

Wyścigi

Materiał opisuje założenia oraz etapy wykonania projektu "Wyścigi" i składa się z sekcji: "Wyścigi – scenariusz projektu", "Projektujemy działania duszków", "Tworzymy aplikację - budujemy skrypty", "Zwiększamy liczbę kibiców", "Zadania uzupełniające".

Materiał zawiera 5 zrzutów ekranu, 1 film, 12 ćwiczeń.

Wyścigi

Wyścigi – scenariusz projektu

Projekt będzie polegał na opracowaniu programu symulującego wyścigi. Od twojego wyboru zależy, jacy zawodnicy lub zawodniczki wezmą udział w konkurencji. Może to być bieg lekkoatletyczny, wyścig zwierząt, samochodów albo motocykli. Zawodom towarzyszyć powinna publiczność dopingująca swoich faworytów. Gdy konkurencja się zakończy, część publiczności będzie się cieszyła ze zwycięstwa swojego zawodnika, druga zaś będzie spokojnie stała po przegranej.
Zobacz przykładowe działanie aplikacji na poniższym filmie.



Film dostępny na portalu epodreczniki.pl

Źródło: Janusz Wierzbicki <janusz.wierzbicki@oeiizk.waw.pl>, licencja: CC BY 3.0.

Film: Przykładowe działanie programu symulującego wyścigi

Polecenie 1

Twoim zadaniem będzie zaprojektowanie i zaprogramowanie własnej wersji symulacji wyścigów.

Polecenie 2

Zaplanuj scenerię dla projektu - zastanów się, gdzie umieścisz publiczność, tory wyścigowe, punkt startu oraz mety. Czy użytkownik będzie na nich patrzył z boku, a może z góry?

Projektujemy działania duszków

Ćwiczenie 1

Zastanów się i zapisz przepis, według którego poruszać się będzie zawodnik. Pamiętaj, żeby zaplanować ruch w taki sposób, by szanse na zwycięstwo obu zawodników były takie same oraz by w czasie wyścigu mogli zmieniać się na prowadzeniu. Symulacja dzięki temu będzie bardziej dynamiczna.

Wskazówka

Możesz losować liczbę kroków, o jaką przesunie się dany zawodnik przed wykonaniem przez niego pojedynczego ruchu. Uwzględnij ustawienie zawodnika w punkcie startu oraz w jaki sposób użytkownik da sygnał do rozpoczęcia wyścigu, czyli uruchomi symulację. Pamiętaj też, że poruszanie się zawodnika powinno zakończyć się, gdy dotrze do mety.

Ćwiczenie 2

Zastanów się, w jaki sposób rozpoznać, że zawodnik osiągnął metę. Uzupełnij przepis.

> **Wskazówka**
>
> Czy zawodnik osiągnął metę można sprawdzić na kilka sposobów - na przykład:
>
> - poprzez sprawdzenie, czy jego współrzędna (*x* lub *y* - w zależności od tego, wzdłuż której osi i w jakim kierunku porusza się zawodnik) jest większa lub mniejsza od ustalonej pozycji mety,
> - dodanie duszka - mety lub oznaczenie mety specjalnym kolorem - gdy duszek - zawodnik dotknie go, rozpozna, że dotarł do mety.

Ćwiczenie 3

Zastanów się, w jaki sposób rozpoznać, że zawodnik wygrał lub przegrał.

> **Wskazówka**
>
> Czy zawodnik dodarł pierwszy, czy drugi, na metę można sprawdzić wykorzystując np. zmienne.
> Pierwszy sposób, to utworzenie dla każdego duszka - zawodnika zmiennej np. *Miejsce pierwszego zawodnika* oraz *Miejsce drugiego zawodnika*. Po uruchomieniu programu, wartości zmiennych należy ustawić na 0. Gdy zawodnik dotrze do mety, wartość zmiennej oznaczającej jego miejsce powinna zostać zmieniona na wartość zmiennej oznaczającej miejsce jakie zajmuje konkurent powiększoną o 1. Wówczas jeśli dotrze pierwszy, wartość ta wyniesie 1, jeśli drugi, to 2.
> Drugi sposób to utworzenie zmiennych przechowujących czas biegu poszczególnych zawodników. Przy każdym ruchu zawodnika można zapisywać czas, jaki upłynął od początku biegu korzystając z klocka stoper. Gdy zawodnicy dotrą do mety, w ich zmiennych będzie zapisany czas ich biegu. Wystarczy sprawdzić, który zawodnik dobiegł w krótszym czasie. W przypadku, gdy czasy są równe - mamy remis.
> Trzecie rozwiązanie polega na sprawdzeniu, gdy zawodnik dotarł do mety, czy rywal znajduje się za nim czy przed nim w danej chwili. Można w tym celu porównać współrzędne *x* lub *y* obu zawodników (w zależności od tego, wzdłuż której osi układu współrzędnych odbywał się wyścig). To rozwiązanie pozwala uniknąć stosowania zmiennych.

Ćwiczenie 4

Zastanów się, w jaki sposób powinien zachowywać się duszek - kibic?

> **Wskazówka**
>
> Kibic powinien dopingować zawodników (np. poruszając się) od momentu ich ustawienia na pozycji startowej, do końca zawodów. Następnie może krzyczeć „Hura! Wygraliśmy" dalej się poruszając lub „Buuu! Przegraliśmy" stojąc nieruchomo - w zależności od tego, czy jego zawodnik wygrał lub przegrał. Miejsce, które zajął dany zawodnik, kibic może sprawdzić, gdy ten dotrze do mety. W tym celu można wykorzystać mechanizm nadawania i odbierania komunikatów.

Jak można było zauważyć na filmie z przykładem działania programu, poszczególnym zawodnikom kibicuje całkiem duża grupa publiczności. Do powielenia pojedynczego kibica można użyć mechanizm klonowania duszków, który zostanie omówiony w dalszej części podręcznika.

Tworzymy aplikację - budujemy skrypty

Mając zaprojektowane działania dla poszczególnych duszków, możesz przystąpić do stworzenia skryptów.

Polecenie 3

Przygotuj tło sceny na potrzeby wyścigów, dodaj dwa duszki - zawodników oraz dwa duszki - kibiców (po jednym dla każdego zawodnika), ewentualnie inne duszki - jeśli zostały uwzględnione w twoim planie. Nie zapomnij odpowiednio nazwać każdego duszka.

Ćwiczenie 5

Stwórz skrypty dla każdego z duszków zgodnie z przygotowanymi algorytmami ich działania. Następnie przetestuj program. Uruchom go parokrotnie. Czy zawodnicy zwyciężają na zmianę? Czy każdego z zawodników dopinguje jeden kibic, który cieszy się, w przypadku zwycięstwa lub jest smutny, w przypadku przegranej?

Zwiększamy liczbę kibiców

Podczas zawodów jednego zawodnika lub drużynę dopinguje zazwyczaj wielu kibiców. Dlatego także w twojej symulacji wyścigów warto zadbać o odpowiednią liczbę widzów.

> **Ćwiczenie 6**
>
> Zastanów się, w jaki sposób możesz zmodyfikować algorytm działania duszka - kibica wykorzystując mechanizm klonowania, aby zwiększyć liczbę widzów dopingujących danego zawodnika.
>
> > **Wskazówka**
> >
> > W celu wykorzystania mechanizmu klonowania do zwiększenia liczby kibiców, należy dodać w algorytmie działania kibica, aby sklonował się określoną liczbę razy. Należy także przewidzieć, co ma zrobić, gdy zaczyna istnienie jako klon. Warto zauważyć, że klon nie wykona poleceń, które duszek wykonał po kliknięciu zielonej flagi. Dlatego w instrukcjach, które wykona, gdy zacznie jako klon, powinny znaleźć się m.in.: polecenie przesuwające go w odpowiednie miejsce na scenie (inaczej zacznie dokładnie w miejscu, w którym znajduje się oryginalny duszek) oraz np. instrukcje animacji jego postaci.

Zadania uzupełniające

> **Ćwiczenie 7**
>
> Dodaj do aplikacji wyświetlanie czasów poszczególnych zawodników w podobny sposób, jak widać na filmie z przykładową aplikacją.

> **Ćwiczenie 8**
>
> Dodaj do aplikacji wyświetlanie miejsca, które zajął dany zawodnik, w podobny sposób, jak widać na filmie z przykładową aplikacją.

Ćwiczenie 9

Utwórz nową wersję aplikacji i dodaj do niej dodatkowego zawodnika (wraz z kibicującymi mu widzami), czasem, który osiągnął na mecie oraz miejscem, które zajął.

Ćwiczenie 10

Dodaj do aplikacji dekorację zawodników na podium, po zakończonym wyścigu. Jeśli w twoim programie ścigają się samochody, dekorowani powinni być ich kierowcy.

Ćwiczenie 11

Przeprowadź kilkanaście wyścigów i zanotuj ile razy wygrali poszczególni zawodnicy. Wyniki porównaj z kolegami. Czy można przyjąć, że przy odpowiednio wielu próbach, każdy z zawodników wygra mniej więcej tyle samo razy?

Ćwiczenie 12

Zmodyfikuj aplikację w taki sposób, aby automatycznie po zakończeniu wyścigu startował następny. Niech program automatycznie zlicza i wyświetla łączną liczbę wygranych dla poszczególnych zawodników. Uruchom program i obserwuj wyniki. Czy twoje wcześniejsze obserwacje potwierdzają się?