===========================================================================

**0.3a - changelog**

* Przy zmianie masy naprawione wyświetlanie optymalnie maksymalnej ilości kul na czerwono, kiedy ilość kul była mniejsza od niej.
* Dodanie w prawym górnym rogu ekranu czerwonego napisu „[ESC] – zatrzymaj skrypt”.
* Dodanie nowej symulacji – 02 z rurką z parametrem określenia długości rurki.

===========================================================================

<https://github.com/grabowski-michal/perfectly-elastic-collisions/tree/master/v0.2b>

**0.2b - changelog**

* Gigantyczna optymalizacja dzięki sekcjom wydzielonym do detekcji kolizji – dzięki temu skok wydajności nawet ponad dwudziestokrotny.

===========================================================================

<https://github.com/grabowski-michal/perfectly-elastic-collisions/tree/master/v0.2a>

**0.2a - changelog**

* Dodanie parametru:
  + **Gradient** – czy użyć gradientu do narysowania kul.
* Po wybraniu przycisku otwarcia przegrody zaczyna być liczony czas z dokładnością do 1 ms.
* Zmiana designu wyświetlenia ilości kul w danym pojemniku.
* Ustatycznienie przesuwanej przegrody po przesunięciu jej.

===========================================================================

<https://github.com/grabowski-michal/perfectly-elastic-collisions/tree/master/v0.1b>

**0.1b - changelog**

* Optymalizacja kodu i zwiększenie wydajności.

===========================================================================

[https://github.com/grabowski-michal/perfectly-elastic-collisions/tree/master/v0.1a](https://github.com/grabowski-michal/perfectly-elastic-collisions/tree/master/v0.2a)

**0.1a - changelog**

* Poprawna funkcjonalność podstawową programu – *Symulacja 1 z przegrodą pionową*.
* Możliwość zmiany parametrów widniejących poniżej:
  + **Długość przegrody** (pionowa) w zakresie 10-100 %,
  + **Szybkość otwierania** w zakresie 1-100 px/s,
  + **Masa kul** w optymalnym zakresie: 0.01 - 10 (jednostek masy),
  + **Ilość kul** (0 - max.),
  + **Przedział długości wektora prędkości** – ustawienie minimalnej i maksymalnej wartości skalarnej długości wektora prędkości. Ta prędkość będzie się generowała losowo na podstawie generatora liczb pseudolosowych.
  + **Kolor kul** (z zakresu palety RGB).

* Skrypt wskazuje **optymalnie maksymalną ilość kul** podczas zmiany masy kul.
* Po wybraniu przycisku „Rozpocznij symulację” – generowane są kule o losowych prędkościach wybranych na podstawie losowo wygenerowanego kąta oraz losowo wygenerowanych szybkościach (|v|) z *przedziału długości wektora prędkości* oraz o promieniu obliczonym na podstawie masy. Z kąta prędkości każdej z kul oraz |v| liczone są składowe wektora prędkości.
* Po uruchomieniu symulacji pojawia się przycisk „Otwórz przegrodę”, który pozwala otworzyć przegrodę, której parametry wybrał użytkownik.
* Pod każdą z komór pojemnika widnieje etykieta wskazująca na ilość kul w danym pojemniku w danej chwili.

Optymalnie maksymalna ilość kul - wskazana maksymalna ilość kul ze względu na ograniczenia wielkości pojemnika (800-400 px). Nie jest to bynajmniej rzeczywista maksymalna ilość kul, ponieważ w rzeczywistości zmieści się ich tam więcej, ale ograniczono odległości startowe pomiędzy krawędziami pojemnika a kulami oraz odległości startowe pomiędzy kulami do połowy promienia każdej z nich.

===========================================================================