



Technical Design Document 2.0

Michal Oláh, Joachim Svítek, Igor Liška



Základné informácie

Cieľová platforma: Nintendo Switch a Nintendo Switch Lite

Ovládanie: Nintendo Switch Joy-Con/Nintendo Switch Pro Controller alebo iné podporované (napríklad Horipad)



Hardvérové požiadavky

PC

Požiadavky pre aktuálny build hry, ktorý je dostupný na platformu PC.

- **CPU:** Dual-core 2GHz i3,
- **RAM:** 4 GB,
- **GPU:** Intel HD 4000,
- **Voľné miesto na disku:** 500 MB
- **OS:** Windows 8/10
- **Ovládanie:** Klávesnica s myšou



Nintendo Switch

Požiadavky pre cieľovú platformu vychádzajú z oficiálnej technickej špecifikácie pre konzolu Nintendo Switch a Nintendo Switch Lite¹. Tieto požiadavky spĺňa každá konzola.

- **CPU/GPU:** NVIDIA Custom Tegra processor
- **Architektúra:** Nvidia second-generation Maxwell
- **RAM:** 4GB LPDDR4
- **Video výstup:** 1080p60
- **Displej:** viac dotyková kapacitný 6.2 palcový IPS LCD s rozlíšením 1280x720,
- **Voľné miesto na disku:** 500MB



Technológie a vývoj

Engine: Unity 2D (aktuálne verzia 2019.2.0f1, predtým vyvíjané vo verzii 2018.2.5f1 64bit) Professional, programovací jazyk C# - Vývoj prebieha na operačnom systéme Windows 10. Unity sme zvolili hlavne vzhľadom na naše skúsenosti s týmto enginom. V budúcnosti je možnou výhodou aj portovanie hry na iné platformy, ktoré Unity ponúka.

Pre vývoj na Nintendo Switch v Unity máme k dispozícii EDEV Nintendo Switch Development Kit. EDEV je vývojový hardvér, ktorý je po vizuálnej aj výkonnostnej stránke totožný s komerčne predávanou konzolou Nintendo Switch. EDEV umožňuje testovať aplikáciu spolu s ladiacou konzolou (angl. debug console) a je plne postačujúci pre vývoj na Nintendo Switch. Do samotného Unity je potrebné dodatočne nainštalovať modul Nintendo Switch Support distribuovaný spoločnosťou Nintendo. Tento modul nie je verejne dostupný a je ho možné stiahnuť až po vytvorení developerského účtu v Nintendo. Po nainštalovaní modulu sme schopní v Unity zvoliť v build settings ako cieľovú platformu Nintendo.



Používané nástroje

Verziovanie: Git (Github), predtým Unity Collaborate

IDE: Visual Studio

Plánovanie a projektový manažment: Jira Cloud

Grafika: Photoshop, Illustrator

Cloud úložisko: Google Drive

Privátny repozitár na GitHubu nám zároveň slúži aj na zálohovanie celého projektu (zdrojových kódov a použitej grafiky v hre). Dokumenty s grafikou (súbory vo formáte .psd, .png a pod.) spolu so všetkými textovými dokumentmi a špecifikáciami (GDD, TDD, ADD atď.) sú uložené na Google Drive, čím je taktiež zabezpečená aj ich záloha a prípadná revízia starších verzií dokumentov. Pribeh projektu je plánovaný a riadený pomocou nástroja Jira Cloud, opäť ide o cloudovú službu, čím je zabezpečená záloha stavu a priebehu projektu.



List požiadaviek

Každá z uvedených požiadaviek (angl. features) vyplývajúcich z GDD, má uvedený stav, na koľko percent je implementovaná a rovnako akú má implementačnú prioritu. Po implementácii každej z požiadaviek do stavu približne 80% prichádza k ich vylepšovaniu (angl. polish) do takého stavu, aby zlepšila hrateľnosť, korešpondovala so zvoleným umeleckým štýlom a konceptom hry a tým vylepšila celkový herný zážitok pre hráča (angl. game feel).

¹ <https://www.nintendo.com/switch/tech-specs/>

Tab. 1 - Tabuľka požiadaviek

#	Požiadavka	Stav	Priorita
Mechaniky hry			
P01	Implementácia postavy a jej základných pohybov - pohyb do strán, skok, pohyb vo vzduchu tak aby prispieval na dynamike hry.	100%	Vysoká
P02	Implementácia základných bojových akcií pre postavu - úder, kop a ich vyváženie v boji.	100%	Vysoká
P03	Implementácia prepísania (angl. override) základných akcií hráča akciami /spôsobmi využitia objektu v prípade, že hráč drží objekt v ruke a držaný objekt disponuje danými spôsobmi využitia.	100%	Vysoká
P04	Implementácia spôsobu ničenia (uberania životov) hráčov a objektov.	85%	Vysoká
P05	Implementácia levelov a ich stavov tak, aby korešpondovali s tematikou a odzrkadľovali ponáranie sa do sveta fantázií.	20%	Vysoká
P06	Implementácia objektov pre každý level - izbu. V každej izbe sa objavujú unikátne objekty len pre daný level respektíve jeho stav.	15%	Vysoká
P07	Implementácia mechaník pre každý level - izbu a pre jednotlivé stavy levelov. Levely a ich niektoré stavy disponujú určitými mechanikami, ktoré sú jedinečné vzhľadom na daný stav (napríklad teleportačné skrine v detskej izbe).	10%	Vysoká
P08	Implementácia možnosti hry pre viacerých hráčov (2 až 6 hráčov).	100%	Vysoká
P09	Implementácia herných módov Capture the Flag / King of the hill.	10%	Vysoká
P10	Implementácia možnosti hry pre jedného hráča (angl. single player).	0%	Vysoká
P11	Implementácia zvukov a zvukových efektov.	5%	Vysoká
P12	Implementácia možnosti hry hráčov v tímoch (napríklad 2 vs 2).	50%	Stredná
P13	Implementácia spôsobu vyhodnocovania víťazstva v jednotlivých hrách.	20%	Stredná
P14	Ako súčasť finálneho vylepšovania zakomponovať interakciu hráčov s predmetmi v leveli, ktoré reagujú na dianie v hre. Príkladom môžu byť ohnivé fakle na stenách alebo rozbíjanie rôznych predmetov a pod.	15%	Nízka
Používateľské rozhranie			
P15	Hlavné menu, ktoré sa zobrazí hráčom po zapnutí hry.	50%	Vysoká
P16	Obrazovka ukončenia hry (GameOver screen), ktorá sa objaví po tom ako jeden z hráčov vyhrá hru (porazí protivníka) - možnosť reštartovania hry/ukončenia hry.	50%	Vysoká

P17	Možnosť výberu levelu samotným hráčom pred spustením hry.	30%	Vysoká
P18	Obrazovka prerušenia hry (Pause screen), ktorá sa objaví keď jeden z hráčov bude chcieť prerušiť hru.	100%	Stredná
P19	Možnosť výberu postáv samotným hráčom pred spustením hry.	0%	Stredná
P20	Ukladanie pokroku hry - uloženie stavov levelov, ktoré má hráč sprístupnené a rovnako pokrok v objavovaní objektov v knihe hračiek.	0%	Stredná
P21	Nastavenia hry ako súčasť menu, nastavenia hlasitosti zvukov v hre.	0%	Nízka
P22	Kniha hračiek - prístupná z hlavného menu. Obsahuje informácie o všetkých objektoch, ktoré sú v hre. Jednotlivé opisy objektov sa budú postupne odomykať podľa toho ako ich hráč prvýkrát v hre objaví.	0%	Nízka
Iné			
P23	Hra viacerých hráčov cez službu Nintendo Switch Online.	5%	Vysoká



Mechaniky

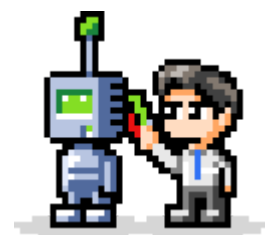
Vizuálny neporiadok

Objekty po použití zanechávajú v leveli rôzne časti ako nábojnice, zapichnuté náboje v stene, pierka z vankúšu a podobne. Je dôležité aby pri veľkom počte objektov v scéne mala hra stále plynulý priebeh a nebola pamäťovo náročná. Preto sa snažíme pri všetkých už nevyužívaných objektoch nezaťažovať fyzikálny systém výpočtami. V prípade, že ostáva objekt v leveli dlhšie, nepoužívame na ňom komponenty ako Rigidbody ani Collider a v leveli ostáva iba jeho snímka. Vďaka tomu sa pre tento objekt nemusia počítať kolízie čo nám dovoľuje mať v leveli väčšie množstvo objektov, ktoré celkovo dotvárajú vizuálny neporiadok.



Umelá inteligencia

Umelá inteligencia sa týka len niektorých špeciálnych objektov v hre, ktorými sú napríklad cínový vojačikovia, strelná vežička pri klincovačke alebo kniha, ktorá naháňa protivníka. Všeobecne AI v hre disponuje akciami ako útok, pohyb za objektom/po svojej predom určenej trajektórii alebo ich kombináciou. AI je najčastejšie implementovaná s cieľom zaútočiť na protivníka. Útok môže byť vykonaný nepriamo, kedy hráč koliduje s objektom, ktorý na základe toho vykoná útok (napríklad cínový vojačikovia) alebo priamo, kedy objekt sám iniciuje útok napríklad tým, že prenasleduje protivníka (napríklad kniha). Niekedy sa rozhodnutie o útoku a pohybe vyhodnocuje napríklad aj na základe blízkosti protivníka k objektu (od akej vzdialenosti od protivníka smie objekt útočiť - napríklad strelná vežička). Pri komplexnejších objektoch, akými sú napríklad záhradní vojačikovia, vypočítame cestu pohybu AI pomocou algoritmu hľadania cesty (path finding). AI si uchováva pozíciu protivníka na ktorého má zaútočiť, pričom sa bude musieť na základe okolia rozhodnúť pre vykonanie akcie z množiny akcií akými sú skok, pohyb do strán alebo samotný útok.



Tvorba levelov

Na vytvorenie vizuálu levelu používame grafické snímky (angl. sprite), ktoré sú rozdelené do jednotlivých vrstiev za sebou. Na oddelenie jednotlivých častí levelu používame 3 vrstvy (angl. sorting layer) - "back", "mid" a "front". Back vrstva slúži ako vrstva pozadia levelu, mid vrstva slúži ako plocha

po ktorej sa postavy pohybujú a front vrstva slúži ako vizuálne prezentačná vrstva, ktorá je nad všetkými ostatnými. Pomocou rozdelenia jednotlivých vizuálov do vrstiev využívame pre umocnenie vizuálneho zážitku z hry parallax efekt, ktorý dodáva do levelu ilúziu hĺbky - kde sa vrstvy vizuálu v popredí pohybujú pred kamerou rýchlejšie ako tie, ktoré sú v pozadí. Pre vytvorenie plochy levelu po ktorej sa môžu hráči pohybovať používame tileset, z ktorého pomocou grid metódy vytvoríme plošiny a steny, kde sa pomocou Tilemap Collideru vygenerujú tvary collideru pre každú časť v zodpovedajúcej tilemape.



Rendering

Jedna jednotka (angl. unit) v Squabble predstavuje 16x16 pixelov. Renderovanie grafických objektov používateľského rozhrania - UI (angl. user interface) obsiahnutých na plátne (angl. Canvas) prebieha pomocou Canvas Renderer. V Canvas používame ako render mode Screen Space - Overlay s nastavením Pixel Perfect, keďže chceme dosiahnuť renderovanie UI bez antialiasingu. V prípade objektov v scéne prebieha renderovanie pomocou Sprite Renderer, ktorý slúži na vizualizáciu objektov pre Unity 2D aj 3D. Pre vytvorenie špecifických prvkov v leveli ako napríklad rôzne farebné tieňe alebo nekonečné 2D tieňe používame nami navrhnuté fragmentové shadery, ktoré sú posledným krokom v "rendering pipeline" a dovoľujú upraviť výslednú farbu pixelu na monitore. Nami navrhnuté shadery využívame aj na post-processingové efekty ako napríklad tlakové vlny alebo deformáciu obrazu. Shadery píšeme v jazyku Cg/HLSL. Unity má niekoľko kompilátorov, ktoré preložia shaderové programy do strojového kódu, podľa platformy pre ktorú je vytvorený "build" hry. V prípade ak by sme chceli kompilátor vybrať manuálne, Unity nám poskytuje preddefinované makrá, ktoré slúžia na tento účel.



Objekty

Na uloženie, konfiguráciu a samotnú prácu s objektmi v hre využívame Prefab systém, ktorý poskytuje Unity. Prefab slúži ako šablóna a tým nám umožňuje jednoduché znovupoužitie objektov naprieč levelmi. Hlavné prefaby v hre sú pre postavu (Character) a objekt (Item), ktoré sú nositeľmi základných vlastností, ktoré musí daný objekt obsahovať. Od týchto prefabov potom odvádzame ich varianty (Prefab variant), ktoré, v našom prípade, slúžia na rozšírenie základného prefabu do jeho zmysluplnej variácie. Z character prefabu vytvárame jednotlivé hrateľné postavy (Tim, Viki, Hugo,...) v hre a z Item prefabu konkrétne objekty (vankúš, kľincovačka,...). Zmeny vykonané nad prefabom sa okamžite odzrkadlia vo všetkých jeho inštanciách, pričom samotné inštancie umožňujú prepísanie jednotlivých hodnôt, ktoré sa líšia od šablóny (napríklad v skriptoch). Tento systém nám uľahčuje manipuláciu a rýchlu konfiguráciu objektov v hre.



Animácie a efekty

Na animácie využívame animačný systém implementovaný priamo v Unity - Animation System niekedy tiež označovaný ako Mecanim. Animácie sa skladajú z jednotlivých snímok (angl. frames), ktoré tvoria celkovú snímkovú animáciu (angl. spritesheet animation), pričom predvolených je 15 snímok za sekundu. Pre každý animovaný objekt v hre je vytvorený animation controller, ktorý sa stará o jednotlivé animácie objektu na základe udalostí a stavov v hre, a púšťa ich na základe spúšťačov (angl. trigger) alebo bool premenných. Pri prefab variantoch využívame Animation Override Controller, ktorý umožňuje rozšíriť základný animation controller nahradením jeho jednotlivých animácií pri zachovaní jeho celkovej logiky, štruktúry a parametrov.

Na niektoré z vizuálnych efektov v hre využívame Particle System, ktorý je implementovaný priamo v Unity a vytvára efekty tým, že animuje veľké množstvo malých 2D objektov v scéne. V Particle systéme využívame vlastné snímky najčastejšie v kombinácii s animáciou na celkové dotvorenie lepšieho dojmu z hry.



Hudba a zvuky

V hre budeme pracovať so zvukmi a dynamickou/adaptívnou hudbou pomocou FMOD. Zvuky sa budú prispôsobovať aktuálnemu daniu a vizuálom v hre, podľa čoho budú prispôsobené a vytvorené udalosti, ktoré budú tieto jednotlivé zvuky spúšťať. Napríklad na začiatku hry bude hudba kľudnejšia pričom s intenzitou boja bude plynule stúpať aj intenzita hudby v hre. Implementácia zvukovej stránky do hry je konzultovaná s externou slovenskou firmou zameranou na produkciu hudby Grand Beats.

Implementácia hudby a zvukových efektov

Na implementáciu zvuku sa bude používať technológia FMOD Studio. Je to 3rd party middleware, ktorý komunikuje s Unity 3D pomocou eventov a parametrov. Tento systém bude slúžiť na:

- Implementáciu zvuku a hudby
- Nastavenie adaptívneho modelu a regulovanie dynamiky v reálnom čase
- Nastavenie randomizácií
- Mix zvuku
- Prípravu na portovanie pre Nintendo Switch

Implementácia SFX

Ozvučovanie akcií a interakcií bude "event-based". Znamená to, že Unity pošle event na základe ktorého FMOD prehrá daný zvuk (náhodný výber z viacerých sample-ov) podľa prednastavených funkcií a ďalších parametrov (pitch-shift modulácia, EQ, layering).

Implementácia hudby

V záujme dosiahnuť prirodzený prechod medzi scénami sa bude hudba implementovať cez systém FMOD a zmeny budú regulované pomocou eventov a parametrov posielané z Unity. Na plynulosť prechodov medzi scénami sa určia takzvané "kľúčové body", pri ktorých je možný prechod bez toho, aby si hráč všimol náhlu zmenu hudby. Znamená to, že po registrovaní eventu pre zmenu hudby sa musí dohrať slučka, ktorá aktuálne hrá a v určitom bode podľa taktu sa zmení alebo prejde cez špecifický fade.

Mix zvuku a hudby

Zvuková úprava (mix) bude vykonaná v systéme FMOD. Pod mixom zvuku sa rozumie: frekvenčné úpravy zvuku a hudby, úpravy hlasitosti, paralelná kompresia zvukových efektov a hudby, modulácie zvuku podľa ďalších parametrov.



Sieťové prvky

Samotný lokálny multiplayer si nevyžaduje pripojenie na internet, nakoľko hráči hrajú spoločne na jednom zariadení.

Pripojenie na internet je potrebné pre funkčnú telemetriu (zber dát z hry za účelom diagnostiky chýb a lepšie pochopenie správania hráča v hre) a internetový multiplayer.

V prípade budúceho rozšírenia hry o internetový multiplayer by bolo potrebné prevádzkovať server, ktorý by slúžil na synchronizáciu herného stavu medzi jednotlivými klientmi (hráčmi), vytvorenie internetovej hry podľa úrovne schopností hráčov (matchmaking) a napojenie na Nintendo Online službu. Prevádzku herného serveru zabezpečíme pomocou cloudovej služby AWS (Amazon Web Services), ktorá vo vhodnej konfigurácii okrem samotnej prevádzky zabezpečuje aj zálohovanie a škálovateľnosť.



Lokalizácia

Hra je vyvíjaná v anglickom jazyku a aktuálne neimplementujeme ďalšie jazyk. Nakoľko samotná hra obsahuje veľmi málo textu, doplnenie a integrácia ostatných jazykov aj v neskoršom štádiu vývoja nebude náročná.

Pre lokalizáciu do Unity by sme zvolili platené rozšírenie I2 Localization² od Inter Illusion, ktorá poskytuje prehľadné rozhranie, funkcionality ako export a import .csv súborov z jazykovými prekladmi z PC alebo Google Drive a podporuje širokú škálu lokalizácií od obrázkov, prefabov až po, pre nás najdôležitejšie, texty.



Optimalizácia a spätná väzba z hry

Na zber spätnej väzby z testovania sme implementovali Unity Analytics, ktoré je priamo zakomponované v Unity a slúži na zber dát o tom, ako a koľko hráči hrajú hru. Tieto dáta nám pomáhajú lepšie a ľahšie identifikovať možné chyby a detailnejšie nám priblížia spôsob interakcie medzi hráčom a hrou. Unity Analytics však bohužiaľ nie je momentálne podporované pre platformu Nintendo Switch. Na optimalizáciu výkonu hry však vieme použiť middleware Telemetry od Radgametools, ktorý kladie dôraz na charakteristiky výkonnosti vo vzťahu k času a stavu hry.



² <https://assetstore.unity.com/packages/tools/localization/i2-localization-14884> - prístup 15.04. 2020