Hra v Pyglet – Vyhýbání dopravě

Jan Koten (kotenjan@fit.cvut.cz)

15. Května, 2020

1 Úvod

Závodní hra, ve které je cílem vyvarovat se po co nejdelší dobu kolizi se soupeři nabrala od doby svého vzniku mnoho podob. Její princip však zůstává stejný, tedy kličkování v husté dopravě za přitěžujících podmínek. V této verzi je kladen důraz na realistický model auta. Při zatáčení tedy auto hráče opisuje kružnice a je schopné jak zrychlení, tak zpomalení, či couvání.
Kromě okolní dopravy ohrožuje hráče náraz do krajnice a tank, který jej pronásleduje po celou dobu hry. Zároveň na hráče každých pár vteřin vystřelí raketu, která mu způsobí značné škody, pokud se jí včas nevyhne. Hra obsahuje dva ukazatele. Červený znázorňuje současné poškození auta, zatímco modrý znázorňuje délku možného přídavného zrychlení. Tím je hráč schopný navýšit svojí rychlost téměř dvojnásobně a překonat tak maximální možnou rychlost tanku. Možnost zrychlení je ovšem rychle vyčerpána a pokud klesne pod určitou hladinu, pak nelze zrychlení použít do opětovného převýšení této hranice. Obnovuje se hráči po celou dobu hry, obšem její využívání je výrazně nákladnější, než její přírůstek a tedy je potřeba jí využívat přerušovaně pro udržení rychlosti nebo pouze v krajních situacích.
Ve hře je dále implementovaná fyzika okolních aut. Náraz do nich hráče zpomalí a nabourané auto vychýlí z dráhy s mírně zvýšenou rychlostí. Okolní auta mohou narážet i jedno do druhého a to i v takové míře, že dojde k jejich zneškodnění. Hráč tohoto může využít a způsobit řetězovou reakci, která mu vytvoří prostor k průjezdu

2 Metody a postupy

Auto hráče má pohyb realizovaný pomocí funkce update. Tato funkce přijímá dva argumenty `x` a `y`. První argument aktualizuje rotaci auta, druhý ovlivňuje samotný posun auta. Samotný posun zajišťují goniometrické funkce, které aktualizují posun auta:

```
x = y * sin(radians(self.sprite.rotation))
y = y * cos(radians(self.sprite.rotation))
```

Zároveň je zde vypočítán poměr posunu mapy v relaci s posunem auta na ose `y`. Mapa se má pohybovat v opačném směru pohybu hráče, kde horní hranicí, nad kterou je aktualizován pouze pohyb mapy je střed obrazovky.
Pokud má hráč nedostatek bodů pro zrychlení, je využita funkce `sign` z knihovny `numpy`, která zamezí přídavnému zrychlení, dokud body pro zrychlení nedosáhnou požadované minimální hranice.
Společně s mapou je také v závislosti na pozici hráčova auta aktualizována poloha veškerých grafických prvků na mapě. Posun mapy je odečten od jejich současné polohy.
Tank, auta a raketa mají implementovanou detekci kolize. V případě aut na dálnici realizovanými třídou `Enemy` se jedná o propracovanější schéma, které při detekci kolize také upraví rychlost, směr a poškození auta. Funkce `collision_update` vypočítá vzdálenost přední části auta od souřadnic v parametru a pokud vyhodnotí kolizi, je za použití knihovny `numpy` vypočítáno zrychlení a jeho směr, který je aplikován na objekt, pro který byla metoda zavolána, dále pokud kolize nastala ve spodní části, upraví se rotace ve směru protivníka, v opačném případě se od něj odchýlí. V závislosti na rychlosti, kterou měl protivník při srážce se upraví poškození auta a

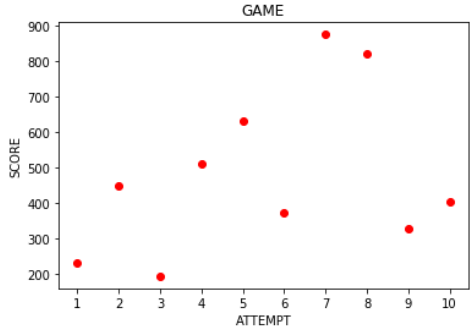
pokud se stane, že poškození klesne pod 0, je auto odstraněno z mapy a nastavena animace zneškodnění. Raketa nemá aktivovanou detekci kolize vůči ostatním autům a kontroluje pouze kolizi se samotným hráčem, zatímco tank likviduje okolní auta při prvním střetu.
Ve hře se dále vyskytují objekty, které nemají implementovanou detekci kolize. Jedná se především o objekty reprezentující animace výbuchů, a explozí. Těmto objektům lze při inicializaci nastavit trajektorii, rychlost a počet snímků, po jejichž dobu bude animace viditelná. Tento počet je defaultně nastaven na 1000 snímků, ale pokud se animace vyskytuje mimo obrazovku, je neprodleně odstraněna.
Objekty ve hře schopny pohybu mají implementovaný podobný model. Každému lze nastavit rychlost po ose x a y, maximálí rychlost a také jim lze nastavit přírůstek rychlosti:

```
y = float(y + self.momentum * 20) / 20.3
```

Toto zajistí plynulé zrychlení a zpomalování.
Tanku a raketě se při aktualizaci pohybu nastavuje takový směr, aby směřovali ke hráči s ohledem na hodnotu maximální rotace. K tomu je využita funkce atan2 z knihovny math.
Ve hře se vyskytuje i ikona srdce, které hráči sníží úroveň poškození. Tato ikona má detekci kolize nastavenou na dvojnásobek její velikosti pro snažší kontakt.

3 Výsledky

Maximální rychlost hráče při dočasném přídavném zrychlení je přibližně o čtvrtinu vyšší, než maximální rychlost tanku a dvojnásobně vyšší, než maimální rychlost hráče bez zrychlení. Maximální počet aut na dráze je konstantní s tím, že jejich poloha je vždy náhodná. Teoeticky tedy nikdy nemusí nastat situace, kdy hráč prohraje. Doposud ale nejvyšší skóre nepřevýšilo hranici 900 a na grafu Graf 1 můžeme vidět, že v jeho výši hraje nemalou část náhoda. Ve většině případů hráč dokázal zničit své auto než ho dostihl tank.



Graf 1: Skóre jednotlivých her

4 Závěr

Hra je navržena tak, aby do ní šly snadno doimplementovat nové grafické prvky. Pro přidání stačí inicializovat instanci třídy Particle, nastavit vhodné parametry a přidat jej metodou append do třídy Particles, která se stará o jejich administraci. Nepřátele se stejným způsobem zpravováni třídou Enemies. Jelikož je v kódu snadný přístup k informacím o poloze a rotaci všech entit včetně hráče, je možné do této hry implementovat genetický algoritmus s cílem dosažení co nejvyššího skóre.