

Szczypta matematyki w tworzeniu gier (część 1)

Matematyka prezentowana w szkołach jest zazwyczaj oderwana od programowania. Niemniej jednak garść wzorów z podręczników okazuje się być całkiem przydatna przy tworzeniu prostych gier. W poniższym artykule przedstawiam kilka z nich.

Przykładowy kod w artykule napisano w JavaScript, ale jestem przekonany, że sprytni czytelnicy poradzą sobie z przetłumaczeniem przedstawionych wzorów również na inne języki programowania.

Zachęcam do zignorowania tekstu tego artykułu i skupieniu się na samodzielnej analizie przykładowego kodu z Listingu 1. Sam artykuł może natomiast służyć za podpowiedź, jeśli cel któregoś z wzorów byłby niejasny.

ZAOKRĄGLANIE, CZYLI OBCIĘCIE LICZBY PO KROPCE

Co zrobić, jeśli mamy liczbę rzeczywistą, np. 12,345, ale interesuje nas jedynie jej część całkowita (tj. 12)?

JavaScript udostępnia trzy podobne, ale różniące się w szczegółach funkcje (patrz Ilustracja 1).

- » `Math.floor(liczba)` – wynikiem zaokrąglenia jest najbliższa liczba całkowita mniejsza lub równa podanej liczbie (jest to tzw. zaokrąglenie w kierunku minus-nieskończoności). Na przykład:

```
Math.floor(2.0) → 2
Math.floor(2.4) → 2
Math.floor(2.6) → 2
Math.floor(-2) → -2
Math.floor(-2.1) → -3
```



CIEKAWOSTKA

Współczesne gry uruchamiane w przeglądarkach korzystają z JavaScriptu i WebAssembly, ale w przeszłości była to domena głównie Adobe Flash. Mimo że technologia ta jest obecnie w odwrocie, oprogramowanie do przygotowywania apletów Flash – Adobe Animate – nadal jest używane do tworzenia wielu animowanych bajek wyświetlanych m.in. na Cartoon Network, Nickelodeon, Disney Channel, YouTube czy Netflix.

- » `Math.ceil(liczba)` – funkcja ta działa podobnie jak poprzednia, ale zaokrąglenie następuje w kierunku liczb większych (tj. w kierunku plus-nieskończoności). Na przykład:

```
Math.ceil(2.0) → 2
Math.ceil(2.4) → 3
Math.ceil(2.6) → 3
Math.ceil(-2) → -2
Math.ceil(-2.1) → -2
```

- » `Math.round(liczba)` – w tym przypadku mamy do czynienia z „prawdziwym” zaokrągleniem, tj. wynikiem będzie liczba całkowita najbliższa podanej liczbie. Na przykład:

```
Math.round(2.0) → 2
Math.round(2.4) → 2
Math.round(2.6) → 3
Math.round(-2) → -2
Math.round(-2.1) → -2
```

- » Co ciekawe, i czasami przydatne, identyczne wyniki da wywołanie funkcji `Math.floor(liczba+0.5)`.

Oprócz funkcji zaokrąglających mamy do dyspozycji jeszcze tzw. konwersję na liczbę całkowitą, która po prostu „usuwa” część ułamkową. W przypadku JavaScript taka konwersja jest dokonywana niejawnie w momencie użycia operacji logicznych na bitach (np. alternatywy – w skrócie: OR, lub alternatywy wykluczającej – w skrócie: XOR). W praktyce korzysta się tutaj z faktu, że alternatywa (operator `x | y`) lub alternatywa wykluczająca (operator `x ^ y`) z liczbą zero nie zmieniają wartości całkowitej zadanej liczby (podobnie jak np. dodanie do dowolnej liczby zera lub pomnożenia jej przez jeden). Na przykład: