

**7.1. Ruchy**, 12 pkt

Bazując na przykładzie z listy 6 dokonaj modyfikacji:

- 1) Kółko jest za duże, zmień jego rozmiar tak by zajmowało  $\frac{1}{4}$  ekranu. (1 pkt)
- 2) Szerokość i wysokość okna oraz promień koła umieść w stałych. Następnie z tych wielkości wylicz, gdzie powinien się znajdować początek aby koło było wycentrowane. (1 pkt)
- 3) Wprowadź zmienne „x” i „y” typu float oznaczające pozycję koła i wykorzystaj funkcję setPosition aby je odpowiednio ustawić: <http://www.sfml-dev.org/tutorials/2.3/graphics-transform.php> (1 pkt)
- 4) Rozszerz program tak, aby zamykał okno po naciśnięciu „Esc”. Wskazówka: kody klawiszy znajdują się w dokumentacji api biblioteki SFML (1 pkt)
- 5) Dodaj do programu zmienne „vx” i „vy” typu float oznaczające prędkość koła. Początkowe wartości mogą być np. 0.05. Dopisz przesuwanie koła:  
 $x = x + vx * dt$ ,  
 $y = y + vy * dt$ . (1 pkt)
- 6) Uwzględnij, że koło może wyjść poza ekran i w takich przypadkach odbij prędkość zgodnie z zasadą, że ruch w kierunku „x” jest niezależny od ruchu w kierunku „y”. (1 pkt)
- 7) Wprowadź stałą „N” oznaczającą ilość elementów. Wykorzystaj tablice jednowymiarowe (wprost zamień zmienne x,y,vx oraz vy na tablice) oraz dodaj tablicę kształtów „shapes” tak, aby obsłużyć N poruszających się kółek, zamiast jednego. (2 pkt)
- 8) Dodaj grawitację, która zmieni w każdym kroku prędkość poruszających się kół (może być najprostsza metoda Eulera) tak, aby punkty odbijały się i poruszały po parabolach, zamiast po prostych. Możesz to zrobić np. wykonując:  
 $x = x + vx * dt$   
 $y = y + vy * dt$   
 $vy = vy + g * dt$  // grawitacja (2 pkt)
- 9) Dopisz do programu coś od siebie, może to być np. oddziaływanie  $N^2$ , uzależnienie koloru od prędkości ciał, dopisz kolizje pomiędzy ciałami itp. etc. (2 pkt)