



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**DEMONSTRACE HERNÍCH MECHANIK PRO KOMBINACI 2D A 3D DISPLEJŮ**

GAME MECHANICS DEMONSTRATION FOR COMBINATION OF 3D AND 2D DISPLAYS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**JAN HANUŠ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. TOMÁŠ CHLUBNA**

BRNO 2024

## Zadání bakalářské práce



153110

Ústav: Ústav počítačové grafiky a multimédií (UPGM)  
Student: **Hanuš Jan**  
Program: Informační technologie  
Název: **Demonstrace herních mechanik pro kombinaci 3D a 2D displejů**  
Kategorie: Počítačová grafika  
Akademický rok: 2023/24

### Zadání:

1. Seznamte se s vhodným API pro vývoj 3D aplikace (OpenGL, Unity, Unreal).
2. Prozkoumejte existující herní mechaniky vhodné pro 3D displeje.
3. Navrhněte vhodnou scénu v omezeném prostoru využívající efektivně změn pohledů holografického displeje.
4. Navrhněte vhodnou kooperaci standardního 2D a 3D displeje pro výsledné demo.
5. Implementujte aplikaci.
6. Zhodnotte zobrazení aplikace na displeji a proměřte efektivitu.
7. Vytvořte video reprezentující výsledky vaší práce.

### Literatura:

- [Looking Glass Factory Documentation](#)
- Gregory, Jason. *Game engine architecture*. crc Press, 2018. ISBN 1351974289, 9781351974288
- Bishop, Lars, et al. "Designing a PC game engine." *IEEE Computer Graphics and Applications* 18.1 (1998): 46-53.
- Adams, Ernest, and Joris Dormans. *Game mechanics: advanced game design*. New Riders, 2012. ISBN 0321820274, 9780321820273
- Takatalo, Jari, et al. "User experience in 3D stereoscopic games." *Media Psychology* 14.4 (2011): 387-414.

Při obhajobě semestrální části projektu je požadováno:  
Body 1 až 4, experimenty vedoucí k bodu 5.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Chlubna Tomáš, Ing.**  
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.  
Datum zadání: 1.11.2023  
Termín pro odevzdání: 9.5.2024  
Datum schválení: 9.11.2023

## Abstrakt

Cílem této práce je demonstrovat mechaniky použitelné pro hry, které využívají holografický displej Looking Glass ve spojení s klasickým displejem. Holografický displej je využitý jako sekundární a jako doplňující zařízení, které poskytuje rozšířené možnosti interaktivního zobrazení 3D scény. Tyto mechaniky jsou demonstrovány pomocí prototypu jednoduché hry, která tyto mechaniky implementuje. Jedná se jak o klasické herní mechaniky, tak o mechaniky využívající vlastností holografického displeje jako je úhel pohledu nebo skrývání informací v jednotlivých pohledech. Přínosem této práce je prozkoumání herních mechanik použitelných pro tuto kombinaci displejů.

## Abstract

The goal of this work is to demonstrate game mechanics that can be used for games that use the Looking Glass holographic display in combination with a traditional display. The holographic display is used as a secondary element and as an additional device, which allows extended options for the interactive view of the 3D scene. These mechanics are demonstrated using a prototype of a simple game that implements these mechanics. These game mechanics include both classical game mechanics and game mechanics made specifically for the holographic display like using view angle and information hidden in some views of the display. The contribution of this work is the exploration of game mechanics that can be used for this combination of displays.

## Klíčová slova

Herní mechaniky, 2D displeje, 3D displeje, Holografický displej Looking Glass, Unity, Herní demo, Hra, Kombinace 2D a 3D displejů

## Keywords

Game mechanics, 2D displays, 3D displays, Looking Glass holographic display, Unity, Game demo, Game, 2D and 3D display combination

## Citace

HANUŠ, Jan. *Demonstrace herních mechanik pro kombinaci 2D a 3D displejů*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Chlubna

# Demonstrace herních mechanik pro kombinaci 2D a 3D displejů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Tomáše Chlubny. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Jan Hanuš  
7. května 2024

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Teorie</b>	<b>3</b>
2.1	Holografický displej Looking Glass . . . . .	3
2.2	Herní engine Unity . . . . .	6
2.3	Blender . . . . .	7
2.4	Point cloud . . . . .	9
2.5	Hloubková mapa . . . . .	9
<b>3</b>	<b>Návrh</b>	<b>11</b>
3.1	Žánr . . . . .	11
3.2	Základní struktura hry . . . . .	12
3.3	Uživatelské rozhraní . . . . .	13
3.4	2D a obecné mechaniky . . . . .	13
3.5	3D mechaniky . . . . .	14
<b>4</b>	<b>Implementace</b>	<b>18</b>
4.1	Použité prostředky . . . . .	18
4.2	Modely . . . . .	18
4.3	Světla a postprocessing . . . . .	19
4.4	Zásilky . . . . .	19
4.5	Holografické pečeti . . . . .	24
4.6	Kontrola obsahu zásilek . . . . .	31
4.7	Hlavní menu . . . . .	34
<b>5</b>	<b>Testování</b>	<b>35</b>
5.1	Testeři . . . . .	35
5.2	Průběh testování . . . . .	35
5.3	Výsledky testování . . . . .	36
<b>6</b>	<b>Závěr</b>	<b>37</b>
	<b>Literatura</b>	<b>38</b>
<b>A</b>	<b>Testovací dotazník</b>	<b>39</b>

# Kapitola 1

## Úvod

V dnešní době je herní průmysl jedním z největších průmyslů nejen v odvětví informačních technologií, ale i v porovnání s průmysly z odvětví odlišných. Herní průmysl je dnes větší než filmový a hudební dohromady [1]. Z tohoto důvodu se dá očekávat velice rychlý vývoj nových technologií využitelných v tomto průmyslu a jedním z velice důležitých jsou zobrazovací systémy, u kterých se očekávají nejen vysoké zobrazovací rychlosti ale hlavně schopnosti zobrazit hry v co nejlepší kvalitě, aby se mohl hráč co nejvíce vžít do hry. V této době se výpočetní technika dostává do fáze, kdy je dostatečně silná a zobrazovací technologie dostatečně kvalitní pro přesun na novou úroveň vžití do hry. To je možné pomocí zobrazení 3D světů ne ve 2D ale v částečném či dokonce úplném 3D. Jako příklad je možné uvést postupy v technologiích virtuální reality a nebo v oblasti 3D displejů jako právě holografický displej Looking Glass [?].

Jelikož technologie holografického displeje Looking Glass ještě není na úrovni 2D displejů jak z hlediska kvality, tak velikosti a cenové dosažitelnosti, pak se dá očekávat že jedním z dobrých způsobů využití těchto displejů je použít je jako sekundární a zobrazovat pouze menší část hry. Takto je možné se vyhnout některým nedostatkům jako je například zobrazení scén s velkou hloubkou, či nižší kvalita výsledného obrazu [4].

Cílem této práce je vytvořit herní demo, implementující některé herní mechaniky, použitelné ve hrách využívajících displej Looking Glass. Tímto způsobem by mělo být možné dosáhnout lepšího zážitku z her a lepší pohlcení do hry [5]. Přesněji jsou v této práci implementovány mechaniky využívající holografického a 3D efektu pro skrytí informací v prostoru. Z podobného důvodu také využívá způsob vykreslování herní scény na holografickém displeji opět pro skrytí některých informací. Úkolem hráče je následně tyto informace opět získat pomocí sledování displeje z různých úhlů pohledu.

V kapitole 2 jsou popsány a vysvětleny informace důležité k projektu a je vysvětlen princip holografického displeje a jeho implementace ve vývojovém prostředí Unity. V kapitole 3 je vysvětleno, jaké v tomto demu budou jednotlivé herní mechaniky. Převážně se jedná o mechaniky spojené s holografickým displejem a o mechaniky kombinující holografický displej s normálním. Dále jsou zde popsány obecné informace o projektu a důvody, které vedly k některým z rozhodnutí. Kapitola 4 se zabývá popisem implementace jednotlivých herních prvků a mechanik. Dále obsahuje popisy některých problémů spojených s cílem této práce a jejich řešení. Konečně v kapitole 5 je popsán průběh několika testů a jaké chyby byly při tomto testování objeveny a jakým způsobem byly odstraněny. Následně popisuje výsledky testování ve spojení s cílem této práce.

# Kapitola 2

## Teorie

Tato kapitola se zabývá teorií podstatnou k této práci. Sekce 2.1 se zabývá funkcionalitou holografického displeje Looking Glass a základními principy, na kterých je tento displej založen. V následující sekci 2.2 je popsán herní engine Unity, jakým způsobem se používá ve spojení s holografickým displejem a důvody, z jakých byl upřednostněn před jinými herními engine. V poslední sekci je popsána teorie relevantní k jednotlivým herním mechanikám integrovaných ve výsledné demonstrační hře.

### 2.1 Holografický displej Looking Glass

Holografický displej Looking Glass je skupina holografických displejů od společnosti Looking Glass Factory založené v roce 2014. Od té doby vydala tato společnost na trh množství displejů různých velikostí, včetně doposud největšího holografického displeje na světě s úhlopříčkou 65 palců a 100 úhly pohledu [2]. Na obrázku 2.1 je podoba displeje Looking Glass, kde menší z těchto displejů byl využit v této práci.

Tyto displeje dokáží zobrazit 3D scénu jako hologram uvnitř displeje bez nutnosti 3D brýlí nebo jiných pomůcek. Potenciální použití je téměř kdekoli od umění přes strojírenské odvětví, až po odvětví informačních technologií kde je možné ho použít jak na hry, tak i na seriózní aplikace.



Obrázek 2.1: Příklad modelů displeje Looking Glass<sup>1</sup>.

---

<sup>1</sup>Zdroj: <https://www.cnet.com/reviews/looking-glass-preview/>

Společnost aktuálně na trhu nabízí pět typů displejů. *Looking Glass Go* je nejmenším nabízeným displejem s 2K rozlišením a úhlopříčkou 6 palců. Dalším nabízeným je *Looking Glass 16" Spacial OLED* s 4K rozlišením a úhlopříčkou 16 palců. Tyto dva displeje jsou zatím ve fázi předprodeje, zbylé tři jsou ve fázi aktivního prodeje. Těmito displeji jsou Displej *Looking Glass 65"* s 8K rozlišením a úhlopříčkou 65 palců, *Looking Glass 32"* také s 8k rozlišením a úhlopříčkou 32 palců a nakonec displej *Looking Glass Portrait* s 2K rozlišením a úhlopříčkou 7.9 palce<sup>2</sup>.

Displej použitý na tuto práci je Looking Glass 8.9", displej první generace, který se již nevyrábí, základní princip je však totožný s ostatními displeji.

### 2.1.1 Jak displej Looking Glass funguje

Displej Looking Glass je vícehledový displej, který v současné době na vodorovné ose umožní zobrazení různého počtu pohledů 3D scény závislejícím na typu displeje. V případě této práce je pohledů 45, v zorném úhlu přibližně 50°, tím je dosaženo 3D efektu, jelikož každé oko vidí jiné pohledy, a tak vidí 3D scénu z jiného úhlu<sup>3</sup>.

Vstupními daty displeje je sada snímků uspořádaných do jednoho obrazu nazývaného Quilt popsaného v podsekcí 2.1.2 a pro správné fungování je nutná aplikace Looking Glass Bridge, která zajišťuje korektní komunikaci s displejem [3].

### 2.1.2 Quilt

Jako quilt se označuje sada snímků, které Looking Glass využívá k zobrazení jednoho holografického snímku jak pro statické obrázky tak i pro videa.

Každý snímek Quiltu reprezentuje pohled kamery z jednoho úhlu a po jeho předání ovladači Looking Glass Bridge je na displeji zobrazena iluze hologramu. Tento formát se používá k zobrazení a sdílení obrázků a videí pro Looking Glass a je vytvořen pro každý snímek hry, to způsobuje značnou grafickou náročnost tohoto displeje. Příklad quiltu je vidět na obrázku 2.2.

### 2.1.3 Princip zobrazení Quiltu na holografickém displeji

Displej při zobrazování prochází každý obrázek ve quiltu a promítá ho do jednotlivých pohledů. Soubor se zobrazuje zleva doprava. Snímky jsou ve quiltu srovnané podle obrázku 2.2 a jednotlivé pohledy jsou vázány vzestupně od jedné.

Na obrázku 2.3 se nachází Quilt s jednotlivými snímky scény a následně zvýrazněný snímek s pořadovým číslem 9 společně s reprezentací úhlu, pod kterým je možné tento snímek vidět na displeji Looking Glass.

### 2.1.4 Paralax a ostření

Displej Looking Glass se pokouší napodobit způsob ostření lidského oka [?]. To se při zaostření na určitý objekt nebo vzdálenost nezaměří na okolní objekty. Tím je dosaženo maximální ostrosti potřebného objektu, ostrost všech ostatních předmětů se však zanedbá a jsou tím pádem rozostřené.

---

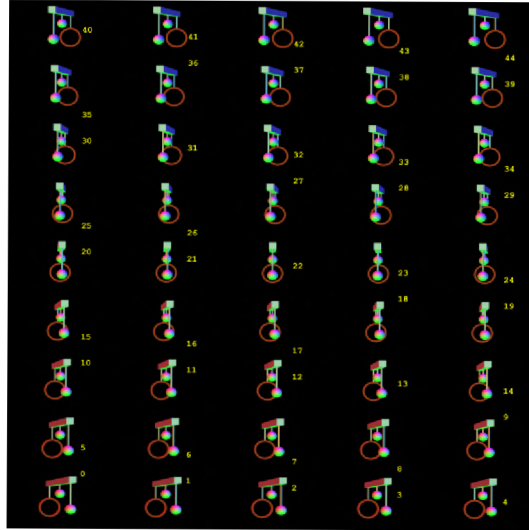
<sup>2</sup>Přehled produktů Looking Glass Factory dostupné z odkazu:

<https://lookingglassfactory.com/product-overview>

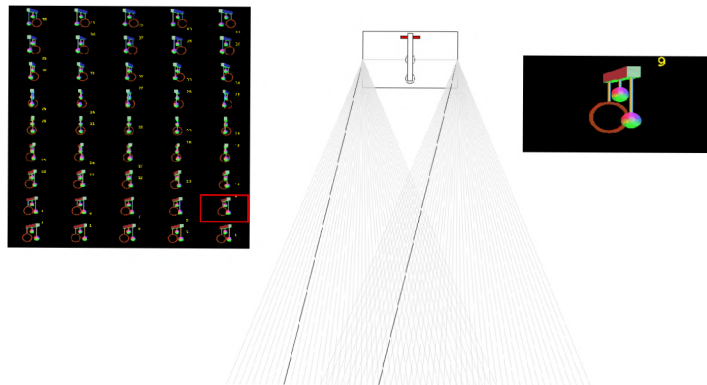
<sup>3</sup>Looking Glass Documentation [online]. 2023 [cit. 2023-9-1]. Dostupné z odkazu:

<https://docs.lookingglassfactory.com/keyconcepts/how-it-works>

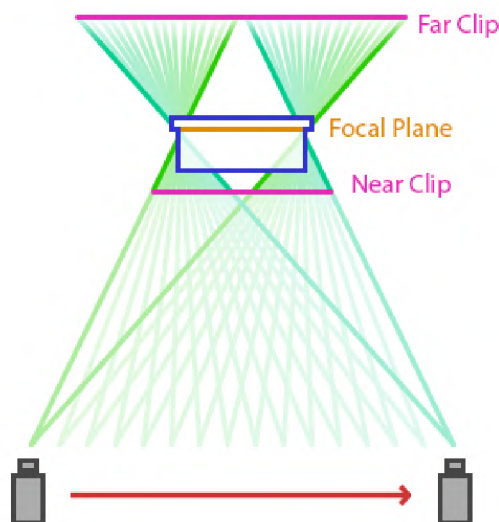




Obrázek 2.2: Příklad formátu quilt, který využívá displej Looking Glass pro zobrazení 3D snímku.



Obrázek 2.3: příklad, jakým způsobem probíhá promítnutí quiltu na displeji



Obrázek 2.4: Znázornění způsobu promítnutí jednotlivých pohledů a jejich ostření

Looking Glass tuto vlastnost napodobuje pomocí zaostřovací roviny, kde její vzdálenost od předmětu určuje míru jeho zaostření, tímto způsobem je dosaženo efektu ostrosti a paralaxy. Rovina, na které leží zaostřovací rovina je na displeji statická a maximálně ostrá. Všechny ostatní roviny jsou s rostoucí vzdáleností od zaostřovací roviny postupně stále více rozostřené a s měnícími se pohledy se stále více posouvají, jelikož mají větší paralaxu.

Na obrázku 2.4 je znázorněno, jakým způsobem se výsledný quilt promítne na displej, zaostřovací rovina leží v rovině displeje a všechny ostatní se od ní opticky vzdalují, dále je zde vidět jak vzniká při zobrazení paralaxa.

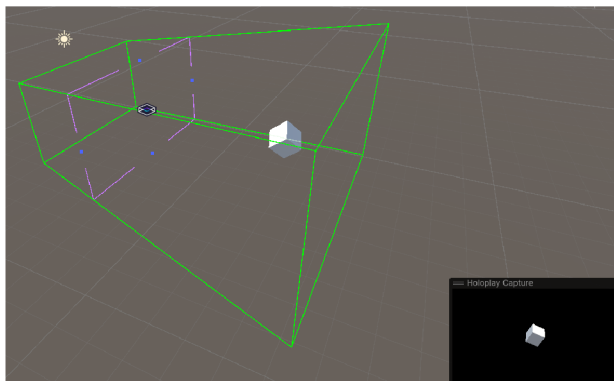
Tento způsob zobrazení může být problematický při zobrazování hodně hluboké scény, jelikož při nárůstu vzdálenosti nad určitou hranici začne být efekt rozostření rušivý až nepřijemný, tento efekt se dá zmírnit použitím post-processingu [4].

### 2.1.5 Nástroje pro Looking Glass displej

Pro Looking Glass existuje hned několik různých nástrojů, nutný je však pouze jeden. Looking Glass Bridge, se dá považovat za ovladač, jelikož bez něho není možné na displeji zobrazit 3D scénu. Displej je však stále použitelný jako 2D displej. Druhým důležitým nástrojem je Looking Glass Studio, který slouží k zobrazení a sdílení obrázků a videí vytvořených pro tento displej. Ostatní nástroje se používají na základě využití displeje, z nichž jsou pro tento projekt důležité převážně vývojářské nástroje pro různé druhy herních engine či 3D editorů jako například Blender, Unreal Engine, či Unity. Jelikož je tento projekt vytvořen v Unity, tak bude tato práce zaměřena převážně na nástroje obsažené v tomto balíčku.

## 2.2 Herní engine Unity

Herní engine Unity je jeden z nejoblíbenějších herních engine [8]. Podobně jako mnoho jiných herních engine, i Unity přebírá část práce za vývojáře, jako je zobrazení a správa scén, správa vstupů a zvuků a simulace fyziky pomocí fyzikálního engine PhysX. Dále umožňuje vývoj



Obrázek 2.5: Objekt *Hologram Camera* v Unity Editoru

her pro většinu dnešních herních platforem, včetně webových aplikací. Dalšími výhodami Unity jsou integrovaný Asset Store, kde mohou vývojáři nabízet a stahovat jednotlivé i celé balíčky assetů, využití jazyka C# pro psaní skriptů, obsáhlá a detailní dokumentace a velká, aktivní komunita vývojářů.

### 2.2.1 Základní struktura hry v Unity

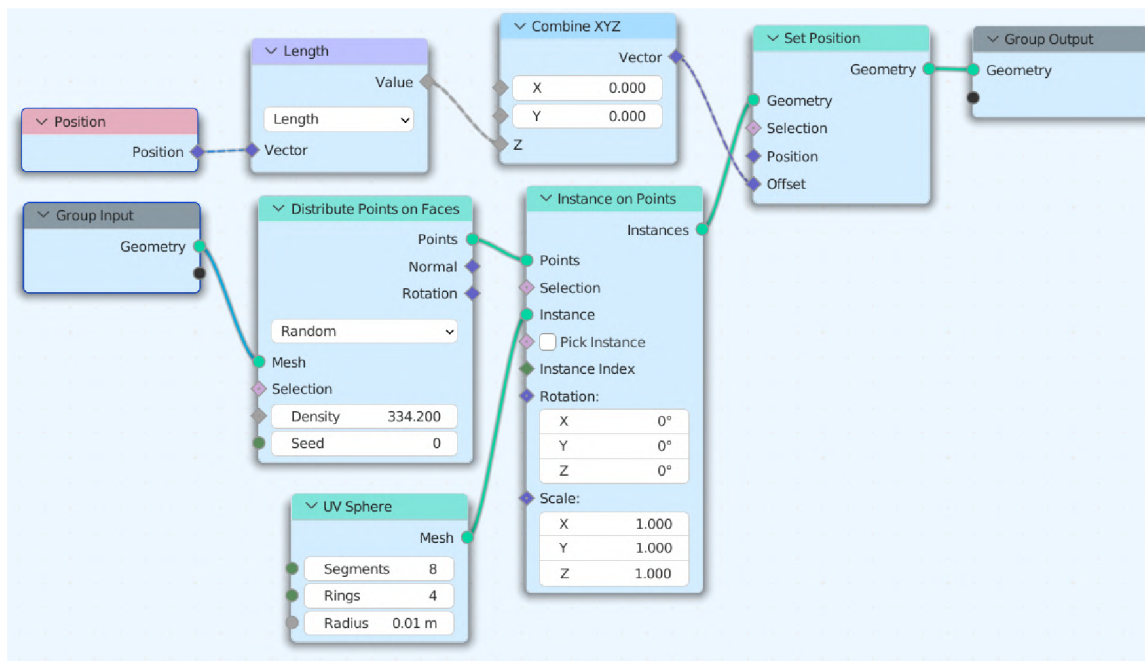
Základem každé hry jsou jednotlivé scény. Každá hra obsahuje alespoň jednu scénu, obvykle se však skládá z více scén. Scény se dále skládají z herních objektů, které mohou obsahovat různé komponenty. Základním prvkem je prázdný herní objekt. Jeho funkce se určuje přidáním komponent, jako jsou renderery, collidery, skripty a animace. Zvláštní komponentou je kamera, která je nezbytná pro každou scénu, ta snímá herní dění a promítá ho na obrazovku. Hlavní kamera obvykle obsahuje také komponentu pro snímání zvuku. Jednotlivé herní objekty nebo skupiny objektů lze následně uložit jako prefabrikáty, které lze následně libovolně vkládat do scén s přednastavenými komponenty. Tyto prefabrikáty lze také využívat ve scriptech, ve kterých je možné přistupovat k jednotlivým komponentům, podobně jako u objektů ve scéně, a následně je vložit do scény.

### 2.2.2 Unity a Looking Glass

Pro vývoj her v Unity, využívajících displej Looking Glass, je nutné importovat balíček se zdroji potřebnými ke správnému snímání a fungování displeje. Ten se dá stáhnout ze stránek Looking Glass Factory. Kromě tohoto balíčku je ještě třeba ze stejných stránek stáhnout a nainstalovat aplikaci Looking Glass Bridge, který zastává roli ovladače schopného zobrazit na displeji hologram. Jednou z hlavních součástí balíčku jsou prefabrikáty. Nejdůležitějším je *Hologram Camera*, který slouží jako kamera snímající 3D scénu pro holografický displej. Na obrázku 2.5 je vyobrazena podoba objektu *Hologram Camera* při editaci scény. Vše co se nachází uvnitř zorného pole vyhrazeného zelenými linkami bude zobrazeno na holografickém displeji se zaostřovací rovinou procházející rovinou vyznačenou fialovou linkou.

## 2.3 Blender

Blender je svobodný 3D modelovací software s otevřeným kódem. Tento software umožňuje provádět všechny kroky 3D návrhu a modelování v jednom nástroji, od samotného modelování přes textury a renderování až po vytváření animací. Dalšími výhodami je podpora



Obrázek 2.6: Síť geometrických uzlů, která vytváří point cloud s body posunutými po ose Z o vzdálenost bodu od počátku geometrie.

jazyka Python pro psaní vlastních nástrojů, možnost definice sítě uzlů popisujících specializované renderery, nebo za pomoci těchto uzlů pracovat přímo s objekty a modely. V tomto projektu je Blender využíván pro tvorbu herních modelů, zejména pro tvorbu holografických pečetí.

### 2.3.1 Geometrické uzly

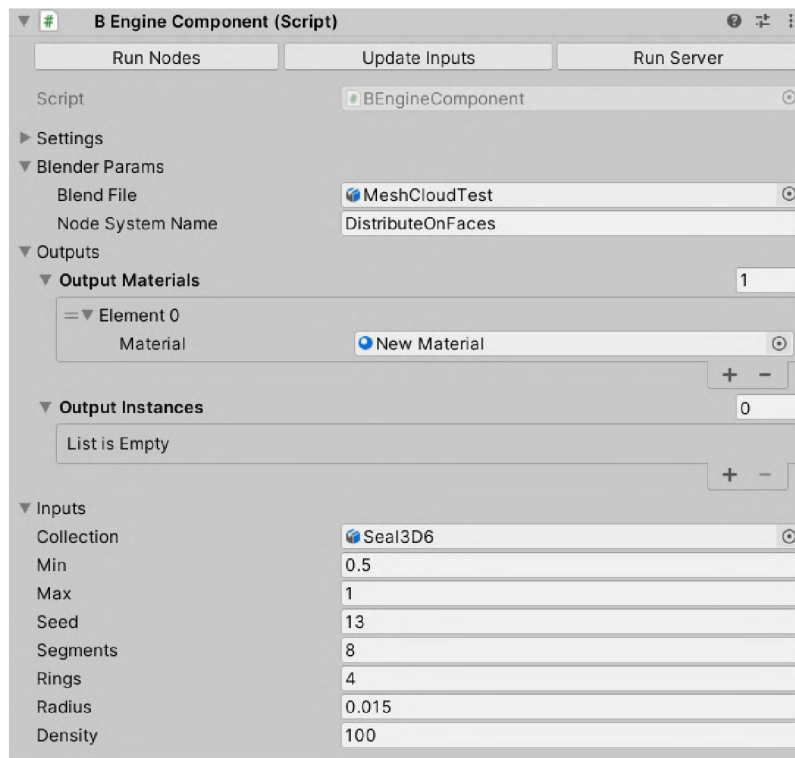
Geometrické uzly jsou jedním z nástrojů Blenderu. Umožňují procedurální kontrolu nad modely a objekty, včetně jejich modifikace a generování nových objektů. Tyto uzly mají specifickou funkci a množinu vstupů a výstupů různých typů. Propojováním výstupů se vstupy s kompatibilním typem se vytváří síť uzlů konající určitě operace nad vybraným objektem či vytváření a modifikace nových objektů.

Na obrázku 2.6 je jedna z využitých sítí geometrických uzlů. Tato síť vygeneruje body na povrchu objektu a přesune je po ose Z do pozice odpovídající její vzdálenosti od počátku této geometrie.

### 2.3.2 BEngine

BEngine je nástroj umožňující využití geometrických uzlů Blenderu přímo v prostředí Unity. Toho je dosaženo spuštěním instance Blenderu [7], otevřením modelu v této instanci, provedením modifikace sítě uzlů a následným uložením změn. Zároveň tento nástroj umožňuje nastavení vstupů sítě přímo v Unity a aplikaci materiálů unity na výstupní objekty k zaručení kompatibility.

Na obrázku 2.7 je vyobrazena konfigurace komponenty BEngine v Unity. Pomocí této komponenty je možné provádět funkce popsané Geometrickými uzly přímo v Unity po



Obrázek 2.7: Příklad BEngine komponenty v unity

definování Blender souboru obsahujícího požadované uzly, vybrání názvu této sítě a doplnění hodnot požadovaných touto sítí uzlů.

## 2.4 Point cloud

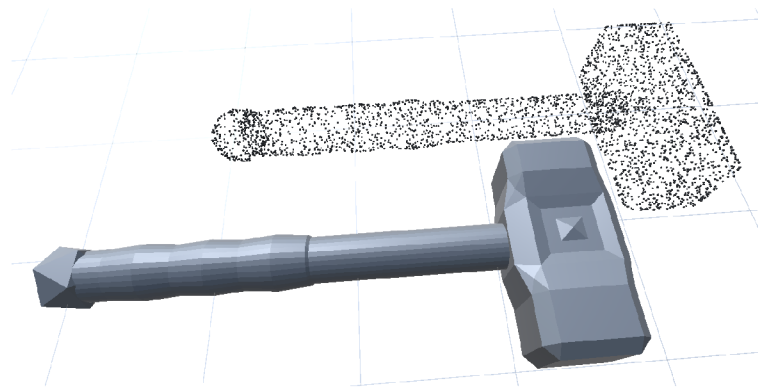
Za point cloud se označuje konečná množina bodů v trojrozměrném prostoru [6]. Často se využívá k reprezentaci topologie, přírodních útvarů či reálných objektů v digitální podobě a obvykle je získán pomocí lidarů, či fotogrammetrie. V tomto případě je využíván k reprezentaci holografických pečeti a získán pomocí geometrických uzlů z předdefinovaného objektu nebo obrázku obsahujícího hloubkovou mapu popisující tvar této pečeti.

Na obrázku 2.8 je point cloud získaný z objektu zobrazeného pod ním za pomoci geometrických uzlů popsaných v podsekcí 2.3.1 a Blenderu.

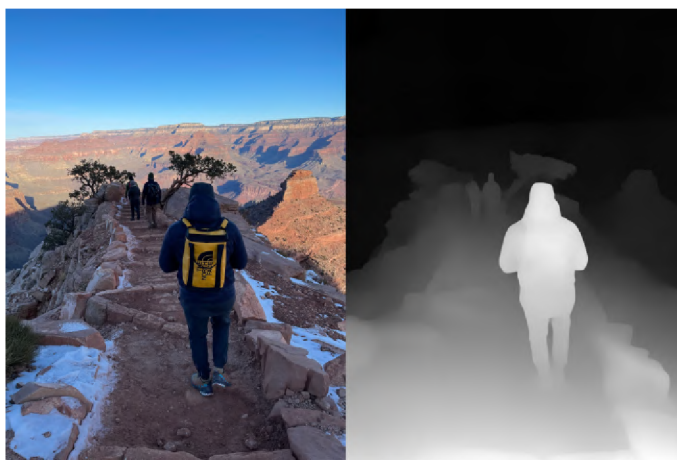
## 2.5 Hloubková mapa

Hloubková mapa je pojem využívaný v počítačové grafice označující obraz nebo mapu, vyznačující vzdálenost bodů od pozorovatele. Každá mapa je tvořena body nebo pixely, které mají určitou hodnotu vyznačující vzdálenost od pozorovatele nebo kamery a může být využita buď jako absolutní hodnota vzdálenosti, nebo může být normalizována a vyznačovat rozmezí vzdálenosti v rozsahu od nuly do jedné [?].

Hloubkové mapy se často využívají v různých postprocesech a efektech ve 3D grafice, příkladem může být rozmazání pozadí, detekce kolizí, zpracování stínů a mnoho dalších.



Obrázek 2.8: Příklad objektu a jeho point cloud reprezentace.



Obrázek 2.9: Příklad Hloubkové mapy a původního obrázku<sup>4</sup>.

Je to klíčový prvek pro vytváření realističtějších a vícevrstevnatých vizuálních scén ve videohrách, simulátorech, filmových efektech a dalších oblastech počítačové grafiky.

Obrázek 2.9 obsahuje dva obrázky, první je foto použité pro generování hloubkové mapy a druhý je z něho vytvořená hloubková mapa.

V tomto projektu se hloubková mapa využívá ke generování holografických pečeti, popsaných v sekci 4.5 s využitím geometrických uzlů Blenderu a nástroje BEngine. Přesněji se na základě poskytnuté hloubkové mapy v podobě obrázku v odstínech šedi vytvoří nová point cloud reprezentace této pečeti.

---

<sup>4</sup>Zdroj: <https://lookingglassfactory.com/blog/depth-maps-best-practices>

# Kapitola 3

## Návrh

V této kapitole je popsán návrh řešení a jednotlivé herní prvky této práce, žánr hry, její prostředí a kontext s důvody, které vedly k těmto rozhodnutím. Následně jsou popsány jednotlivé herní mechaniky a způsoby, jakými využívají klasický a holografický displej.

Cílem této práce má být demonstrační hra, demonstrující využití normálního a holografického displeje zároveň. Přesněji, implementující mechaniky využitě na klasickém displeji a mechaniky využitě na holografickém displeji v jedné hře a jistý způsob jejich propojení do vzájemně spolupracujícího celku.

### 3.1 Žánr

Žánr hry je do značné míry ovlivněný limitacemi displeje a samotným tématem práce. Bylo tedy rozhodnuto vytvořit hru kombinující 2D i 3D herní grafiku, kde klasický displej zobrazuje 2D část hry a holografický zprostředkovává 3D část.

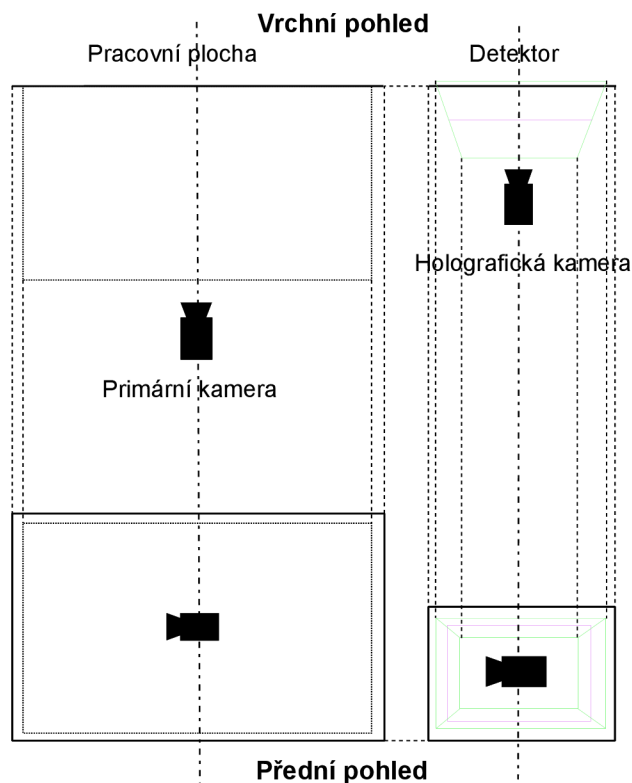
Nakonec byl vybrán žánr simulace s prvky skrytých předmětů/informací, inspirovaný hrou *Papers, Please*<sup>1</sup>, snímek z které je vidět na obrázku 3.1. Hráč je postaven do role kontrolora, který přijímá a zpracovává různé zásilky a k nim přiložené dokumenty. Cílem je ověřit správnost informací, případně nalézt chyby a podle toho zpracovat zásilku.



Obrázek 3.1: Snímek ze hry *Papers please* použité jako inspirace demonstrační hry<sup>2</sup>.

<sup>1</sup>Hra dostupná v obchodě steam společnosti valve: <https://store.steampowered.com/app/239030/PapersPlease/>

<sup>2</sup>Zdroj: <https://store.steampowered.com/app/239030/PapersPlease/>



Obrázek 3.2: Diagram znázorňující rozložení prostoru herní scény.

### 3.2 Základní struktura hry

Základní strukturou hry jsou dvě obrazovky, první obrazovka umístěna vpravo je určena pro klasický displej a má podobu kanceláře. Tato obrazovka je určena k přijímání, odesílání a přesouvání zásilek, včetně do druhé obrazovky, a ke kontrole textových údajů na zásilce a přiložených dokumentech. Dále umožňuje sledování času a jiných informací a nahlédnutí do nápovědy.

Druhá obrazovka určená pro holografický displej má podobu detektoru. Na této obrazovce probíhá kontrola obsahu zásilky a hledání případného kontrabandu a ke kontrole různých druhů holografických pečetí. Zásilka je na tuto obrazovku vložena jejím přesunutím z první obrazovky na pravý kraj pracovní plochy. Dále je možné přepínat jednotlivé režimy a požádat o navrácení zásilky detektoru pomocí panelu zobrazeného v pravém horním rohu první obrazovky, který se zobrazí při vložení zásilky do detektoru a po jejím návratu na pracovní plochu se skryje.

Na obrázku 3.2 je znázorněn diagram herní scény s první obrazovkou snímanou pomocí ortografické kamery a druhou využívající objekt *Hologram Camera* poskytnutý v balíčku nástrojů pro holografický displej. Scéna je rozdělena na dvě spolu komunikující ale nepřekrývající se části.

Úkolem hráče je přijímat zásilky na první obrazovce a následně ověřit správnost jejich informací a informací na přiložených dokumentech. Následně je možné provést na detektoru



kontrolu obsahu zásilky a holografické pečeti. V případě správného rozhodnutí hráče o správnosti zásilky je hráči udělen bod a při špatném rozhodnutí je udělena informace o chybách v zásilce. Po vypršení limitu varování je hráči udělena bodová pokuta. Tyto body jsou následně využity jako podmínka pro postup do následujícího dne.

### 3.3 Uživatelské rozhraní

Uživatelské rozhraní je pouze jednoduché a obsahuje absolutní minimum prvků pro zajištění správného fungování hry. Rozhraní je tvořeno základními objekty rozhraní Unity, obsahuje několik panelů, které se zobrazují, nebo mizí, podle aktuálního stavu a pozice zásilky. Tyto panely obsahují tlačítka pro ovládání detektoru a odevzdání zásilky či požádání o další zásilku. Kromě samotného herního rozhraní obsahuje demonstrační hra také rudimentární hlavní menu a obrazovku značící konec dne.

#### 3.3.1 Režimy detektoru

Po vložení zásilky do detektoru jejím přesunutím na požadované místo na kraji zásilky, se zásilka přesune na detektor zobrazený na holografickém displeji a na klasickém se zobrazí panel umožňující jeho navrácení na pracovní plochu a tlačítka pro změnu režimu detektoru. V současné podobě jsou implementovány tři režimy detektoru:

- **Základní režim:** V tomto režimu je zásilka viditelná ve stejné podobě jako na pracovní ploše
- **Režim kontroly obsahu:** V tomto režimu je zobrazen obsah zásilky a případný kontraband ohraničený reprezentací obalu zásilky
- **Režim kontroly pečeti:** V tomto režimu je zobrazena pouze holografická pečeť pro umožnění maximální velikosti pečeti a tak nejlepší možnosti správné kontroly zásilky. Tento režim je vynechán pokud zásilka neobsahuje pečeť.

Mezi těmito režimy se přepíná stiskem tlačítka cyklicky jedním směrem a při kliku na tlačítko návratu zásilky je nejprve přepnuto zpět do normálního režimu.

#### 3.3.2 Náповědy a návody

Uživatelské rozhraní kromě jednoduchých ukazatelů dat jako je váha zásilky, datum dne a čas do konce dne, také umožňuje zobrazit jednoduchou nápovědu, která hráči sdělí správné názvy států a jejich regionů a pečeti, tyto informace jsou nutné ke správnému rozhodnutí o bezchybnosti zásilky, tato nápověda je hráči přístupná podržením klávesy H.

### 3.4 2D a obecné mechaniky

Tato sekce se zabývá 2D mechanikami, tyto mechaniky jsou tvořeny primárně pro klasický displej, je možné jich ale využít i na holografickém.

#### 3.4.1 Přijetí a odeslání zásilky

Pokud hráč právě nezpracovává zásilku, může si zažádat o novou. Ta je k němu dopravena po dopravě na jeho hlavní pracovní plochu. Po provedení kontroly hráč umístí zásilku

zpět na dopravník a rozhodne, zda je v pořádku nebo obsahuje chyby. Následně je zásilka odvezena pryč. Pokud hráč udělal chybu, je mu podána informace o nesprávném rozhodnutí.

### 3.4.2 Správa času

Hráč je omezen časem, po který trvá jeden den, a jeho cílem je zpracovat největší možné množství zásilek. Po vypršení času do konce dne má hráč možnost dokončit poslední přijatou zásilku, a následně je den ukončen a vyhodnocen. Programátor má možnost tento čas měnit, pro každý den je však tento čas identický.

### 3.4.3 Kontrola údajů zásilky

Každá přijatá zásilka může obsahovat jednu či více chybných informací. Hráč má za úkol tyto chybné údaje odhalit buď porovnáním informací, které by měly být shodné, nebo pomocí příručky, obsahující nápovědy k ověření údajů, jako jsou správné kraje či známky pro jednotlivé státy původu zásilky. Všechny generované hodnoty jsou vymezeny limity či hodnotami definovanými vývojářem. Každý den má tyto hodnoty, až na několik výjimek, definované samostatně. Je tedy možné mít každý den jiné náhodné hodnoty, včetně chybných hodnot.

Dále Hráč musí ověřit platnost jednotlivých dokumentů a platné datum odeslání zásilky pomocí kontroly s aktuálním datem, které bude hráči poskytnuto.

### 3.4.4 Ovládání

Hra se ovládá téměř výhradně pomocí myši jediným případem využití klávesnice je zobrazení panelu s pomocnými informacemi a návrat do hlavního menu pomocí klávesy *Escape*.

### 3.4.5 Úrovně

Jednotlivé úrovně mají stejnou podobu, liší se však daty zásilek, vývojář má možnost definovat jednotlivé dny, jaké hodnoty se v nich budou generovat a jaké jsou povolené či zakázané hodnoty. Vývojář má také možnost definovat, jaké dokumenty přiložené k zásilce se budou generovat a jaké jsou pravděpodobnosti na chybu v jednotlivých parametrech zásilky a dokumentu.

## 3.5 3D mechaniky

Tato sekce se zabývá 3D mechanikami, tyto mechaniky jsou určeny exkluzivně pro holografický displej.

### 3.5.1 Holografické pečeti

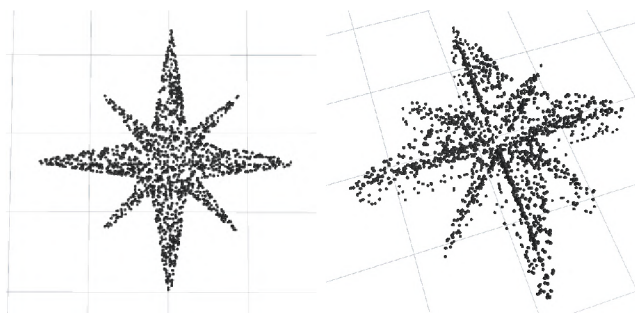
Zpracovávané zásilky mohou obsahovat holografickou pečeť jejíž platnost je nutné kontrolovat za použití holografického displeje.

Tyto pečeti jsou rozděleny do několika typů pomocí způsobu jejich zobrazení a kontroly. Všechny pečeti jsou reprezentovány point cloudy vytvořenými pomocí geometrických uzlů popsanych v sekci 2.4.

Tyto pečeti slouží jako způsob ověření platnosti a bezchybnosti zásilky a jsou jednou z hlavních mechanik této práce. Kromě možnosti záměny správné pečeti za pečeť náleží



Obrázek 3.3: Příklad ploché pečeti před dalším zpracováním.

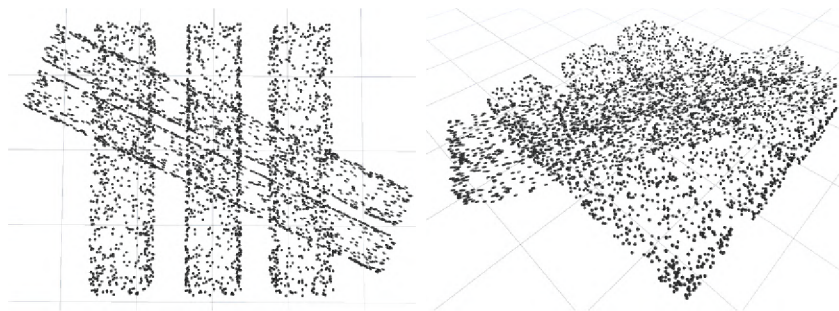


Obrázek 3.4: Příklad pečeti využívající hlobkovou mapu.

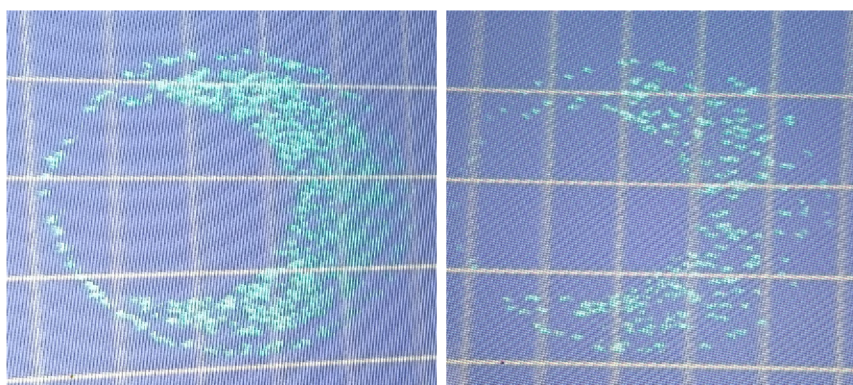
špatnému státu je možné zobrazení pečeti, u které nebude možné rozpoznat její původní podobu a tedy bude také chybná. Správná pečeť by tedy měla být rozpoznatelná pouze z jednoho úhlu pohledu a měla by náležet státu původu zásilky.

Programátor má možnost výběru ze tří typů pečeti, které bude také možné kombinovat, případně přidávat nové, a v případě jednoho z typů také tři druhy úpravy pečeti před jejím zobrazením. Tyto typy jsou:

1. **Plochá pečeť** - Tato pečeť je tvořena point cloudem tvořící obraz pečeti umístěné na jedné rovině. Tento typ je určený k další modifikaci před jejím zobrazením s cílem umožnit její pozorování pouze ze správného úhlu pohledu. Způsoby úpravy jsou popsány níže v podsekcí 3.5.2. Jsou vytvořeny na základě obrazu, který byl předán geometrickým uzlům s násobičem hloubky nastaveným na nula. Podoba jedné z těchto pečeti je znázorněna na obrázku 3.3.
2. **Pečeť tvořená hlobkovou mapou** - Tyto pečeti jsou vzhledově a původem podobné plochým pečetím, nejsou však umístěny na rovině ale jejich point cloud má i určitou hloubku. Tyto pečeti jsou opět tvořeny na základě obrazu, ale jejich obrazy jsou tvořeny odstíny šedi a při jejich generování pomocí geometrických uzlů byl násobič hloubky nastaven na nenulovou hodnotu. Podoba jedné z těchto pečeti je znázorněna na obrázku 3.4.
3. **Trojrozměrné pečeti** - Tyto pečeti jsou opět reprezentovány point cloudem, ale jejich základem je trojrozměrný objekt, který byl na point cloud převeden jinou sítí geometrických uzlů. Podoba jedné z těchto pečeti je znázorněna na obrázku 3.5.



Obrázek 3.5: Příklad pečeti založené na objektu.



Obrázek 3.6: Snímky z demonstrační hry ukazující pečeť využívající maskování pohledů z různých úhlů, na prvním snímku z nekompletního pohledu a na druhém z kompletního.

### 3.5.2 Způsoby úpravy plochých holografických pečetí

V této podsekcí jsou popsány způsoby modifikace plochých pečetí před jejím zobrazením ve hře. Cílem těchto modifikací upravit pečeť takovým způsobem, který umožní rozpoznat její původní tvar pouze ze správného úhlu pohledu.

#### Maskování pohledů

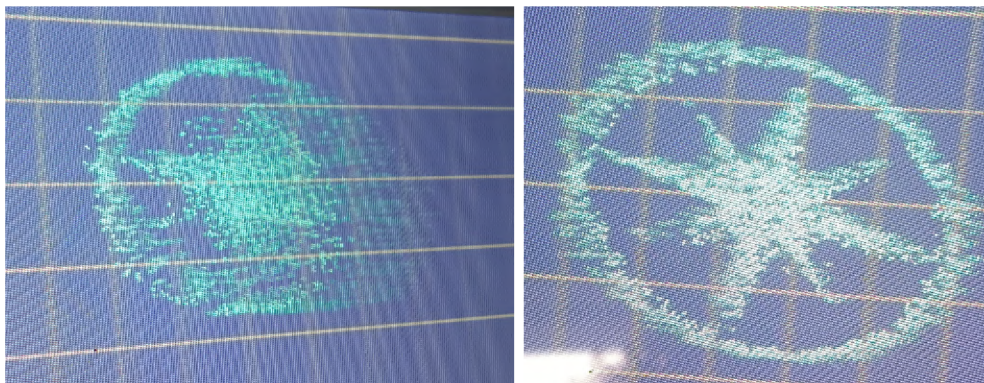
Nastavení vrstvy objektu a zobrazovacích masek pohledů displeje umožňuje vývojáři skrýt objekty na vybraných pohledech displeje. Tímto způsobem se dá efektivně využít holografického displeje a přimět hráče se dívat na displej z více úhlů. Na obrázku 3.6 je zobrazena jedna z takto upravených pečetí ze dvou různých úhlů pohledu.

Takto je dosaženo rozložení částí pečeti do různých pohledů displeje a zajistí se tak její pozorovatelnost v celku pouze na určitých pohledech a tedy určitých úhlech pohledu.

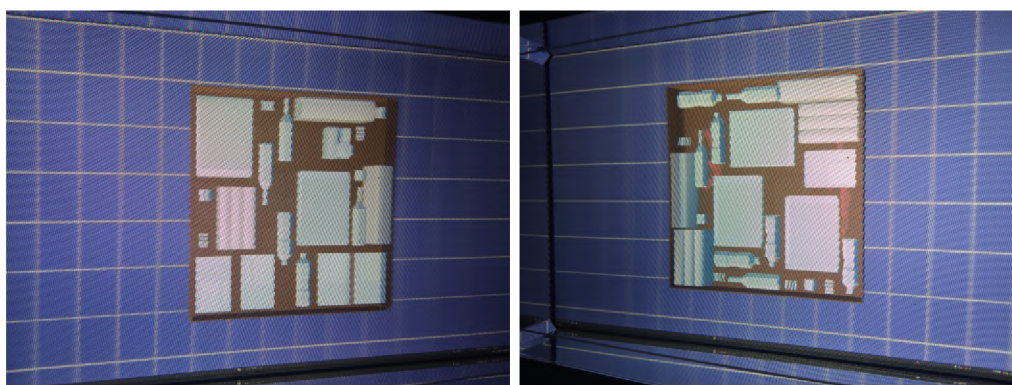
V případě chybné pečeti může nastat situace, kdy není možné pozorovat pečeť v celku z jakéhokoliv úhlu pohledu.

#### Využití úhlu pohledu

Rozložením jednotlivých částí holografické pečeti do prostoru lze dosáhnout oblaku, který umožní pozorovat původní podobu pečeti pouze ze správného úhlu. Je tedy nutné aby hráč prohlédl pečeť z více úhlů pohledu aby mohl vyhodnotit původní podobu. V případě



Obrázek 3.7: Snímky z demonstrační hry ukazující pečeť využívající úhly pohledu z různých úhlů, na prvním snímku ze špatného, na druhém ze správného.



Obrázek 3.8: snímky z demonstrační hry ukazující zásilky s různým obsahem, zásilka na druhém snímku obsahuje kontraband

chybné pečeti mohou některé části pečeti chybět nebo pečeť nemusí z žádného úhlu tvořit smysluplný tvar. Příklad této pečeti z různých úhlů je na obrázku 3.7.

### 3.5.3 Kontrola obsahu zásilky

Obsah zásilky bude generován náhodně do 2D mřížky, tyto objekty budou mít jak náhodnou pozici tak rotaci. Tento obsah se následně bude moci porovnat s dokumentem obsahujícím seznam obsahu, nebo bude možné skrýt kontraband pod jeden z objektů. Pomocí 3D efektu bude cílem pro hráče tento kontraband pomocí sledování zásilky z různých úhlu nalézt a nahlásit. Na obrázku 3.8 je možné vidět dva příklady obsahu zásilky, kde druhý obrázek obsahuje také kontraband.

## Kapitola 4

# Implementace

Tato kapitola se zabývá řešením a implementací jednotlivých částí tohoto projektu. Jsou zde vysvětleny řešení některých problémů a postupy tvorby jednotlivých herních mechanik.

K implementaci hry byl použit herní engine Unity verze 2022.3.10f1 a k implementaci kódu bylo využito Visual Studio 2022 společnosti Microsoft. Dále byl využitý Loking Glass plugin verze v2.0.0-alpha13 a Looking Glass Bridge verze 2.3.5.

### 4.1 Použité prostředky

Většina assetů je vlastní výroby, pro vytvoření 2D assetů byl využit program Krita a pro vytváření 3D assetů byl využit Blender.

Návrhy pečetí byly vytvořeny Karolínou Kavanovou a obsah zásilky a kontraband je až na pár výjimek tvořený objekty z několika různých balíčků s assety, tyto balíčky jsou:

1. *Trash Low Poly Cartoon Pack*<sup>1</sup>
2. *FREE Lomo Camera*<sup>2</sup>
3. *Speakers PBR*<sup>3</sup>
4. *Voicebox*<sup>4</sup>
5. *Guns Pack: Low Poly Guns Collection*<sup>5</sup>

Využitým objektům byl nastaven identický materiál, jehož cílem je změnit podobu těchto objektů na obraz vytvořený nahlédnutím do zásilky bez jejího otevření podobně jako na letištních snímačích zásilek. jeden z těchto objektů je na obrázku 4.1 včetně jeho varianty s vlastním materiálem.

### 4.2 Modely

Většina modelů jsou pouze jednoduché modely vytvořené v Blenderu, jedinou výjimkou jsou modely pečetí probírané v sekci 4.5.

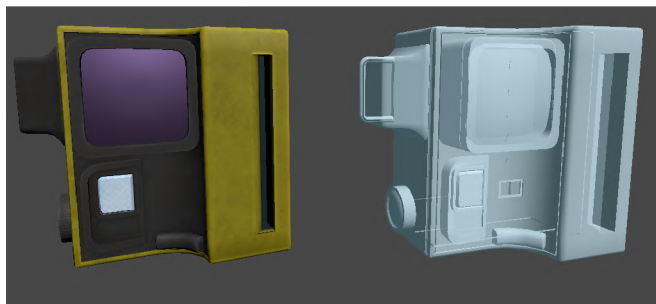
<sup>1</sup>[assetstore.unity.com/packages/3d/trash-low-poly-cartoon-pack-66229](https://assetstore.unity.com/packages/3d/trash-low-poly-cartoon-pack-66229)

<sup>2</sup>[assetstore.unity.com/packages/3d/props/electronics/free-lomo-camera-186797](https://assetstore.unity.com/packages/3d/props/electronics/free-lomo-camera-186797)

<sup>3</sup>[assetstore.unity.com/packages/3d/props/electronics/speakers-pbr-111606](https://assetstore.unity.com/packages/3d/props/electronics/speakers-pbr-111606)

<sup>4</sup>[assetstore.unity.com/packages/3d/props/electronics/voicebox-110022](https://assetstore.unity.com/packages/3d/props/electronics/voicebox-110022)

<sup>5</sup>[assetstore.unity.com/packages/3d/props/guns/guns-pack-low-poly-guns-collection-192553](https://assetstore.unity.com/packages/3d/props/guns/guns-pack-low-poly-guns-collection-192553)



Obrázek 4.1: Příklad modelu použitého jako objekt v zásilce před a po aplikování vlastního materiálu

Projekt také obsahuje sadu objektů, které mohou být umístěny do zásilky, které musí odpovídat velikosti mřížky, do které jsou vkládány. Tyto objekty jsou předdefinované v podobě prefabrikátů který má nastavené posunutí, rotaci a velikost takovým způsobem, který umožní tyto objekty následně bez problémů vložit do zásilky.

### 4.3 Světla a postprocessing

Všechna světla ve scéně jsou pouze statická, jelikož se jak normální, tak holografická kamera nepohybují, není třeba použít dynamická světla.

Jelikož je scéna cíleně málo hluboká není nutné využít postprocessing pro zjemnění rozostření ve vyšších vzdálenostech od holografické kamery.

### 4.4 Zásilky

Zásilky jsou jednou z hlavních herních mechanik tohoto projektu a vyhodnocení jejich správnosti je hlavním cílem každého dne.

Po započetí nového dne hráč opakovaně získává zásilky, které mohou, ale nemusí, obsahovat chyby. Zásilky jsou buď předdefinované vývojářem, nebo vygenerované náhodně a kromě samotné zásilky je hráči zasláno také určité množství dokumentů, které mají své hodnoty vázané k dané zásilce a jsou dalším způsobem rozhodnutí o její bezchybnosti. Zásilka se štítkem s daty je vyobrazena na obrázku 4.2. Každá zásilka má tuto podobu, liší se pouze v datech na štítku.

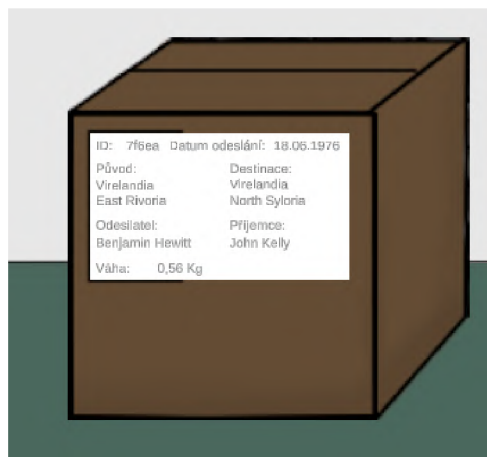
V současnosti jsou předdefinované zásilky implementované pouze částečně a jejich plánovaná. Jediným způsobem získání zásilky je tedy jejich náhodné generování.

#### 4.4.1 Generování zásilek

Po každém zažádání o další zásilku je buď vybrána předdefinovaná zásilka ze seznamu, nebo je vygenerována nová na základě hodnot aktuálního dne a omezujících hodnot jednotlivých parametrů zásilky a přiložených dokumentů.

Předdefinovaná zásilka je vybrána v případě že její index je stejný nebo vyšší jako pořadí následující zásilky, pokud není k dispozici žádná vhodná předdefinovaná zásilka je vygenerována nová, náhodná. Postup generování zásilek je rozdělen do několika kroků:

1. Rozhodnutí o správnosti zásilky



Obrázek 4.2: Podoba zásilky na pracovní ploše

2. Vyhodnocení chybných parametrů
3. Generování hodnot zásilky
4. Generování jednotlivých dokumentů

#### 4.4.2 Vyhodnocení chybných parametrů

O správnosti zásilky rozhoduje jednoduchá podmínka založená na náhodné hodnotě. Pokud je zásilka chybná, je dále vyhodnoceno, ve kterých parametrech zásilky se nachází chybná hodnota. Tyto parametry jsou omezeny na takové, které mohou obsahovat chybu, a jsou aktivní v daný den, přičemž je u každého parametru zohledněna jeho šance na chybu. Pravděpodobnost chybného parametru  $P(p)$  je vyjádřena jako poměr numerické hodnoty šance na chybu parametru  $H(p)$  a celkové sumy všech možných šancí na chybu  $\sum_{i=1}^n H_i$  pro všechny možné parametry. Podle této rovnice je pravděpodobnost chybného parametru určena podle vztahu:

$$P(p) = \frac{H(p)}{\sum_{i=1}^n H_i}$$

, kde:

- $P(p)$ : Pravděpodobnost chyby parametru  $p$ .
- $H(p)$ : Numerická hodnota nebo pravděpodobnost chyby parametru  $p$ .
- $\sum_{i=1}^n H_i$ : Celková suma všech hodnot šancí na chybu  $H_i$  pro celkový počet  $n$  parametrů.

Z těchto parametrů je vybrán náhodný počet parametrů omezený hodnotami proměnných *minErrCount* a *maxErrCount* a vybrané parametry jsou uloženy do seznamu chyb, pomocí kterého jsou při následném generování vybrané hodnoty vygenerovány chybně.

Tímto způsobem je možné přesně stanovit množství chyb a jaké chyby se generování chyby nezávisle na pořadí.



### 4.4.3 Generování jednotlivých parametrů

Po vyhodnocení chybných parametrů popsaných v podsekcí 4.4.2 se generují jednotlivé hodnoty parametrů zásilky, a následně jednotlivých dokumentů.

Samotná zásilka obsahuje několik hodnot, které nemohou být chybné, jelikož jsou srovnávány hodnotami v dokumentech. Mezi tyto parametry patří:

- **Id:** identifikátor unikátní v rámci zásilky. Každý identifikátor je vygenerován pomocí systémové třídy *Guid*, která se využívá ke generování unikátních identifikátorů. Tento identifikátor je následně převeden na řetězec a zkrácen na pět prvních znaků. Jelikož je identifikátor využit pouze v rámci zásilky a dokumentů není nutné zaručit jeho originalitu. V případě generování chybného identifikátoru v dokumentech, popsaných v podsekcí 4.4.4, se navíc nově vygenerovaný identifikátor generuje opakovaně dokud není rozdílný od tohoto. Tím se vyloučí jakákoliv možnost náhodného vygenerování stejného identifikátoru.
- **Váha:** Hodnota vyjadřující hmotnost zásilky, generovaná v určitém rozsahu omezeném proměnnými nastavenými pro daný den. Hodnota váhy může být špatně, chyba je však uložena do samostatné proměnné také omezené minimální a maximální hodnotou.
- **Výchozí a cílový stát:** Státy, ve kterých je zásilka odesílána a doručována, jsou vybírány z předem definovaného seznamu možných států, který je podmnožinou všech definovaných států. Tyto seznamy jsou popsány v podsekcí 4.4.5.
- **Jména příjemce a odesílatele:** Jména, která jsou přiřazena příjemci a odesílateli zásilky, jsou náhodně vybrána ze seznamů jmen a příjmení a spojena do jednoho řetězce.

Tyto parametry se generují vždy a následně se tyto hodnoty kopírují i do jednotlivých dokumentů v případě správnosti těchto parametrů v dokumentech, nebo se z jejich hodnoty mohou generovat chybné hodnoty v dokumentech.

U ostatních parametrů se nejdříve kontroluje jejich správnost nahlédnutím do seznamu chyb. Pokud se parametr nalezne v seznamu, je vygenerována chybná hodnota v opačném případě je hodnota buď vygenerována korektní nebo se vezme hodnota ze zásilky. Tyto parametry jsou výchozí a cílový region a datum odeslání.

Regiony fungují na podobném principu jako státy, jejich definice jsou součástí každého státu, využívají tedy stejný systém popsaný v podsekcí 4.4.5.

Při generování jsou buď vybrány korektní regiony náležící danému státu, nebo v případě chyby jsou vybrány buď korektní nebo nekorektní regiony z jiného státu, nebo nekorektní ze správného státu.

Datum odeslání je založeno na vlastní třídě značící datum a jeho hodnota je buď menší nebo shodná se současným datem. V případě chyby je jeho hodnota vyšší než aktuální datum.

Zásilka obsahuje také speciální parametry, jako je obsah zásilky, případný kontraband a holografické pečeti. Tyto parametry se také řídí správností ze seznamu chyb, ale jejich způsob generování a zobrazení ve hře se výrazně liší a budou blíže popsány v následujících sekcích.

### 4.4.4 Dokumenty

Dokumenty jsou součástí systému, který tyto dokumenty generuje zobrazuje a spravuje, každý dokument je potomkem základního dokumentu a každý má funkci na vygenerování a instancování ve hře.

Povolení k průchodu státem	
ID:	Datum platnosti: 29.08.1976
Země původu Utopria	Cílový stát Virelandia
Původní region East Rivoría	Cílový region North Syloría
Odesílatel Benjamin Hewitt	Příjemce John Kelly

Obrázek 4.3: Příklad dokumentu zobrazeného ve hře

Tyto dokumenty se vytvářejí v souladu s nastavenými rozsahy hodnot a případně vygenerovanými chybnými hodnotami, které jsou uloženy v seznamu chyb.

Na rozdíl od zásilky je možnost chyb i v parametrech, které v zásilce chybně být nemohly. Podle požadavků je však možné dokumenty definovat i s parametry bez možnosti chybných hodnot.

### Generování dokumentů

Každý dokument má vlastní definované hodnoty a tedy i vlastní metodu pro generování obsahu, každý dokument tedy dědí metodu *Generate*, která má jako vstupní data data zásilky a třídu *RandomPackageRanges* obsahující hodnoty pro generování hodnot zásilek a dokumentů v daný den.

Generování jednotlivých hodnot tedy závisí na typu hodnoty a dokumentu, zásadně jsou však buď převzaty z hodnot zásilky nebo generovány obdobně jako hodnoty samotné zásilky popsané v sekci 4.4.3, tedy v případě numerické hodnoty se generují náhodně v rozmezí určeném třídou *RandomPackageRanges* a v případě jména či státu nebo regionu se vybírá hodnota ze seznamů předdefinovaných hodnot.

V případě korektního dokumentu se naprostá většina hodnot přebírá z dat zásilky, jelikož pokud se jedná o stejný parametr, musí nutně mít i shodnou hodnotu. V případě chybné hodnoty se hodnota generuje náhodně a srovnává se s hodnotou v zásilce, aby nemohl nastat případ, kdy má hodnota být chybná, ale generovaná hodnota je přesto shodná se správnou hodnotou.

Rozhodnutí o správnosti hodnoty je rozhodnuto před samotným generováním hodnot a to kontrolou seznamu chyb popsaným v sekci 4.4.2. Pokud se hledaná hodnota nachází v seznamu chyb, má vygenerovaná hodnota být chybná.

V současné verzi demonstrační hry je implementován pouze jeden typ dokumentu, tím jsou povolení k přesunu. Tato povolení jsou rozdělena na čtyři druhy dokumentu mezi kterými je vybrán jeden podle původu a destinace zásilky, těmito dokumenty jsou:

- **Povolení k lokálnímu přesunu:** Toto povolení je generováno pokud je jak výchozí tak cílový stát domovský
- **Povolení ke vstupu:** Toto povolení je generováno v případě pouze cílového státu nastaveného za domovský.

- **Povolení k výstupu:** Toto povolení je generováno v případě pouze výchozího státu nastaveného za domovskými
- **Povolení k přesunu:** Toto povolení je generováno pokud ani jeden ze států není domovský.

## Vykreslování dokumentů

Podobně jako generování dokumentů i jejich vykreslování je individuální podle typu dokumentu. To je zajištěno třídami *Drawer*, které všechny dědí od společného předka. Každá varianta této třídy má množinu parametrů určující jednotlivá textová pole odpovídající hodnotám dokumentu.

Jednotlivé varianty těchto tříd jsou v herní scéně umístěny jako komponenta samostatnému objektu, který následně slouží jako reprezentace daného dokumentu. Dále jsou každé třídě *Drawer* poskytnuta jednotlivá textová pole, do kterých se při vykreslení vypisují hodnoty dokumentu. Pokud není dokument třeba je pouze deaktivován a opět se aktivuje pokud nově generovaná zásilka tento dokument obsahuje.

Při zavolání jednotné funkce *DrawDocument* je nejdříve zkontrolován typ vstupního dokumentu a v případě typu shodného s požadovaným je do každého textového pole vepsána hodnota z přiloženého dokumentu. Po vyplnění dat do textových polí je objekt dokumentu přesunut na pozici ve scéně určenou jako vstupní bod pro dokumenty a aktivován, aby byl zobrazen. Výsledný dokument je tedy ve hře vyobrazen jako jednoduchý objekt s textovými poli naplněnými jednotlivými hodnotami parametrů dokumentu. Příklad jednoho takového dokumentu je na obrázku 4.3.

## Manipulace s dokumenty

Každý dokument je po vytvoření umístěn na specifické místo v herní scéně specifikované vývojářem. Po umístění na dokument začne působit simulovaná gravitace. Pokud tedy není dokument v kontaktu s jedním z nastavených colliderů vyznačujících stůl, dokument v každém časovém okamžiku zrychluje o specifikované množství až do dosažení nastavené terminální rychlosti. Po vstupu do kolize se stolní deskou je pohyb zastaven.

Dokument je také možné přetahovat pomocí kurzoru myši, to hráči umožňuje jistou míru organizace a manipulace s dokumenty. Samotný pohyb však není důležitý pro určení správnosti dokumentu. Manipulaci s dokumenty zajišťuje třída *Drawer* přiřazená danému dokumentu a pro všechny dokumenty je tato funkcionalita definovaná v rodičovské třídě.

### 4.4.5 Správa států

V projektu je integrován správce států, který umožňuje definici a modifikaci jednotlivých států, přidávat a odebírat jednotlivé regiony tohoto státu a definovat a modifikovat chybné názvy těchto států a regionů. Dále každý stát obsahuje sklad pečeti, který slouží obdobně k uložení různých typů pečeti a jejich chybných verzí

Tento správce je definován ve scéně hlavního menu a následně označen za perzistentní. Při následném spuštění hry se tedy neodstraní a díky staticky přístupné instanci je přístupný pro každý jiný skript, který by mohl potřebovat přístup k jednotlivým státům.

K těmto datům se přistupuje převážně ze skriptů zásilky a dokumentů, které mají definované ukazatele na státy, jejich názvy a jiná data jsou však v naprosté většině případů čtena z tohoto správce.



Obrázek 4.4: Některé z předloh pro ploché holografické pečeti před převedením na point cloudy

Jako poslední funkce správce státu je možnost nastavení domovského státu, tento stát je pak považován za stát, ve kterém se hráč a jeho kancelář nachází a umožňuje jisté změny v parametrech zásilky na základě jeho hodnoty.

## 4.5 Holografické pečeti

Holografické pečeti jsou jednou z hlavních herních mechanik a jsou jedním ze způsobů podle kterých je možné rozhodnout o bezchybnosti zásilky.

V projektu je využito tří typů pečeti, plochých pečeti, pečeti založených na hloubkových mapách a 3D pečeti, podle způsobu, kterým byly vytvořeny. Ploché pečeti jsou založené na plochem objektu nebo obrázku, ze kterých jsou následně pomocí geometrických uzlů generovány point cloudy využitě v samotné hře. Pečeti založené na hloubkové mapě jsou založeny a tvořeny téměř stejným způsobem, jejich základem jsou místo jednoduchého obrázku hloubková mapa a při jejich generování je využita stejná síť geometrických uzlů s drobnými rozdíly v jejich vstupních hodnot. Zatímco 3D pečeti jsou založeny na klasických 3D objektech a na tvorbu point cloudu je využita jiná síť geometrických uzlů. Tyto point cloudy jsou v Unity zobrazeny pomocí umístění objektu na každý z jejich bodů a zpracován klasickým způsobem.

Úkolem hráče je rozhodnout o správnosti těchto pečeti. U pečeti využívajících hloubkové mapy a 3D pečeti není tak obtížné vidět její pravou podobu. Z tohoto důvodu jsou tyto pečeti buď poškozené, náležící jinému státu, nebo kombinace těchto možností. U plochých pečeti je více využito vlastností holografického displeje a kromě způsobu ověření jako u předešlých je ještě třeba pokusit se nalézt správný úhel pohledu pro zobrazení její původní podoby.

### 4.5.1 Generování pečeti

V této podsekcí je popsáno, jakým způsobem byly vytvořeny jednotlivé verze pečeti. V průběhu vývoje byly nejdříve vytvořeny ploché pečeti a následně společně s návrhem využít point cloud k reprezentaci pečeti, bylo navrženo využít i 3D pečeti.

První verze plochých pečeti byla tvořena plochým objektem, který byl následně v Blenderu rozdělen pomocí jednotlivých hran na samostatné objekty. To vytvořilo množinu objektů tvořících původní pečeť, které bylo možné dále zpracovat před zobrazením na holografickém displeji.

Následující verze začaly využívat geometrické uzly Blenderu pro vytvoření point cloud reprezentace původního objektu.

### Využití geometrických uzlů

Každá pečeť reprezentována point cloudem je vytvořena v samostatném Unity projektu pomocí Blender geometrických uzlů a nástroje BEngine, následně jsou z těchto point cloudů



Obrázek 4.5: Předlohy pro ploché holografické pečeti využívající hloubkovou mapu před převedením na point cloudy

vytvořeny prefabrikáty, exportovány z projektu a importovány do primárního projektu a jako prefabrikáty uloženy do polí správce pečeti pro zobrazení ve hře.

V projektu bylo využito tří různých geometrických uzlů, dva jejichž vstupem je kolekce objektů a jeden jehož vstupem je obrázek, počítá se s odstíny šedi, ale teoreticky je možno využít i barevného.

### Generování z geometrie

Kromě vstupu se tyto uzly dají rozdělit i do dvou kategorií podle způsobu získávání pozic bodů na povrchu objektů. První způsob využívá uzel *Mesh to Points*, který ze vstupní geometrie vytvoří množinu bodů, tyto body mají čtyři možnosti umístění.

- **Vertexy:** Na pozici každého vertexu geometrie je umístěn bod.
- **Hrany:** Bod je umístěn do středu každé hrany geometrie.
- **Stěny:** Bod je umístěn do středu každé stěny geometrie.
- **Rohy:** Bod je umístěn na každý rohový vertex geometrie.

Druhý způsob převodu využívá uzel *Distribute Points on Faces*, který rozdělí body na povrchu geometrie náhodně. Je možnost buď využít čistě náhodnou distribuci, nebo využít distribuci *Poisson disc*, který rozdělí body náhodně, ale vezme v potaz minimální vzdálenost mezi body pro rovnoměrnější rozdělení bodů po povrchu.

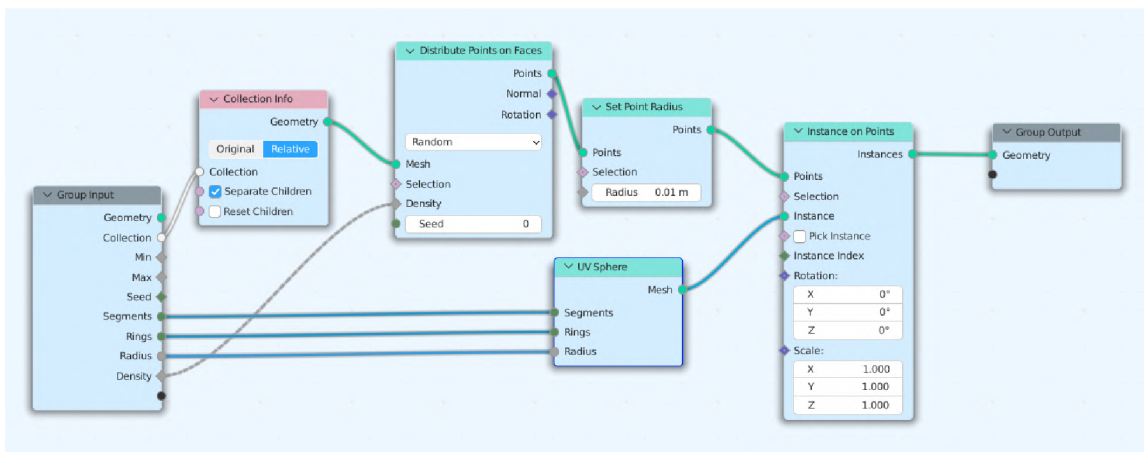
Jelikož druhý způsob využívá náhodnou distribuci bodů po povrchu geometrie, byl vybrán tento způsob s čistě náhodnou distribucí bodů pro tvorbu všech point cloud reprezentací pečeti.

Zbytek generování bodů je v případě generování z kolekce geometrií totožný

1. Z kolekce geometrií se vyberou jednotlivé geometrie.
2. Na povrchu každé geometrie se vytvoří množina bodů.
3. Na každém bodu se vytvoří instance koule jejíž parametry jsou nastavitelné při generování.
4. Jednotlivé instance se realizují a nastaví se jim velikost v rozmezí nastaveném při generování.

### Generování z obrazu

Generování pečeti z obrazu probíhá podobným způsobem, jako generování z geometrie popsané v předchozí podsekci. Stejně jako u generování z geometrie je na povrchu obrazu



Obrázek 4.6: Síť geometrických uzlů využitá pro generování pečetí z geometrie

vytvořena množina bodů pomocí uzlu *Distribute Points on Faces*, následně se však proces liší a začíná být brán v potaz vložený obraz. Předlohy jednotlivých pečetí jsou vidět na obrazech 4.4 a 4.5. První z těchto obrázků obsahuje předlohy plochých pečetí, na druhém obrázku jsou pak tyto obrázky upravené pro generování pečetí z hloubkové mapy. Základ generování bodů po geometrii je stejný, část sítě, která tyto body generuje je na obrázku 4.8

Pro každou pozici bodu je nahlédnuto do obrazu a vybrána barva na této pozici, část sítě, která má na starosti získat pixel z obrazu je na obrázku 4.7. Pokud je hodnota barvy v tomto bodě větší než určený práh, pak je generování bodu zastaveno, v opačném případě je bod posunut na ose Z o určitou vzdálenost určenou hodnotou barvy modifikovanou násobícím vektorem, výpočet posunu je na obrázku 4.9. Pokud je hodnota tohoto vektoru nula je výsledná pečeť opět na jedné rovině, je však převedena na point cloud. Vstupním obrazem může taky být hloubková mapa, po jejíž převedení je vytvořena point cloud reprezentace této mapy obrazu s reprodukcí původní hloubky.

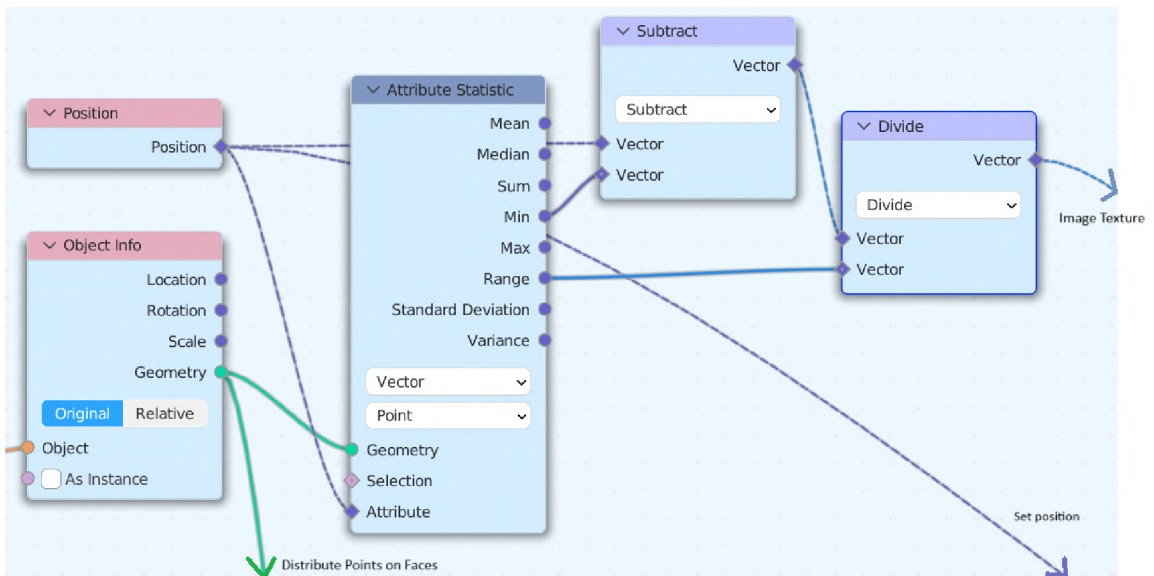
## Generování 3D pečetí

3D pečeť jsou tvořeny point cloud reprezentací kolekce 3D objektů tvořící určitý objekt, tato reprezentace je stejně jako v předešlých podsekcích generována pomocí geometrických uzlů a obdobně se pro jejich generování využívá jak uzel *Mesh to Points* i uzel *Distribute Points on Faces*, Hlavním rozdílem jsou tedy vstupní data. Na obrázku 4.10 Je vidět 3D pečeť společně s její point cloud reprezentací využitě v projektu.

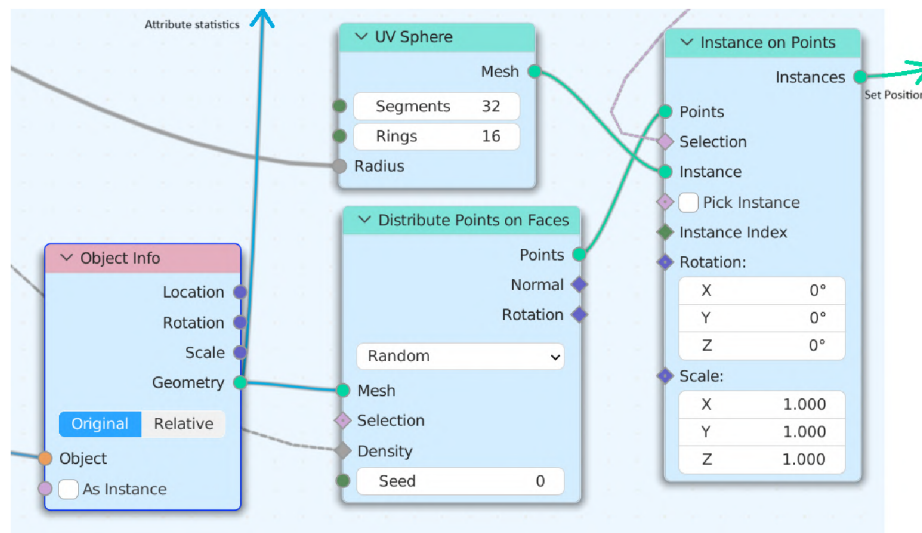
### 4.5.2 Správa pečetí

Po vygenerování pečetí jsou pečeť importovány do hlavního projektu a následně jsou rozděleny do seznamů pečetí v datech států. Každý stát obsahuje tři pole do kterých může být každá pečeť rozdělena. Tyto pole jsou:

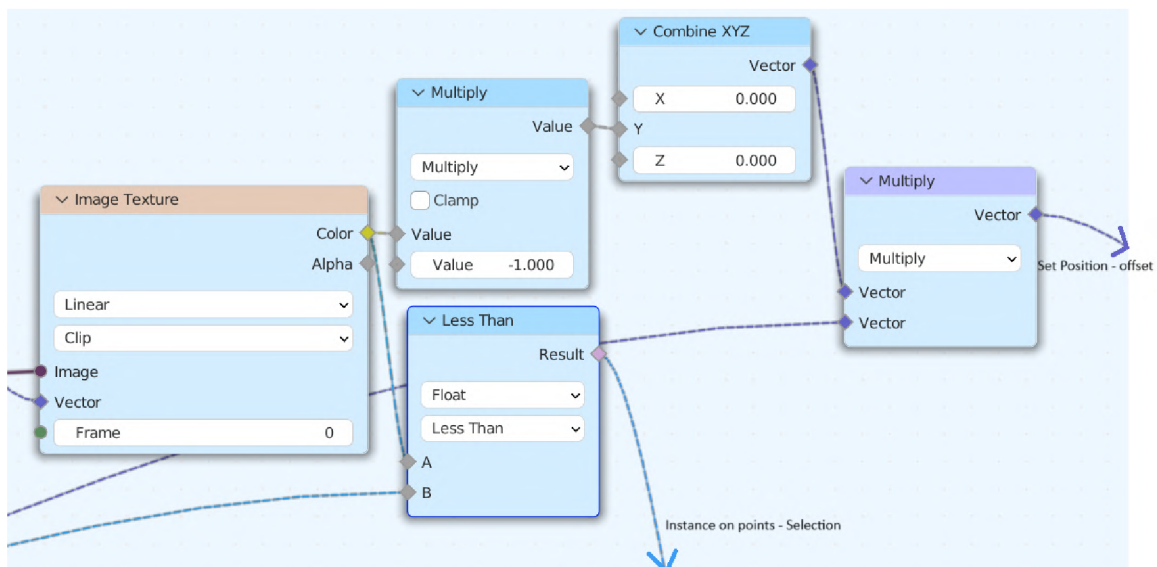
- Ploché pečeť
- Pečeť vytvořené pomocí hloubkové mapy
- 3D pečeť



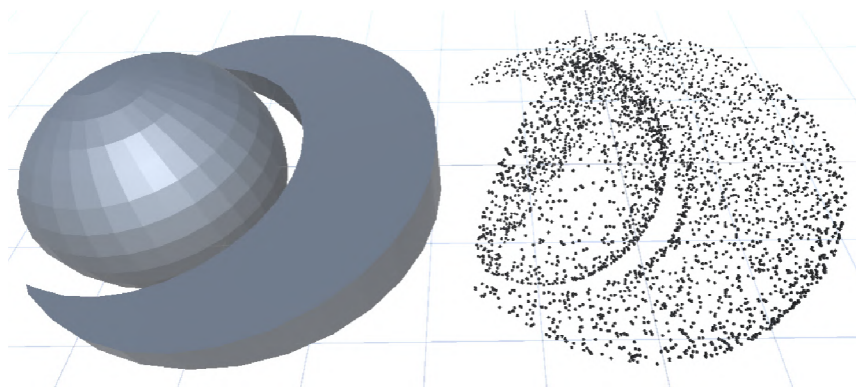
Obrázek 4.7: Část sítě pro generování point cloudu z obrázku, která převádí pozici bodu na pozici pixelu v obrázku.



Obrázek 4.8: Část sítě pro generování point cloudu z obrázku rozdělující body po obrázku.



Obrázek 4.9: Část sítě pro generování point cloudu z obrázku, která převádí hodnotu pixelu obrázku na posun.



Obrázek 4.10: Jedna z objektových pečeti v podobě jak původního objektu tak její point cloud reprezentace.



Výběr a zobrazení pečeti probíhá v několika krocích:

1. Správce pečeti vyhodnotí, zda-li má vygenerovaná pečeť být správná nebo poškozená.
2. Správce pečeti vyhodnotí, jaký typ pečeti se má použít na základě hodnoty v datech dne a případné náhodné hodnoty.
3. Správce vybere náhodnou pečeť z příslušných polí. Tato pečeť je následně instancována, kde předkem této instance je přednastavený objekt se správnou pozicí a velikostí pro zobrazení této pečeti.
4. Vybraná pečeť je zpracována funkcí *SealPreprocess()*, která každému bodu pečeti nastaví objekt reprezentující tento bod, materiál a každému bodu také normalizuje pozici pro zajištění konzistence velikosti mezi pečeti a tím i konzistenci v následné úpravě.
5. Pečeť je v případě potřeby upravena na základě jejího typu, nastavené hodnoty v datech dne a případné náhodné hodnoty.

### Úprava plochých pečeti

V případě že vybraným typem pečeti je plochá pečeť, je před jejím zobrazením každý bod upraven, aby nebylo řešení pečeti triviální. Jsou implementovány dva druhy úpravy, úprava rozložením a úprava maskováním.

Úprava rozložením přesune každý bod po ose Z na náhodné místo omezené do prostoru koule vyhrazené přednastaveným poloměrem.

Velikost posunu je vyhodnocena pomocí náhodné hodnoty omezené na úsečku procházející aktuálním bodem, rovnoběžná s osou X, jejíž oba body se dotýkají povrchu koule, jelikož se bod musí nacházet ve středu této úsečky jsou omezeními náhodné hodnoty kladná a záporná vzdálenost od středu k bodům této úsečky.

Jelikož je při výpočtu těchto vzdáleností dostupný dostatek informací, stačí pro výpočet Pythagorova věta, kde přepona je poloměr koule  $r$  a jedna přepona je vzdálenost bodu od středu koule, kde  $\mathbf{x}$  a  $\mathbf{y}$  určují souřadnice aktuálního bodu. Druhá přepona je hledaná vzdálenost  $h$ . Jelikož je tato vzdálenost kolmá k rovině na které se nachází každý bod pečeti, vznikne pravoúhlý trojúhelník. Výsledná rovnice je tedy

$$h = \sqrt{\left| r^2 - \sqrt{x^2 + y^2} \right|^2}$$

Po posunu těchto bodů je celá pečeť otočená o náhodnou hodnotu mezi pozitivními a zápornými patnácti stupni pro změnu úhlu pohledu, ze kterého je možné vidět původní podobu pečeti. Tyto omezující hodnoty jsou určeny vlastnostmi holografického displeje, který více než tyto úhly nedokáže zobrazit.

Úprava maskováním je způsob zahalení původní podoby pečeti na holografickém displeji. Zahalení je dosaženo nastavením některé z vybraných vrstev každému z bodů pečeti a následným nastavením zobrazovacích masek jednotlivým sektorům displeje. Tímto způsobem je možné zobrazit na každém ze sektorů pouze část pečeti, aby však byla pečeť alespoň z nějakého úhlu pozorovatelná, je nutné nastavit alespoň jeden sektor na speciální masku, která zobrazí všechny vybrané vrstvy.

V ideálním případě by měl každý pohled displeje vlastní vrstvu, v tomto případě jsou omezením vlastnostmi Unity, které umožňuje maximálně 32 zobrazovacích vrstev, jelikož

vybraný holografický displej má 45 pohledů je tato možnost prakticky nemožná. Ve výsledném řešení se využívá dvacet vrstev a tedy dvacet sektorů. Tímto způsobem nejsou všechny zobrazovací vrstvy využity výhradně pro tuto mechaniku a zároveň je zachované dostatečné rozlišení pro nutnost sledovat displej z více úhlů pohledu, který zároveň není triviální.

O rozdělení bodů do masek se stará opět správce pečeti, ten projde všechny přímé potomky vybrané pečeti a přidělí jim jednu náhodnou vrstvu z přednastaveného seznamu vrstev. Po rozdělení jednotlivých částí pečeti do vrstev provede vytvoření zobrazovacích masek pro sektory displeje, tyto masky jsou tvořeny náhodným počtem masek ze stejného seznamu, které jsou přes logický OR spojeny do jedné. Počet masek je omezený minimem a maximem nastaveném opět v datech jednotlivých dnů. Tyto masky jsou uloženy do seznamu a po dokončení jejich definice jsou předány skriptu holografického displeje, který byl upraven aby tuto funkcionalitu podporoval.

Tímto způsobem je dosaženo obrazu pečeti, který je na holografickém displeji v celku pozorovatelný pouze z jednoho úhlu pohledu.

Další možností je kombinace těchto dvou mechanik, kde je každý bod pečeti nejenom přesunut, ale také přiřazen do náhodné zobrazovací vrstvy. Tímto způsobem je možné ještě více ztížit odhalení původní podoby pečeti jinde než na vybraném úhlu pohledu.

## Úprava chybné pečeti

Při vyhodnocení pečeti jako chybné je několik možností, jakým způsobem pečeť poškodit a zbavit její validity.

Nejjednodušší možností je vybrat pečeť stejného typu, ale ze seznamu chybných pečeti. Tímto způsobem dále stačí s pečeti zacházet jako s korektní a upravit jí standardním způsobem.

Další možností je vybrat pečeť ze seznamu pečeti jiného státu. V tomto případě je pak možno vybrat jak ze seznamu korektních tak i nekorektních pečeti, jelikož samotná chyba v pečeti není její validita z pohledu celistvosti, ale z hlediska náležitosti špatnému státu.

Dále je možné pečeť poškodit odstraněním určitého množství bodů a tedy znemožnit pozorovat pečeť v její celistvosti. Tato možnost se využívá v kombinaci s ostatními způsoby, jelikož není možné zaručit dostatečné poškození, do stavu ze kterého hráč nebude schopen odhadnout její původní tvar.

Body jsou odstraněny smyčkou, která opakovaně odstraňuje náhodného přímého potomka aktuální pečeti do dosažení počtu odstraněných bodů, který odpovídá procentuální hodnotě všech bodů zaokrouhlené na celé číslo, která se nachází v datech aktuálního dne.

Poslední možností je poškodit pečeť během její úpravy postupy popsány výše, kde v případě využití maskovací metody není nastaven žádný sektor, který by umožňoval zobrazit všechny díly pečeti a v případě využití metody posunu se každý bod pečeti kromě osy Z posouvá i po ostatních osách o náhodnou vzdálenost určenou vzdáleností buď od středu pečeti, nebo vzdáleností od poloměru pomyslné koule. Z těchto vzdáleností je vybrána menší hodnota, jako pokus o omezení bodů pouze v rámci této pomyslné koule.

## Modifikace kódu holografické kamery

Aby bylo možné využít maskovací metodu úpravy pečeti, bylo nutné upravit kód holografické kamery. Tato kamera bez úpravy vytváří jednotlivé quilty pomocí jedné kamery a smyčka, která kameru posune na příslušné místo a následně snímek kamery zakomponuje na určité místo ve vytvářeném quiltu.

Do tohoto kódu bylo zakomponován seznam masek, kde jednotlivé masky jsou určeny pro jednotlivé sektory displeje, tyto sektory jsou určeny počtem jednotlivých snímků, ze kterých je tvořen výsledný quilt a počtu masek v seznamu. Důležité změny v kódu holografické kamery byly provedeny v souboru *MultiViewRendering.cs* mezi řádky 225 a 240.

## 4.6 Kontrola obsahu zásilek

Obsah zásilky je další z mnoha možností, které mohou být testovány pro správnost zásilky, navíc se nabízí jako ideální možnost využití holografického displeje.

Obsah je tvořený množinou objektů umístěných do reprezentace zásilky, uspořádaných do dvourozměrné mřížky. Pod normálním obsahem se také může nacházet kontraband. cílem hráče je v takovém případě kontraband odhalit a označit zásilku za chybnou. Kontraband nemusí být viditelný z přímého úhlu pohledu a je tedy nutné prohlédnout zorné pole displeje pro jeho odhalení.

Na chyby v obsahu zásilky může poukazovat váha zásilky. Ta v případě rozdílu s dokumenty může vyznačovat přítomnost kontrabandu nebo přebytečného obsahu. Je zde ale stále možnost pouze chyby v dokumentaci, která není důvodem k zastavení doručení zásilky.

### 4.6.1 Generování obsahu

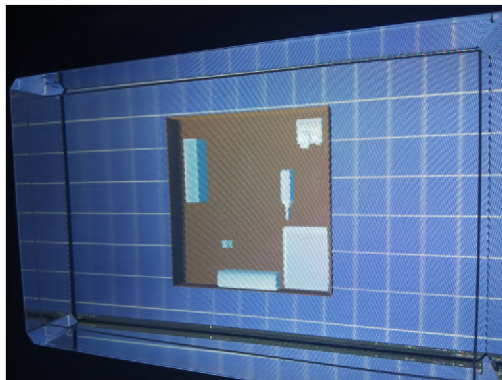
Obsah zásilky je generován pomocí náhodného vkládání náhodných přednastavených objektů do dvourozměrného pole. Tyto objekty jsou vkládány dokud není dosaženo požadovaného množství objektů v zásilce, nebo není dosaženo nastaveného množství neúspěšných pokusů o vložení objektu do pole.

Samotné objekty jsou definovány podobně jako tvary v tetrisu, mají tedy pole s relativními pozicemi a při pokusu vložení tohoto objektu jsou kromě pozice určené souřadnicemi objektu testovány i všechny pozice určené relativními pozicemi tohoto objektu. Podobně jako u tvarů v tetrisu je také možnost celé toto pole relativních pozic opakovaně otočit o 90 stupňů kolem původní pozice. Počet těchto otočení je zaznamenán pro následné vložení objektu, reprezentujícího tento tvar do zásilky. Tyto tvary jsou definovány programátorem v třídě *ShapeManager* společně s hodnotami určenými ke vkládání reprezentací do zásilek.

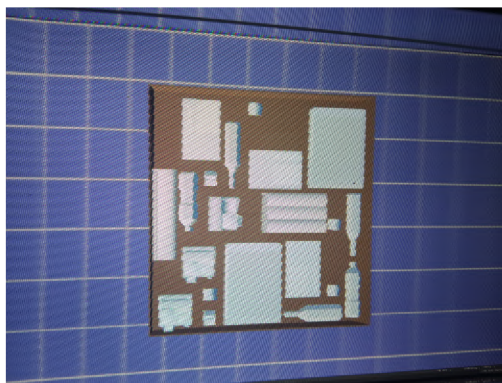
Požadované množství objektů je náhodná celočíselná hodnota omezena v rozmezí nastaveném v datech aktuálního dne. Při nastavení přepínače *LayoutFill* v datech dne je tato hodnota ignorována a obsah zásilky je generován až do dosažení maximálního množství neúspěšných pokusů. Na obrázku 4.11 je příklad zásilky s nízkými hodnotami určujícími množství objektů v zásilce. Na obrázku 4.12 je zásilka se středními hodnotami těchto parametrů a na obrázku 4.13 je v parametrech nastaven přepínač pro úplné naplnění zásilky.

O generování obsahu zásilek se stará třída *Packagelayout*, která obsahuje jak funkce na generování obsahu tak i výsledné objekty a případný kontraband, ta po zavolání funkce *GenerateRandomShapes* vytvoří nové dvourozměrné pole objektů v zásilce, prázdný seznam objektů a počet objektů ke generování a začne generovat tvary v souladu se vstupními parametry, nastavujícími podmínky zmíněné výše. Dvourozměrné pole je určeno pouze pro testování volných pozic při jeho plnění. Je složeno z Booleových hodnot označující, zdali se na vybrané pozici nachází část nějakého objektu, nebo je tato pozice volná. Vložení tvaru do zásilky probíhá v několika krocích:

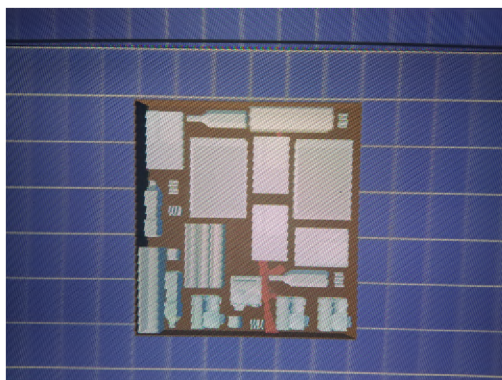
1. Nejdříve se vybere náhodný tvar ze seznamu předdefinovaných tvarů a vytvoří se jeho kopie pro vložení do pole.



Obrázek 4.11: Příklad zásilky s nízkým množstvím objektů.



Obrázek 4.12: Příklad zásilky se středním množstvím objektů.



Obrázek 4.13: Příklad zásilky plné objektů.

2. Vygeneruje se náhodné počáteční hodnoty posunu a otočení tvaru pro zlepšení náhodnosti umístění.
3. Pomocí smyčky se script snaží opakovaně ověřit dostupnost pozic v poli obsazených pozic na pozicích určených souřadnicemi objektu a jeho relativních pozic. Pokud je pokus neúspěšný jsou vykoušeny všechny možné rotace objektu a pokud je vložení stále neúspěšné je objekt posunut v poli o jednu pozici dále. Pokud je vložení neúspěšné v jakékoliv pozici a rotaci je objekt zahozen a pokud ještě nebyl dosažen maximální počet pokusů, je generování opakováno od prvního kroku.

V případě že je vložení do pole úspěšné jsou hodnoty v poli volných pozic na místech určených pozicí objektu a relativními pozicemi jeho součástí nastaveny na hodnotu *true* naznačující obsazenost této pozice v zásilce. Následně je vkládaný objekt přidán do pole ostatních objektů již vložených do zásilky a hodnota neúspěšných pokusů je nastavena zpět na nulovou hodnotu.

#### 4.6.2 Vykreslování obsahu zásilky

Po vygenerování obsahu zásilky popsané v předchozí sekci je do reprezentace zásilky ve hře vložena reprezentace každého vloženého objektu.

Vykreslování zajišťuje správce zásilek, Jelikož *Packagelayout* není potomek třídy *MonoBehaviour* a nemá tedy možnost provádět potřebné změny přímo ve hře. Správce zásilek tedy po vygenerování nové zásilky, včetně vygenerování jejího obsahu a obnovení ostatních dat v zásilce, začne vkládat reprezentace objektů do této zásilky.

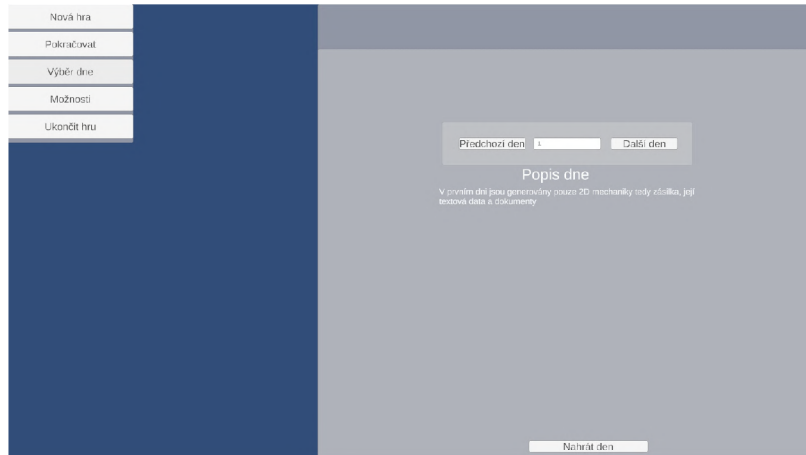
Prvním krokem je odstranění všech případných objektů, které v zásilce zůstanou po jejím odevzdání v předešlém cyklu, Následně je pro každý objekt zavolána funkce *PlaceRepresentation* která obstarává samotné vložení objektu do zásilky. Tato funkce ke vkládání využívá jak data objektu, tak i hodnoty uložené ve správci tvarů. Hodnoty ve správci tvarů slouží jako modifikátory pozice, rotace a velikosti objektu, nutné pro korektní vložení reprezentace do zásilky způsobem, který odpovídá umístění objektu v poli obsazených pozic a zamezí překrývání s ostatními reprezentacemi.

Při vykreslování každé reprezentace objektu se nejprve vytvoří prázdný objekt s rodičovským objektem nastaveným na objekt značící oblast, kam se mají vykreslit jednotlivé objekty. do tohoto nového objektu se následně vytvoří nová instance reprezentačního objektu nastaveného v každém právě vykreslovaném objektu, s rodičovským objektem nastaveným na výše zmíněný prázdný objekt. Po vytvoření této instance je nadřazený objekt opakovaně otočen o 90 stupňů podle počtu otočení nastavených v datech objektu a následně je posunut na správnou pozici. Pozice objektu je vypočtena ze souřadnic objektu, kde každá souřadnice pozice je vynásobena vektorem, určujícím směr a velikost posunu po této ose. Nakonec je objektu nastavena pozice hloubky určené programátorem a nastavené ve správci tvarů.

#### 4.6.3 Kontraband

Generování kontrabandu je založeno opět na přítomnosti konkrétní chyby v seznamu chyb popsaném v sekci 4.4.2. Pokud je hledaný záznam nalezen je do zásilky přidán kontraband.

Generování kontrabandu je téměř totožné s generováním obsahu zásilky popsaném v předchozích podsekcích. Hlavním rozdílem je jiný seznam před definovaných objektů, specifických pro kontraband, využití vlastního pole volných pozic, jiných objektů a při vykreslování těchto objektů je vybrán rozdílný parametr *Z* specifický pro kontraband. Tento



Obrázek 4.14: Snímek hlavního menu s právě otevřeným panelem pro nahraní hry.

parametr je zamýšlený jako způsob, pomocí kterého je kontraband vykreslen pod normální obsah. Je zde možnost vykreslit kontraband v jiných výškách včetně nad povrchem normálního obsah. Tento způsob zobrazení však trivializuje pro hráče problém.

## 4.7 Hlavní menu

Hlavní menu je omezeno na nejnútnejší ovládací prvky a některé nedokončené prvky a je vyhrazeno do vlastní scény. Hlavní menu umožňuje hráči provést několik možných akcí.

1. **Nová hra:** Výběr této možnosti vymaže aktuální uložená data, vytvoří nový soubor s uloženými daty a spustí první den ze seznamu dní.
2. **Pokračovat:** Tato možnost vezme uložená data, pokud žádná neexistují vytvoří nové, a nahraje následující den ze seznamu dní od posledního dokončeného dne v datech.
3. **Nahrát:** Tato možnost zavře všechny jiné otevřené panely a zobrazí panel s možnostmi nahraní libovolného dne ze seznamu dní, který již hráč dokončil, nebo následující od posledního dokončeného. každý den obsahuje krátký popis.
4. **Možnosti:** Tato možnost zobrazí panel s nastavením a uzavře jiné otevřené panely. V současnosti nejsou v tomto panelu žádné možnosti k nastavení.
5. **Konec:** Tato možnost je zamýšlena pro ukončení hry.

Obrázek 4.14 obsahuje snímek z hlavního menu demonstrační hry. V menu je také právě otevřený panel s možnostmi nahraní hry.

## Kapitola 5

# Testování

V této kapitole je popsán způsob, průběh a výsledky testování jednotlivých herních mechanik uživateli. Data byla získána v přítomnosti autora a pomocí dotazníku vytvořeného pomocí Google Forms sepsána pro následné vyhodnocení.

### 5.1 Testeři

Testování se zúčastnili tři testeři. Žádný neměl zkušenosti s holografickým displejem a dva měli zkušenosti se hrou, která sloužila jako inspirace pro tuto demonstraci.

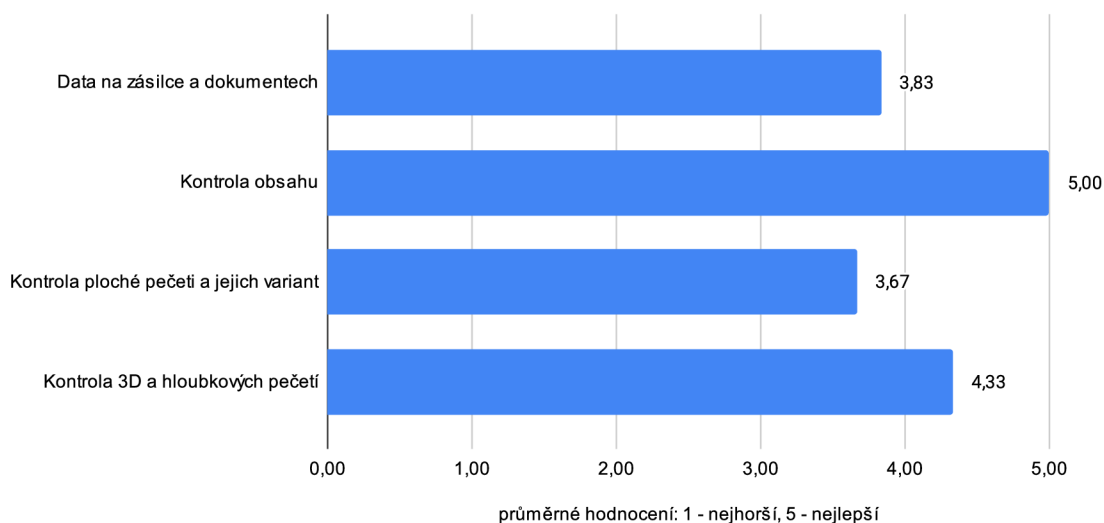
### 5.2 Průběh testování

Před začátkem každého testu byl daný tester seznámen se hrou, jejím ovládním a mechanikami, jelikož je hra relativně komplexní, neobsahuje žádnou formu tutoriálu a strohé množství vysvětlivek, bylo testerům umožněno podávat otázky k ovládním a kontrole dat.

Následně byli testerů spuštěny jednotlivé dny, které byly zaměřeny na specifické herní mechaniky. Každý tester měl možnost vyplňovat výstupní formulář po každém dni postupně, nebo celý po dokončení posledního dne, který kombinoval testované mechaniky do jednoho celku.

V průběhu testů bylo objeveno několik chyb, většina z nich však byla relativně nepodstatná a bylo možné je během ne dlouhé doby opravit. Příkladem může být nefunkční element s aktuálním datem, chybějící informace o pořadí aktuálního dne a těžko čitelné informace na zásilkách a dokumentech. Jednou z důležitých objevených chyb byla chyba, která nastala po neočekávaném přesunutí zásilky přímo na detektor namísto položení do příslušné zóny. Tímto způsobem se znemožnila další manipulace se zásilkou jelikož mimo pracovní plochu není možné zásilku uchopit. Tato chyba byla opravena implementací hranic, na kterých se pohyb zásilky zastavil a znemožnil tak opuštění pracovní plochy.

První část dotazníku se zabývala hodnocením jednotlivých mechanik. Výsledky těchto odpovědí jsou vyobrazeny v grafu na obrázku 5.1 a naznačují relativně vysokou spokojenost s jednotlivými mechanikami. Zajímavé na výsledcích může být například vysoká spokojenost s hledáním kontrabandu v zásilce. Dále je zajímavý rozdíl v hodnocení plochých pečetí a 3D a hloubkových pečetí. Někteří testeři naznačili větší spokojenost se snadno pozorovatelnými, ale zajímavě zobrazenými, pečetěmi, než s plochými pečetěmi, upravenými pro pozorování z určitého úhlu.



Obrázek 5.1: Výsledky hodnocení jednotlivých herních mechanik.

V druhé části dotazníku byli testerům podány otázky o jejich zážitku při hře. Přesněji byl hodnocen požitek ze hry a přirozenost užívání holografického displeje. Přirozeností je myšleno jak pohodlné či nepříjemné pro ně bylo používání holografického displeje při hře.

Žádný z testovaných nenaznačil žádné obtíže s využitím displeje a všichni byli schopní pracovat s mechanikami pro něho určenými bez větších obtíží. Poslední z otázek bylo hodnocení umístění displeje. Ten byl při testování umístěn na pravé straně klasického displeje a ve srovnatelné výšce s ním. Jeden z testovaných této otázce dal neutrální hodnocení, ostatní odpověděli pozitivně.

### 5.3 Výsledky testování

Výsledky testování naznačují, že tato konfigurace displejů je při hře tohoto druhu hry vhodná, a to i pro hráče, kteří mají omezené zkušenosti s hraním her. Tomuto faktu napomáhá vybraný způsob využití displeje. Který umožňuje využívat displej pouze pro mechaniky pro tento typ displeje vhodný.



## Kapitola 6

# Závěr

Cílem této práce bylo vytvořit hru, využívající jak klasický, tak holografický displej Looking Glass a vytvořit jak mechaniky dedikované pro holografický displej tak mechaniky určené pro displej klasický. Cílem nebylo nutně vytvořit nové mechaniky, ale využít jejich kombinaci k vytvoření hry, ve které oba displeje spolupracují. Pro tento projekt bylo třeba se seznámit s geometrickými uzly Blenderu, jejich využití pro tvorbu point cloudu, seznámit se s prostředím Unity, jeho nástroji a funkcemi a alespoň částečně kódem holografické kamery pro její modifikaci.

Výsledkem této práce je demonstrační hra, obsahující sedm dní reprezentujících herní úroveň. Každý den obsahuje kombinaci různých mechanik s různými hodnotami reprezentujícími možné využití jednotlivých mechanik v plnohodnotné hře. Tyto mechaniky jsou zaměřeny na využití buď normálního nebo holografického displeje a jejich propojení. Součástí projektu je také možnost definice dalších dní a integrace dalších mechanik, některé v různých stádiích implementace.

# Literatura

- [1] JONE. *How Big is the Video Game Industry Compared to Other Industries?* [online]. Červen 2022 [cit. 2023-9-1]. Dostupné z: <https://thoughtco.net/how-big-is-the-video-game-industry-compared-to-other-industries/>.
- [2] KAMPS, H. J. *At 65 inches and 8K, the world's largest holographic display is here* [online]. Červen 2022 [cit. 2023-9-1]. Dostupné z: <https://techcrunch.com/2022/06/07/looking-glass-factory-large-holographic-display/>.
- [3] SHAWN FRAYNETUNG, P. L. *Superstereoscopic display with enhanced off-angle separation* [online]. Duben 2018 [cit. 2024-5-1]. Dostupné z: <https://patents.google.com/patent/US10298921B1/en>.
- [4] TOMÁŠ CHLUBNA, T. M. . P. Z. How Capturing Camera Trajectory Distortion Affects User Experience on Looking Glass 3D Display. *Multimedia Tools and Applications*. 2023, [cit. 2024-5-1]. Dostupné z: <https://doi.org/10.1007/s11042-023-16350-5>.
- [5] SCHILD, J., LAVIOLA, J. a MASUCH, M. Understanding user experience in stereoscopic 3D games. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2012, s. 89–98. CHI '12. DOI: 10.1145/2207676.2207690. ISBN 9781450310154. Dostupné z: <https://doi.org/10.1145/2207676.2207690>.
- [6] COMPANY, L. G. *Depth Maps: How Software Encodes 3D Space* [online]. Leden 2023 [cit. 2024-5-2]. Dostupné z: <https://lookingglassfactory.com/blog/depth-map>.
- [7] MIFTH. *Bengine For Unity* [online]. 2022 [cit. 2024-5-2]. Dostupné z: <https://blendermarket.com/products/bengine-for-unity>.
- [8] SOFTWARE, P. *What Are the Best Game Engines?* [online]. Leden 2023 [cit. 2024-5-2]. Dostupné z: <https://www.perforce.com/blog/vcs/most-popular-game-engines>.

# Příloha A

## Testovací dotazník

Každý z testerů byl po dokončení testů požádán o vyplnění dotazníku určeného pro sběr dat k následnému vyhodnocení. Tento dotazník byl definován pomocí google forms a kromě jména obsahoval dvě skupiny otázek. První skupina byla určena k hodnocení jednotlivých implementovaných herních mechanik a jejich kombinace. V druhé části byly otázky směřované na užitek ze hry a přirozenost využití holografického displeje ve spojení s klasickým. Tyto otázky byly:

1. Hodnocení kontroly dat na záznamech
2. Hodnocení kontroly dokumentů
3. Hodnocení kontroly obsahu
4. Hodnocení kontroly ploché pečeti a jejich variant
5. Hodnocení kontroly 3D a hloubkových pečeti
6. Hodnocení zábavy při řešení
7. Hodnocení přirozenosti využívání holografického displeje

První dvě otázky byly při shrnutí dat spojeny do jedné jelikož jsou téměř shodné. Každý testovaný měl také možnost přidat vlastní poznámky či nalezené problémy. Žádný testovaný však tuto možnost nevyužil. Důvodem byl fakt, že při testování byly tyto poznámky rovnou sepsány pro následné opravy.