

## Technický popis kódu

Kód je strukturován do tří tříd a hlavního herního cyklu spolu se základním nastavením některých objektů (obrazovka, skupiny třídy Group,...). V kódu jsem se snažila stručnými popisky vysvětlit jednotlivé funkce a proměnné, zároveň všechny důležité proměnné mají názvy odpovídající jejich funkci. Nebudu se jimi tedy dopodrobna zabývat v dokumentaci.

Jedinou použitou externí knihovnou je PyGame.

### Třída Game

Reprezentuje všechny akce hry od kontroly kolizí a počítání score po vykreslování jednotlivých obrazovek.

- funkce update – tato funkce je volaná v hlavním cyklu hry v každé iteraci a provádí kontrolu kolize herních objektů pomocí funkce check\_collisions, v případě zachycení komety (start\_timer == True) spouští funkci fast a navíc kontroluje polohu planet a odečte bod, pokud propadnou z herního pole
- funkce draw – je funkcí, kterou stejně jako update voláme stále dokola v herním cyklu a je zodpovědná za statické objekty, které se zobrazují na obrazovce. Jde vlastně o „nudné“ umisťování a vykreslování textu/obrazce.
- funkce check\_collisions – díky vestavěné funkci na kontrolu kolize tato funkce uloží objekt, se kterým koliduje hráč a podle jeho typu provede akce, které mohou být: přičtení bodů, odebrání života, přehrání doprovodného zvuku, nastavení start\_timer = True, aby mohla být funkcí update spuštěna funkce fast, nehledě na typ je planeta vrácena na místo, odkud může opět padat a je zvýšena její rychlost. Touto funkcí je také kontrolováno, zda nedošlo ke konci hry ztrátou životů, a případně tato funkce zavolá funkci pause\_game a reset\_game
- funkce fast – pokud chytíme kometu je funkcí update zavolána tato funkce (jak je zmíněno výše). Na obrazovce se objeví nový text s odpočtem času a rychlost hráče se zvýší. Tato funkce je volána po dobu 15s (tedy 15\*60 díky nastavenému fps), a pak je její volání díky kontrole podmínky if uvnitř jí samotné ukončeno.
- funkce pause\_game – funkce nám vykreslí zcela jinou obrazovku než při samotné hře a díky while cyklu se celý herní cyklus zastaví, dokud se uživatel nerozhodne, jak chce pokračovat.
- Funkce draw\_image – je menší pomocná funkce, kterou jsem si napsala, abych si ušetřila zdoluhavé a monotónní vykreslování statických objektů ve třídě Game. Jediná její vada je, že nelze použít při umisťování v závislosti na něčem jiném, než je střed objektu. Tuto vadu jsem se pokoušela odstranit, avšak na jednoduché řešení jsem nepřišla a vyhodnotila jsem, že objektů, na které nelze aplikovat je v kódu menšina.
- Funkce help\_screen – Při vymýšlení této funkce jsem si vzpomněla, že každá hra má svůj návod a ne každý pro něj chce procházet soubor s dokumentací. Tuto funkci přivoláte kdykoli v průběhu spuštění aplikace stisknutím klávesy h a odejdete z ní stiskem klávesy enter. Funkce díky while cyklu pozastaví hru (podobně jako funkce pause\_game) a zobrazí nápovědu (obrázky všech planet, jejich popis a co hráči přinese jejich chycení). V podstatě se jedná opět o nudné vykreslení všech obrázků na statickou obrazovku (zde se velmi hodí funkce draw\_image).
- Funkce reset\_game – je volána, pokud chce hráč po ukončení hry pokračovat hrou novou, navrátí díky zavolání funkcí reset pro objekty tříd Player a Planet hru do původní podoby (vyjma proměnné best, která si pamatuje hráčův osobní rekord)

## **Třída Planet**

Slouží k vytvoření jednotlivých planet, přiřadí jim jejich typ a s ním spojenou vizuální podobu, rychlost a místo startu, zkrátka vše, co námi potřebná planeta potřebuje. V třídě jsem programovala pouze tři malé funkce, pár užitečných jsem si také zajistila dědičností z vestavěné třídy Sprite.

- Funkce update – zajišťuje pohyb planety, v hlavním herním cyklu je volána stále dokola, pro zdání plynulého pohybu. Zde jsem řešila problém, jak zajistit, aby planety padaly ve zdánlivě nahodilých intervalech po sobě. Toho se mi podařilo docílit odlišným začátkem „letu“ nad obrazovkou, tudíž čas, kdy se dostanou do viditelného pole je různý, ač začaly své putování ve stejný okamžik. Ve spojení s pozdější odlišnou rychlostí, jednotlivých planet to dostává kýžený.
- Funkce reset\_place a reset\_speed – z názvu je zřejmé, k čemu se používají, jsou volány funkcí reset\_game ve třídě Game.

## **Třída Player**

Slouží k vytvoření postavy hráče s vlastnostmi, které pro naši postavu potřebujeme – vizuální vykreslení, doprovodné zvuky, životy, rychlost,... Opět jsou zde jen dvě menší funkce a dědičnost třídy Sprite.

- Funkce update – je volána stále dokola v herním cyklu a zajišťuje pohyb postavy hráče pomocí kláves šipek.
- Funkce reset – slouží k navrácení objektu do původního „nastavení“

Potom kód obsahuje opět poněkud nudné, ale nutné, vytvoření objektů podle jednotlivých tříd. Zde jsem nakonec zvolila založení pomocí třídy Group, protože vestavěné funkce této třídy se mi hodí do funkčnosti kódu (např. kontrola kolizí, vykreslování,...). A následuje už jen hlavní herní cyklus, který udržuje hru „při životě“ a umožňuje while cyklem její běh, aniž by program hned doběhl.

Na závěr bych dodala, že jsem hru podrobila několika hodinám samotestování a dolazování detailů typu: Jak rychle mají planety na začátku padat? Odkud a jak často mají padat? Je ten text čitelný? Atd. Můj osobní názor je, že všechny takové uživatelské detaily jsou v pořádku, avšak neměla jsem možnost sesbírat názory více hráčů, což by bylo pro tvorbu hry úplně nejcennější. Kód jsem se tudíž snažila psát tak, aby případná změna detailů byla snadno proveditelná a dělala se jen na jednom místě.

## Zdroje:

- Ikony/obrázky: <https://www.flaticon.com> konkrétně:
  - Cyclops - [https://www.flaticon.com/free-icon/cyclops\\_4925896?term=alien&related\\_id=4925896](https://www.flaticon.com/free-icon/cyclops_4925896?term=alien&related_id=4925896)
  - Comet - [https://www.flaticon.com/free-icon/comet\\_3119838?term=comet&page=2&position=2&origin=search&related\\_id=3119838](https://www.flaticon.com/free-icon/comet_3119838?term=comet&page=2&position=2&origin=search&related_id=3119838)
  - Earth - [https://www.flaticon.com/free-icon/earth\\_4238090?term=earth&page=1&position=4&origin=search&related\\_id=4238090](https://www.flaticon.com/free-icon/earth_4238090?term=earth&page=1&position=4&origin=search&related_id=4238090)
  - Meteor - [https://www.flaticon.com/free-icon/meteor\\_5821231?term=meteor&page=1&position=47&origin=search&related\\_id=5821231](https://www.flaticon.com/free-icon/meteor_5821231?term=meteor&page=1&position=47&origin=search&related_id=5821231)
  - Planet1 – [https://www.flaticon.com/free-icon/planet\\_661593?term=planet&page=1&position=58&origin=search&related\\_id=661593](https://www.flaticon.com/free-icon/planet_661593?term=planet&page=1&position=58&origin=search&related_id=661593)
  - Planet2 – [https://www.flaticon.com/free-icon/planet\\_3336008?term=planet&page=1&position=3&origin=search&related\\_id=3336008](https://www.flaticon.com/free-icon/planet_3336008?term=planet&page=1&position=3&origin=search&related_id=3336008)
  - Planet3 - [https://www.flaticon.com/free-icon/planet-earth\\_1374275?term=planet&related\\_id=1374275](https://www.flaticon.com/free-icon/planet-earth_1374275?term=planet&related_id=1374275)
  - Planet4 - [https://www.flaticon.com/free-icon/mars\\_1146343?term=planet&related\\_id=1146343](https://www.flaticon.com/free-icon/mars_1146343?term=planet&related_id=1146343)
- Zvuky: <https://www.leshylabs.com/apps/sfMaker/#examples>