

2D multiplayer shooter v Pyglet

Martin Beneš

ČVUT–FIT

benesm44@fit.cvut.cz

15. května 2022

1 Úvod

Semestrální práce byla vypracována v rámci předmětu BI-PYT v letním semestru 2021/2022.

Cílem práce bylo vytvořit jednoduchou střílečku pro více hráčů, kteří se mohou připojit přes lokální síť. Pravidla hry jsou jednoduchá, hrají proti sobě dva týmy a vyhrává ten tým, který úspěšně vystřelí všechny protivníky. Prázdná místa v řadách hráčů jsem se rozhodl zaplnit jednoduchou umělou inteligencí. Tohoto všeho jsem se rozhodl dosáhnout přes Python framework Pyglet.

2 Hra

Na začátku hry jsou oba týmy náhodně promíchány a umístěny do protějších okrajů herní mapy. Cílem hráče je střílet protivníky a, pokud možno, nestřílet spoluhráče. To komplikuje fakt, že kulky, které hráč vystřelí, se odráží od stěn. A to dokud nenarazí do nějakého hráče nebo další kulky. Díky tomu jsou hráči (kteří chtějí vyhrát) vystaveni před dilema, zda-li vystřelit a riskovat smrt spoluhráče, nebo raději čekat a nijak neznevýhodnit protivníky.

3 Hráč

Hráč má omezený počet životů, který se zobrazuje v pravém dolním rohu obrazovky. Jakmile tento počet klesne k nule, hráč umírá a už do hry nemůže dále zasahovat.

K dispozici má pohyb pomocí WASD a střelbu stiskem tlačítka myši. Dále má schopnost *dash*, která na krátkou dobu zvýší hráčovu rychlost. Tato schopnost se dobíjí, její dobítí je signalizováno změnou barvy hráče. Tato schopnost se aktivuje stiskem mezerníku.

4 Umělá inteligence

Umělá inteligence je ve hře implementována velmi jednoduše. Má dva stavy: *search* a *KILL*.

Ve stavu *search* se hledá nepřítel, kterého by bot s chutí zastřelil. Toho je docíleno tím, že má bot přiřazen pomyslný kruh, který reprezentuje jeho

dohled. Pokud se nepřítel nachází v tomto kruhu a nekryje jej žádná zeď, pak bot přechází do stavu *KILL*.

Ve stavu *KILL* bot střílí po své oběti a zároveň od ní udržuje optimální vzdálenost. Při kolizi se stěnou bot náhodně změní směr chůze, což samozřejmě není optimální, bohužel nebylo dost času na implementaci pathfinding algoritmu.

5 Server

Server má CLI, přes které lze přepínat mezi stavy. U serveru se také rozlišuje vícero stavů, které určují chování: *lobby*, *start*, *game*, *gameOver*.

Ve stavu *lobby* mají hráči možnost se připojit.

Ve stavu *start* je vytvořena herní instance na serveru.

Ve stavu *game* server vyhodnocuje data zaslaná klientem a aktualizuje podle nich stav hry.

Ve stavu *gameOver* se klientům zasílá výsledek hry.

6 Klient

Klient čeká, než mu server zašle informaci, že je ve stavu *game*, načež spustí lokální instanci hry, kde zobrazuje data zaslaná ze serveru a posílá zpět vstup ze strany hráče.

7 Metody/postupy/algoritmy

Ve hře je hojně využita dědičnost na herních objektech. Všechny objekty dědí od třídy **Object**, která se dále větví na **ObjectCircle** a **ObjectRect**.

Pro komunikaci mezi serverem a klientem je použita knihovna *socket*. Pro obsluhu této komunikace se používá *threading*. S oběma těmito knihovnami jsem pracoval poprvé a asi to podle toho bude i vypadat, server i klient by určitě snesly lepší zpracování.

8 Výsledky

Nejnáročnější částí a zároveň hlavním cílem pro mě u této práce bylo dostat hru do stavu, kdy je možné ji hrát přes síť. Bohužel při testování mezi dvěma zařízeními jsem narazil na chybu s modulem *pickle*,

kterou se mi nepodařilo vyřešit. Na jednom zařízení ale aplikace běží bez problému.

Parametry jsou uloženy v souboru *utils/config.py* a je možné si s nimi pohrát.

Na hodně věcí, které jsem měl původně v plánu nakonec nezbyl čas. Například mapa je hardcoded na danou velikost okna.



Obrázek 1: Ukázka hry

9 Závěr

Samozřejmě by hru šlo ještě hodně zdokonalit. Například přidat vícero herních map, vylepšit umělou inteligenci nebo zavést nějaký systém na hraní vícero kol. O prázdninách bych se rád k projektu vrátil a některé z těchto prvků doimplementoval (i když bez termínu odevzdání se k tomu asi nedokopu).

Zároveň jsem se naučil pracovat s frameworkem Pyglet, což pro mě bylo také přínosné. V minulosti jsem udělal pár her v Pygame, ale Pyglet jsem zvolil, abych měl vlastní srovnání. Pyglet bych určitě doporučil, byť k němu na internetu nenaleznete tolik informací, v dokumentaci se orientuje mnohem lépe. Nakonec bych řekl, že dělat hry mě baví, proto jsem si tvorbu semestrální práce užil. Jak už jsem zmiňoval, tak nejobtížnější pro mě osobně bylo navrhnout a implementovat server a klienta. Celkově z toho ale mám dobrý pocit a jsem rád, že si hru mohu zahrát přes lokální síť s přáteli.

Reference

- [1] Pyglet documentation. online, 2022. [cit. 2022–05–15] <https://pyglet.readthedocs.io/en/latest/>.
- [2] Socket documentation. online, 2022. [cit. 2022–05–15] <https://docs.python.org/3/library/socket.html>.