Przykładowe kolokwium

Zasady kolokwium

Przebieg

- Kolokwium odbywa się w formie on-line. Czas przeznaczony na rozwiązanie kolokwium to 1 godzina 30 minut. O wyznaczonej godzinie w systemie eKursy zostanie udostępniona treść kolokwium.
- Od momentu udostępnienia treści, przez 1,5 h aktywne będzie zadanie w systemie eKursy w którym należy umieścić rozwiązanie w formie plików tekstowych z rozszerzeniem `.cpp` / `.h` / `.hpp`.
- Nie ma preferencji co do organizacji kodu względem podziału na pliki - zadanie może być przesłane w jednym pliku `.cpp` lub podzielone na wiele plików w zależności od indywidualnych preferencji/wygody.
- Tylko rozwiązania umieszczone w wyznaczonym zadaniu na eKursy będą akceptowane, nie dopuszcza się innych form dostarczenia rozwiązania (wraz z przykładowym kolokwium udostępnione zostało zadanie na eKursy).
- Istniej możliwości wielokrotnego wgrywania rozwiązania w ramach limitu czasu.

Ocenianie

- Błędy kompilacji dyskwalifikują z dalszej oceny i skutkują wynikiem 0 pkt.
- Pełna ocena za zadanie zakłada nieużywanie zmiennych globalnych oraz niekopiowanie argumentów tam, gdzie nie jest to konieczne (proszę korzystać z referencji).
- Jeśli w treści są podane nazwy funkcji, klas itd. należy się ich trzymać.
- Unikaj tworzenia publicznych pól klas, staraj się odpowiednio przygotować interfejs klasy.
- Rozwiązania zostaną poddane kontroli antyplagiatowej - proszę korzystać tylko z kodu swojego autorstwa, jeżeli przygotowują Państwo *gotowe* fragmenty programów proszę przygotować je samodzielnie.

Kontakt z prowadzącym

W trakcie kolokwium prowadzący dostępny będzie na platformie Zoom, pod tym samym linkiem, pod którym odbywają się standardowe zajęcia. W przypadku wystąpienia problemów technicznych z portalem eKursy lub ewentualnych błędów w treści zadań macie Państwo możliwość natychmiastowego zgłoszenia takiej sytuacji. Logując się proszę użyć pełnego imienia i nazwiska.

Treść zadania

Do zadania dołączono zestaw tekstur do wykorzystania. Rozpakuj je i umieść folder tekstury w folderze uruchamiania programu (w przypadku Qt Creator będzie to *build folder* projektu), korzystaj tylko z ścieżek względnych.

Stwórz trzy klasy: `Thor`, `KamienNieskonczonosci` oraz `StworOutrider`, dziedziczące po wspólnej klasie `AvengerSprite`, reprezentującej oteksturowany obiekt. Zastanów się czy klasa bazowa `AvengerSprite` nie powinna być klasą pochodną klasy, która już implementuje oczekiwane funkcjonalności. Korzystając z SFML zaimplementuj następującą funkcjonalność:

1. Dodaj tekstury do stworzonych klas oraz przeskaluj je z argumentem: `(0.1, 0.1)`, skorzystaj z dostarczonych tekstur.
2. Umieść na środku ekranu obiekt klasy `Thor`.
3. Umieść na scenie 20 obiektów klasy `StworOutrider` i oraz 6 obiektów klasy `KamienNieskonczonosci` w losowych miejscach (poza środkiem ekranu, wylosuj takie wartości, aby obiekty nie wystawały poza krawędź ekranu), przechowuj unikalne wskaźniki do nich w jednej wybranej kolekcji.
4. Dodaj do klasy `AvengerSprite` pola opisujące prędkością pionową i poziomą, ustaw je domyślnie tak, aby:
    - `Thor`: poruszał się jednostajnie w prawo (np. 150),
    - `KamienNieskonczonosci`: poruszał się jednostajnie w dół (np. 50),
    - `StworOutrider`: poruszał się jednostajnie w lewo lub w prawo (kierunek losowany w momencie tworzenia obiektu; np. 100 lub -100).
5. Dodaj do klasy `AvengerSprite` metodę `void Animuj(const sf::Time &elapsed)`, implementującą ruch obiektu z zapisaną w obiekcie prędkością pionową/poziomą.
6. Rozszerz metodę `Animuj` o wykrywanie kolizji z krawędziami ekranu, w przypadku dotknięcia krawędzi, każdy obiekt przenosi się na drugą krawędź ekranu.
7. Dodaj graczowi możliwość zmiany kierunku poruszania się `Thor`'a za pomocą naciśnięcia klawiszy w, s, a, d (góra, dół, lewo, prawo). Kierunek ruchu powinien zostać utrzymany nawet po puszczeniu klawisza.
8. Dodaj w klasie `Thor` obsługę żyć (początkowa wartość 3) oraz punktów (początkowa wartość 0).
9. Dodaj wykrywanie kolizji `Thor`'a z pozostałymi obiektami na scenie:
    - usuwaj obiekt, z którym wystąpiła kolizja,
    - kolizja z `StworOutrider`'em: przenieś `Thor`'a na pozycję startową i odejmij mu jedno życie,
    - kolizja z `KamienNieskonczonosci`: dodaj `Thor`'owi 100 punktów.
10. Przerwij grę, jeśli `Thor` straci wszystkie życia (komunikat w konsoli *PRZEGRANA*) lub uzbiera 600 punktów (komunikat w konsoli *WYGRANA*).
11. Dodaj obsługę myszki - kliknięty lewym klawiszem myszy `KamienNieskonczonosci` powiększa się dwukrotnie aż do osiągnięcia skali 8x.