

Temat: Wyścig na czas

Przedmiot: Informatyka

Autor: Sylwester Zasoński

Czas trwania: 1-2h lekcyjne kl 4-6

Opis scenariusza:

Nauczysz się programować robota odmierzającego czas przejazdu toru wyścigowego. Codey rocky będzie symulował start wyścigu na czas a także zapisze i pokaże czas przejechanego okrążenia. Jeżeli wcześniej twoi uczniowie nie uczyli się programować sterowania robotem przeprowadź z nimi najpierw zajęcia według scenariusza "Sterujemy robotem z poziomu komputera" dostępnego na stronie [kodowanienaekranie.pl](https://www.kodowanienaekranie.pl). Poniżej bezpośredni link do scenariusza:

<https://www.kodowanienaekranie.pl/scenariusze/codeyrocky/sterujemy-robotem-z-poziomu-komputera.pdf>

Cele ogólne:

- Rozwijanie kompetencji miękkich (umiejętność pracy zespołowej, logiczne, algorytmiczne myślenie)
- Pobudzanie kreatywności
- Łączenie kluczowych obszarów rozwoju

Cele operacyjne:

Uczeń:

- planuje i realizuje własne projekty/prace; realizując te projekty/prace współdziała w grupie
- posługuje się komputerem lub innym urządzeniem cyfrowym oraz urządzeniami zewnętrznymi przy wykonywaniu zadania
- współpracuje z uczniami, wymienia się z nimi pomysłami i doświadczeniami, wykorzystując technologię

Metody:

Poszukujące, eksponujące, praktycznego działania

Środki dydaktyczne:

1. Minimum 2 roboty Codey Rocky
2. Komputer i aplikacja mBlock5

Przygotowanie:

Przygotuj materiały do stworzenia toru wyścigowego np. taśma malarska do wyklejenia trasy na podłodze.

Przebieg zajęć:

Zajęcia zacznij od pogadanki na temat sportów motorowych/kolarstwa. Spytaj uczniów czy mieli okazję takowe oglądać/uczestniczyć na żywo. Pokieruj rozmową tak aby pojawił się element kwalifikacji do wyścigów (np. Formuła 1) lub jazdy na czas jako elementy wyścigów kolarskich. Ponieważ projekt, którym się zajmiecie będzie dotyczył właśnie jazdy pojedynczego robota na czas (okrażenie) a nie wyścigu wielu robotów.

Do projektu użyte zostaną 2 roboty, pierwszy będzie tym jeżdżącym, drugi natomiast posłuży jako start i meta i programowaniem tego drugiego zajmiesz się przy pomocy tego scenariusza.

Po skończonym wstępie i wyjaśnieniu koncepcji zadania pora na uruchomienie aplikacji mblock5. Na początek procedura startu. Przyjmijmy, że będzie to odliczanie od 3 (3..2..1...start), które zostanie uruchomione w momencie naciśnięcia **A**

kiedy przycisk A jest wciśnięty

na robocie.

Zapytaj jakie 2 bloki będą potrzebne? Co wyświetli robot i ile ma czekać pomiędzy wyświetlaniem?

Najprostszy sposób to użycie i powielenie dwóch poniższych bloków. Gdzie robot pokazuje odliczanie i czeka 1 sekundę pomiędzy kolejnymi cyframi odliczanymi 3,2,1.



Aby było “bardziej widowiskowo” kod możesz rozbudować o kolory np.
Czerwony->żółty->zielony.

Gotowy kod startu powinien wyglądać tak:



Dalszą część dopiszesz do istniejącego już fragmentu, całość będzie wywoływana przyciskiem A.

Zastanów się wspólnie z uczniami, które z czujników robota mogą posłużyć do mierzenia czasu i jak to mogło by wyglądać?

W większości wyścigów na mecie stosuje się czujniki rejestrujące czas jej przekroczenia, czy to w postaci nacisku na linię/pas, użyciu specjalnego nadajnika wykrywanego przez urządzenie rejestrujące czy chociażby kamer.

W przypadku naszego Codey Rocky czujnik, który można użyć to czujnik wykrywania przeszkody.

Dalszy kod powinien wyglądać następująco:

Uruchom stoper – kiedy zobaczę przeszkodę z przodu (mijający mnie robot) zatrzymaj stoper i wyświetl wynik.



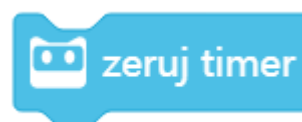
Bloki potrzebne do utworzenia tej części kodu znajdują się w kategoriach kontrola i czujniki.

Zerknijmy najpierw, które bloki z czujników będą potrzebne.

Skoro robot ma odmierzać czas to oczywiście:



ale przyda się również



ponieważ na

początku musimy ustawić wartość stopera na 0.

przeszkody z przodu?

Trzecim blokiem będzie , posłuży on do wykrycia przejeżdżającego robota.

Teraz pora na kategorię kontrola. Po pierwsze potrzebujemy warunek jeżeli



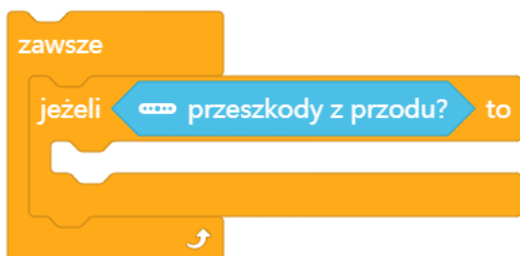
Ten warunek ma za zadanie wywołać pokazanie przejechanego czasu na ekranie robota służącego za start/metę - *Jeżeli przeszkoda z przodu wykryta pokaż czas z timera.*

Zastanówcie się wspólnie czy warunek ma być sprawdzony tylko raz czy ma być sprawdzany nieustannie w oczekiwaniu na nadejście robota?

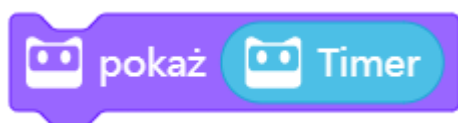
Oczywiście pojedyncze wywołanie tego warunku nic nam nie da, bo zostanie sprawdzony zaraz po wystartowaniu, kiedy robota jeszcze nie będzie, zatem potrzebna jest pętla zawsze.



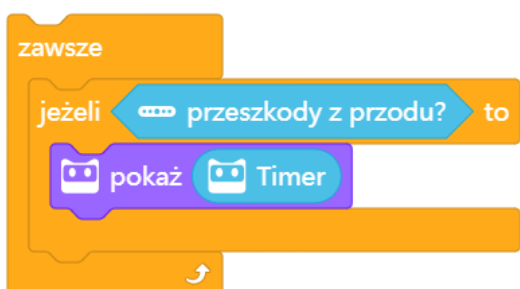
Warunek umieszczamy w środku pętli, a czujnik wykrywania przeszkody zagnieżdżamy w warunku.



Teraz pora na zaprogramowanie tego co się wydarzy, kiedy warunek zostanie spełniony. Pamiętaj, robot ma pokazać czas, który weźmie z timera.



Powyższy blok umieść w środku warunku



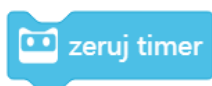
A cały fragment dołącz do istniejącego już kodu. Możesz śmiało przetestować to co do tej pory zostało napisane.

Działa?

Początek powinien działać dobrze, ale przejechany czas różni się od prawidłowego.

Teraz zajmijmy się tym co w programowaniu nazywa się **debugging** czyli wychwytywaniu błędów w kodzie i ich naprawianiu.

Zabrakło istotnego elementu na początku drugiej części kodu, zaraz przed pętlą



zawsze. Jest to pokazane wcześniej , które powoduje wyzerowanie stopera zaraz po ukazaniu się na ekranie napisu "start". Dodanie tego blokcza powinno naprawić cały kod. Możesz spróbować ponownie, ale...

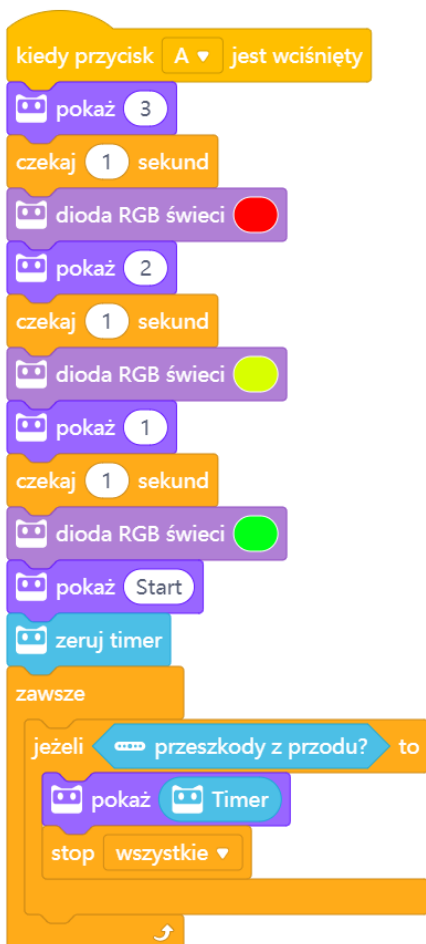
Próbowaleś/aś (po zliczeniu czasu) ponownie uruchomić kod przyciskiem A?

Nie działa, prawda? A to dlatego, że potrzebujesz jeszcze jednego bloku z kategorii kontrola, a jest to zatrzymaj wszystkie kody



Powinien zostać wstawiony w warunek zaraz po pokazaniu wyniku czyli bloczku pokaż.

Kompletny i gotowy kod wygląda następująco:



Teraz możesz się już ścigać na czas. Dobrej zabawy!