

1 Tvorba aplikácie pre virtuálnu realitu

Cieľom aplikácie pre virtuálnu realitu, ktorá využíva prvky herného žánru, je prezentovať používateľovi situácie, ktoré v ňom vyvolávajú fóbické reakcie. Aplikácia slúži ako prototyp možnej terapie na potlačenie strachu z fóbického prostredia, v ktorom je používateľ nútený splniť zadanú úlohu.

V tejto kapitole sa budeme venovať tvorbe VR aplikácie, ktorá by mala slúžiť na terapiu pre pacientov trpiacich hypsofóbiou a hylofóbiou. Súčasťou práce je predstavenie vývojového prostredia, samotná tvorba aplikácie a v závere príde jej finalizácia.

1.1 Špecifikácia aplikácie

Popularita virtuálnej reality v zdravotnej starostlivosti napreduje veľkou rýchlosťou a prichádzajú s ňou aj čoraz kvalitnejšie aplikácie. Táto skutočnosť bola hlavným motívom tvorby aplikácie, a preto sme sa ju rozhodli navrhnuť tak, aby bola pre používateľa atraktívna a zároveň adrenalínová s cieľom potlačiť jeho fóbické reakcie. Keďže VR je schopná veľmi vierohodne a efektívne simulovať rôzne fóbické prostredia, rozhodli sme sa aplikáciu rozdeliť na dve časti – jednu na terapiu pacientov trpiacich strachom z výšky (hypsofóbia) a druhú na simuláciu tmavého lesa (na liečbu hylofóbie).

Aplikáciu sme vytvárali v operačnom systéme Windows 10 pomocou herného nástroja Unity 3D, ktorý si neskôr ešte detailnejšie predstavíme. Dôležitá informácia je, že finálny produkt nášho snaženia bol vytvorený iba pre operačný systém Android. Je to z jednoduchého dôvodu a tým je zariadenie Oculus Quest, ktorým disponuje KI FPV UMB a pre ktorý sa aplikácia vyvíjala. Vďaka Unity 3D však existuje možnosť transformácie finálneho produktu aj do iných platforiem. V aplikácii bolo vytvorených viacero prostredí, nielen samotný les a imitácia výšky.

Pre ilustráciu si zoberieme hlavné menu aplikácie. Tvorí ho „miestnosť“, v ktorej sa ocitneme po spustení aplikácie. Môžeme sa v nej otáčať, pohybovať a pomocou ovládačov aktivovať tlačidlá nachádzajúce sa v tejto „miestnosti“, ktoré nás budú navigovať v ovládaní aplikácie, a teda aj vo výbere jedného z našich dvoch hlavných prostredí. Súčasťou hlavného menu je aj návod, ako sa pohybovať v konkrétnom prostredí a byť v interakcii s predmetmi, ktoré sú súčasťou splnenia úloh.

Ďalšie prostredia sú už uvedený les (hylofóbia) a miesto, v ktorom si používateľ otestuje svoj strach z výšok (hypsofóbia). Vďaka implementovaným úlohám, ktoré sa musia v týchto prostrediach splniť, nám pribudnú k hlavnému menu ešte dve prostredia. Jedno reprezentuje úspešné splnenie úlohy a druhé nesplnenie úlohy. V posledných dvoch prípadoch sa používateľ ocitne v „miestnosti“, ktorá mu oznámi či úlohu splnil alebo nie a tiež bude obsahovať tlačidlá, pomocou ktorých si úlohu bude môcť zopakovať, vypnúť aplikáciu alebo sa vrátiť do hlavného menu. Z vyššie uvedeného vyplýva, že aplikácia obsahuje 5 prostredí.

Hylofóbia je časť aplikácie, ktorá simuluje prostredie tmavého lesa, v ktorom sme vytvorili veľké množstvo stromov, lesné cestičky, zvuky patriace k takémuto prostrediu, horiacu priepasť, ktorá je súčasťou splnenia úlohy, 3 kocky (tiež sú súčasťou úlohy). Po obvode celého prostredia sa vytvorilo vysoké pohorie, ktoré tvorí hranicu a používateľovi sa nebude dať táto hranica prekročiť. Spomínané kocky sú rozmiestnené vo vnútri lesa a úlohou používateľa je ich nájsť, vziať a hodiť do horiacej priepasti, čím uhasí oheň. Úloha je splnená, keď sa podarí uhasiť oheň na dne priepasti. Nakoľko prostredie lesa je rozsiahle, tak pozícia kociek bola navrhnutá tak, aby bola prvá kocka pri pozornom sledovaní okolia viditeľná zo štartovacej pozície používateľa. Akonáhle ju používateľ spozoruje a premiestni sa k nej, tak z tej pozície bude viditeľná druhá kocka. Tento istý postup platí aj pre tretiu kocku. Používateľova štartovacia pozícia sa nachádza pri horiacej priepasti, pričom si musí uvedomiť, že tak ako v reálnom svete aj v aplikačnom prostredí, má dve ruky a tak dokáže vziať naraz len dve kocky. Preto musí zvoliť správny postup, ako ich všetky tri dopraviť do cieľa bez toho, aby zablúdil v tmavom lese. Malou pomôckou je, že kocky majú v sebe nastavené bodové svetlo, a tak sú ľahko identifikovateľné v danom prostredí. Terén je však veľmi členitý a ľahko sa môže stať, že sa kocky stratia z dohľadu. Pre pohyb v tomto prostredí sme zvolili skript „swinging arms“, ktorý nás posúva v smere nášho pohľadu na základe pohybu rúk, v ktorých používateľ drží ovládače.

Hypsofóbia je prostredie v ktorom sa používateľ ocitne na vysokom brale s vežou, na ktorej vrchole je umiestnená doska. Aj v tomto prípade hrajú kocky hlavnú úlohu. Nachádzajú sa na konci dosky a sú hneď 4. Štartovacia pozícia používateľa je na začiatku dosky ešte v útrobach veže, a tak nie je od začiatku zrejmé, v akej výške sa nachádzame. Používateľovi to bude jasné hneď po prvých krokoch, nakoľko úlohou v tomto prípade je vziať všetky kocky z dosky a priniesť ich späť na určené miesto. Počet kociek nie je náhodný a číslo 4 je udané z dôvodu, vyššej náročnosti pre používateľa, ktorý sa ešte raz musí vrátiť,

aby všetky 4 kocky dopravil do cieľa. Musí si uvedomiť, že ak mu spadne čo i len jedna kocka, alebo on sám prekročí okraj dosky, spadne a úloha nebude splnená, tak aplikácia mu to oznámi tým, že zmení prostredie a ocitne sa v „miestnosti“- nesplnená úloha. Okolie je obohatené ešte o priepasť pod bralom, zvukmi vetra, pohorím, ktoré opäť tvorí hranicu prostredia a vodou na dne priepasti. Pre pohyb v tomto prostredí sa musí používateľ pohybovať aj v reálnom svete, aby sme dosiahli dokonalý pohlcujúci zážitok, kvôli čomu musí byť aj správne nastavená miestnosť v zariadení, aby nedošlo ku kolízii s predmetmi v miestnosti.

Po finalizácii aplikácie otestujeme funkčnosť každého prostredia vybranými respondentami, pričom budeme skúmať ich závraty. Overíme si aj samotné ovládanie, zistíme všetky prípadné chyby a nedostatky, na odstránení ktorých budeme následne pracovať.

1.2 Vývojové prostredie – Unity 3D

Herný nástroj Unity 3D je jedným z najpoužívanejších na svete a je vyvíjaný spoločnosťou Unity Technologies. Prvá verzia bola vytvorená v roku 2005 a v dnešnej dobe podporuje viac ako 26 platforiem vrátane VR a AR. Oproti ostatným nástrojom je jednoduchší a rýchlejší. Keďže Unity má obrovskú komunitu a dobre spracovanú dokumentáciu, tak skriptovanie v jazyku C# je dobre zvládnuteľné. K dispozícii máme aj Asset Store čo je obchod s rôznymi balíčkami, ktoré uľahčujú prácu v editore. Niektoré sú spoplatnené a iné zas voľne dostupné. Pri vytváraní nášho prostredia sme tiež využili niektoré voľne dostupné balíčky.

Pre prácu v Unity 3D je potrebné mať nainštalovaný nástroj Unity Hub, ktorý má na starosti správu projektov a inštalácií rôznych modulov slúžiacich na rozšírenie kompatibility pre viaceré platformy. Nový projekt si v ňom vieme ľahko vytvoriť. Kliknutím na tlačidlo „New“ nám ponúkne výber z rôznych šablón potrebných pre projekt a spýta sa nás, kde na disku sa majú výsledky našej práce uložiť. Po týchto nastaveniach stačí kliknúť na tlačidlo „Create“ a vytvorí sa nám prostredie v ktorom môžeme začať pracovať – **Unity editor**.

1.2.1 Unity editor

Pomocou Unity editora môžeme vyvíjať aplikácie alebo hry s využitím grafických prvkov. Sprevádzal nás celou praktickou časťou bakalárskej práce, čím sme si podrobne osvojili prácu s ním a na základe týchto informácií si predstavíme jednotlivé časti editora.



Obrázok 1 : Unity editor

1. **Hlavné menu** – pozostáva s položiek *File*, *Edit*, *Asset*, *GameObject*, *Component*, *Window* a *Help*. Tvorí základ editora a jednotlivé položky nám prinášajú možnosť pridávania scén, assetov, objektov a mnohé ďalšie funkcionality.
2. **Pracovné nástroje** – slúžia na editáciu objektov v našom prostredí. Vďaka nim ich dokážeme presúvať, otáčať, zväčšovať, zmenšovať alebo ich uchopiť. Je tu možnosť spustenia scény v editore.
3. **Hierarchia** – tvorí ju zoznam všetkých súčastí scény.
4. **Scéna** – „pracovný stôl“ editora, v ktorom vytvárame aplikácie. Jedná sa o trojrozmerný priestor do ktorého môžeme vkladať 3D modely a vymodelovať si tak prostredie podľa vlastných predstáv.
5. **Kamera** – každá scéna obsahuje kameru, ktorú môžeme voľne v scéne presúvať. Zobrazuje pohľad na scénu v reálnom čase a pri spustení aplikácie v rámci jej testovania v editore.
6. **Inšpektor** – zobrazuje všetky dostupné informácie označeného objektu (meno, pozíciu, vrstvy, animácie, skripty a mnohé ďalšie).

7. **Projekt** – tvorí ho adresárová štruktúra, v ktorej sú uložené všetky modely, textúry, audio súbory, materiály a mnohé ďalšie súčasti celého projektu. Vieme sem presúvať súbory z disku a pracovať s nimi.

1.3 Implementácia

V tejto časti práce sa pozrieme na tvorbu aplikácie a dodržiavanie kritérií, ktoré sme zadefinovali v špecifikácii. Do implementácie spadá tvorba projektu, ktorý pozostáva z 5 prostredí: z prostredia hylfóbie, hypsofóbie, hlavného menu, splnenej a nesplnenej úlohy. V čase vytvárania aplikácie sme používali verziu Unity 3D - 2019.4.14f1.

1.3.1 Vytvorenie a nastavenie projektu

Pred vytvorením projektu je dôležité nainštalovať modul „*Android Build Support*“, vďaka ktorému získame podporu vývojového prostredia pre naše zariadenie zobrazujúce virtuálnu realitu. Modul sa dá jednoducho nainštalovať v **Unity hube** cez „*Install > Version > Add Modules*“. Po vytvorení nového projektu so šablónou 3D sa zobrazí **Unity editor**. V aktuálnom stave ešte nie je pripravený pre vývoj interaktívnej aplikácie pre platformu Oculus Quest a je potrebné vykonať niekoľko dôležitých nastavení.

Ako prvé je nutné stiahnutie balíčka z AssetStoru s názvom „*Oculus Integration*“, ktorý prináša pokročilé vykresľovanie, podporu vývoja sociálnych sietí, platforiem, zvuku, avatarov a mnohých ďalších súčastí pre zariadenia Oculus VR.

V ďalšom kroku sa presunieme do nastavení projektu cez hlavné menu „*File > Build Settings > Player Settings*“. V paneli najprv klikneme na ikonu „*Android settings*“, čím ju aktivujeme a všetky ostatné zmeny nastavení, ktoré ideme vykonať, budú prispôbené pre platformu Android. Následne v tom istom paneli klikneme na nastavenie „*XR Settings*“, kde aktivujeme podporu pre virtuálnu realitu. V tomto nastavení musíme pridať podporu aj pre našu platformu Oculus cez „*List is Empty > + > Oculus*“. Nastavenia nás upozornia, že nemáme podporu pre Vulkan Graphics API¹. Presunieme sa v paneli do „*Other Settings > Graphics APIs > Vulcan > - (mínus)*“, čím ho odstránime. V nastavení „*Other Settings*“ tiež

¹ Vulkan Graphics API – multiplatformové aplikačné rozhranie pre trojrozmernú grafiku

nastavíme „*Minimum API Level na Android 5.0 (API Level 21)*“.

Posledné a zároveň najdôležitejšie nastavenie je zmena samotnej platformy pre ktorú sa aplikácia vyvíjala. Opäť sa presunieme do nastavení projektu cez hlavné menu „*File > Build Settings*“, kde v paneli „*Platform*“ klikneme na „*Android*“. Tu sa nám ponúknu ďalšie nastavenia, v ktorých urobíme zmenu len v „*Texture Compression > ASTC*“ na základe odporúčaní Unity pre Oculus zariadenia. Následne klikneme na tlačidlo „*Switch Platform*“, ktoré začne kompilovať celý projekt aj s našimi nastaveniami pre platformu Android.

Po uložení všetkých vyššie uvedených nastavení sa môžeme vrátiť späť do Unity editora a začať vytvárať prostredie aplikácie. Každú zmenu prostredia, ktorú vykonáme, si môžeme skontrolovať v zariadení Oculus Quest cez „*File > Build Settings > Build And Run*“. Unity editor následne vytvorí z projektu Android aplikáciu a nakopíruje ju do zariadenia Oculus, ktoré máme k počítaču pripojené USB káblom. V zariadení sa objaví naša aktuálna aplikácia, v ktorej si môžeme skontrolovať vykonané zmeny. Proces je časovo náročný a pri každej kontrole zmien sa čaká približne 10 minút na skompilovanie a nakopírovanie aplikácie do zariadenia.

Pri vytváraní aplikácie sme opakovali tento postup niekoľkokrát. Záchranou bola aktualizácia operačného systému zariadenia, ktorá umožňovala vstúpiť do nášho prostredia v reálnom čase. Aktualizácia sa objavila približne v polovici tvorby aplikácie.

1.3.2 Hylofóbia

V tejto časti predstavíme tvorbu prvého hlavného prostredia našej aplikácie s názvom Hylofóbia. Popíšeme podrobne postup realizácie jednotlivých položiek a predstavíme aj použité skripty, ktoré sme vytvárali.

Modelovanie terénu

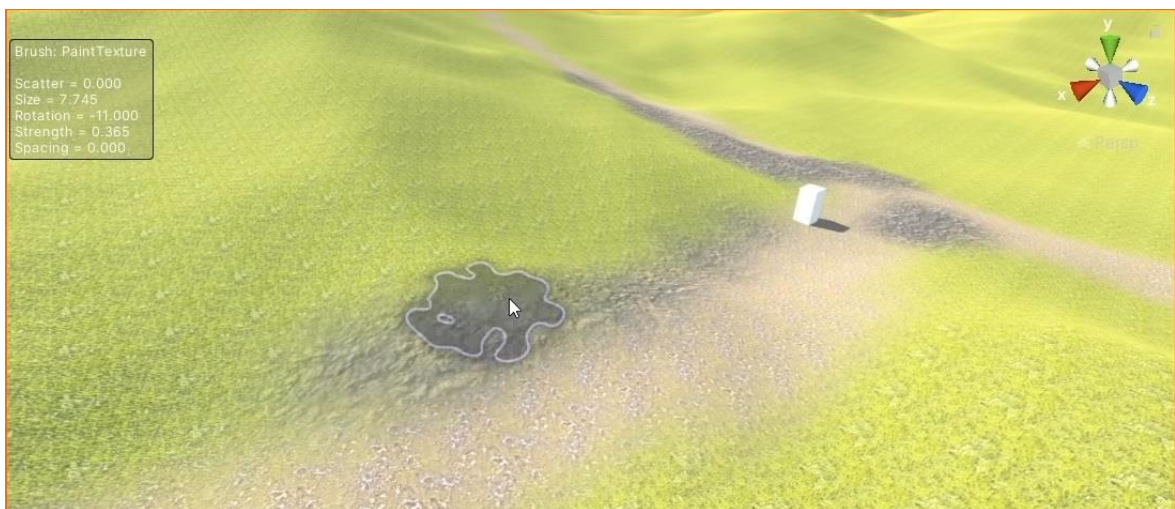
Unity 3D nám ponúka dve možnosti tvorby členitosti terénu. Prvá je manuálne modelovanie, ktorú sme použili v našom prostredí. Druhou možnosťou je vloženie súradníc zemského povrchu do špeciálneho nástroja, ktorý na základe reálnych dát vytvorí digitálny model reliéfu.

Pri manuálnom modelovaní umiestnime terén do scény v Unity editore cez položku hlavného menu „*GameObject > 3D Object > Terrain*“. V scéne sa vytvorí platňa bez textúr o veľkosti 1000 x 1000 Unity jednotiek. V nástrojoch môžeme jej veľkosť jednoducho meniť, tak ako sme to predstavili v časti **Unity editor**. Prednastavená veľkosť platne je pre náš projekt dostačujúca. Tvorbu jej členitosti začneme cez „*Inspector > Terrain > Paint Terrain > Raise or Lower Terrain*“. V tejto časti máme na výber množstvo foriem členitosti terénu. V Unity 3D sú tieto formy označované ako „*Brushes*“ (štetce). Cez nastavenie „*Brush Size*“ a „*Opacity*“ vieme zadať editoru veľkosť a nepriesvitnosť štetca. Následne držením ľavého tlačidla myši a prechádzaním po platni zvyšujeme jej reliéf. V prípade potreby zníženia reliéfu podržíme pravé tlačidlo myši so súčasným držením tlačidla „*Shift*“ na klávesnici. Reliéf upravujeme týmto spôsobom, až kým nedostaneme vhodný tvar. Pre naše prostredie sme zvolili terén členitejší, aby les pôsobil dôveryhodnejšie. Najvyššie hodnoty reliéfu sme nastavili po obvode platne, čím sme vytvorili nepriechodnú hranicu terénu, ktorá bude po pridaní textúr vyzeráť ako skalnaté pohorie. Ak nám pri tvorbe reliéfu nevyhovujú dostupné formy štetcov, máme možnosť ich rozšíriť bezplatným balíčkom „*Terrain Tools Sample Asset Pack*“ z *Asset storu*.

Ďalší krok tvorí pridávanie textúr. Cez „*Inspector > Terrain > Paint Terrain > Paint Texture*“ si nastavíme jednotlivé vrstvy textúr, ktoré budeme používať pri vytváraní zelene, chodníkov, hliny a štrku. Textúry tohto typu nie sú automatickou súčasťou editora, a preto je potrebné z *Asset storu* pred ich pridávaním stiahnuť a importovať balíka „*Outdoor Ground Textures*“. Po kliknutí na tlačidlo „*Add Layer*“ nám po nainštalovaní balíka pribudne zoznam rôznych druhov textúr, ako napríklad „*snow, sand, dirt, rock*“ a mnohé ďalšie.

Ako základ pre náš terén sme vybrali druh textúry „*moss*“, ktorá nám najvernejšie pripomínala zeleň. V inšpektore môžeme nastavovať ďalšie vlastnosti textúry, ako napríklad jej hustotu. V položke „*Normal Scale*“ prepíšeme hodnotu 5 na hodnotu 1 a hustota textúry sa zjemní, pričom povrch terénu vyzerá lepšie. Pred pridávaním ďalších textúr je dôležité nastaviť mierku, v ktorej budeme dotvárať prostredie. Bez nastavenia mierky by mohlo dôjsť k situácii, že prostredie by nemalo prirodzenú veľkosť vzhľadom na avatara. Myslíme tým veľkosť stromov, chodníkov a objektov ako takých nachádzajúcich sa v prostredí. Pre vytvorenie správnej mierky použijeme objekt „*kočka*“, ktorý vložíme do scény cez položku hlavného menu „*GameObject > 3D Object > Cube*“. Kocka má automaticky nastavené rozmery 1 x 1 x 1 Unity jednotiek, ktoré budeme brať v pomere 1:1 s metrom štvorcovým.

V inšpektore nastavíme „scale“ Y – osi na hodnotu 2, čím dostaneme kváder s podstavou 1 x 1m a výškou 2m. Kváder nám svojimi rozmermi pripomína dospelého človeka (približná veľkosť avatara) a môžeme dotvárať prostredie adekvátne ku veľkosti kvádra. Začneme chodníkmi, pre ktoré sa najviac hodia textúry „dirt“ a „scree“. Prejdeme cez „Inspector > Terrain > Paint Terrain > Paint Texture > Add Layer“ na pridanie vrstvy a vložíme požadované textúry. Pre lepšiu identifikáciu si vrstvy môžeme pomenovať vlastným menom a uložiť ich do okna projekt. Vďaka veľkosti kvádra vieme presne nastaviť veľkosť štetca, ktorým nanášame textúry chodníkov do scény. Textúra „dirt“ slúži na vykreslenie stredu chodníka a „scree“ na jeho okraje. Ako bolo spomínané, rozmer platne nášho terénu bol prednastavený na hodnotu 1000 x 1000 Unity jednotiek, čo v našom prípade v pomere 1:1 predstavuje 1000 x 1000 m². Ide o skutočne veľkú scénu a jej naplnenie je časovo náročné, a to minimálne pri tvorbe chodníkov. Pomocou nastavených textúr a veľkosti štetca začneme vytvárať chodníky. Snažíme sa ich rovnomerne rozmiestniť po celom teréne. Pre lepšiu ilustráciu vytvorenia členitosti terénu, aplikovania jeho textúry, vytvárania chodníkov, a to všetko pri dodržiavaní mierky vzhľadom na veľkosť kvádra, nám posluží obrázok č. 20.



Obrázok 2 : Modelovanie prostredia Hylofóbia 1

Ďalšiu fázu tvorí rastlinstvo - pridávanie stromov, kvetov, trávy. Opäť ide o modely, ktoré nie sú súčasťou editora a je potrebné ich stiahnuť v podobe balíčkov z *Asset storu*. Pre vkladanie stromov sme použili balík „Conifers“ a na kvety a trávu „Grass Flowers Pack“. Začneme vkladat' stromy cez „Inspector > Terrain > Paint Trees“. Editor, a to aj napriek stiahnutiu balíčka, modely stromov stále nevidí (No trees defined). Preto ich musíme manuálne vložiť podobne, ako sme to robili pri vkladaní textúr. Stále sa nachádzame v paneli „Paint Trees“ a pokračujeme cez „Edit Trees > Add Tree > Tree Prefab“, kde sa nám otvorí okno s modelmi patriacimi balíčku „Conifers“. Balíček ponúka 4 druhy stromov a po

kliknutí na požadovaný tvar ho stačí potvrdiť tlačidlom „Add“. Podobne, ako pri nanášaní textúr, aj v tomto prípade používame pri vkladaní stromov štetec. Môžeme nastaviť jeho veľkosť a zároveň pribudli možnosti pre editáciu stromov, kde nastavíme ich výšku a hustotu. Opäť editujeme stromy tak, aby ich veľkosť bola adekvátne vzhľadom na rozmery kvádra. Samotné vkladanie prebieha podobne ako pri textúrach, a to podržaním ľavého tlačidla myši. Textúrovanie chodníkov v teréne o rozlohe 1000m² bolo náročné. Postupovať pri vkladaní stromov podobným spôsobom je z časového hľadiska oveľa náročnejšie, nakoľko chceme, aby sa les rozprestieral po celej oblasti. Preto použijeme funkciu „*Mass Place Trees*“, ktorá nám rovnomerne rozmiestni stromy po celej oblasti. Stačí v nej zadať počet stromov v možnosti „*Number Of Trees*“. Podobne vložíme aj iné druhy stromov z ponuky. Po skontrolovaní scény zistíme, že stromy sa skutočne nachádzajú rozložené po celej oblasti. Problém je v tom, že sú aj na vytvorených chodníkoch, a to v našom prostredí nechceme. Kombináciou ľavého tlačidla myši s tlačidlom „Ctrl“ na klávesnici môžeme nechcené stromy z oblasti vymazať. Preto nastavíme veľkosť štetca adekvátne k šírke chodníka a spomínanou kombináciou tlačidiel prejdeme v editore všetky chodníky. Tento úkon, vzhľadom na dĺžku chodníkov, je tiež časovo náročný, ale zároveň je podstatne rýchlejší ako manuálne vkladanie stromov do celej oblasti. Rovnakým spôsobom vkladáme do scény aj kvety a trávu cez „*Inspector > Terrain > Paint Details > Edit Details > Add Grass Texture > Detail Texture*“. Po zvolení požadovaného modelu nastavíme jeho veľkosť, šírku a farbu. Základné nastavenia rastlín majú výrazné sfarbenie, čo nepôsobí prirodzene, preto využijeme možnosť editácie sfarbenia a upravíme sýtosť v možnosti „*Healthy Color*“ a „*Dry Color*“. Pokračujeme v úpravách a vkladaní rastlín rovnako ako pri stromoch. Výsledný efekt nám ukáže obrázok č. 21.



Obrázok 3 : Modelovanie prostredia Hylofóbia 2

Posledné dve fázy modelovania terénu predstavujú objekty súvisiace s úlohou, ktorú musí používateľ splniť a navodenie správnej atmosféry. Medzi objekty spadá horiaca priepasť a 3 kocky, ktoré sú v oblasti rozmiestnené tak, ako sme to definovali v špecifikácii aplikácie. Vloženie kocky do scény prebieha cez položku hlavného menu „*GameObject > 3D Object > Cube*“. Kockám upravíme rozmery na (0,2 x 0,2 x 0,2) a pridáme požadovanú textúru. Približne v strede oblasti upravíme reliéf cez „*Inspector > Terrain > Paint Terrain > Raise or Lower Terrain*“ pomocou známej kombinácie ľavého tlačidla myši a tlačidla „*Shift*“ na klávesnici. Vyberieme vhodný štetec a začneme reliéf znižovať. Úpravu reliéfu modelujeme na rozlohe približne 10 x 10 Unity jednotiek (m²). Vytvoríme priepasť, do ktorej podľa zadania úlohy musí používateľ hodiť všetky kocky, čím zároveň uhasí oheň. Následne začneme vytvárať na dne priepasti oheň. Pre vytvorenie ohňa potrebujeme do scény vložiť „particle systém“ cez „*GameObject > Particle System*“. V inšpektore vpravo hore, vedľa zámky, je možnosť prepnúť režim z „*debug módu*“ na „*normal mód*“. Po prepnutí tohto módu začneme editovať v inšpektore vytvorený „*Particle systém*“.

Možností nastavenia je veľa, ale nás zaujíma len „*Start speed*“, ktorý upravíme na hodnotu 2 a „*Start Rotation*“ na hodnotu 60. Táto úprava doieli v konečnom dôsledku prirodzený vzhlád ohňa. Ďalšia položka je „*Start Color*“, kde vyberieme tmavo oranžovú farbu. Presunieme sa v inšpektore do zložky „*Shape*“, v ktorej máme možnosť upravovať tvar systému a jeho „*Radius*“ (polomer). Ten nastavíme na hodnotu 1,5. Nastavovanie hodnôt je vždy individuálne a závisí od potrieb každého používateľa. V nasledujúcom kroku, v okne projektu, vytvoríme materiál cez pravé tlačidlo myši a „*Create > Material*“. Vytvorenému materiálu v inšpektore pridáme pripravenú textúru ohňa, ktorú sme si voľne stiahli z internetu. Opäť ide o individuálny výber textúry a je len na používateľovi, aká sa mu do prostredia hodí. Pre lepšiu identifikáciu pomenujeme takto pripravený materiál názvom „*oheň*“ a známym „*drag and dropom*“ presunieme na náš „*Particle System*“. Textúra sa objaví aj na scéne, ale stále nemá požadovaný vzhlád.

Posledný krok sa nachádza opäť v inšpektore „*Particle Systemu*“. Prejdeme na „*Shader > Particles > Additive*“. Po tomto nastavení, dostane vytvorený systém vzhlád horiaceho ohňa. Môžeme jednoducho meniť jeho veľkosť, pozíciu a aj ho kopírovať. Aby sme jeho vzhlád neporušili menením jeho aktuálnej mierky, tak ho skopírujeme, v našom prípade 3 - krát a presunieme v scéne na dno priepasti. Posledným krokom v rámci priepasti je vytvorenie objektu, ktorý po kolízii s kockami alebo avатарom vyvolá určitú reakciu (zmenu scény). V našom prípade ide o splnenie úlohy (v prípade hodenia kociek do priepasti uhasí oheň) alebo nesplnenie úlohy (spadnutie avatara do priepasti). Objekt si vymodelujeme

jednoducho z kocky, ktorú vložíme do scény. Rozmery prispôbíme veľkosti priepasti a odstránime jej základnú textúru v inšpektore cez „*Mesh > Remove Component*“. Nastavením docielime, aby objekt nebol v scéne viditeľný a zároveň bude schopný kolízie s inými objektmi. Pripravený objekt presunieme v scéne na dno priepasti a cez inšpektor vložíme do neho jednoduchý skript, ktorý vyvolá zmenu scény podľa toho či sa ho dotknú kocky z úlohy (zmena scény na splnená úloha), alebo či sa ho dotkne avatar (používateľ spadol do priepasti a scéna sa zmení na nesplnenú úlohu).

Na záver začneme pracovať na atmosfére, ktorú nám pomôže navodiť hmla, nočná obloha a zvuky. Hmlu nastavíme cez položku hlavného menu „*Window > Lighting > Fog*“. Zaškrtnutím tlačidla „*Fog*“. Unity editor pridá hmlu do celého prostredia. V nastavení si upravíme jej hustotu (Density) na hodnotu 0,02 a farbu ponecháme základnú. Pre vytvorenie nočnej oblohy potrebujeme v okne projektu vytvoriť materiál cez pravé tlačidlo myši a následne „*create > material*“. Klikneme na vytvorený materiál a nastavíme cez „*Inspector > Shader > Skybox > Procedural*“ položku „*Atmosphere Thickness*“ na hodnotu 5, čím docielime nočnú oblohu. Už postačuje len prenesenie materiálu štýlom „*drag and drop*“ na oblohu, ktorá automaticky stmavne. Pre pridanie zvukov do scény si potrebujeme najprv vhodné zvuky zaobstarat' a nahrat' ich do okna projektu. Následne cez „*GameObject > Audio > Audio Source*“ nám v hierarchii pribudne objekt „*Audio Source*“ a na scéne sa vytvorí ikona reproduktora. Zvuky, ktoré sme nahrali do okna projektu, reprezentujú atmosféru nočného lesa. Aby sme ich počuli v našej scéne, tak ich musíme vložiť do objektu „*Audio Source*“ cez „*Inspector > Audio Source > AudioClip*“ opäť štýlom „*drag and drop*“ z okna projektu. Daný úkon zabezpečí, že v okolí reproduktora v scéne budú zvuky počuteľné. V inšpektore môžeme nastaviť dosah zvuku na vzdialenosť, akú potrebujeme, ale rovnaký zvuk v celom prostredí by nebol prirodzený. Preto v hierarchii skopírujeme objekt „*Audio Source*“ toľkokrát, koľko zvukov chceme vložiť do scény. V každom novom objekte („*Audio Source 2, Audio Source 3*“) už len vkladáme vybrané zvuky rovnakým spôsobom.

Avatar

Aby sme sa mohli v prostredí pohybovať a interagovať s objektmi potrebujeme vytvoriť Avatara (počítačovú reprezentáciu používateľa). Jednoduchou cestou by bolo vloženie do scény „*OVRPlayer Controllera*“, ktorý je súčasťou balíka „*Oculus Integration*“. Tento Avatar je plne prispôbený na pohyb v našom prostredí a obsahuje mnoho

užitočných skriptov, ktoré by sa dali použiť v našom prostredí, ale nie je schopný bezproblémovo fungovať so skriptom „*Swinging Arms*“. Uvedený skript sme vytvorili špeciálne pre prostredie Hylofóbia a je založený na pohybe v smere pohľadu používateľa, ktorý súčasne prirodzene hýbe rukami (v nich drží ovládače), podobne, ako pri bežnej chôdzi. Z uvedeného dôvodu si vytvoríme vlastného Avatara, na ktorom skript funguje.

Do scény vložíme „*OVRCamera Rig*“ z okna projektu. Ide o kameru, ktorá bude reprezentovať náš pohľad v scéne. Následne v hierarchii cez pravé tlačidlo myši jej pridáme cez „*3D Object > Capsule*“ telo, ktoré bude tvoriť kapsula. Je to dôležitý krok, bez ktorého by náš Avatar nemohol mať kolíziu s terénom a okolitým prostredím. Podobne, ako pri kocke na dne priepasti, aj tu odstránime základnú textúru kapsuly v inšpektore a nastavíme jej „*Radius*“ na hodnotu 0,2 (hodnota nastavuje polomer kapsuly, ktorá slúži na nastavenie vzdialenosti, do ktorej nás kolider kapsuly pustí k objektu). V ďalšom kroku vložíme kamere v hierarchii prázdny objekt cez „*Create Empty*“ a pomenujeme ho „*Feet*“. Objekt bude slúžiť ako pomyselné nohy a vďaka nemu vieme Avatarovi povedať, kde sa nachádza podlaha a naša kamera nespadne pri spustení scény na zem. Aby proces fungoval, tak pre kapsulu v inšpektore vytvoríme skript a pomenujeme ho „*BodyColider*“. Skript bude obsahovať dve premenné, ktoré si nazveme „*head*“ a „*feet*“ a jednoduchú funkciu, ktorá bude obsahovať len jeden riadok („*gameObject.transform.position = new Vector3 (head.position.x , feet.position.y, head.position.z*“)). Pridaním skriptu kapsule v inšpektore vzniknú dve premenné: „*head*“ a „*feet*“. Z hierarchie presunieme do premennej „*head*“ položku „*CenterEyeCamera*“, ktorá je súčasťou „*OVRCamera Rigu*“ a do premennej „*feet*“ vytvorený prázdny objekt, ktorému sme dali rovnaké meno „*feet*“. Docielili sme, že kamera sa bude držať v rovnakej výške a bude meniť svoje súradnice spolu s kapsulou a objektom *feet* (pomyselné nohy). Avatar je funkčný a na základe nášho pohybu v reálnom svete sa už pohybujeme aj v scéne. Prostredie má ale veľkú rozlohu a pohybovať sa v ňom týmto štýlom je nemožné. Z uvedeného dôvodu sme vytvorili spomínaný skript „*SwingingArms*“, na základe ktorého sa budeme pohybovať pomocou hýbania rúk.

1.3.3 Hypsofóbia

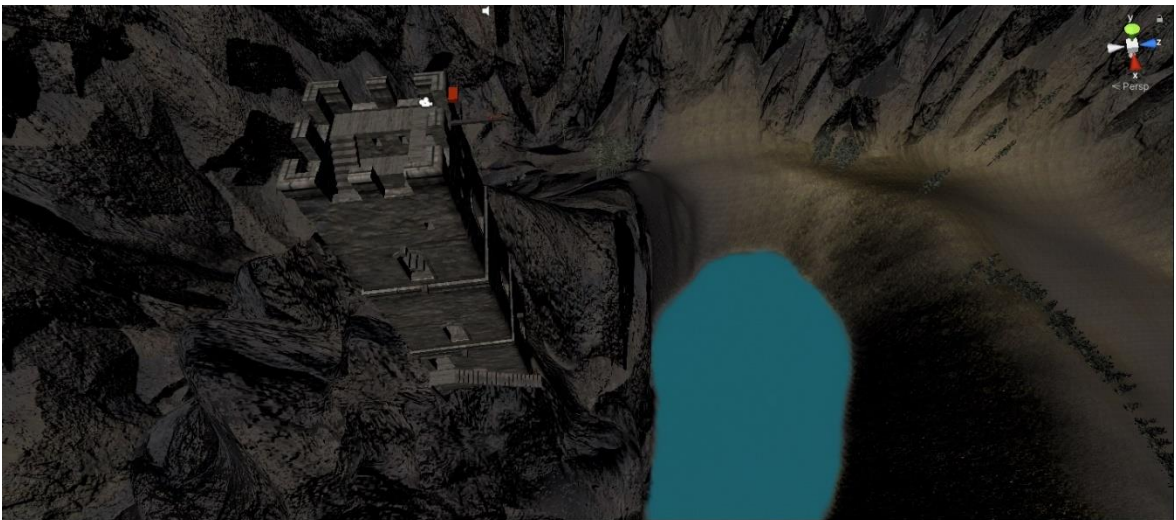
Druhé hlavné prostredie aplikácie, ktoré má navodzovať strach z výšky je Hypsofóbia. Ako vyplýva zo špecifikácie aplikácie, budeme klásť dôraz na vernosť prostredia, aby v používateľovi skutočne vyvolalo fóbické reakcie. Mnohé úpravy terénu budú podobné

ako v prostredí Hylofóbie, a z tohto dôvodu nie je potrebné všetky detailne opisovať.

Modelovanie terénu

Modelovanie terénu až na niekoľko výnimiek, je takmer identické ako v prostredí Hylofóbia. Opäť vložíme terén do scény, použijeme rovnakú textúru a zachováme jeho pôvodné rozmery. V tomto prostredí sa používateľ nemusí veľa pohybovať, ale bude sa nachádzať na vysokom bode a vďaka základnej veľkosti terénu jeho pohľad na rozľahlú krajinu dodá scéne autenticitu. Ďalším krokom je úprava členitosti. V tejto časti tvorby dochádza k prvej zmene. Najprv si podobným spôsobom, ako v prvom prostredí, upravíme reliéf terénu, aby bol členitý a po jeho obvode vytvoríme opäť pohorie. Zmena nastane v strede scény, kde vytvoríme vysoké bralo. Stačí na to ten istý štetec ako pri tvorbe členitosti, avšak na jednom mieste podržíme pravé tlačidlo myši dlhší čas, pričom terén začne vystupovať do výšky. Potrebujeme viac času a skúšania pri modelovaní, aby bralo pôsobilo dôveryhodne a nevystupovalo nad okolité pohoria. V prípade, že ho vytvoríme príliš vysoké, známou kombináciou pravého tlačidla myši a klávesy „*Shift*“ vieme terén opäť znížiť. Kombináciu kláves na zníženie terénu použijeme následne v ďalšom kroku a podobne, ako v prostredí Hylofóbie, začneme pod bralom vytvárať priepasť. Priepasť vytvoríme väčšiu a na jej dno použijeme textúru „*Water*“, ktorá v používateľovi vyvolá dojem, že sa na dne priepasti nachádza voda. Proces vytvárania klasickej vody v Unity editore je náročný, a nakoľko sa používateľ reálne v tomto prostredí k nej nedostane, tak nám stačí použiť aj spomínanú textúru „*Water*“, ktorá je voľne stiahnuteľná z *Asset Storu*. Prostredie máme pripravené, a ostáva nám posledná úprava týkajúca sa miesta, kde sa odohráva úloha, ktorú musí používateľ splniť. Z *Asset Storu* stiahneme balíček „*Towers_PBR*“, ktorý obsahuje modely veží. Po importovaní balíčka zvolíme vežu „*Stone_Tower*“ a umiestnime na vrch brala tak, aby sa nachádzala tesne nad priepasťou. Veže nemajú vytvorené *kolidery*, a teda sa nedokážu dostať do kolízie s inými objektmi. V našom prípade to nie je nutné, nakoľko sa nami vytvorený Avatar nebude pohybovať po samotnej veži, ale vytvoríme dosku, ktorú si jednoducho vymodelujeme z objektu kocka a zvolíme jej vhodnú textúru. Dosku umiestnime na vrch veži tak, aby do nej zasahovala a nezabudneme zvoliť aj správnu stranu veže, aby sme sa nachádzali nad priepasťou. Na začiatok dosky (vnútri veži) umiestnime Avatara. Na jej opačný koniec umiestnime 4 identicky vytvorené kocky podobne, ako v prvom prostredí. V inšpektore kociek pridáme skript „*Grababble*“, ktorý je súčasťou Unity editora a vďaka nemu sa dajú kocky chytiť

a prenášať. Posledným krokom je vytvorenie dvoch plošín. Obe si vymodelujeme z objektu kocka. Jednu menšiu, ktorú umiestnime pri štartovacej pozícii Avatara. Zvolíme jej červenú textúru a pridáme skript, ktorý po kolízii s kockami prehodí scénu na „*Splnená úloha*“. Druhá plošina podobne ako v prvom prostredí bude slúžiť na aktivovanie scény „*Nesplnená úloha*“. Odstránime jej textúru a umiestnime ju pod dosku približne v strede veže. Skript na nej bude meniť scénu vždy, keď sa dostane do kolízie s ktoroukoľvek kockou alebo



Obrázok 4 : Modelovanie prostredia Hypsofóbia

Avatarom. Pre lepšiu ilustráciu prostredia nám posluží obrázok č. 22.

Avatar

Pre vytvorenie Avatara použijeme identický postup ako v prvej časti. Jediný rozdiel je v tom, že skript „*Swinging Arms*“ neaplikujeme, pretože v tejto časti aplikácie bude veľmi dôležitá presnosť každého pohybu, resp. každého kroku po doske. Tým, že sa používateľ bude reálne pohybovať, vyvoláme v ňom väčšiu ilúziu vnorenia sa do vytvoreného prostredia.

1.3.4 Hlavné menu, splnená a nesplnená úloha

Posledné scény, ktoré potrebujeme vytvoriť sú tie, v ktorých sa používateľ ocitne pri spustení aplikácie a splnení, prípadne nesplnení úlohy. Tvorba a nastavenie spomínaných scén je takmer identická. Budú sa odlišovať len pozadím a tlačidlami, ktoré bude mať používateľ k dispozícii.

Ako prvé importujeme potrebný balíček (XR Interaction Toolkit) cez „*Window* >

Package Manager > XR Interaction Toolkit > Install". Pridáme do projektu novú scénu cez „*File > New Scene*". Scéna je vytvorená a pridáme do nej „*XR Rig*“ cez pravé tlačidlo myši v hierarchii a následne „*XR > Stationary XR Rig*“, ktorý bude slúžiť na interakciu s naším prostredím. Súčasťou *XR Rigu* je aj kamera, vďaka ktorej sa môže používateľ v prostredí rozhliadať a teda hlavnú kameru, ktorú Unity editor automaticky vloží do novej scény môžeme v hierarchii vymazať. V ďalšom kroku prejdeme do inšpektora položky „*LeftHand Controller*“, ktorá je tiež súčasťou *XR Rigu*. Nastavíme položku „*Max Raycast Distance*“ na hodnotu 300 a aktivujeme položku „*Enable Interaction with UI GameObject*“. Identicky to nastavíme aj pre „*RightHand Controller*“. Práve sme docielili, že keď sa používateľ ocitne na scéne, automaticky bude mať k dispozícii laserové lúče slúžiace k potvrdzovaniu tlačidiel, napríklad aj v hlavnom menu.

V ďalšom kroku vytvoríme panel hlavného menu pravým tlačidlom myši v hierarchii cez „*UI > Canvas*“. Na scéne sa objaví zatiaľ iba obrys panela. V hierarchii pridáme panelu opäť cez položku „*UI*“ pozadie („*Image*“) a nadpis („*Text*“). V inšpektore vložíme pozadie panelu (pripravený obrázok v okne projektu) cez položku „*Source Image*“ a nadpis premenujeme na „*Fóbie*“. Umiestnenie a farbu nadpisu upravíme adekvátne k pozadiu panela v inšpektore. Na záver vytvoríme tlačidlá, v hierarchii pravým tlačidlom myši cez „*UI > Button*“. Na paneli sa objaví tlačidlo, ktoré v inšpektore môžeme editovať. Zmeníme mu nadpis na názov prvého prostredia – Hylofóbia. Pozadie tlačidla pridáme rovnako cez položku „*Source Image*“. Do položky „*OnClick*“ vložíme pripravený skript, ktorý po stlačení tlačidla aktivuje scénu s prostredím Hylofóbia. Rovnako postupujeme pri tvorbe zvyšných tlačidiel hlavného menu. Jediný rozdiel je v tlačidle „*Koniec*“, ktoré obsahuje zmenu v skripte a po jeho aktivovaní nedôjde k zmene prostredia, ale k vypnutiu aplikácie.

Identickým spôsobom postupujeme aj pri tvorbe scén splnená a nesplnená úloha. Rozdiel je len v pozadiach panelov a funkčnosťou tlačidiel. Scéna splnená úloha bude obsahovať informáciu pre používateľa o úspešnom splnení úlohy a k dispozícii bude mať tlačidlá s možnosťou návratu do hlavného menu, vypnutia aplikácie alebo reštartovania úlohy.