

Soubor `atoms.py` obsahuje třídu `Atom`, která reprezentuje kružnice, které se vykreslují na plátně, a třídu `ExampleWorld`, která se stará o vytváření atomů a operace s nimi. Atom je reprezentován čtveřicí hodnot uložených v n-tici (`pos_x`, `pos_y`, `radius`, `color`). Tyto čtveřice se vrací ve funkci `tick()` - počet čtveřic odpovídá počtu atomů. Tato metoda je v časových intervalech volána metodou `run` v souboru `playground.py`. V tomto souboru je zajištěno vykreslování v GUI a není nutné jej měnit.

1. (1b) V konstruktoru třídy `ExampleWorld` je předán parametr, který reprezentuje počet atomů, které se mají vytvořit, k tomu výška a šířka plátna. Vytvořte odpovídající počet instancí třídy `Atom` a uložte do seznamu. Pozici, poloměr a barvu atomů generujte náhodně (při pozici využijte šířku a výšku plátna). K tomu vytvořte metodu `random_atom`, kde atom s náhodnými vlastnostmi vytvoříte a vrátíte – v konstruktoru třídy `ExampleWorld` pak tedy budete volat tuto metodu a přidávat atomy do seznamu (pro generování můžete využít funkci `randint` z balíčku `random`, pro barvu lze použít třídu `playground.Colors`).

2. (1b) Rozpohybujte atomy, a to tak, že k souřadnici x/y budete přičítat číselnou hodnotu (rychlost pro směr x/y). Rychlosti předejte v konstruktoru třídy `Atom` (dohromady tedy bude mít konstruktor parametry `pos_x`, `pos_y`, `radius`, `color`, `speed_x`, `speed_y`).

Ve třídě `Atom` vytvořte metodu `move()` která posune daný atom o vektor rychlosti - tuto metodu je pak potřeba volat v metodě `tick()` - v každém zavolání metody je tedy atom posunut a převeden na n-tici, což způsobí, že se atom bude po plátně pohybovat. V metodě `move()` kontrolujte, zda se atom dostal na(za) hranici plátna - v okamžiku, kdy se tak stane, otočte směr jeho pohybu - otočte vektor rychlosti ve směru `x` nebo `y`, podle toho, na kterou hranu atom narazil (velikost plátna předejte do funkce v argumentech).

3. (2b) Vytvořte třídu `FallDownAtom` dědící z třídy `Atom`. Tato třída bude obsahovat dvě **třídní** proměnné `g` a `damping` (například s hodnotami 3.0 a 0.8) – gravitační zrychlení a útlum.

Třída `FallDownAtom` dědí metodu `move`. Tu je však potřeba změnit tak, aby na tento atom působila gravitace. Při každém zavolání této metody se provede přičtení hodnoty `g` k vertikální rychlosti (jelikož je počátek osy `y` nahoře, přičítáním hodnoty `g` k rychlosti bude atom směrem dolů zrychlovat, přičítáním k záporné rychlosti `y` – tedy při stoupání atomu vzhůru, zase bude atom zpomalovat, až se dostane rychlost do kladných hodnot a začne opět klesat). Atom navíc při nárazu na spodní hranici zpomalí ve vertikální i horizontální rychlosti (`rychlost * damping`) – to způsobí, že se atomy nebudou pohybovat do nekonečna po plátně.

4. (0.5b) Zajistěte, aby u třídy `Atom` i `FallDownAtom` nedošlo k úniku atomů z plátna (všimněte si, že lze měnit velikost plátna) a to tak, že když by k tomu mělo dojít, nastavte jejich pozici na hranici plátna (pozice je dána středem, takže nastavení na spodní hraně je `pos_y = height - rad`).

5. (0.5b) Ve třídě `ExampleWorld` změňte metodu `random_atom()` tak, aby také náhodně generovala instanci `Atom` nebo `FallDownAtom` (např. Generováním náhodného čísla 0 nebo 1 – podle toho vytvoříte instanci).