

## Souhrn

Zadání mě velmi bavilo. 😊 Mám to pojmenované jako Vampire Game. Ale upíři tam k vidění nejsou, takže by hra měla být spíše pojmenovaná „Vampire Game - without vampires“

Snaha byla vytvářet co nejvíce komponent ale ne zase aplikovat jak šílenec SOLID. Pokud mám ve třídě RequireComponent počítám s tím, že se pravděpodobně komponenta nebude měnit. Serialize Fieldy jsem dával věcem, u kterých je potenciál něco měnit (projektily apod).

Inspiroval jsem se trochu z Unrealu – tím, jak má postavené třídy a co mají za úkol. Player i enemy dědí z abstract Actoru, který představuje něco „živého“ a má tedy health.

Health třída zachytává všechnu interakci se zdravím a pomocí Unity Eventů může dobře předávat různé informace o proběhnutých akcích.

Hlavním principem, co jsem se snažil držet u hráče, je mít rozdělenou logiku na tři části. Samotný Player drží jen všechny důležité komponenty. Movement komponent pak přímo nastavuje jeho pohyb a Player Controller zachytává input a předává jej do movementu. Hráči jsem mohl dát pro ušetření práce charactercontroller, ale abych si vyzkoušel více funkcí v unity tak jsem pohyb dělal „ručně“ přes rigidbody.

Enemy je postaven podobně ovšem při absenci inputu má pouze jeden controller. Detekci hráče jsem mu udělat v coroutine aby neběžela logika každý frame.

Pro objekty, kterých by bylo potenciálně hodně jsem dělal jejich info přes ScriptableObject aby se předávaly pouze reference. Především tak slouží pro Enemies a projektily. Nějakou dobu jsem nemohl přijít na kloub kolizím v Unity, až mě to překvapilo...

Řekněme největší třídou je GameManager. Jedná se o singleton, který má za úkol držet důležité informace o hráči. V základu hráče samotného a PlayerController. Myšlenka byla taková, že GameManager je persistent napříč scénami. Stará se spawnování hráče, o spouštění scény a přepínání inputů, případně sloužící k fungování UI – taková spojka ale ne zase monolitická třída.

Tady jsem strávil nejvíce času a opravdu hodně velkou část jsem pouze zkoušel co a jak. Bojoval jsem s lifecyclem objektů a s tím souvisejícími inicializacemi apod. Docela jsem si s tímhle nápadem podřezával větev. Není to v podobě, kterou jsem si představoval – hodně různých oklik a natvrdo dělaných fixů (viz MainMenu helper). Chtěl jsem více a elegantněji používat eventy.

## Coby kdyby

Kvůli času jsem nestihl particle impacty a counter. S tím, jak to teď funguje by to nebyl problém implementovat, ale bohužel jsem si to nešťastně rozvrhl a věnoval se dlouho debugu a zjišťování, jak to vůbec funguje. Ještě teď při psaní koukám, že sinus movement je trochu hýbnutý (inicializace asi začíná, když je enemy ve špatné pozici). Pak by se dalo dělat ještě hodně samozřejmě ...třeba poolování.

## Časově

**hráč** - 8h

**enemy** – 6h

**UI, scény, pickup heal, přídatný movement:** kolem 11h

pak ještě navíc další učení se – mohlo to být celkem 5hodin. (new input action, fyzika v unity, kolize, nějaké best practices, Ai atd.).

stažený je pouze úvodní obrázek. Nějaké malé kousky kódu z fór apod.