Report

Jan Jonáš

Zpracování

1. Hráč

1.1. New Input System

Uživatelský vstup jsem implementoval pomocí New Input Systému od Unity, se kterým jsem dosavaď nepracoval. Vytvořil jsem InputActions asset a v něm jednoduchou action mapu, ve které byly namapovány vstupy pro WASD, Space a LMB.

Dále jsem si udělal singleton PlayerController, která vstupy zpracovává, případně vyšle event (mouse down/up).

1.2. WASD pohyb

Hodnoty snímaného Vectoru2 jako WASD pohybu kombinuji jako Vector3, již s předchozí y hodnotou rychlosti pro zachování momenta při výskoku a pádu

1.3. Skok

Systém uzemnění pomocí raycastu pod sebe skrz určitou vrstvu, pokud je hráč uzemněn a zaregistruje se SPACE, tak se přidá síla k rigidbody směrem nahoru

1.4. Manuální střílení

Pokud je zaznamenáno podržení levého tlačítka myši, spustí se Coroutina, ve které se konvertuje pozice kurzoru na raycast, který je cílený na specifický plošný child trigger collider, a pak se pozice střetnutí použije jako cílová pozice pro výstřel. Pokud se tlačítko myši pustí, coroutine se zruší a zapne se opět manuální střílení

1.5. Automatické střílení

ve FixedUpdate OverlapSphereNonAlloc na určitou vrstvu, pokud se chytne nějaký collider, tak se vyšle event s tímto gameObjectem. Pokud se nenajde nic, vyšle se event pro ztrátu.

Další script tyto eventy odposlouchává, pro event chycení se spustí coroutine, kde se vytvoří kulka a vyšle se směrem na pozici chyceného gameObjectu.

Pokud přijde event ztráty, coroutine se zruší.

1.6. HUD

Hráč vidí labely Killed a počet životů v horních rozích obrazovky.

1.6.1. Killed Label

Pokud hráč zabije nepřítele, tak se přidá pod Killed typ nepřítele a skóre, do budoucna se přidává skóre k již zobrazeným typům nepřítele.

Pokud hráč nic nezabije během daného časového úseku, skóre se zresetuje a label Killed se vyčistí na původní stav.

1.6.2. HP

Zobrazuje počet životů ve formátu počet/max.

Životy jsou uloženy v odděděném scriptu od Heath, kde pokud dojdou životy tak se zruší collidery a přehraje se particle systém.

Po smrti se hra přepne zpět na úvodní menu.

2. Nepřátelé

2.1. Spawner

Spawner je prefab, který v časovém intervalu nad sebou spawnuje náhodné nepřátele, které jsou na prefabu uložené v seznamu

2.2. Pohyb

Nepřátelé se pohybují pomocí MovePosition směrem k hráči, jehož pozici získávají pomocí PlayerController singletonu

Dalé je pomocí metody upravena pozice, a to aby vždy rigidbody zůstalo ve stejné výšce.
Tuto metodu lze přepsat v oddděných třídách, například pro sinusoidní pohyb.

2.3. Střílení

Střílení funguje na stejném principu jako automatické střílení u hráče, akorát s jiným prefabem pro kulku (jiná vrstva pro kolize).

2.4. Životy a dropy

Stejně jako u hráče jsou životy v oddděném scriptu, akorát pokaždé co nepřítel dostane damage, tak lehce zprůhlední. Vypočítáno jako počet životů / max.

Pokud nepřítel zemře, spawne určitý počet health dropů, které přidají zpět životy pokud je hráč sebere.

Sebehodnocení a časový rozklad

Ovládání hráče

Na PlayerControlleru jsem strávil asi nejvíce času, s fyzikou a pohybem hráče obecně jsem dlouho nedělal, takže mi chvilku trvalo než jsem si vše osvojil. Problém nastal i se skokem hráče, jelikož se občas teleportoval a pak pomalu padal dolů.

Naučení New Input Systému a osvojení/implementace fyziky mi trvalo přibližně **6h.**

Detekce pro automatické střílení

Detekci jsem nejprve řešil pomocí trigger enter/exit na child collideru, ale tam nastaly chyby při ničení objektů, takže jsem přešel na Physics.OverlapSphere.

To mi i práci ulehčilo, jelikož byly všechny scripty na jedné hiearchické úrovni.

Vytvoření logiky pro detekci a implementace mi trvalo přibližně **4h.**

Manuální střílení

Opět mi trvalo chvíli domyslet, nakonec jsem implementoval konverzi vstupu myši (x,y kurzoru) na Ray, který vystřelím na určitou vrstvu. Bod průniku, upravený pro výšku výstřelu se určí jako cíl střely.

Zde byl ze začátku problém s úhlem, jelikož původně byla vrstva zem. Jenže kvůli naklonění kamery nad hráčem nebyl výstřel cílený tam, kam je namířen kurzor.

Vyřešil jsem to runtime-vytvořeným box-colliderem na child gameObjectu, který má nastavenou vrstvu. Collider je ve výšce odkud hráč manuálně střílí, takže jsou výsledky přesnější.

(Také jsem ze začátku nepochopil zadání, takže jsem původně manuálně střílel jako postava v 3. osobě.)

Iterace a implementace různých způsobů mi trvala tak **5h**

Zbytek záležitostí z povinných úkolů mi celkově trvala cca **4h**, ale tolik jsem se na nich nezasekl.

Particle Systém, Reset timer, Health pickupy, spawnpoint prefaby občas měly lehké potíže, ale celkově mi dohromady trvaly přibližně **4h.**

Také jsem udělal procedurální animaci hráče, která pomocí stavového automatu přechází mezi módy změny v RGB hodnotách, plus změna průhlednosti nepřátel při změně životů. tato úloha mi trvala přibližně **2h.**

Zbytek času byl stráven kontrolou kódu, hledání různých chyb a opravami.

Úkoly jsem nedělal po jednom, pokud jsem se zasekl tak jsem šel na jiný. Tím pádem se časy úkolů prolínaly, nebo doplňovaly, protože řešení v úkolu jiném mě přivedlo na řešení v úkolu prvním.

I tak jsem hru stihl do 28h, byly věci na kterých jsem se více nebo méně zasekl, ale díky tomu jsem se i dost přiučil.
(New Input System, práce s vrstvami nebo detekce pomocí OverlapSphere.)