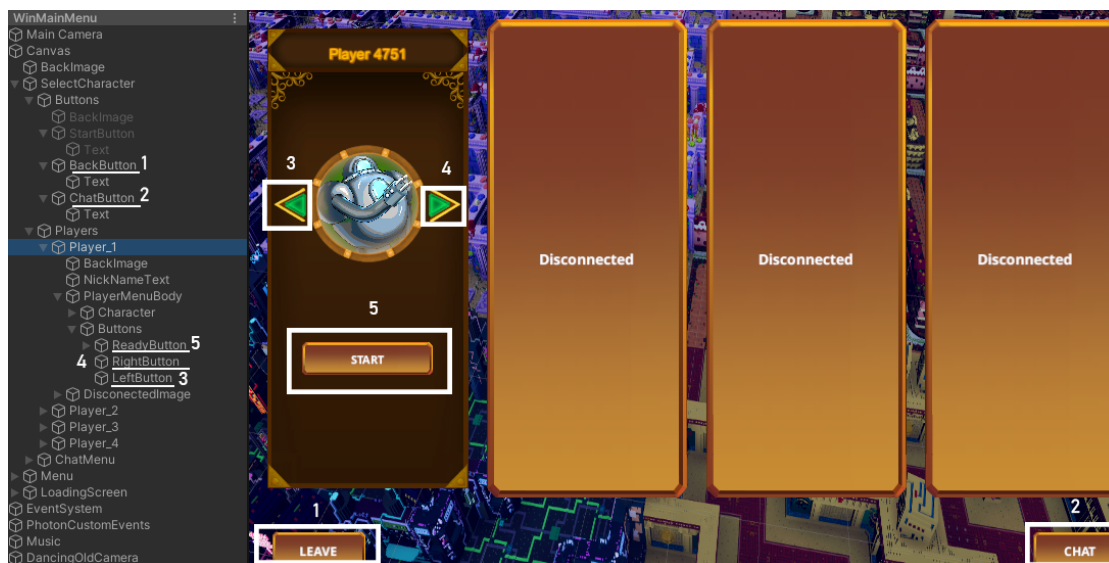
Implementované třídy a jejich význam:

1. *CreateRoomWindowScript* a *JoinRoomByNameScript* zabývají se připojením klientů k místnostem.
2. *EventManager* zpracovává všechny základní události hráčů.
3. *SwitchMenuScript* přepíná mezi menu.
4. *RoomsButtonScript* vytváří tlačítko s informacemi o místnosti a umožňuje připojení.
5. *PUNRoomsCustomManager* obnovuje seznam aktuálních místností a umožňuje připojení.
6. *PlayersInfo* a *PlayersInfoManager* synchronizují informace o hráči v menu výběru postavy.
7. *ChatSampleSceneLogic* a *ChatSelectCharacterLogic* definují logiku chatů ve hře a v menu výběru postavy.
8. *MainMenuLoadingLogic* a *ChooseHeroLoadingLogic* odpovídají za logiku načítání obrazovky hlavního menu.
9. *LeaveButtonScript* ukončuje běh programu.
10. *ChangeCanvasGroupScript* odpovídá za zobrazení grafických prvků menu v herní scéně.
11. *ObservePlayersInfo* spravuje informaci o hráči při výběru herní postavy.
12. *WinMenu* a *WinMenuController* odpovídají za funkce herního menu.
13. *WinMazeMap* odpovídá za načtení mapy.
14. *WinDeadMenu* zobrazuje zprávu o smrti s možností vzkříšení postavy.
15. *TitleFPSSwitch* odpovídá za zobrazení FPS (*frames per second*).
16. *DraggingMap* vykresluje posunutou mapu bludiště.
17. *ReplaceMap* odpovídá za posunutí mapy na obrazovce.
18. *ObservePlayersStatus* popisuje aktuální stav hráče (*score*, smrt).

Způsob implementace funkcí uživatelského rozhraní bude ukázán na příkladě menu výběru herní postavy (další materiály a zdrojové kódy lze nalézt v příloženém médiu). Scéna tohoto menu obsahuje následující funkce:

1. zpátky do hlavního menu (tlačítko *Leave*),
2. otevřít chat (tlačítko *Chat*),
3. zvolit jinou herní postavu (2 boční tlačítka/šipky),
4. zahájit hru (tlačítko *Start/Ready*).

Každá funkce má přiřazený vztah k určitému tlačítku na obrazovce, k němuž je přidělen konkrétní skript s funkcionalitou. Na Obrázku 2.2 je k vidění hierarchie daných tlačítek v Unity ve scéně výběru postavy.



■ Obrázek 2.2 VoxelMaze, reprezentace grafických prvků v Unity

2.1.1 Odejít

Tlačítko „Odejít“ (*Leave*) je propojeno s funkcí *LeaveRoom()* třídy *PlayersInfoManager*, která využívá již implementovanou statickou třídu *PhotonNetwork* se stejnojmennou funkcí od PUN:

■ **Výpis kódu 2.1** Opuštění aktuální místnosti

```
public void LeaveRoom()
{
    PhotonNetwork.LeaveRoom();
}
```

2.1.2 Otevřít chat

Tlačítko „Chat“ je propojeno s funkcí *OpenClose(bool)*, která ukazuje okénko na tzv. plátně (canvas); funkcionalitu chatu realizuje RPC (síťová) funkce *SendMessage(string, string, int)* stejné třídy *ChatSelectCharacterLogic* s parametry udávajícími jméno, číslo hráče a text zprávy:

■ **Výpis kódu 2.2** Chat

```
public void OpenClose(bool active)
{
    canvasGroup.alpha = active ? 1 : 0;
    canvasGroup.interactable = active;
    canvasGroup.blocksRaycasts = active;
    isActive = active;
}

[PunRPC]
private void SendMessage(string player_name, string text, int player_id)
{
    ChatContainer.Message message;
    message = ChatContainer.AddNewMessage(player_name, text, player_id);
    CreateNewTextObject(message);
}
```

2.1.3 Zvolit jinou herní postavu

Přepínat se mezi postavami dovoluje funkce *ChangeCharacter(bool)* třídy *PlayersInfoManager* s parametrem *next* definujícím, zda byla zmáčknuta levá nebo pravá šipka odpovídající za přepínání herních postav:

■ Výpis kódu 2.3 Změna postavy

```
public void ChangeCharacter(bool next)
{
    if (!PhotonNetwork.IsMasterClient)
        self_status.ready = false;
    self_status.chosen_character += next ? 1 : -1;
    self_status.chosen_character += characters_num;
    self_status.chosen_character %= characters_num;
    PlayersInfo.self.UpdatePlayerStatus(current_index, self_status);
}
```

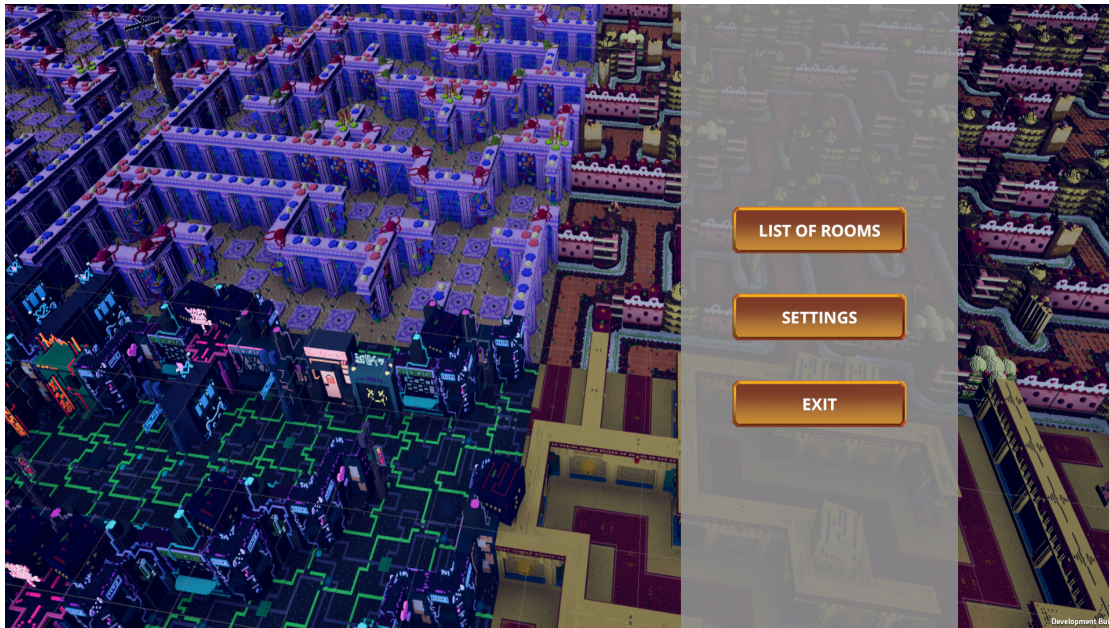
2.1.4 Připojit se ke hře

Za připojení se ke hře odpovídá tlačítko „Ready“ s napojenou funkcí *Ready()* třídy *PlayersInfoManager*. Funkce nejdříve kontroluje, zda uživatel mačkající tlačítko je tzv. master klientem — klient synchronizující ostatní připojené klienty a spouštějící funkce hry. V případě, že hráč není master klientem, funkce pouze mění jeho status na připraven/nepřipraven. V opačném případě se kontrolují stavy ostatních hráčů a když budou všichni připraveni, spouští se událost zahájení hry:

■ Výpis kódu 2.4 Příprava ke hře

```
public void Ready()
{
    if (!PhotonNetwork.IsMasterClient)
    {
        self_status.ready = !self_status.ready;
        PlayersInfo.self.UpdatePlayerStatus(current_index, self_status);
    }
    else
    {
        bool ready = true;
        for (int i = 0; i < PhotonNetwork.PlayerList.Length; i++)
        {
            ready = ready && PlayersInfo.self.GetStatusByIndex(i).ready;
        }

        if (ready && !start_load_scene)
        {
            start_load_scene = true;
            PhotonNetwork.CurrentRoom.IsOpen = false;
            PhotonNetwork.RaiseEvent(7, null,
                new RaiseEventOptions
                {Receivers = ReceiverGroup.All},
                new SendOptions
                {Reliability = true });
        }
    }
}
```



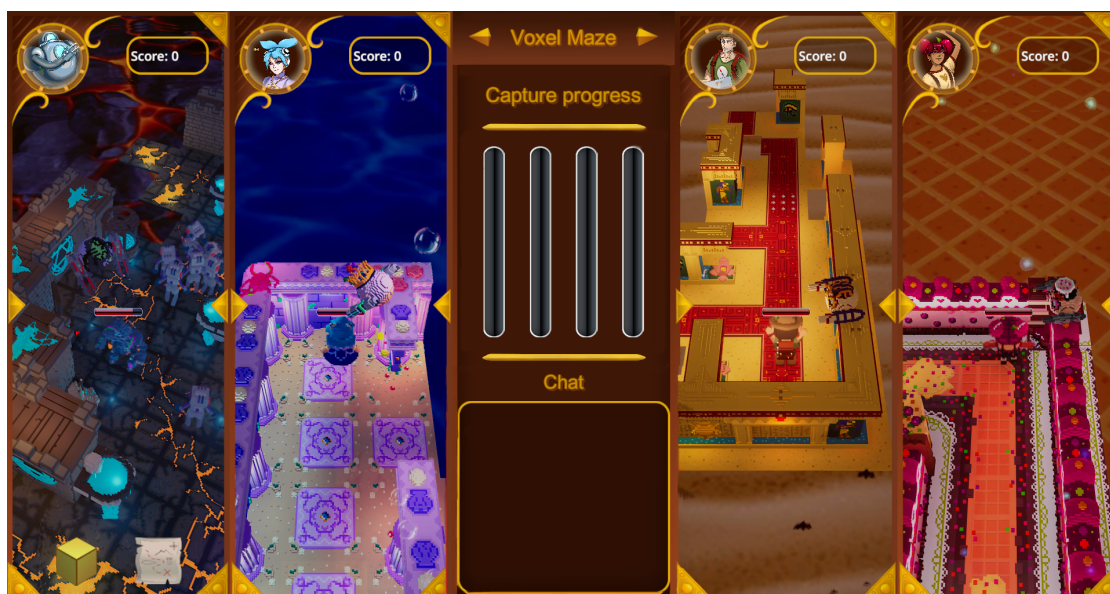
■ Obrázek 2.3 VoxelMaze, realizace UI hlavního menu



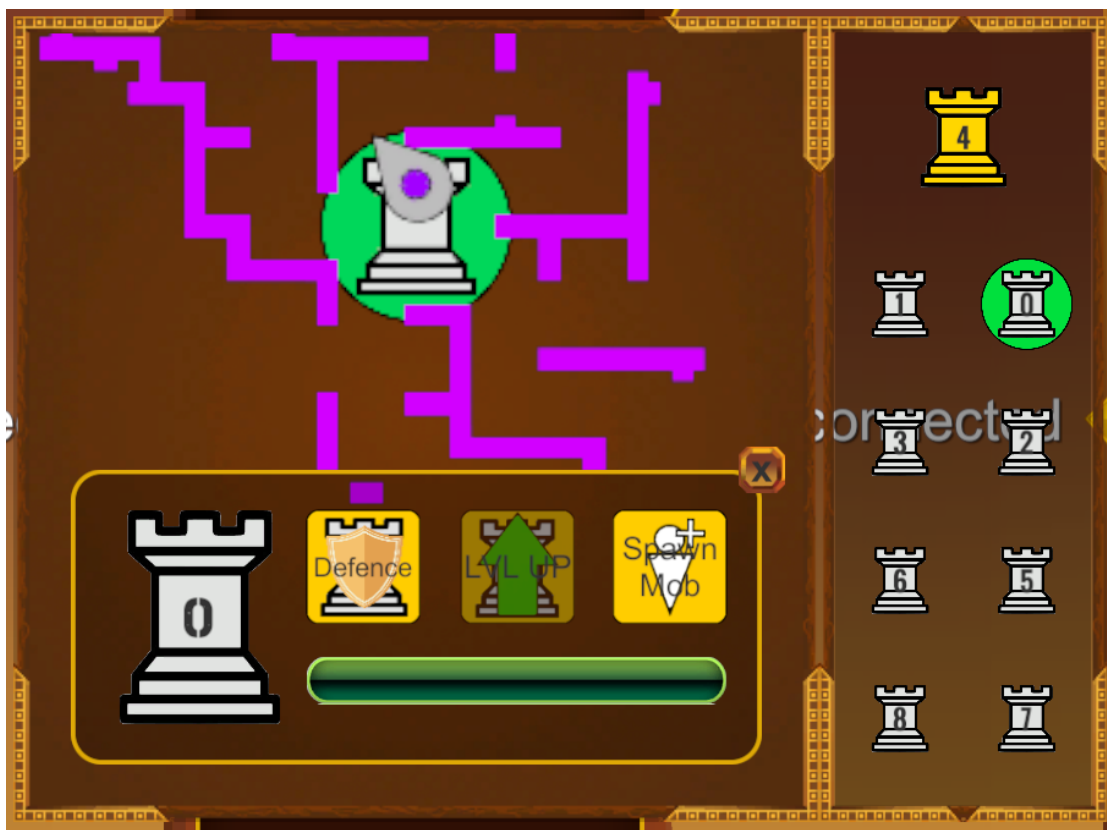
■ Obrázek 2.4 VoxelMaze, realizace UI vytvoření místnosti



■ Obrázek 2.5 VoxelMaze, realizace UI výběru postavy



■ Obrázek 2.6 VoxelMaze, realizace UI hry



■ Obrázek 2.7 Voxelmaze, realizace UI mapy



■ Obrázek 2.8 Voxelmaze, realizace UI výhry



■ Obrázek 2.9 VoxelMaze, realizace UI pro hry

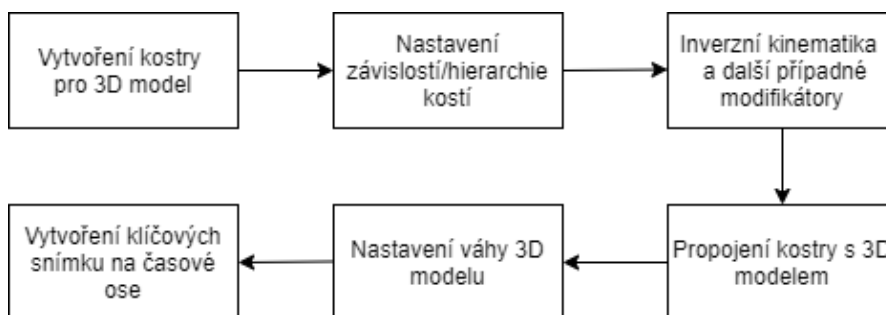
2.2 Animace

Postup realizace animace pro všechny stavy herních postav je stejný. Nejdříve je potřeba vytvořit pro postavu kostru, která umožní snadné ovládání 3D modelu dle potřeb. Jelikož každá postava má odlišný tvar modelu, např. zástupkyně vodního biomu Alice má na hlavě plovoucí rybu, která také musí být animována, není možné vytvořit stejnou kostru pro všechny modely. Některé budou mít více kostí, některé méně. Postup vytvoření kostry a její propojení s modelem vypadá následovně:

1. vytvoření kostry pro 3D model,
2. nastavení závislosti/hierarchie kostí,
3. použití inverzní kinematiky a další případné modifikátory,
4. propojení kostry s 3D modelem,
5. nastavení váhy 3D modelů.

Až model bude mít nastavené váhy pro každou kost, je potřeba pro všechny animace (chůze, běh atd.) vytvořit klíčové pózy (snímky) na časové ose, které Blender následně interpoluje a vytvoří plynulou animaci. Výsledná pipeline práce je zobrazena na Obrázku 2.10.

Realizace animací všech postav bude ukázána na příkladu herní postavy Alice, hotové animace



■ Obrázek 2.10 pipeline vytvoření kostry

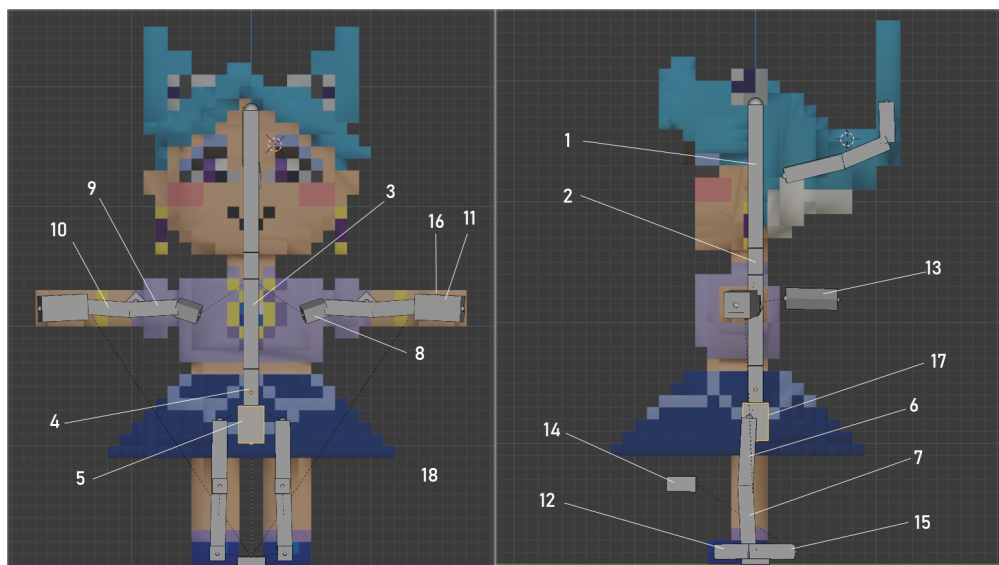
ostatních postav lze najít v příloženém médiu.

2.2.1 Vytvoření kostry

Pro realizaci přirozené animace herní postavy je důležité správně vytvořit kostru, která bude reprezentovat soubor kostí spojených dohromady. Každá kost bude zodpovídat za konkrétní část objektu a bude mít odlišné natočení a pozici v závislosti na časovém parametru během animace. Kostra postavy Alice na Obrázku 2.11 je vytvořena z pánevní kosti (středu kostry) pomocí instrumentu *Extrude* v Blenderu. Pomocí instrumentu lze vytvořit další potřebné kosti ve výsledku dávají celostní kostru postavy. Každá postava má minimální požadovanou sadu kostí:

- kost pro hlavu (číslo 1),
- kost pro krk (číslo 2),
- kost pro trup (číslo 3),
- kost pro dolní část zad (číslo 4),
- kost pro pánev (číslo 5),

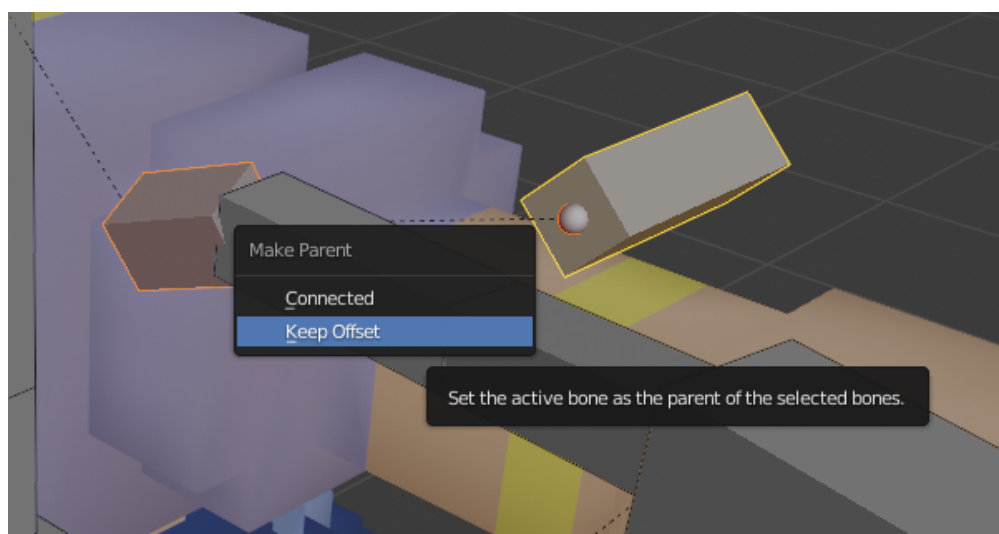
- kost pro horní část nohy (číslo 6),
- kost pro dolní část nohy (číslo 7),
- kost pro pohyb ramena (číslo 8),
- kost pro horní část ruky (číslo 9),
- kost pro dolní část ruky (číslo 10),
- kost pro dlaň (číslo 11),
- kost pro chodidlo (číslo 12),
- kost odpovídající za rotaci lokte (číslo 13),
- kost odpovídající za rotaci kolena (číslo 14),
- kost odpovídající za pohyb chodidla (číslo 15),
- kost odpovídající za pohyb dlaně (číslo 16),
- kost odpovídající za pohyb pánve (číslo 17),
- kontrolní kost řídící polohu ostatních kostí (číslo 18).



■ Obrázek 2.11 vytvoření kostry

2.2.2 Nastavení závislosti kostí

Každá kost musí mít svého „rodiče“ pro zděnění transformací. Kost číslo 8 (rameno) je rodičem kosti číslo 13 (loket), tak změna pozice a natočení kosti číslo 13 bude ovlivňovat natočení kosti číslo 8. Nastavit takovou závislost je možné výběrem kosti „dítěte“, následným výběrem kosti „rodiče“ a zvolením funkce „Keep offset“ (viz Obrázek 2.12). Stejným způsobem jsou nastavené závislosti kostí číslo 8 a 3, 14 a 15, 6 a 17, 17 a 18, 16 a 18, 15 a 18.



■ Obrázek 2.12 nastavení rodičovství kosti

2.2.3 Inverzní kinematika

Pro vytvoření dojmu plynulého pohybu některé kosti, např. dolní části ruky a nohy, budou mít aplikovaný modifikátor Inverzní kinematika (*Inverse kinematics*), dále jenom IK. IK na základě údajů poslední kosti v řetězci umožňuje automaticky vypočítat a nastavit pozici a natočení mezilehlých kostí. Při standardním postupu bez použití daného modifikátoru se poloha každé kosti nastavuje zvlášť po jedné, protože po provedení změn nad rodičovskou kostí „děti“ dědí transformace a také mění svoji pozici a natočení. Provést menší úpravy po rozmístění všech kostí je dost problematické.

Pro nastavení kostem modifikátoru IK stačí definovat první a poslední kosti v řetězci animované části kostry. V případě ruky to budou kosti číslo 8 a 16, v případě nohy to jsou kosti číslo 6 a 15.

2.2.4 Nastavení váhy

Nastavení váhy (*Weight paint*) je nástroj, který umožňuje uživateli vytvořit tepelnou mapu pro vrcholy objektu. Pomocí tepelné mapy lze určit, jak velký vliv bude mít pohyb konkrétní kosti 3D modelu na sadu jeho vrcholů, každý vrchol dostane svou hodnotu váhy. Kreslení vah se dělá pro každou kost zvlášť pomocí štětce v režimu „Weight Paint Mode“. Rozsah barev na modelu je možný od tmavě modré do tmavě červené, kde tmavě modrá znamená žádný vliv kosti a tmavě červená znamená maximální vliv kosti na obarvenou část modelu.

Na začátku model je obarven defaultně do tmavě modré barvy, pomocí štětce Kreslit (*Draw*) lze upravit barvu potřebné části. Na Obrázku 2.13 je vidět již hotovou tepelnou mapu pro hlavu (kost číslo 10). Takhle je potřeba zpracovat každou kost 3D modelu.

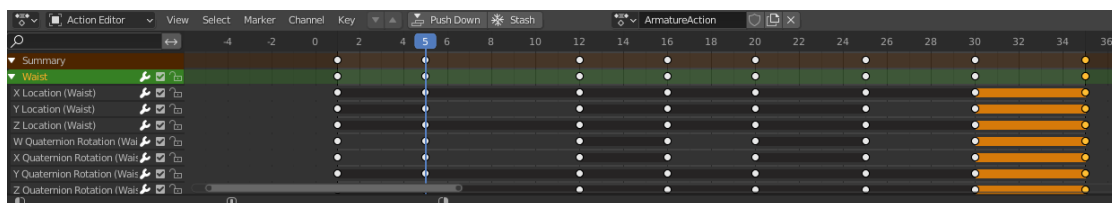
2.2.5 Klíčové snímky

Počáteční a koncové pózy každé animace modelu musí být vždy stejné, aby byl možný plynulý přechod mezi různými stavy postavy. Proces definice klíčových snímků je jednoduchý: v režimu *Pose Mode* na časové ose je potřeba vybrat konkrétní časový parametr (viz Obrázek 2.14), pro který budou provedeny změny, nastavit pozici a natočení jednotlivých kostí. Jediné potřebné transformace kostí jsou rotace a změna pozice, proto po zmáčknutí klíčové klávesy *I* (nastavit klíčový snímek) v otevřeném interaktivním okénku s více možnostmi je potřeba vybrat *Location* *ℰ*



■ Obrázek 2.13 nastavení vah

Rotation (viz Obrázek 2.15). Po nastavení změn pro všechny klíčové časové parametry je možné animaci spustit pomocí tlačítka mezera (*Space*) a opravit případná problémová místa. Hotová animace je přístupná v příloženém médiu.



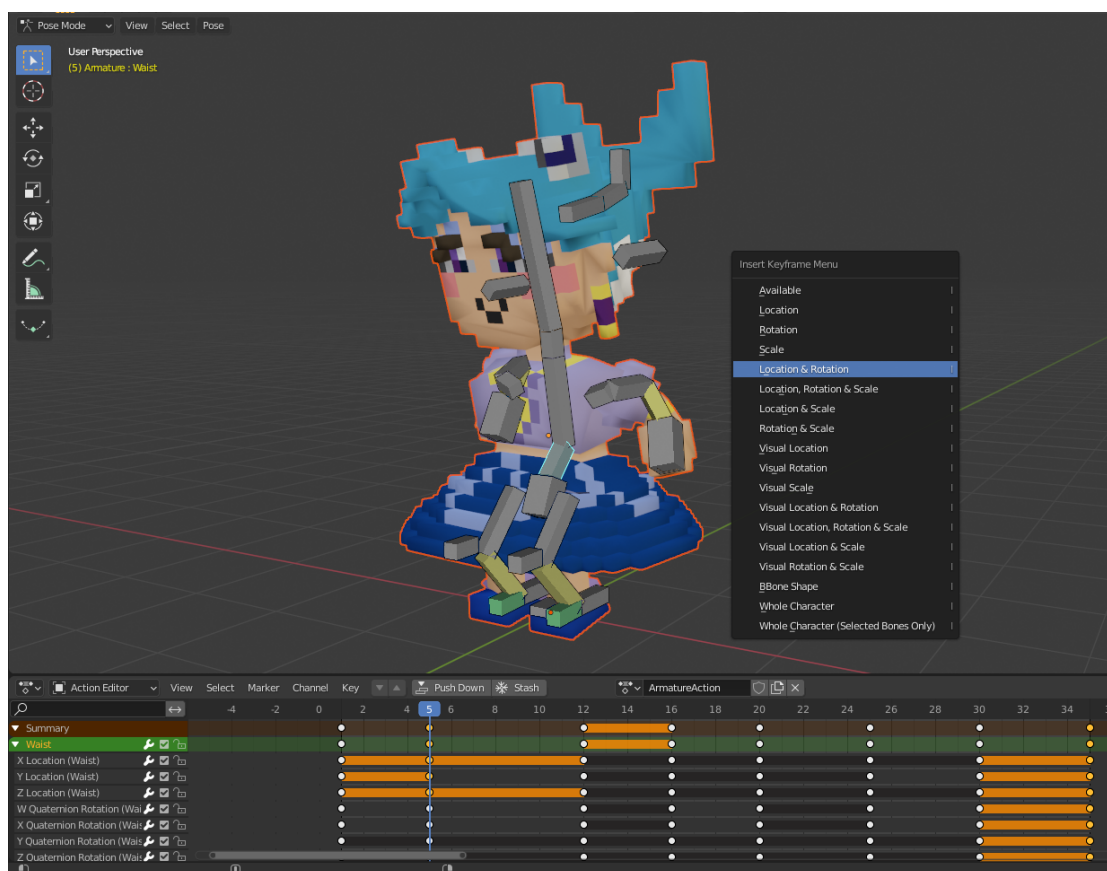
■ Obrázek 2.14 časová osa animace

2.3 Testování

Smyslem testování aktuální verze je načerpat objektivní zpětnou vazbu od zkušených hráčů her stejného typu a to na základě konkrétně mířených dotazů. Za cíl je položeno zjištění možných vylepšení pro další verze hry a zvýšení tím její hráčskou oblíbenost.

Zjišťování bude provedeno formou webového nástroje Google, který umožňuje jednoduše vytvářet dotazníky s různými typy otázek. Testeři budou vybráni nejen dle zmíněné zkušenosti s hraním obdobných typů her, ale budou na ně položeny další 2 přísná kritéria. Jedná se o:

1. věk 15–25 let, co odpovídá věku cílové skupiny,
2. nepodílení se na tvorbě a přípravě této hry — potřebuji objektivní a nezaujaté názory.



■ Obrázek 2.15 klíčové snímky animace skoku

2.3.1 Dotazník

Dotazník obsahuje 12 otázek na jednotlivé okruhy hráčského rozhraní, kde se zjišťuje uživatelská přívětivost z pohledu vizuální stránky, pochopení jednotlivých ovládacích prvků, odezvy hry, komunikace hry s hráčem, aj. Odpovědi jsou koncipovány následovně: v případě, že uživatel zvolí kladnou odpověď „ano“, není zjišťována žádná další interakce; v případě negativní odpovědi je požadována odpověď v otevřené formě, kde je možné zjistit možná vylepšení.

Otázky z formuláře:

1. Bylo jednoduché zahájit, spustit hru?

- Ano,
- Ne, protože...

Zjišťuji, zdali bylo pro hráče jednoduché hru spustit, nebo mu uživatelské rozhraní kladlo překážky a tím mohla hra ztrácet uživatelskou přívětivost.

2. Bylo zřejmé co který ovládací prvek vykonává za akci?

- Ano,
- Ne, nepochopil jsem, jakou funkci splňuje...

Zjišťuji, zdali vytvořené ovládací prvky jsou popsány a graficky uzpůsobeny k tomu, aby uživatel bez delšího uvažování pochopil, co prvek vyvolá za akci při jeho stisknutí. Např.: tlačítko „Leave“ vyvolá ukončení hry.

3. Bylo rozmístění ovládacích prvků přehledné?

- Ano,
- Ne, nelíbilo se mi rozmístění...

Zjišťuji, zdali rozmístění jednotlivých ovládacích prvků bylo přehledné, nebo uživatelé některý z prvků očekávali v jiné části obrazovky.

4. Je retro stylizace menu vhodná?

- Ano,
- Je potřeba ji změnit na novější/jinou

Zjišťuji, zdali uživatelé vyhovuje z vizuální stránky vzhled prvků v menu, nebo realizované rozhraní neodpovídá stylovému očekávání.

5. Jste spokojený/á s barvou pozadí, barvou ovládacích prvků?

- Ano,
- Ne, nevyhovovalo mně...

Zjišťuji, zdali existují prvky rozhraní, pohled na které uživatel shledal vyrušujícími, neharmonickými.

6. Byl jasné, co se od postavy ve hře očekává?

- Ano,
- Ne, chyběla mně nápověda o ...

Zjišťuji, jak je kvalitní komunikace hry s hráčem, zdali nechybí vizuální pomůcky, pomáhající se zorientovat začátečníkům: zprávy o interakci s herním prostředím: zabití nepřátel, množství získaných bonusů, cíl postavy v aktuální okamžik aj.

7. Jste spokojený/á s defaultním nastavením, nebo je potřeba předvyplnit některé parametry jinak?

- Ano, jsem spokojený/á,
- Ne, chtěl(a) bych změnit parametry...

Zjišťuji, zdali uživatel má potřebu změnit defaultní nastavení základních parametrů pro založení nové hry.

8. Byla dostatečně zvýrazněná interakce s uživatelským rozhraním? (Rozsvícení tlačítek)

- Ano,
- Ne, chtěl(a) bych, aby...

Zjišťuji, zdali uživatelé pro lepší interakci s aplikací chybí reakce rozhraní na události.

9. Používal(a) jste integrovanou mapu pro orientaci v bludišti? Věděl(a) jste o její existenci?

- Ano, věděl(a) jsem o mapě a používal(a) jsem ji,
- Ne, mapu jsem nevyužíval(a), protože ...

Zjišťuji, zdali uživatelé stačí znalosti z předchozích zkušeností s podobnými hrami, nebo je potřeba explicitně uvést informaci s popisem speciálních kláves.

10. Byl(a)-li jste připojen(a) k online hře, používal(a) jste chat pro komunikaci s ostatními hráči?

- Ano, používal(a) jsem chat,
- Ne, chat jsem nevyužil(a), protože ...

Zjišťuji, zdali uživatel má potřebu v komunikaci se spoluhráči, případně zda bude používat lokální chat, nebo jakoukoli jinou aplikaci.

11. Připojilo-li se ke hře více hráčů, měl(a) jste potřebu pozorovat jiné hráče? Jsou další okénka s ostatními účastníky hry potřeba, nebo chcete pozorovat jenom svoji postavu?
- Tuto funkci nepotřebuji,
 - Ano, chci vidět další hráče...

Zjišťuji, zdali uživatel má potřebu pozorovat i jiné hráče a zda mu pomůže ve výběru směru informace o lokaci jiné postavy.

12. (pokračování předchozí otázky) Přál(a) byste si přidat možnost posunutí hranic rámečku, nebo kompletně odstranit okénka s ostatními uživateli?
- Mně vše vyhovuje,
 - Přál(a) bych si ...

Zjišťuji nejvíc vyhovující pro uživatele variantu vizuálního obsahu hrací plochy.

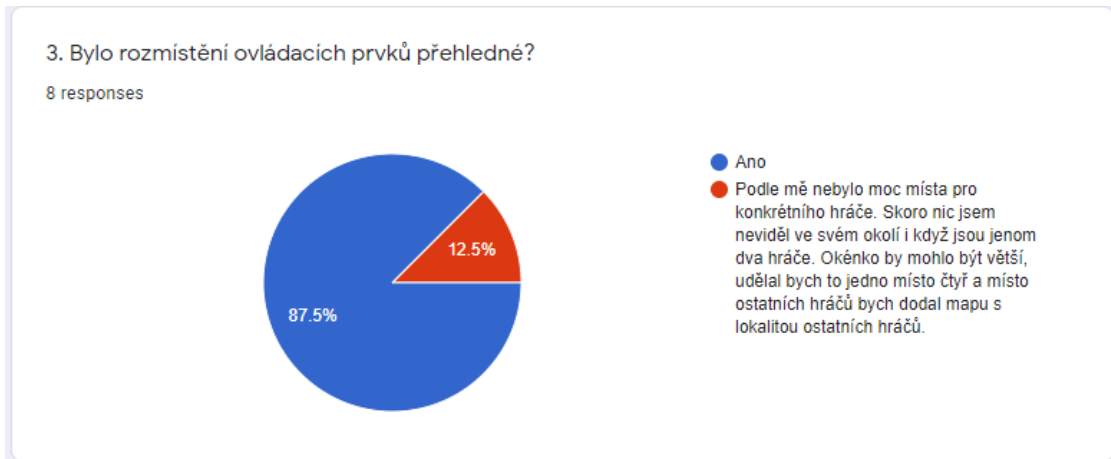
13. Je ještě něco co by jste chtěl(a) doplnit? Zjišťuji další případná doporučení ohledně změny uživatelského rozhraní.

Dotazník je anonymní, aby testující necítil žádná omezení a odpovídal zcela upřímně. Forma s dotazy byla poslána deseti přihlášeným se zájemcům z fakulty FIT.

2.3.2 Výsledky

Anonymního testování se nakonec zúčastnilo 8 respondentů, níže jsou představené výsledky:

- otázky 1., 2., 4., 7. a 8. mají jednoznačný výsledek — 100 % hlasovalo pro odpověď „Ano“.
- Na 3. otázku většina testerů odpověděla „Ano“, jeden člověk však dodal komentář o zbytečnosti všech 4 rámců s hracím polem (viz Obrázek 2.16) a nabídl jako alternativu přidat na obrazovku mapu s aktuální lokací ostatních hráčů.
- Dle odpovědí na otázku číslo 6 princip hry nebyl jasný 50 % testerům, jeden člověk před nalezením mapy nevěděl o existenci věží (viz Obrázek 2.17).
- Výsledek odpovědí na 9. otázku nás informuje o problému s mapou (viz Obrázek 2.18): 1 člověk o ní na začátku nevěděl, dalšímu člověku se v ní nepodařilo zorientovat.
- Odpovědi 10. otázky jsou v pořádku, 1 z hráčů nepoužíval chat, jelikož hrál sám (viz Obrázek 2.19).
- 3 z 8 lidí v odpovědích k 11. otázce vyjádřili názor, že definované místo (rámečky) pro vlastní postavu je příliš malé (viz Obrázek 2.20), 1 účastník odpověděl, že by vidět ostatní postavy nechtěl.
- 50 % participantů ve 12. otázce souhlasilo, že by bylo dobré přidat možnost posunutí nebo odstranění rámečků oddělujících hráče (viz Obrázek 2.21).
- V otázce číslo 13 padly 2 zajímavé nápady, jeden se týče herní mechaniky a bude předán programátorům, druhý nápad je v realizaci úvodního popisu cílů na začátku hry (viz Obrázek 2.22).



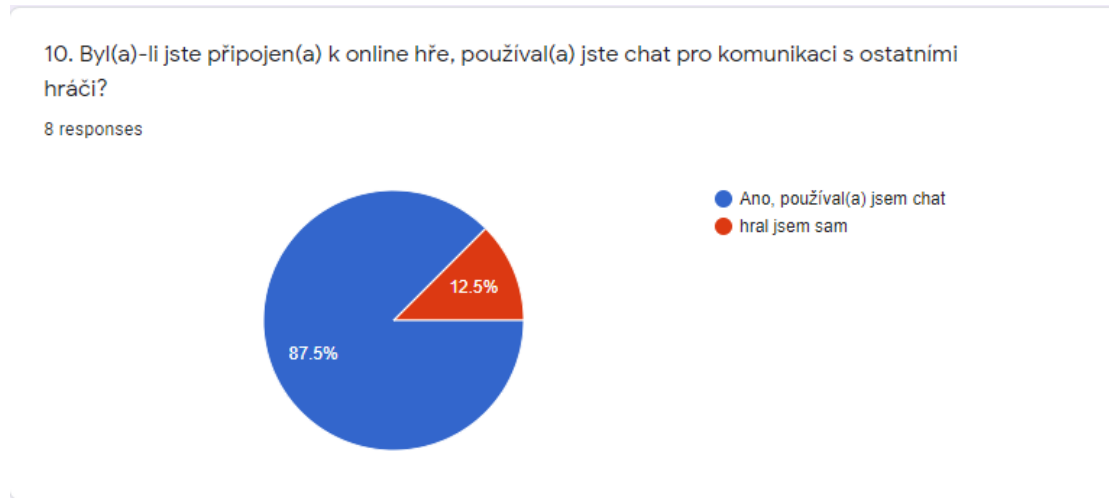
■ Obrázek 2.16 Výsledek testování, 3. otázka



■ Obrázek 2.17 Výsledek testování, 6. otázka



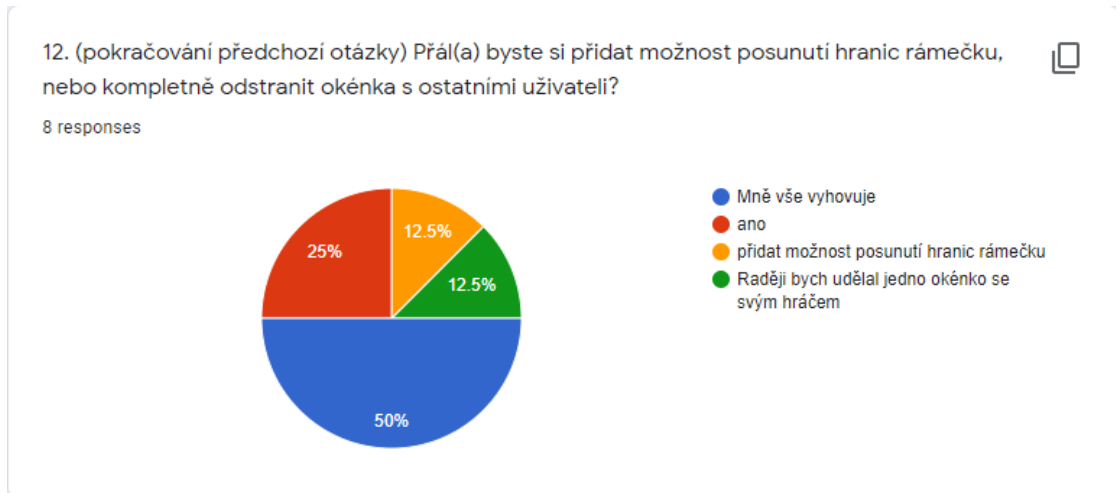
■ Obrázek 2.18 Výsledek testování, 9. otázka



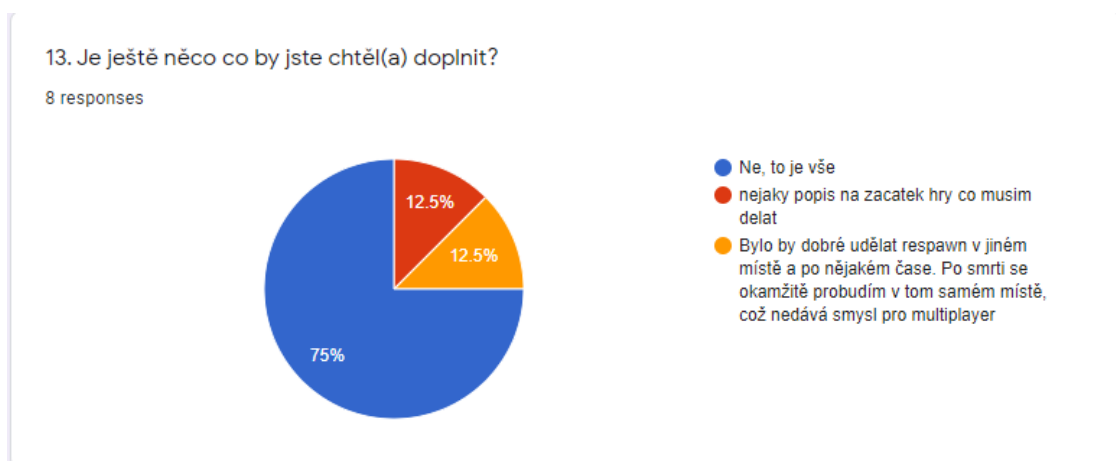
■ Obrázek 2.19 Výsledek testování, 10. otázka



■ Obrázek 2.20 Výsledek testování, 11. otázka



■ Obrázek 2.21 Výsledek testování, 12. otázka



■ Obrázek 2.22 Výsledek testování, 03. otázka

Výsledkem mé práce jsou splněné cíle stanovené výše: implementované přehledné uživatelské rozhraní pro počítačovou aplikaci VoxelMaze odpovídající požadavkům uživatele a realizované animace všech hlavních herních postav. Dle provedeného testování provedená práce zlepšila celkový dojem uživatelů z aplikace. Také je důležitý otevírající se prostor pro nové nápady vylepšení vyvíjené aplikace pro další případné práce. Pomocí této práce aplikace VoxelMaze se posunula dopředu na schodech vývoje a jsem si jistá, že v brzké době ji čekají i další velké změny. V aplikaci samozřejmě vždy bude místo pro vylepšení implementace

Cílem této bakalářské práce byl návrh a implementaci uživatelského rozhraní

3.1 Doporučení pro další práce

Pro případné další práce nad podobným tématem aplikace VoxelMaze bych doporučila opravit realizaci některých prvků uživatelského rozhraní. V herní scéně je třeba udělat rámečky oddělující oblasti hry všech hráčů buď interaktivní: pohyb doleva a doprava / možnost odstranění, nebo je komplexně odstranit již na začátku a na uvolněné místo dát mapu, se kterou není potřeba interagovat (otevírat/zavírat) během hry.

V případě, že budou implementovány další herní mechaniky a celkový počet stavů herních postav a jejich náročnost (např. interakce s předměty) poroste, doporučila bych zaměřit se na implementaci animace s inverzní kinematikou, která by se vypočítávala v reálném čase a vypadala mnohem lépe.

Na základě provedeného mnou testování bylo také zjištěno, že uživatelům chybí úvodní představení hry s bližším popisem herního mechanismu a cílů. Ve výsledku to může být trailer uvádějící hráče do hlavní myšlenky hry a vysvětlující postup pro výhru, nebo jenom úvodní nepovinný (který lze přeskocit) interaktivní režim, který by zvýrazňoval konkrétní grafické prvky rozhraní a vysvětloval jejich význam.

Literatura

- [1] Mojang 4J Studios, Markus Persson. Minecraft [software], 18. 11. 2011. [přístup 8. 12. 2019]. [Požadavky na systém: procesor Intel Pentium 4 2 GHz, operační systém Windows XP/Windows 7/Windows Vista, videokarta NVIDIA GeForce 6600, volné místo na disku 400 MB, operační paměť 512 MB]. URL: www.minecraft.net.
- [2] N. Bevan. Iso standards for the user interface. In *IEE Colloquium on User Interfaces and Standardisation*, pages 1/1–1/3, 1989.
- [3] Gerald Farin. 21 - surfaces with arbitrary topology. In Gerald Farin, editor, *Curves and Surfaces for CAGD (Fifth Edition)*, The Morgan Kaufmann Series in Computer Graphics, pages 377–398. Morgan Kaufmann, San Francisco, fifth edition edition, 2002. URL: <https://www.sciencedirect.com/science/article/pii/B9781558607378500211>, doi:<https://doi.org/10.1016/B978-155860737-8/50021-1>.
- [4] James D Foley, Foley Dan Van, Andries Van Dam, Steven K Feiner, John F Hughes, Edward Angel, and J Hughes. *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional, 1996.
- [5] G. P. Johnson; G. D. Abram; B. Westing; P. Navr'til; K. Gaither. Sage2, 2012. URL: <https://sage2.sagecommons.org/>.
- [6] Xabier Garikano, Mikel Garmendia, Angel P Manso, and Eneko Solaberrieta. Strategic knowledge-based approach for cad modelling learning. *International journal of technology and design education*, 29(4):947–959, 2019.
- [7] id Software. Quake iii arena [software], 2. 12. 1999. [Požadavky na systém: procesor Pentium 266 MHz, operační systém Microsoft Windows, GNU/Linux, Mac OS, Sega Dreamcast, PlayStation 2, Amiga, Xbox 360, Symbian, Android, OpenGL, volné místo na disku 420 MB, operační paměť 64 MB]. URL: <https://ioquake3.org/get-it>.
- [8] Innersloth. Among us [software], 2018. [přístup 16. 11. 2018]. [Požadavky na systém: procesor SSE2 instruction set support, operační systém Windows 7 SP1+, DirectX v. 10, volné místo na disku 1 GB, operační paměť 1 GB]. URL: <https://www.bluestacks.com/apps/action/among-us-on-pc.html>.
- [9] Carlos Eduardo Vaisman Muniz, Anselmo Montenegro, Marcos Lage, and Cristina Nader Vasconcelos. Polygonal mesh extraction from digital voxel art. In *2013 XXVI Conference on Graphics, Patterns and Images*, pages 171–178, 2013. doi:10.1109/SIBGRAPI.2013.32.
- [10] Mario Russo. *Polygonal modeling: basic and advanced techniques*. Jones & Bartlett Learning, 2006.

- [11] W. Shengli and Z. Chongming. Nurbs surface generation by control points. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 544–547, 2011. doi:10.1109/ICCSN.2011.6014784.
- [12] Unity Technologies. Unity [software], 8. 6. 2005. URL: <https://unity.com/products/unity-platform>.
- [13] Valve. Counter-strike [software], 24. 7. 2019. [přístup 8. 12. 2019]. [Požadavky na systém: procesor 500 MHz, operační systém Windows® 7/Vista/XP, videokarta Intel® Core™ 2 Duo E6600 nebo AMD Phenom™ X3 8750, volné místo na disku 8 MB, operační paměť 2 GB]. URL: <https://counter-strike.en.uptodown.com/windows>.
- [14] A. Willis, J. Speicher, and D. B. Cooper. Surface sculpting with stochastic deformable 3d surfaces. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 249–252 Vol.2, 2004. doi:10.1109/ICPR.2004.1334147.

Obsah přiloženého média

	readme.txt	stručný popis obsahu média
	exe	adresář se spustitelnou formou implementace
	src		
		anim implementace animace
		ui zdrojové kódy implementace
	text	text práce
		thesis.pdf text práce ve formátu PDF