

# The Binding of Pac-Man

Jan Tichý

ČVUT-FIT

tichyj12@fit.cvut.cz

4. ledna 2022

## 1 Úvod

Semestrální práce byla vypracována v rámci předmětu BI-PYT v semestru 2021/22.

Prvotní myšlenkou bylo vytvořit roguelike RPG hru inspirovanou hrou The Binding of Isaac s tím, že udělám možnost pohledu do útrob hry a vytvořím 2 možnosti renderování: renderování hitboxů a renderování hry s texturami. To vše v Python frameworku Pyglet. Neodhadl jsem časovou náročnost, takže se hra zmenšila pouze na prostý generátor náhodných levelů a střílení nepřátel s možností přepínání renderingu.

Cílem této práce je tedy vytvořit techdemo roguelike hry.

## 2 Úroveň

Samotné úrovně (levely) jsou vygenerovány z 3x3 patternů, které hra zpracuje v `app/assets/level`. Je možné přidat další patterny, ale soubory musí zachovat formát 3x3 ASCII s novým řádkem na konci (**z toho plyne, že velikost herního pole musí být větší než nula a obě velikosti musí celočíselně dělitelné třemi**). Po každé úrovni se náhodně vygeneruje další. V úrovni jsou 2 druhy podlahových kostek. Jedna je prázdná, druhá je ohnivá. Ta způsobuje poškození hráči a nepřátelé chodící po zemi se jí vyhýbají.

## 3 Hráč

Hráč má definované různé statistiky (rychlost, poškození, rychlost střelby...) v `utility/config.py` souboru, které se dají libovolně měnit. Má také k dispozici teleportaci, která přeskočí kus herního pole.

## 4 Nepřátelé

Hra obsahuje dva druhy nepřátel:

**Ground enemies** chodí po zemi a vyhýbají se ohnivému podlaze. K hráči hledají cestu pomocí algoritmu **A\***, ve kterém využívám knihovnu **Numpy**.

**Flying enemies** létají a kostky je nepopálí.

Nepřátelé mají 3 typy střelby: **hvězda** - střílí do

všech směrů, **kříž** střílí diagonálně a **hodiny** střílejí projektily ve směru hodinových ručiček.

## 5 Ovládání

Hráč ovládá pohyb postavy pomocí WSAD a střelbu pomocí šipek. Teleportace vyžaduje držet pohybovou klávesu a stisk mezerníku. Možnost změny renderování uživatel provede stiskem klávesy **Q**.

## 6 Metody/postupy/algoritmy

Pozemní nepřátelé využívají **A\*** algoritmus pro nalezení cesty k hráči. S tímto algoritmem mám dobré zkušenosti z předchozích projektů, takže jsem se rozhodl pro něj.

Vyžívám dědičnosti napříč třídami, stěžejní třídy jsou **Renderable**, **Hitbox** a **Entity**, které pohánějí celou hru. Přepínání scény (menu/hra) funguje na základě přepínání dvou **Instance** objektů, které mají své renderování a jinou funkcionalitu.

## 7 Renderování

Stěžejní bod mé práce je (alespoň pro mne) renderování. Naučil jsem se a osahal si pár technik vykreslování a dalo mi to chuť zabývat se grafickým rozhraním OpenGL, které pyglet používá.

Základní třídou je **Renderable**, která obsahuje dvě metody: `__cg_render` a `__debug_render`. Každá dědičí třída je musí implementovat pro fungující render.

Ukázalo se, že grafika je ta nejpomalejší věc z celého projektu a byť jsem se snažil optimalizovat sebevíc, při větším množství objektů ve scéně hra klesá na nízké snímky.

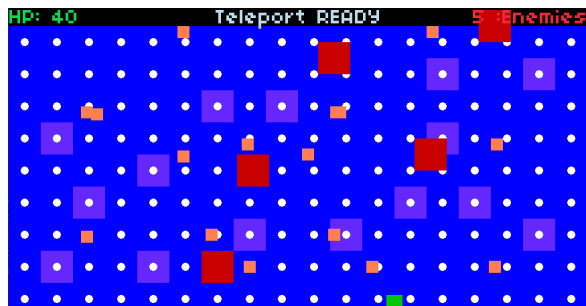
Pro nepohyblivé renderované objekty (**Sprite**, **Label**) jsem používal **Batch**, který zapeče všechny objekty a rapidně sníží časové požadavky pro vykreslování. Toto ale nefunguje pro pohyblivé objekty.

## 8 Výsledky

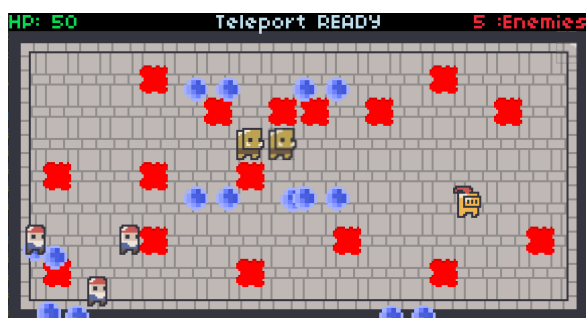
Hru jsem naimplementoval do hratelného stavu v nekonečném režimu (jediný konec je smrt). Dle re-

cenzí betasterů (kamarádů) je hra zábavná a mohla by být za pár eur na Steamu (takže když nevyjde FIT, půjdu dělat hry v Pygletu na steam).

Hra je konfigurovatelná přes konfigurační soubor, generuje náhodné místnosti, náhodné rozpořádání nepřátel a je hratelná. Cíl s renderingem jsem také splnil, takže já jsem spokojený.



Obrázek 1: Grafika v *debug* módu



Obrázek 2: Grafika v *cg* módu

Na snímcích obrazovky 1 a 2 je možné vidět oba režimy renderingu.

## 9 Závěr

Největším přínosem pro mě byly 3 věci:

1. grafika a její zpracování je super
2. baví mě dělat hry
3. jsem absolutně neschopný dělat časové odhady okolo náročnost projektu, na kterém pracuji

Projekt hry s tímto námětem pro mě byl víc než ambiciózní. Bohužel jsem neodhadl časovou náročnost a proto jsem nesplnil některé z požadavků a hodně z mých představ o hře. Je ale skvělé, že jsem se seznámil s novým frameworkem pyglet, zkusil si něco jako “*procedurální generaci*” a stvořil jsem, sice hloupou a ne moc dobrou, ale hru. Ať už hodnocení dopadne jakkoliv, já jsem s výsledkem spokojen.

Hra se dá rozšiřovat o nové nepřátele skrze config file, což považuji za dobrou featuru pro uživatele, dají se přidávat presety úrovní a další konfigurace. Doplnění bossů by bylo relativně snadné, ale v čase,

který jsem měl, jsem to nedokázal. Koncept Rende-rable classy se mi myslím povedl a díky němu jsem si práci usnadnil pro další implementaci renderování hry.

## 10 Poděkování

Nakonec bych chtěl poděkovat svým cvičícím za ochotu a super cvika, která mi předala potřebné znalosti.

A taky mým kamarádům, kteří mě podporovali a hru testovali, konkrétně Okskimu, Securitymu, Urdontovi a Tireneali.

## Reference

- [1] Popis a-star. [https://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](https://en.wikipedia.org/wiki/A*_search_algorithm).
- [2] analogstudios. Textury entit. <https://analogstudios.itch.io/forestsprites>.
- [3] analogstudios. Textury podlahy. <https://analogstudios.itch.io/dungeon>.
- [4] Pyglet. Dokumentace knihovny pyglet. <https://pyglet.readthedocs.io/en/latest/>.