

Programowanie gier w środowisku **Kodu Game Lab**

Podręcznik dla uczniów klas szkoły podstawowej

Autor: Wojciech Drankiewicz

Spis treści

Spis treści	1
Od autora	2
Instalacja Kodu Game Lab	2
Programowanie w Kodu	3
Pierwsze uruchomienie programu	3
Budowa okna	3
Narzędzia do edycji	4
Programowanie obiektów	5
Budowanie ścieżek	6
Korzystanie z wyników - zmienne	7
Dodawanie stron w skrypcie	7
Priorytetowanie zadań	7
Dynamiczne tworzenie obiektów	8
Zagnieżdżanie akcji i warunków	8
Zajęcia 1 - Kodu poznaje świat	9
Zajęcia 2 - Labirynt	11
Zajęcia 3 - Wyścig	13
Zajęcia 4 - Piłka Nożna	15
Zajęcia 5 - Pacman	17
Zajęcia 6 - Owczy Pęd	20
Zajęcia 7 - Gwiezdna Walka	23
Zajęcia 8 - Gra Platformowa	26
Zajęcia 9 - Obrona Fabryki	31
Zajęcia 10 - Dwie Wieże	36
Zajęcia 11 - Kodu City - tworzenie świata	39

Od autora

Niniejszy skrypt powstaje w oparciu o pracę z darmowym programem *Kodu Game Lab* wydanym przez firmę Microsoft. W dniu rozpoczęcia pracy nad materiałem aktualną wersją była 1.5.47.0. Warto zwrócić uwagę, że wraz z biegiem czasu realizowanie prezentowanych przykładów może nieco różnić się od stanu faktycznego.

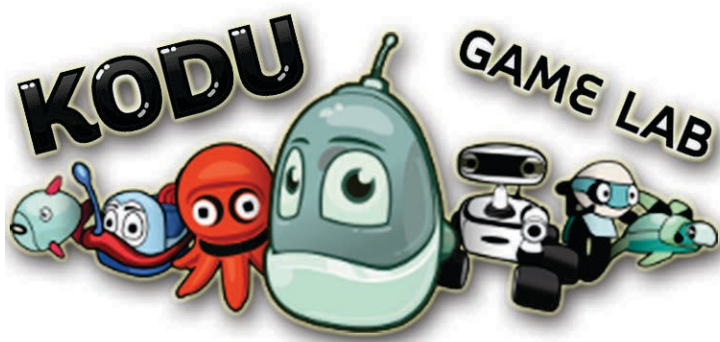
Podręcznik obejmuje krótki opis instalacji programu oraz instrukcję 15 zajęć, na realizację których należy poświęcić 2 godziny dydakcyjne.

Instalacja Kodu Game Lab

Kodu Game Lab to aplikacja, która pozwala tworzyć gry z pomocą prostego języka programowania. Korzystanie z *Kodu* rozwija kreatywność, pomaga radzić sobie z rozwiązywaniem problemów, uczy zarówno tworzenia nietuzinkowych historii, jak i kodowania. Z tego programu może korzystać każdy - nawet te osoby, które nie mają żadnego doświadczenia w projektowaniu czy skryptowaniu gier.

Aplikacja została wydana w 2013 roku i do dziś dzień uzyskuje regularne aktualizacje.

Producent zaleca, by korzystały z niej osoby w wieku powyżej 7 lat.



Plik instalacyjny pobrać można ze strony:

<https://www.microsoft.com/en-us/download/confirmation.aspx?id=10056>.

W przypadku, gdy link stanie się nieaktywny, wystarczy odwiedzić domenę <https://www.microsoft.com> i tam z pomocą opcji "wyszukaj" odnaleźć interesujący nas program.

Pobrany na komputer plik uruchamiamy i postępujemy zgodnie z poleceniami instalatora. Po kilku minutach jesteśmy gotowi na nasze pierwsze spotkanie z aplikacją *Kodu Game Lab*.

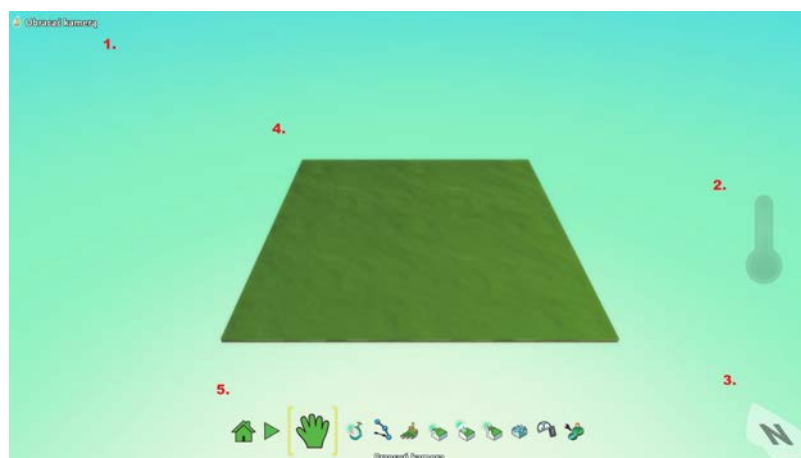
Programowanie w Kodu

Pierwsze uruchomienie programu



Po uruchomieniu programu pojawia się przed nami ekran powitalny, na środku którego widnieje menu główne. Możemy się po nim poruszać zarówno z pomocą klawiszy strzałek, jak i kursora myszy. Pracę rozpoczynamy wybierając opcję *NEW WORLD*, która tworzy nowy projekt.

Budowa okna



Po utworzeniu nowego projektu na ekranie pojawia się gotowa do edycji plansza. Warto zatem przyrzeć się budowie okna. Składa się z następujących elementów:

1. Podpowiedzi
2. Wskaźnik zasobów
3. Kompas
4. Wykreowany świat
5. Narzędzia do edycji

1. Podpowiedzi

W lewym górnym rogu okna znajduje się krótki opis akcji wraz z ikonami dotyczącymi klawiszy myszy i klawiatury, jakie można w danym momencie zrealizować.

2. Wskaźnik zasobów

Znajdujący się po prawej stronie ekranu “termometr” informuje zarówno użytkownika jak i twórcę gry o tym czy dana liczba obiektów w projekcie pozwala na jego płynne działanie. Jeżeli wskaźnik pozostaje w zielonym zakresie, praca nie powinna napotykać przeszkód takich jak klatkowanie czy niespodziewane wyłączenie edytora.

3. Kompas

Strzałka w prawym dolnym rogu ekranu ułatwia nawigację w projekcie. Tworzenie gier w środowisku 3D bywa czasem kłopotliwe. Kierunek wskazywany przez kompas powinien rozwiewać wszelkie wątpliwości.

4. Wykreowany świat

Podgląd świata zawiera zarówno budowę planszy, jak i wszystkie programowalne obiekty.

5. Narzędzia do edycji

Dokładny opis znajduje się poniżej.

Narzędzia do edycji

Home menu

Jest to Menu Główne. Po kliknięciu w zielony domek na ekranie wyświetlają się następujące opcje:

- Restart - resetuje postęp w rozgrywce poza strefą edycji.
- Edytuj świat - przywraca do strefy edycji.
- Zapisz mój świat - zapisuje postęp w kreacji projektu.
- Wczytaj świat - wczytuje wcześniej zapisany świat.
- Nowy pusty świat - tworzy nowy, pusty projekt.
- Drukuj kod dla świata - drukuje listę wszystkich komend każdego z zaprogramowanych obiektów.
- Wyjście do menu głównego - przywraca użytkownika do Menu powitalnego edytora.



Graj grę

Wybierając zielony znaczek "play" użytkownik uruchamia aktualnie przygotowany projekt.



Przesuń kamerą

Zaznaczając rączkę użytkownik może swobodnie zmieniać ustawienie kamery, która obserwuje kreowany świat - zarówno jej pozycję, oddalenie jak i kąt widzenia.

Przytrzymując lewy przycisk myszy zmieniamy współrzędne kamery. Korzystając z prawego klawisza zmieniamy perspektywę obserwacji. Za pomocą scroll'a oddalamy i przybliżamy widok.



Dodaj i programuj obiekty

Wybierając Kodu, twórca gry może dodać oraz programować obiekty dostępne w bibliotece środowiska. Narzędzie również pozwala na:

- Zmianę ustawień dla konkretnego bota.
- Wycinanie i kopiowanie obiektów.
- Zmianę parametrów obiektów.



Dodaj i edytuj ścieżkę

Wybranie ścieżki pozwala na tworzenie drogi. Dokonuje się tego poprzez wyznaczenie połączonych ze sobą punktów. Tak utworzona trasa może zostać wykorzystana do sterowania dowolnego obiektu. Dodatkowe opcje narzędzia to:

- Dodaj więcej węzłów.
- Zmień wysokość (pozwala na podniesienie ścieżki w górę, tak aby zbudować most).
- Zmień rodzaj – udostępnia różne faktury ścieżek: drogi, barierki.
- Usuń węzeł.



Rysuj, dodaj lub usuń podłoże

Narzędzie pozwala na niemal dowolne kreowanie planszy, na której toczyć się będzie rozgrywka. W lewym górnym rogu znajduje się lista materiałów, z jakich można utworzyć podłoże. Domyślnie materiał składa się z prostopadłościanów. Używając klawisza *strzałka w dół*, zmienia się na kształt obły.

Podmenu w prawym górnym rogu pozwala na zmianę kształtu pędzla. Dodatkowo rozmiar pędzla można ustalić klawiszami strzałek.



Podnieś lub obniż podłoże

Narzędzie pozwala na wymodelowanie wzniesień o prostokątnym lub łagodnie zaokrąglonym szczycie. Podobnie, jak w przypadku dodawania podłoża, użytkownik może zmieniać pędzel, z którego korzysta.



Flatten: Make ground smooth or level – Wyrównywanie: wygładzenie oraz tworzenie jednolitego poziomu terenu

Narzędzie to, tak jak nazwa wskazuje, pozwala w zależności od przycisków myszy na:

- Lewy przycisk – wygładzenie wzgórz,
- Środkowy przycisk (scroll) – wygładzenie względem terenu,
- Prawy przycisk – tworzenie jednolitego poziomu.



Roughen: Create spikey or hilly ground – Wypiętrzenie: górzysty bądź pagórkowaty teren

Jest to narzędzie, które pozwala na podniesienie terenu ostrymi lub łagodnymi wypiętrzeniami, tworząc tym samym pagórki:

1. Lewy przycisk myszy – ostre szczyty wzniesień,
2. Środkowy przycisk (scroll) – wygładzenie względem terenu,
3. Prawy przycisk – tworzenie pagórków.



Dodaj, podnieś lub obniż wodę

Wybór tego narzędzia umożliwia dodanie różnych cieczy do struktury terenu. Z pomocą podmenu można dokonać wyboru umieszczanej cieczy.



Usuń obiekty

Narzędzie pozwala na szybkie usunięcie obiektów ze wskazanego przez użytkownika obszaru. W prawym górnym rogu, znajduje się wybór kształtu pędzla.

Zmień ustawienia świata

Wybierając tę ikonę uzyskuje się dostęp do wielu opcji, które w zależności od ustawień, zmieniają obraz świata oraz jego poszczególne funkcje.



Programowanie obiektów

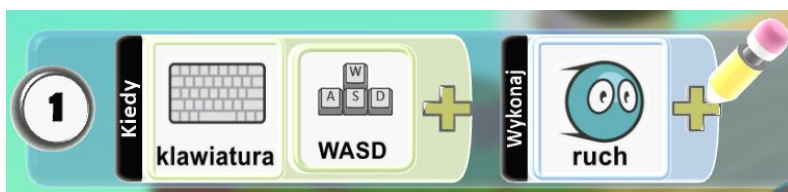
Programowanie w *Kodu Game Lab* polega na kodowaniu obiektów znajdujących się w grze. Skrypty te przygotowywane są na zasadzie *akcji i reakcji*. To znaczy, że umieszczony na planszy obiekt zareaguje wtedy i tylko wtedy, gdy użytkownik doprowadzi do sytuacji, kiedy jeden z warunków przygotowanych przez programistę zostanie spełniony.

Przykładem takiego programu niech będzie skrypt znajdujący się poniżej.

Każda z linii skryptu składa się z dwóch elementów: pola KIEDY oraz pola WYKONAJ.

Pierwsze pole służy do utworzenia warunku, jaki musi zostać zrealizowany, aby obiekt, którego dotyczy skrypt, zareagował w pożądanym sposób. Należy zwrócić uwagę, że warunek ten

buduje się wielostopniowo - od ogółu do szczegółu. Taka metoda formułowania pola KIEDY daje możliwość na



szybkie rozeznanie się w tym, na co zareagować może programowany przez nas obiekt. Wystarczy wybrać odpowiednią grupę, a następnie rozszerzyć komendę klikając w pojawiający się obok plus.

Pole WYKONAJ przygotowuje się w analogiczny sposób. To jednak reprezentuje akcję, jaką wykona programowany przez nas obiekt.

Baza komend KIEDY oraz WYKONAJ jest inna dla różnych obiektów.



Budowanie ścieżek

Ścieżki w *Kodu Game Lab* pozwalają na dodanie do planszy elementów takich jak ulice, ogrodzenia czy kwiaty. Ważniejszym aspektem jest jednak możliwość wykorzystania ich w celach programistycznych - wskazanie obiektom drogi po jakiej mają się poruszać.

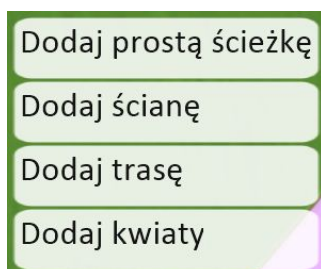
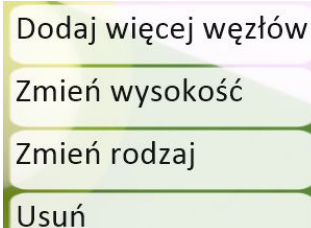


Gdy aktywne jest narzędzie *Dodaj i edytuj ścieżkę*, kursorem myszy należy wskazać miejsce, w którym ma się znaleźć węzeł. Kliknięcie lewym przyciskiem myszy sprawi, że na planszy pojawi się migocząca kula - element ścieżki pozostający niewidoczny w trybie rozgrywki. Tak długo, jak nie użyjemy prawego przycisku, nowe punkty będą łączyły się tworząc spójną strukturę.

Jeżeli gotowa ścieżka zostanie wskazana kursorem, z pomocą klawiszy LEWO-PRAWO, istnieje możliwość zmiany jej koloru. Ma to znaczenie podczas programowania, gdyż obiekty rozróżniają trasy na podstawie ich barwy.



Użycie prawego klawisza myszy na węźle otwiera podręczne menu, które pozwala na dodanie nowych punktów do istniejącej ścieżki, zmianie ich wysokości oraz rodzaju.



Wciśnięcie prawego przycisku myszy w miejscu, w którym ma znaleźć się węzeł nowej ścieżki sprawi, że otworzy się nowe menu, które pozwala użytkownikowi na dodanie elementu innego

rodzaju: ściany, trasy lub kwiatów. Te dalej będą rozpoznawane przez obiekty jako ścieżki o konkretnym kolorze, lecz w trybie rozgrywki staną się widoczne i przyjmą wybraną przez autora formę.





Korzystanie z wyników - zmienne

Zmienne są szeroko znanymi strukturami programistycznymi. W *Kodu Game Lab* kryją się pod postacią wielokolorowych wyników, punktów globalnych oraz prywatnych oznaczonych literami dużymi i małymi. Niestety, zmienne w takiej formie mogą przechowywać jedynie wartości liczbowe oraz jest ich skończona liczba, lecz na potrzeby nauki programowania zasób ten jest wystarczający.

Do wyników możemy odwoływać się za pomocą skryptów pisanych dla obiektów. Z punktu widzenia pola KIEDY możemy sprawdzić i porównywać ich wartości. Wykonać możemy następujące działania: ustawić wartość, dodać do oraz odjąć od wartości.



Sposób wyświetlania zmiennej możemy zmienić w narzędziu *Zmień ustawienia świata*.

Dzięki tak przygotowanej strukturze, programista może przysłać informacje z pomocą wartości liczbowych i w taki sposób wyzwać część skryptów.



Dodawanie stron w skrypcie

Kod przygotowywany dla obiektu nie musi ograniczać się tylko do jednego trybu pracy. Jeżeli obiekt ma działać w dany sposób tylko w określonych sytuacjach, a w przypadku zaistnienia odpowiednich warunków porzucić te zachowania, to skorzystanie z nowej strony jest najlepszym rozwiązaniem.

Obiekt realizuje tylko te komendy, które znajdują się na jego aktywnej stronie. To pozwala na takie napisanie kodu, aby wybrana postać realizowała zupełnie inne zadania dla konkretnych warunków globalnych.

Przykładem zastosowania stron jest Duch w projekcie Pacman. Ten najpierw ściga protagonistę, by po chwili panicznie uciekać, gdy bohater zje specjalny owoc.



Priorytetowanie zadań

Numer znajdujący się przy wierszu nie jest jedynie liczbą porządkową. Określa on również wagę przyporządkowanego do niego zadania. Komenda, przy której znajduje się niższa wartość, posiada wyższy priorytet. W wielu przypadkach nie jest to niezauważalne podczas realizacji skryptu, lecz sytuacja ta ulega zmianie, gdy kilka wierszy w ramach swoich reakcji odnosi się do rozkazów z tej samej grupy.

Przykład wykorzystania priorytetów znajduje się na zamieszczonym obok obrazku. We wszystkich wierszach obiekt w polu "Wykonaj" ma zrealizować rozkaz "ruch", lecz te różnią się względem siebie kierunkiem oraz prędkością. Gdy spełniony jest więcej niż jeden warunek "Kiedy", postać nie byłaby w stanie poruszać się na dwa sposoby naraz. W takim wypadku zostanie wykorzystana zasada priorytetów i to wiersz o najwyższym z nich zostanie wywołany.



Dynamiczne tworzenie obiektów

W *Kodu Game Lab* istnieje możliwość dynamicznego tworzenia obiektów w trakcie trwania rozgrywki. Oznacza to tyle, że programista może przygotować jeden z nich jako wzór, a następnie powielać go w zależności od potrzeb rozgrywki zamiast dodawać każdego z nich ręcznie.

Aby było to możliwe, wybrany obiekt wzorcowy musi mieć zaznaczoną opcję “Do utworzenia” w ustawieniach. Wówczas ten stanie się niewidoczny i nieaktywny w podglądzie rozgrywki tak długo, jak nie zostanie utworzony.

W celu utworzenia obiektu należy w skrypcie w polu “Wykonaj” z komend typu “akcje” wybrać “utwórz”, a następnie dodać obiekt z puli “tworzone”.



Zagnieżdżanie akcji i warunków

Podczas przygotowania skryptu programista może wybrać dowolny wiersz i wsunąć go głębiej. Skutkuje to tym, że linia, która ma większy margines jest zależna bezpośrednio od komendy poprzedzającej, która została osadzona płycej. Oznacza to tyle, że komenda wsunięta będzie rozpatrywana przez obiekt tylko wtedy, gdy warunek nadrzędnej linii będzie spełniony.

Na przykładzie po prawej przyznanie punktów czerwonych z wiersza czwartego nastąpi dopiero, kiedy warunek linii trzeciej zostanie spełniony.

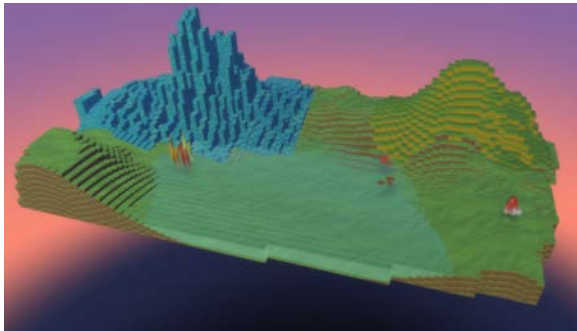


W przykładzie po prawej widnieje sytuacja, w której komendy są zagnieżdżone podwójnie.

Odejście punktów żółty w drugiej linii nastąpi dopiero, gdy warunki z wierszy pierwszego oraz drugiego zostaną spełnione. To samo obostrzenie nałożone jest na komendy z trzeciego i czwartego wiersza.



Zajęcia 1 - Kodu poznaje świat



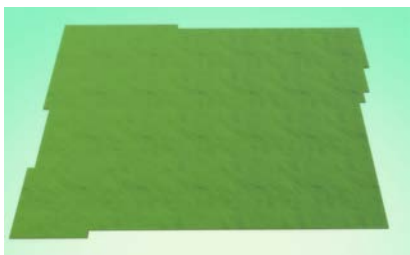
Cel

Podczas realizacji tego projektu uczniowie mają okazję zapoznać się z wykorzystaniem nowo poznanych narzędzi do kreowania świata.

Efekt

Uczniowie przygotowują nieduży świat z zaprogramowanym głównym bohaterem, którym będą mogli eksplorować utworzoną planszę.

Konstrukcja poziomu



Krok 1

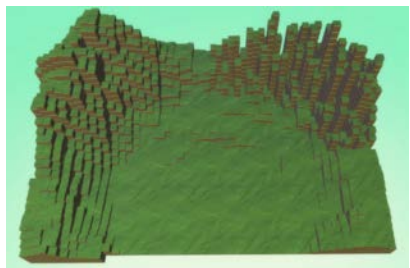
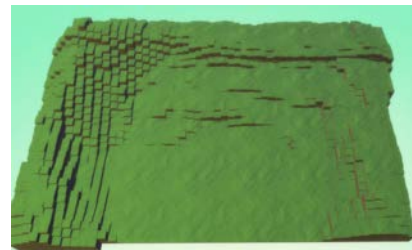
Z pomocą narzędzia *Rysuj, dodaj lub usuń podłoże* należy przygotować niewielki obszar, który zostanie poddany dalszej edycji.

Obrys obszaru może być dowolny - nada się zarówno owalny, jak i prostokątny. To samo dotyczy faktury terenu - choć w przykładzie wykorzystano jedną, można pokusić się o wielobarwną powierzchnię. Sugerowanym kształtem pędzla jest pędzel okrągły lub kwadratowy.

Krok 2

Wykorzystując narzędzie *Podnieś i obniż podłoże*, wybierając dowolny pędzel, należy wypiętrzyć obrys powstałej planszy.

Wypiętrzenie wykonuje się naciskając lewy przycisk myszy - wówczas miejsce wskazane przez pędzel podniesie się. Jeżeli skorzysta się z "miękkiego" pędzla, powierzchnia będzie podnosiła się w różnym tempie - szybciej w centrum, wolniej na obrzeżach.

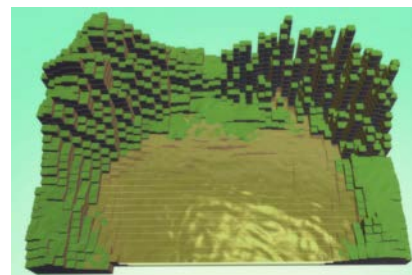


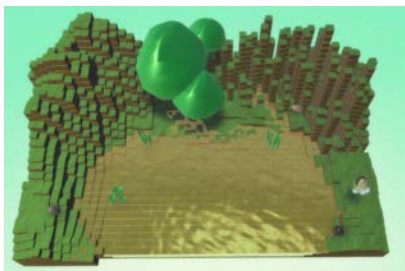
Krok 3

Używając narzędzia *Roughen - Wypiętrzenie* wzbogacamy krajobraz o ostre i łagodne wzniesienia. Pędzlem należy wskazać obszar wcześniej już wypiętrzony (prawy górny róg planszy). Można następnie wykorzystać narzędzie *Flatten - wyrównanie*, aby przemienić ostre grzbiety w strzeliste wzniesienia (lewy górny róg).

Krok 4

Narzędziem *Dodaj wodę* należy wypełnić uzyskane we wcześniejszych krokach wgłębienie. Kolor cieczy jest w tym wypadku dowolny. Lewym przyciskiem myszy można podnieść poziom cieczy, prawym natomiast opuścić.





Krok 5

Korzystając z *Dodaj obiekty* umieszczamy na przygotowanej planszy dowolne elementy otoczenia takie jak drzewa, rośliny i kamienie. Należy umieścić również Kodu.

W tym celu należy wskazać kursorem miejsce na planszy i użyć lewego przycisku myszy. Z otwartego menu należy wybrać pożądany obiekt.

Przytrzymując lewy przycisk na powstałym obiekcie możemy go przesunąć zmieniając położenie kursora myszy.

Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/ przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.

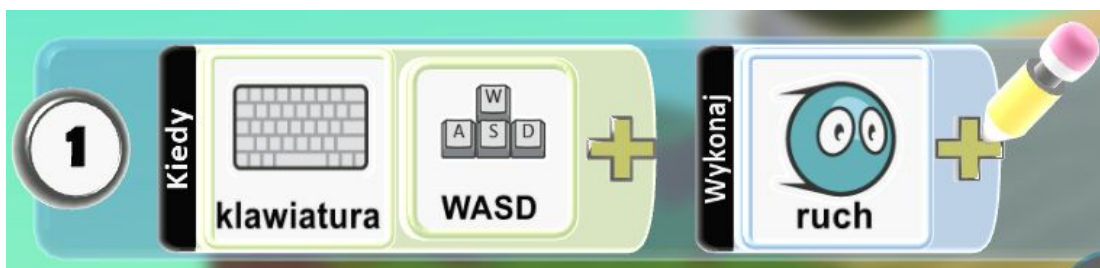


Kodu

Dla obiektu Kodu należy utworzyć jedną linijkę programu wybierając następujące komendy:

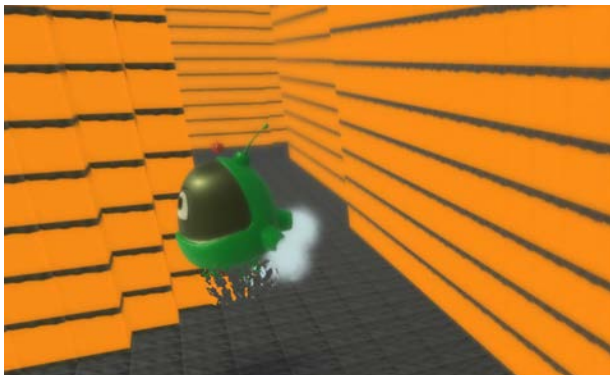
- 1) Kiedy: klawiatura + WASD; Wykonaj: ruch

Tak przygotowany kod sprawi, że gracz wciskając klawisze W, A, S oraz D wprawi w ruch programowany obiekt.



Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 2 - Labirynt



Cel

Podczas realizacji tego projektu uczniowie poznają w jaki sposób przebiega programowanie obiektów w *Kodu Game Lab*.

Efekt

Uczniowie przygotowują labirynt z zaprogramowanym głównym bohaterem, którym będą mogli eksplorować utworzoną planszę.

Konstrukcja poziomu



Krok 1

Z pomocą narzędzia *Rysuj, dodaj lub usuń podłoże* należy przygotować kilka płaskich korytarzy. Na początku, używając prawego przycisku myszy, należy "wymazać" teren powstały przy uruchomieniu nowego projektu.

Do tworzenia korytarzy należy wybrać pędzel, który można "przeciągać". Korzystając z niego wystarczy wskazać dowolne miejsce na planszy i przytrzymać lewy przycisk myszy. Przesuwając kursor można dostrzec, że cień pędzla rozciąga się. Po zwolnieniu przycisku powstaje podłoże w obszarze, który został w ten sposób zaznaczony.

Krok 2

Wykorzystując narzędzie *Podnieś i obniż podłoże* należy wybierać magiczny pędzel. Zaznacza on cały obszar wskazanego kursorem myszy rodzaju. Po przytrzymaniu lewego przycisku myszy cała plansza podnosi się.



Krok 3

Nie zmieniając aktywnego narzędzia, używając klawisza strzałki w lewo należy zmniejszyć rozmiar magicznego pędzla. W ten sposób obrys korytarzy przestanie być zaznaczony. Przytrzymanie prawego przycisku myszy sprawi, że obszar ponownie się obniży.

Krok 4

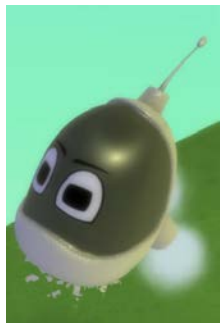
Za pomocą narzędzia *Dodaj obiekty* należy umieścić na planszy następujące obiekty:

- Kodu - w centrum planszy,
- Chmura - jeden z rogów planszy,
- kilka Jablek - dowolne pozycje.



Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy najechać kursorem myszy na postać lub przedmiot, który chcemy zaprogramować, a następnie kliknąć prawy przycisk myszy.

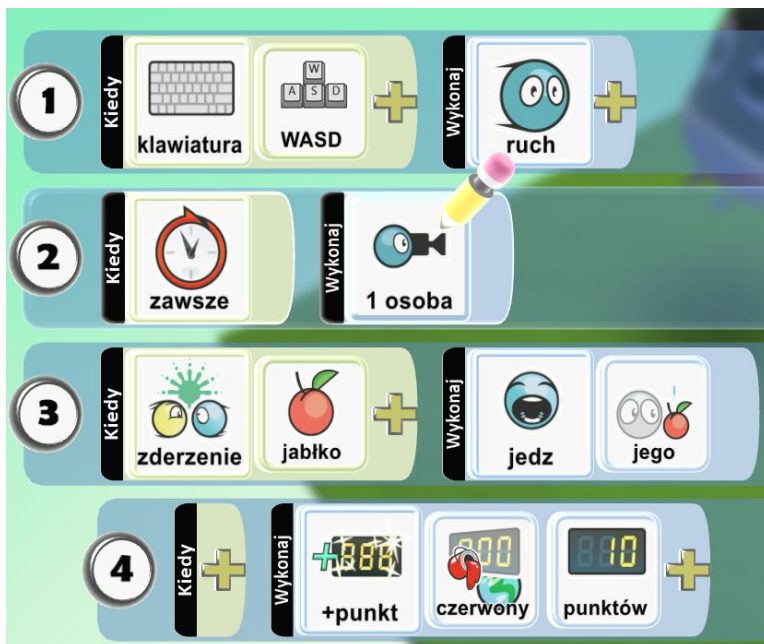


Kodu

Dla obiektu Kodu przygotowujemy kilka linii programu wybierając następujące elementy:

- 1) Kiedy: klawiatura + WASD; Wykonaj: ruch
- 2) Zawsze: kamera Pierwsza osoba
- 3) Kiedy zderzenie z Jabłkiem - zjedz je
- 4) ORAZ dodaj punkty - czerwone - 10 punktów

Pierwsza linia kodu oznacza, że kiedy gracz korzysta z klawiszy W, A, S oraz D, Kodu się porusza. Drugi wiersz wymusza na kamerze, aby ta zawsze wykonywała rzut z pierwszej osoby. Trzeci i czwarty wiersz są ze sobą związane. Jeżeli jedna linia zostanie wsunięta, oznacza to, że ta staje się zależna od linii poprzedniej. Konstrukcja omawiana w tym przykładzie sprawia, że kiedy spełni się jeden warunek, wykonają się dwie instrukcje. Tutaj skrypt realizuje się, gdy Kodu zderzy się z dowolnym jabłkiem - to zostaje wówczas zjedzone, a punkty Czerwone zwiększają się o 10.



Chmura

Jeżeli obiekt znajduje się zbyt wysoko, aby mógł wejść w interakcję z Kodu, wystarczy zmienić jej wysokość. Można tego dokonać z aktywnym narzędziem *Dodaj i programuj obiekty* i klikając na chmurze prawym przyciskiem myszy.

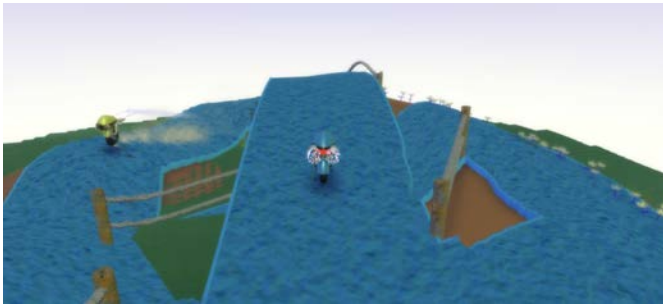
- 1) Kiedy zderzenie z Kodu - gra: Wygrana

Skrypt chmury sprawia, że kiedy ta wejdzie w kolizję z Kodu, gra kończy się zwycięstwem. Krótko pisząc - odnalezienie chmury staje się celem gracza.



Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 3 - Wyścig



Cel

Podczas realizacji tego projektu uczniowie poznają w jaki sposób buduje się ścieżki oraz w jaki sposób można je wykorzystać.

Efekt

Uczniowie przygotują trasę wyścigu, na której będą mogli zmierzyć się z zaprogramowanym botem.

Konstrukcja poziomu

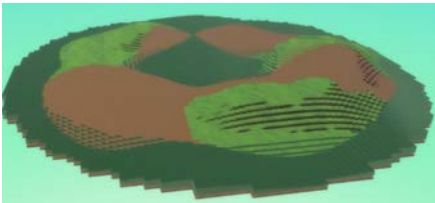
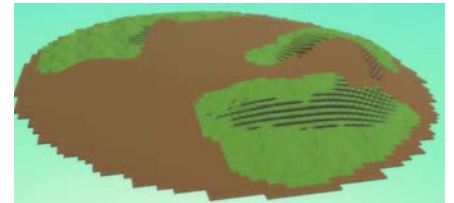
Krok 1

Z pomocą narzędzia *Rysuj, dodaj lub usuń podłoże* należy przygotować teren, na którym ustawiona będzie trasa wyścigu. Kształt oraz rodzaj tekstury może pozostać dowolny - należy jednak pamiętać, aby plansza była w stanie pomieścić trasę.



Krok 2

Wykorzystując narzędzie *Podnieś i obniż podłoże* można nadać planszy głębi - wzniesienia nie powinny być nadbyt strome.



Krok 3

Używając narzędzia *Dodaj wodę* można na planszy umieścić niski poziom ciecży.

Krok 4

Za pomocą narzędzia *Dodaj ścieżkę* należy nanieść na planszy różne trasy. Tę obowiązkową - tor wyścigu - można umieścić wybierając z menu, które otwiera się przy pomocy prawego przycisku myszy, opcję *Dodaj trasę*.

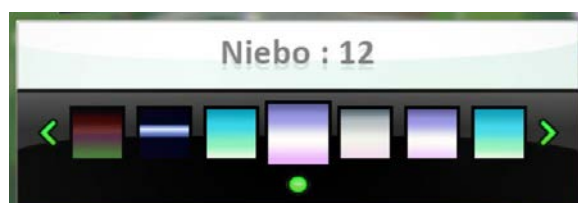
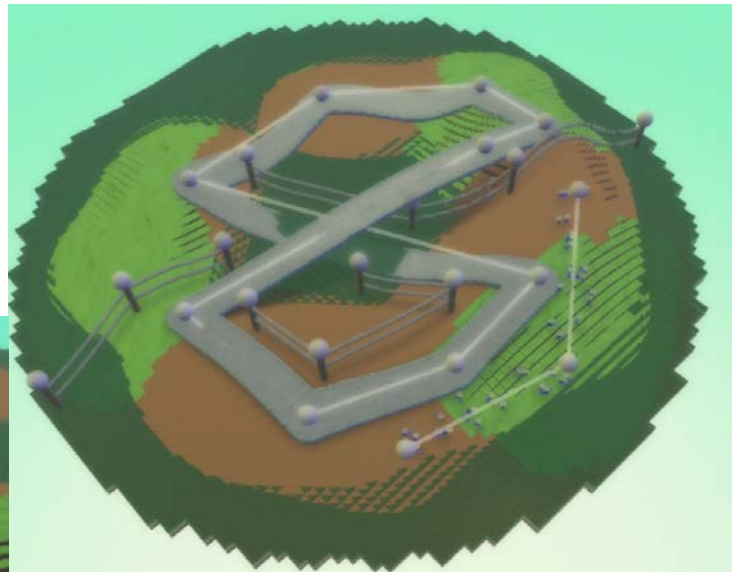
Na planszę należy nanieść tor wyścigu, ograniczenia w postaci ścian oraz kwiaty, które ubogacają krajobraz.

Dodaj prostą ścieżkę

Dodaj ścianę

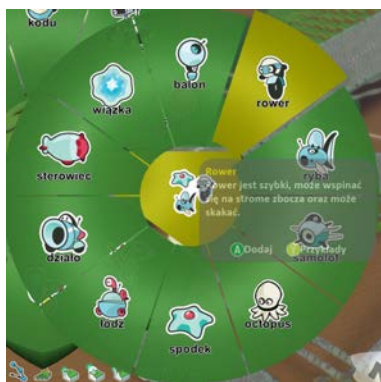
Dodaj trasę

Dodaj kwiaty



Krok 5

Wybierając narzędzie *Zmień ustawienia świata* można przestawić pozycję "Niebo", aby to przybrało inną barwę.



Krok 6

Na trasie wyścigu należy umieścić dwa obiekty typu Rower - każdy o innej barwie.

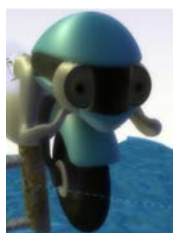


Krok 7

Zmieniając rodzaje ścieżek można dostosować wizualnie tor. Trasie wyścigu należy przyporządkować niepowtarzalną barwę, aby ta była łatwo rozpoznawalna przez boty.

Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy najechać kursorem myszy na postać lub przedmiot, który chcemy zaprogramować, a następnie kliknąć prawy przycisk myszy.

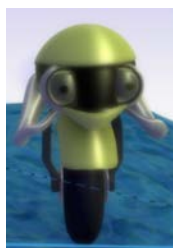


Rower - gracz

Dla obiektu Rower przygotowujemy jedną linijkę programu wybierając następujące elementy:

- 1) Kiedy: klawiatura + Arrows; Wykonaj: ruch

Napisany kod sprawia, że obiekt porusza się zgodnie z wciskanymi przez gracza strzałkami na klawiaturze.



Rower - przeciwnik

Dla obiektu, który ma się poruszać samodzielnie przygotowujemy następujący skrypt:

- 1) Wykonaj: ruch + na ścieżce + niebieski



Tak dobrane komponenty sprawiają, że postać po uruchomieniu rozgrywki będzie poruszała się wzdłuż niebieskiej ścieżki. Warto zaznaczyć, że nieuzupełnienie pola KIEDY skutkuje tym, że polecenia WYKONAJ są realizowane przez cały czas.

W tym przykładzie została zrealizowana pierwsza forma automatycznego sterowania obiektów w grze - poruszanie z pomocą ścieżek.

Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 4 - Piłka Nożna



Cel

Podczas realizacji tego projektu uczniowie mogą poznać zmienne, które zostaną wykorzystane jako warunek końca gry.

Efekt

Uczniowie przygotowują boisko z zaprogramowanymi zawodnikami oraz piłką do gry.

Konstrukcja poziomu



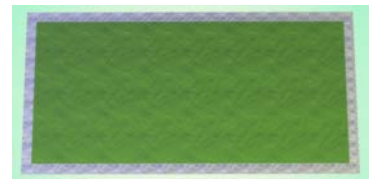
Krok 1

Z pomocą narzędzia *Rysuj, dodaj lub usuń podłoże* należy przygotować prostokątny obszar wykonany z białej tekstury (np. blok ze śniegiem).



Krok 2

Na planszę trzeba nanieść mniejszy prostokąt z trawiastej faktury.



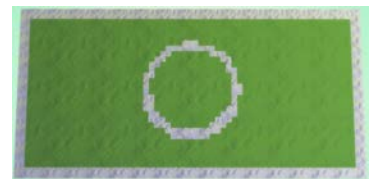
Krok 3

Na środku boiska powinien znaleźć się okrągły plac z białą teksturą.



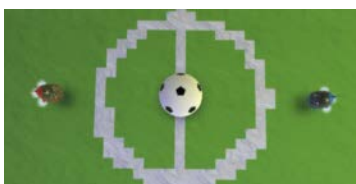
Krok 4

Na białym kole należy utworzyć okrągłą teksturę z trawą.



Krok 5

Trzeba rozdzielić połowy białą linią.



Krok 6

Ostatecznie z pomocą różnych tekstur tworzy się dwa pola reprezentujące bramki.

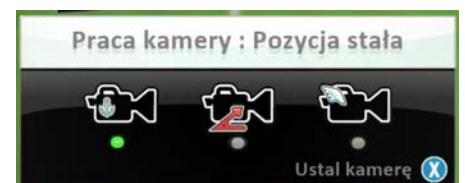


Krok 7

Na boisku powinny znaleźć się trzy obiekty: piłka na środku boiska, dwa Kodu skierowane do środka.

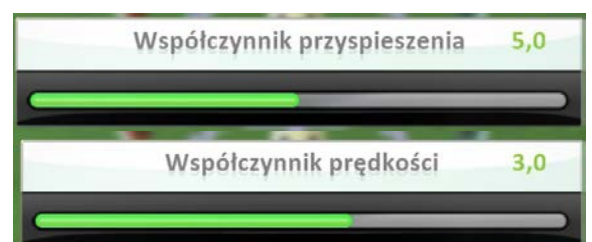
Krok 8

Należy zmienić parametr świata: Praca kamery na Pozycję stałą.



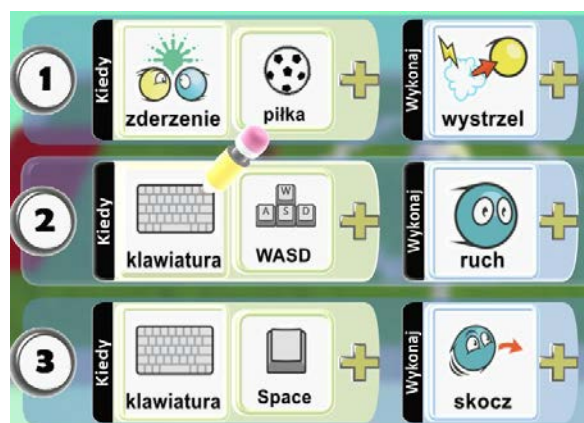
Krok 9

Należy zmienić ustawienia obu obiektów Kodu: Współczynnik prędkości oraz Współczynnik przyspieszenia.



Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy najechać kursorem myszy na postać lub przedmiot, który chcemy zaprogramować, a następnie kliknąć prawy przycisk myszy.



Kodu - gracz

Dla obiektu Kodu należy utworzyć następujący skrypt:

- 1) Kiedy: zderzenie + piłka; Wykonaj: wystriel
- 2) Kiedy: klawiatura + WASD; Wykonaj: ruch
- 3) Kiedy: klawiatura + spacja; Wykonaj: skocz



Skrypt przewiduje trzy zdarzenia: sterowanie z pomocą klawiszy W, A, S oraz D, wykonanie skoku z pomocą klawisza spacji oraz wystrzelenie (wykopnięcie) piłki, gdy Kodu się z nią zderzy.



Kodu - rywal

Dla obiektu Kodu należy utworzyć poniższy skrypt:

- 1) Kiedy: oczy + piłka; Wykonaj: ruch + skieruj
- 2) Kiedy: zderzenie + piłka; Wykonaj: wystriel + jego



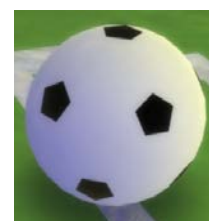
Program sprawia, że Kodu działa samodzielnie i reaguje na piłkę. Gdy ją widzi, zmierza w jej kierunku oraz podczas kolizji z nią, wystrzeliwuje ją przed siebie.



Piłka

Obiekt, z uwagi na swój neutralny charakter, zawiera poza własnym skrypcem warunki końca gry. Linie kodu są następujące:

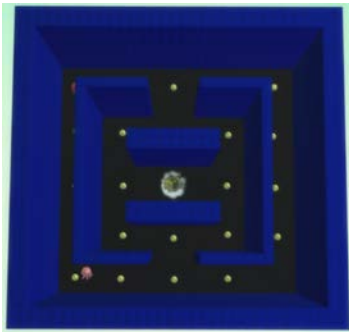
- 1) Kiedy: na podłożu + rodzaj (niebieskie); Wykonaj: +punkt + czerwony + 1
- 2) ORAZ Wykonaj: resetuj + świat
- 3) Kiedy: na podłożu + rodzaj (czerwone); Wykonaj: +punkt + niebieski + 1
- 4) ORAZ Wykonaj: resetuj + świat
- 5) Kiedy: punkty + niebieski + równy + 5 punktów; Wykonaj: koniec
- 6) Kiedy: punkty + czerwony + równy + 5 punktów; Wykonaj: wygrana



Kod piłki sprawdza czy znajduje się na jednym z pól bramkowych. Jeśli tak jest, to przyznaje punkty odpowiedniej stronie (niebieskiej lub czerwonej), a następnie sprowadza obiekty na początkowe pozycje. Gdy któryś z wyników osiągnie wartość 5, gra kończy się.

Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 5 - Pacman



Cel

Podczas realizacji tego projektu uczniowie poznają czym są strony skryptu i zastosują je w praktyce.

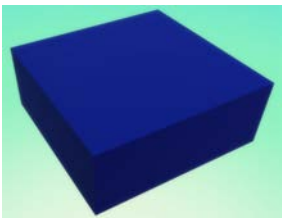
Efekt

Uczniowie przygotują projekt, który imituje klasyczną grę "Pacman".

Konstrukcja poziomu

Krok 1

Z pomocą narzędzia *Rysuj, dodaj lub usuń podłoże* należy przygotować kwadratowy obszar, który zostanie poddany dalszej edycji.



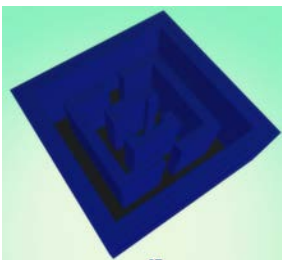
Krok 2

Wykorzystując narzędzie *Podnieś i obniż podłoże*, wybierając magiczny pędzel, należy wypiętrzyć obrys powstałej planszy.



Krok 3

Ponownie korzystając z narzędzia *Rysuj* należy nanieść na planszę korytarze wykorzystując ciemną teksturę.



Krok 4

Wykorzystując narzędzie *Podnieś i obniż podłoże*, wybierając magiczny pędzel, należy obniżyć poziom korytarzy. Jeśli mury są zbyt wysokie, zamiast ich obniżać, należy podnieść poziom podłogi.



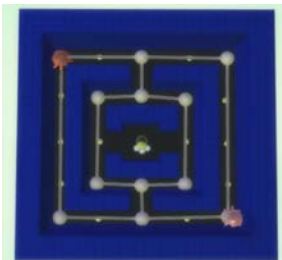
Krok 5

Korzystając z *Dodaj obiekty* na planszy trzeba umieścić następujące obiekty: Kodu - postać gracza, dwie ośmiornice - duchy oraz wiele żółtych jabłek - reprezentujące kulki zbierane przez Pacmana.



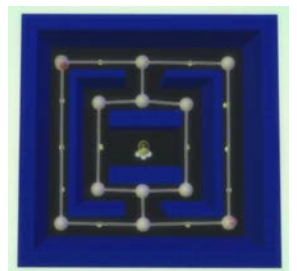
Krok 6

Na obszarze rozgrywki powinny znaleźć się niewidoczne ścieżki, po których będą poruszały się duchy ścigające Pacmana.



Krok 7

Należy sprawdzić, czy duszki swobodnie mieszczą się w korytarzach. Jeśli są zbyt wielkie, można zarówno zmniejszyć ich rozmiary, jak i poszerzyć korytarze.



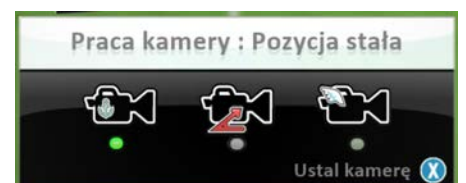
Krok 8

Kilku jabłkom trzeba zmienić odcień na inny np. zielony.



Krok 9

Należy zmienić parametr świata: Praca kamery na Pozycję stałą.



Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.



Kodu - Pacman

Dla obiektu Kodu należy utworzyć następujący skrypt:

- 1) Kiedy: klawiatura + WASD; Wykonaj: ruch
- 2) Kiedy: zderzenie + jabłko + zielony; Wykonaj: jedz + jego
- 3) ORAZ +punkt + biały + 1 punkt
- 4) Kiedy: zderzenie + jabłko; Wykonaj: jedz + jego
- 5) ORAZ +punkt + czerwony + 1 punkt
- 6) Kiedy: punkty + czerwony + równy + 20 punktów + 2 punkty; Wykonaj: wygrana



Przygotowany skrypt pozwala na sterowanie Kodu za pomocą klawiszy W, A, S oraz D. Gdy obiekt zderzy się z dowolnym jabłkiem, gracz zdobędzie czerwony punkt. Kiedy zjedzone jabłko jest zielone, dodawany jest 1 biały punkt - to pozwoli na sterowanie skryptem duchów. W przypadku, gdy gracz zbierze 22 jabłka, gra kończy się zwycięstwem.

Ośmiornica - Duch

Skrypt ośmiornicy zawarty jest na dwóch stronach. Pierwsza dotyczy normalnego funkcjonowania, druga sytuacji, kiedy Kodu-Pacman zje zielony owoc.



Pierwsza strona wygląda następująco:

- 1) Kiedy: oczy + kodu + nie + line of sight (linia widzenia); Wykonaj: ruch + na ścieżce
- 2) Kiedy: oczy + kodu + line of sight; Wykonaj: ruch + skieruj
- 3) Kiedy: punkty + białe + większe i równe + 1 punkt; Wykonaj: przejdź + strona 2
- 4) Kiedy: zawsze; Wykonaj: kolor + czerwony + raz
- 5) Kiedy: zderzenie + kodu; Wykonaj: koniec



Duch podąża po ścieżce tak długo, jak nie widzi kodu - jego wzrok jest ograniczony ścianami. Gdy to się zmieni, ośmiornica zacznie ścigać Pacmana. Gdy punkty białe osiągną wartość większą od zera, duch przejdzie na drugą stronę,

zmieni tryb pracy. Białe punkty zmieniają się, gdy Kodu zje zielone jabłko.

Duch ustawia swój kolor na czerwony - tę instrukcję wykonuje tylko raz. Jeżeli zderzy się z Kodu, gra się kończy.



Druga strona składa się z następujących bloków:

- 1) Wykonaj: ustaw wynik + biały + 0 punktów
- 2) Kiedy: oczy + kodu; Wykonaj: ruch + z dala
- 3) Kiedy: czas + 10 sekund; Wykonaj: przejdź + strona 1
- 4) Wykonaj: kolor + raz + biały
- 5) Kiedy: zderzenie + kodu; Wykonaj: zanikanie + ja

Wraz ze zmianą strony, resetuje się biały wynik, który narzuca zmianę trybu pracy ośmiornicy. Ta natychmiast oddala się od Kodu, a po 10 sekundach skrypt wraca na pierwszą stronę. Jednokrotnie duch zmieni kolor na biały, a w przypadku zderzenia z Kodu, ośmiornica znika.

Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 6 - Owczy Pęd



Cel

Realizując ten projekt uczniowie będą mogli poznać nowy sposób programowania sztucznej inteligencji z pomocą listy priorytetów.

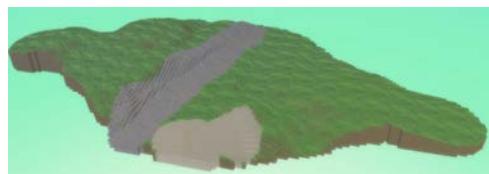
Efekt

Uczniowie przygotują projekt, w którym główna postać ma za zadanie bezpiecznie przeprowadzić stado.

Konstrukcja poziomu

Krok 1

Wykorzystując dowolne tekstury należy przygotować zarys planszy. Centralna część powinna pozostać owalna, a po przeciwległych stronach nieco rozciągnięta. Teren będzie przypominał kształtem cytrynę.

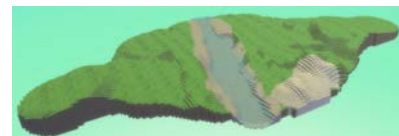


Krok 2

Na planszy powinny znaleźć się nieregularne wzniesienia. Należy zwrócić uwagę, by zbocza były wystarczająco łagodne, aby obiekty mogły się na nich bez przeszkód poruszać.

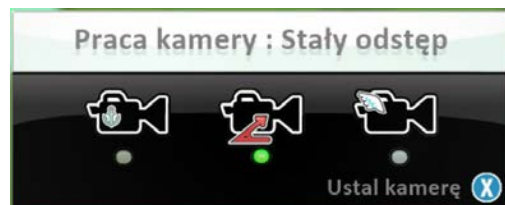
Krok 3

Na teren rozgrywki można nanieść nieco wody.



Krok 4

Na planszy należy umieścić obiekty, które zarówno wzbogacą teren, jak i umożliwią przeprowadzenie rozgrywki. Elementami statycznymi mogą być: drzewa, kwiaty (utworzone z pomocą ścieżek), skały. Obiektami programowalnymi będą: trzy kodu - czerwony, czarny i biały oraz chatka. Chatkę należy umieścić na jednym skraju planszy, a po przeciwległej stronie kodu białe i czarne. Kodu czerwone powinno znaleźć się w centrum planszy. Rozmiary obiektów można dostosować wedle własnego uznania.



Krok 5

Należy zmienić parametr świata: Pracę kamery na tryb Stałego odstępu.



Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.

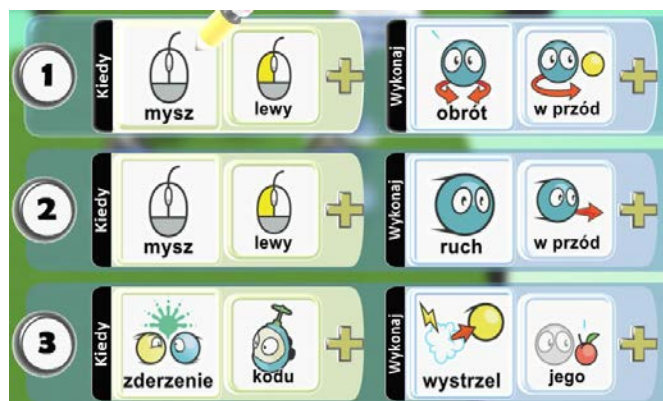
Kodu - Owczarek

Dla obiektu Kodu należy utworzyć następujący skrypt:

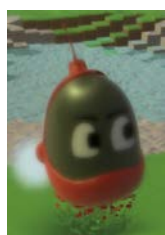
- 1) Kiedy: mysz + lewy; Wykonaj: obrót + w przód
- 2) Kiedy: mysz + lewy; Wykonaj: ruch + w przód
- 3) Kiedy: zderzenie + kodu; Wykonaj: wystrzel + jego

Przygotowany skrypt pozwala na sterowanie Kodu za pomocą myszy. Kiedy użytkownik przytrzyma lewy klawisz, obiekt zarówno obraca się we wskazanym przez kursor kierunku, jak i porusza się przed siebie. To sprawia, że Kodu zbliża się do wybranej pozycji tak długo, jak wciskany jest przycisk myszy.

Trzecia linia skryptu pozwala na wystrzeliwanie (wybijanie w powietrze) każdego Kodu, z którym zderzy się sterowany przez gracza obiekt.



Kodu - Owczarkowi warto zmienić następujące ustawienia, aby mógł łatwo gonić pozostałe obiekty:



Kodu - Wilk

Skrypt Wilka sprawia, że ten porusza się na różne sposoby w zależności od tego, jaki obiekt znajduje się w jego bliskim sąsiedztwie. Numer linii każdego wiersza stanowi układ priorytetów. Na przykład, jeśli Wilk bardziej boi się Owczarka niż chce zjeść Owce, to uciekanie przed nim powinno posiadać najwyższy priorytet, czyli znajdować się w pierwszej linii.

Kod prezentuje się następująco:

- 1) Kiedy: oczy + kodu + czarny + blisko; Wykonaj: ruch + z dala + szybko + szybko
- 2) Kiedy: oczy + kodu + biały + blisko; Wykonaj: ruch + skieruj + szybko + szybko
- 3) Kiedy: oczy + kodu + nie + blisko; Wykonaj: ruch + spacer
- 4) Kiedy: zderzenie + kodu + biały; Wykonaj: znikanie + jego
- 5) ORAZ + Wykonaj: +punkt + czerwony + 1 punkt



Wilk porusza się w zależności od tego, jaki kodu znajduje się w jego sąsiedztwie. Gdy jest to czarny owczarek, postać będzie przed nim uciekać. Kiedy jest to biała owca, rzuci się za nią w pogoń. Jeżeli żaden nie znajduje się w bliskim sąsiedztwie, wilk swobodnie spaceruje. Dodatkowo, gdy zwierzak zderzy się z białym kodu, tamten znika, a gra dodaje jeden punkt do czerwonego wyniku. Jest to licznik zjedzonych owiec.



Kodu - Owca

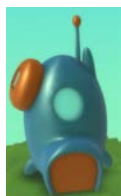
Wszystkie owce korzystają z tego samego skryptu. Stąd też najlepiej najpierw zaprogramować jeden z obiektów, a dopiero w kolejnym kroku go powielić. Rozmiar owiec można zwiększyć.

Kod prezentuje się następująco:

- 1) Kiedy: oczy + kodu + czarny + blisko; Wykonaj: ruch + z dala + szybko + szybko
- 2) Kiedy: oczy + kodu + czerwony + blisko; Wykonaj: ruch + z dala + szybko + szybko
- 3) Kiedy: oczy + kodu + biały; Wykonaj: ruch + skieruj + powoli + powoli
- 4) Kiedy: oczy + kodu + nie + blisko; Wykonaj: ruch + spacer



Owca jest zwierzęciem płochliwym i stadnym. Stąd jej reakcje na kodu w odpowiednich kolorach. Ta będzie uciekać zarówno przed wilkiem, jak i owczarkiem, a gdy dostrzeże swoich kuzynów, to będzie zbijała się z nimi w grupę. Osamotniona owca błąka się bez celu.



Chatka

Obiekt stanowi cel, do którego gracz powinien doprowadzić owce. W ramach swojego skryptu sprawdza również warunki końca gry.

Kod prezentuje się następująco:

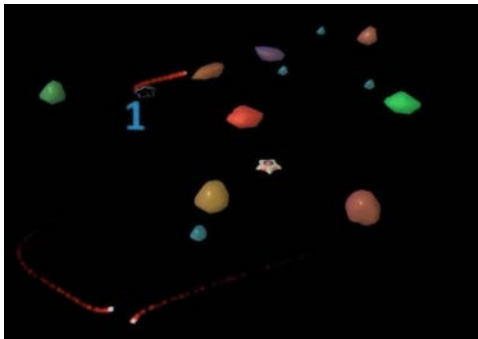
- 1) Kiedy: zderzenie + kodu + biały; Wykonaj: znikanie + jego
- 2) ORAZ Wykonaj: +punkt + zielony
- 3) Kiedy: punkty + czerwony + większe, równe + 3 punkty; Wykonaj: koniec
- 4) Kiedy: punkty + zielony + większe, równe + 3 punkty; Wykonaj: wygrana



Gdy owca zderzy się z chatką, znika i dodawany jest jeden punkt do zielonego wyniku. Obiekt sprawdza też rezultat rozgrywki. Jeżeli wynik czerwony - licznik zjedzonych owiec - będzie wynosił więcej niż 2, to nastąpi przegrana. Gdy to wynik zielony - licznik odprowadzonych owiec - będzie większy od 2, gracz zwycięża.

Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 7 - Gwiezdna Walka



Cel

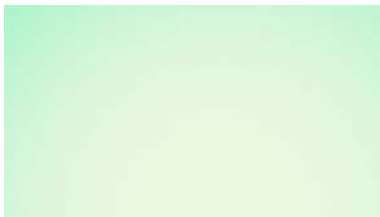
Realizując ten projekt uczniowie poznają w jaki sposób tworzyć dynamicznie obiekty w grze.

Efekt

Uczniowie przygotowują projekt, w którym dwa spodki ścigają się w zdobyciu kolorowych wiązek.

Konstrukcja poziomu

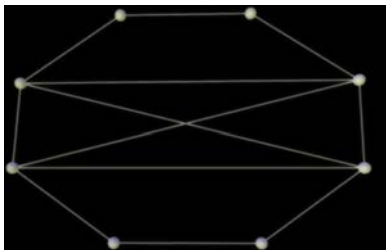
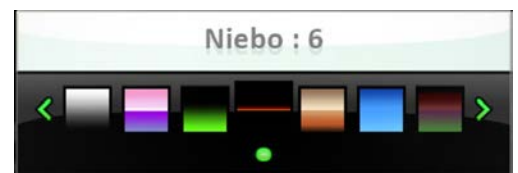
Krok 1



Pierwszy etap konstrukcji poziomu polega na pozbyciu się wszelkiej tekstury z planszy.

Krok 2

Należy zmienić ustawienie planszy "Niebo". Najlepiej przyjąć jeden z ciemnych wzorów, który lepiej odda charakter przestrzeni kosmicznej.

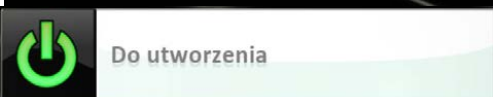
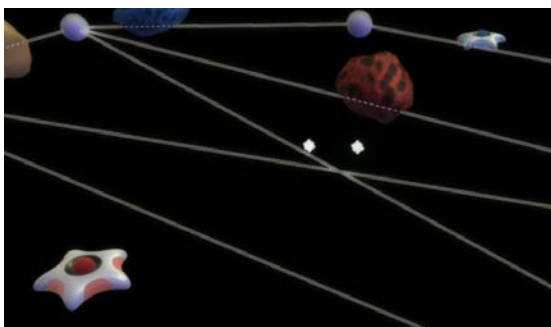
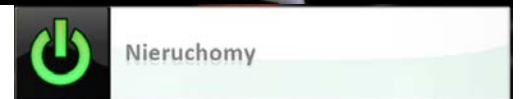
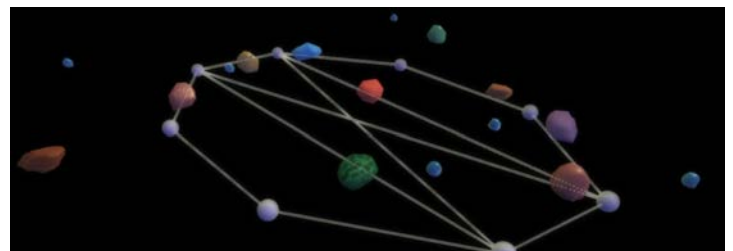


Krok 3

Przy pomocy narzędzia do budowania ścieżek należy utworzyć prosty układ skrzyżowań, które będą stanowić ramę rozgrywki.

Krok 4

Przestrzeń ma zostać wypełniona dowolnymi skałami. Im więcej rozmiarów, kolorów, rodzajów, tym lepiej. Trzeba również zwrócić uwagę, aby obracać je w różnych kierunkach, dzięki czemu uzyskuje się iluzję tego, że każdy z kamieni jest inny. Wysokość wszystkich kamieni powinna być zbliżona do siebie, a w ustawieniach każdego z nich należy włączyć opcję "Nieruchomy".

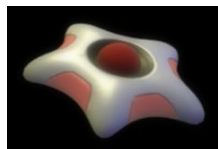


Krok 5

Ostatnimi obiektami, które powinny się znaleźć na planszy są dwa spodki oraz dwie wiązki. Po jednym z każdego obiektu należy pomalować na czerwono, a pozostałe na niebiesko. W ustawieniach wiązek należy zaznaczyć również opcję "Do utworzenia". Wysokość obiektów powinna być zbliżona do wysokości dryfujących skał.

Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.



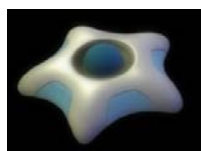
Spodek - Gracz

Dla obiektu Spodek należy utworzyć następujący skrypt:

- 1) Kiedy: klawiatura + WASD; Wykonaj: ruch
- 2) Wykonaj: (kamera) podąż
- 3) Kiedy: zderzenie + wiązka + czerwony; Wykonaj: +punkt + czerwony + 1 punkt
- 4) Kiedy: zderzenie; Wykonaj: ogłusz + ja



Spodek sterowany jest przez gracza z pomocą klawiatury, a kamera w rozgrywce jest ustawiona tak, aby podążała za tym obiektem. Jeżeli statkowi kosmicznemu uda się zderzyć z czerwoną wiązką, gracz zdobywa czerwony punkt. Kiedy dojdzie do kolizji z dowolnym obiektem, spodek staje się na chwilę ogłuszony - to wymusza na gracz, aby uważnie latał pomiędzy dryfującymi skałami.



Spodek - przeciwnik

Dla obiektu Spodek należy utworzyć następujący skrypt:

- 1) Kiedy: oczy + wiązka + niebieski + nie; Wykonaj: ruch + spacer
- 2) Kiedy: oczy + wiązka + niebieski; Wykonaj: ruch + skieruj
- 3) Kiedy: zderzenie + wiązka + niebieski; Wykonaj: +punkt + niebieski + 1 punkt
- 4) Kiedy: zderzenie; Wykonaj: ogłusz + ja



Spodek zaprogramowany jest tak, by poruszał się za niebieską wiązką. Jeżeli żadnej nie widzi, lata w przypadkowych kierunkach. Jeżeli statkowi kosmicznemu uda się zderzyć z niebieską wiązką, naliczany jest niebieski punkt. Kiedy dojdzie do kolizji z dowolnym obiektem, spodek staje się na chwilę ogłuszony.



Wiązka - Czerwona

Obiekt wzorcowy czerwonej wiązki. Kod prezentuje się następująco:

- 1) Wykonaj: ruch + na ścieżce
- 2) Kiedy: zderzenie + spodek + czerwony; Wykonaj: znikanie + ja



Wiązka na początku gry jest nieobecna. Będzie natomiast regularnie tworzona przez skały. Ich jedynym celem jest poruszanie się po przygotowanej ścieżce i znikanie po kolizji z czerwonym spodem.



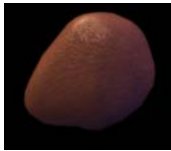
Wiązka - Niebieska

Obiekt wzorcowy niebieskiej wiązki. Kod prezentuje się następująco:

- 1) Wykonaj: ruch + na ścieżce
- 2) Kiedy: zderzenie + spodek + niebieski;
Wykonaj: znikanie + ja



Wiązka na początku gry jest nieobecna. Będzie natomiast regularnie tworzona przez skały. Ich jedynym celem jest poruszanie się po przygotowanej ścieżce i znikanie po kolizji z niebieskim spodkiem.



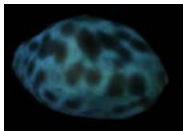
Skała I

Spośród utworzonych wcześniej skał należy wybrać jedną ze znajdujących się na skraju ścieżki, a następnie przygotować jej poniższy kod:

- 1) Kiedy: czas + 5 sekund + 3 sekundy; Wykonaj: utwórz + Wiązka 1
- 2) Kiedy: czas + 5 sekund; Wykonaj: utwórz + Wiązka 2
- 3) Kiedy: punkty + czerwony + większe równe + 10 punktów; Wykonaj: wygrana + Wiązka 1
- 4) Kiedy: punkty + niebieskie + większe równe + 10 punktów; Wykonaj: koniec



Skała, co pewien odgórnie określony czas wyrzuca z siebie dwa rodzaje wiązek. Dodatkowo sprawdzany jest warunek końca gry: zwycięstwo, jeśli to gracz zbierze 10 wiązek lub przegrana, gdy zrobi to przeciwnik.



Skała II

Spośród utworzonych wcześniej skał należy wybrać jedną ze znajdujących się na skraju ścieżki, a następnie przygotować jej poniższy kod:

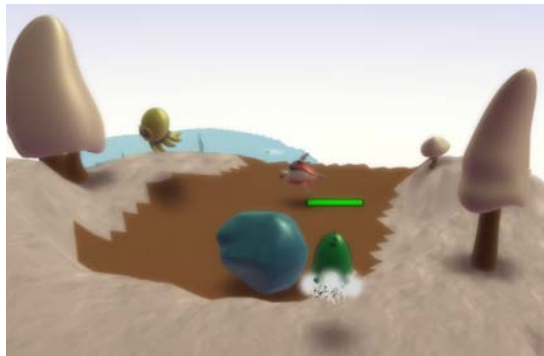
- 1) Kiedy: czas + 5 sekund + 3 sekundy; Wykonaj: utwórz + Wiązka 2
- 2) Kiedy: czas + 5 sekund; Wykonaj: utwórz + Wiązka 1



Skała, co pewien odgórnie określony czas wyrzuca z siebie dwa rodzaje wiązek.

Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 8 - Gra Platformowa



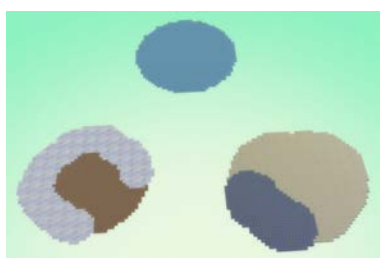
Cel

Realizując ten projekt uczniowie ugruntuja nabytą wiedzę wykorzystując wszystkie poznane zagadnienia w praktyce.

Efekt

Uczniowie przygotowują projekt gry platformowej, w którym postać teleportuje się, zbiera monety oraz zмага się z przeciwnikami.

Konstrukcja poziomu

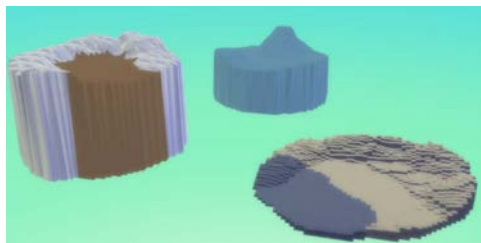


Krok 1

Pierwszy etap polega na przygotowaniu trzech mniejszych plansz wykorzystując przy tym dowolne tekstury..

Krok 2

Należy podnieść i nadać głębi wszystkim trzem platformom.

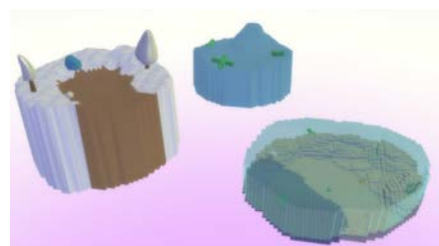


Krok 3

W narzędziu "Zmień ustawienia świata" można dowolnie dobrać parametry "Niebo" oraz "Oświetlenie".

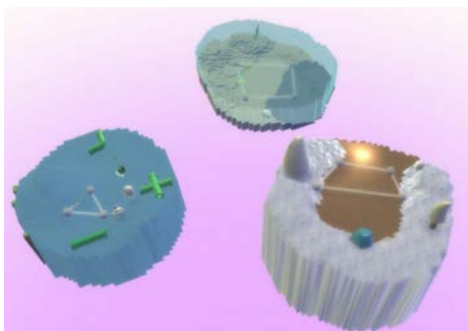
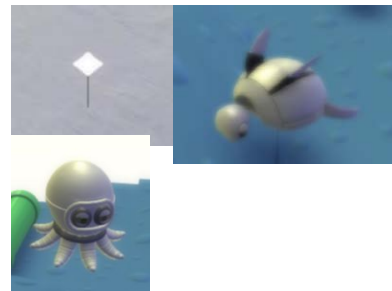
Krok 4

Z pomocą statycznych obiektów oraz cieczy można uzupełnić krajobraz platform.



Krok 5

Na jednej z platform należy umieścić 3 obiekty: Kodu, Serce oraz Monetę. W ustawieniach tych obiektów trzeba zaznaczyć opcję "Do utworzenia". Następnie potrzebne będą kolejne obiekty: Wiązka, Ośmiornica, Żółw, które będą w przyszłości kopiowane.



Krok 6

W ostatnim kroku należy utworzyć po jednej prostej ścieżce na każdej z przygotowanych platform.

Programowanie obiektów

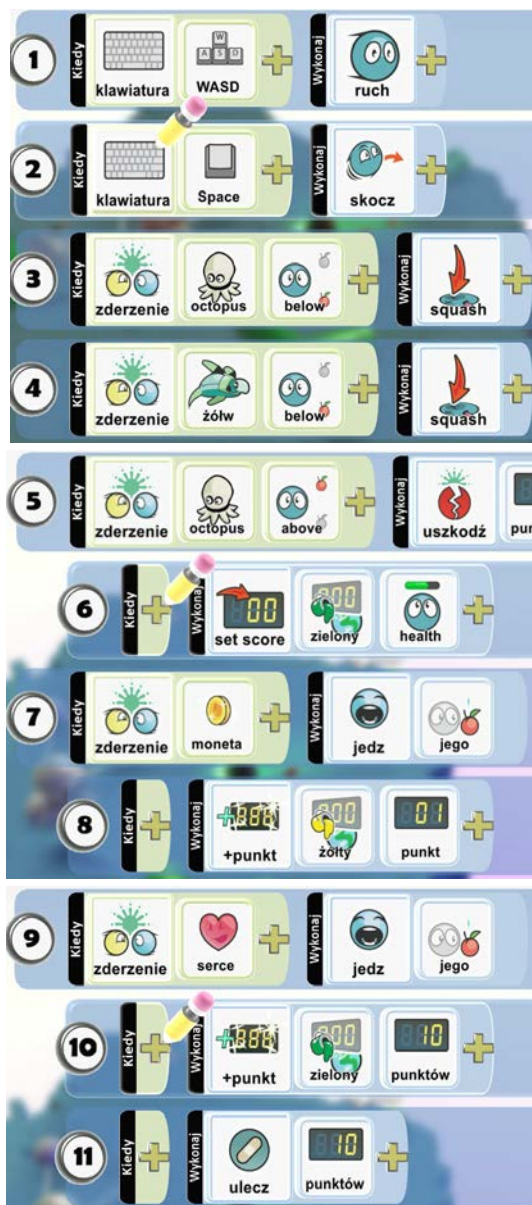
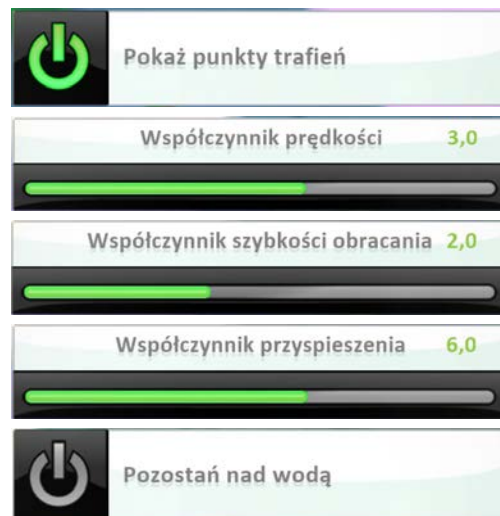
Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.



Kodu - Gracz

Kodu jest obiektem do utworzenia. Cecha ta jest ważna w przypadku, gdy chcemy, by obiekt "teleportował się" między różnymi pozycjami na planszy. Dodatkowo można mu zmienić następujące parametry, aby łatwiej nim sterować:

- Pokaż punkty trafień: ON
- Współczynnik prędkości: 3.0
- Współczynnik szybkości obracania: 2.0
- Współczynnik przyspieszenia: 6.0
- Pozostań nad wodą: OFF



Skrypt dla Kodu prezentuje się następująco:

- 1) Kiedy: klawiatura + WASD; Wykonaj: ruch
- 2) Kiedy: klawiatura + Space; Wykonaj: skocz
- 3) Kiedy: zderzenie + octopus + below; Wykonaj: squash
- 4) Kiedy: zderzenie + żółw + below; Wykonaj: squash
- 5) Kiedy: zderzenie + octopus + above; Wykonaj: uszkodź + 10 punktów + ja
- 6) ORAZ Wykonaj: set score + zielony + health
- 7) Kiedy: zderzenie + moneta; Wykonaj: jedz + jego
- 8) ORAZ Wykonaj: +punkt + żółty + 1 punkt
- 9) Kiedy: zderzenie + serce; Wykonaj: jedz + jego
- 10) ORAZ Wykonaj: +punkt + zielony + 10 punktów
- 11) ORAZ Wykonaj: ulecz + 10 punktów

Kodu jest obiektem sterowanym przez gracza. Gdy wskoczy na jednego z dwóch przeciwników, to ci zostaną spłaszczeni. Jeżeli dojdzie do kolizji od boku stworzenia, to to wyrządza szkody Kodu. Wraz z tym następuje aktualizacja punktów zielonych na obecny stan punktów zdrowia obiektu gracza. Wynik ten jest potrzebny do sprawdzania warunku końca gry.

Zderzenie z monetą lub sercem kończy się dla Kodu pozytywnie. W obu przypadkach obiekt jest zjadany i dodawane są punkty - żółte dla pieniążka i zielony dla serca (dodawane są również punkty życia). Wynik żółty kontroluje możliwość teleportowania się między platformami.



Żółw - posiadacz monet

Obiekt ten kryje w sobie monety, których potrzebuje gracz do przejścia każdej z platform. Wśród ustawień należy wyłączyć opcję "Pozostań nad wodą". Po przygotowaniu kodu dla żółwia, należy go skopiować i umieścić po jednej sztuce na każdej platformie.



Skrypt wygląda następująco:

- 1) Wykonaj: kolor + losowy + raz
- 2) Wykonaj: ruch + na ścieżce
- 3) Kiedy: życie + mniejsze równe + 0 punktów; Wykonaj: utwórz + Moneta 1

Żółw przy każdej rozgrywce będzie miał inne zabarwienie i zawsze będzie poruszał się po ścieżce. Kiedy jego punkty życia spadną do zera (np. zostanie spłaszczony przez Kodu) utworzy monetę.



Ośmiornica - przeciwnik

Obiekt stanowi przeciwnika dla postaci sterowanej przez gracza. Można go pokonać wskakując na niego. Po wykonaniu jednej ośmiornicy należy ją skopiować i umieścić w dowolnej liczbie na każdej platformie.



Skrypt wygląda następująco:

- 1) Kiedy: zderzenie + kodu + below; Wykonaj: wystrzel + jego
- 2) Wykonaj: kolor + losowy + raz
- 3) Wykonaj: ruch + spacer
- 4) Kiedy: życie + mniejsze równe + 0 punktów; Wykonaj: utwórz + Serce 1

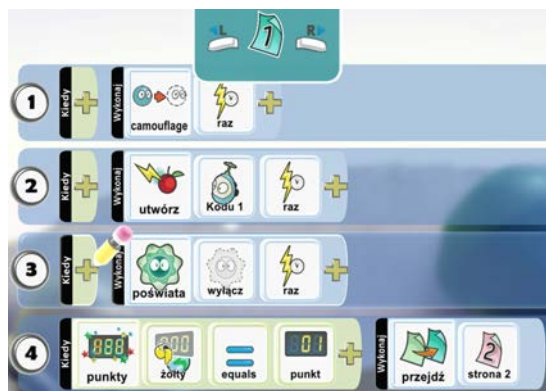
Ośmiornica przy zderzeniu z Kodu, kiedy jest wyżej od niego, wystrzeliwuje go w powietrze. Podobnie jak żółw na początku rozgrywki przyjmuje losowy kolor i odtąd porusza się w przypadkowy sposób. Gdy jego punkty życia spadną do poziomu 0, tworzy serce, które

uzdrowia Kodu.



Wiązki - teleportery

Obiekty należy umieścić pojedynczo na każdej platformie. Ich skrypty będą się między sobą nieznacznie różniły, lecz zasady działania będą podobne. W trakcie realizacji projektu można zatem przygotować pierwszy obiekt, a następnie go kilkakrotnie skopiować i odpowiednio zmodyfikować. Wśród ustawień należy zwrócić uwagę na wysokość na jakiej wiązka się znajduje - kodu musi być w stanie wejść z nią w kolizję.



Skrypt wiązki na pierwszej platformie:

Strona pierwsza:

- 1) Wykonaj: camouflage + raz
- 2) Wykonaj: utwórz + Kodu 1 + raz
- 3) Wykonaj: poświęta + wyłącz + raz
- 4) Kiedy: punkty + żółty + equals + 1 punkt; Wykonaj: przejdź + strona 2

Strona druga:

- 1) Wykonaj: poświęta + raz + losowy
- 2) Wykonaj: uncamouflage
- 3) Kiedy: zderzenie + kodu; Wykonaj: zanikanie + jego
- 4) ORAZ Wykonaj: +punkt + niebieski + 1 punkt



Pierwsza strona skryptu ustawia parametry początkowe i oczekuje na zdobycie przez gracza jednego żółtego punktu (wiąże się to ze zdobyciem jednej monety), aby przejść na drugą stronę. Parametry początkowe to: przejście w stan kamuflażu, utworzenie kodu oraz wyłączenie poświęta. Druga strona cofa te zmiany i czeka na kolizję z kodu. Gdy ta nastąpi, postać gracza znika i dodawany jest jeden niebieski punkt - ten natomiast wyzwala pojawienie się kodu na nowej platformie.



Skrypt wiązki na drugiej platformie:

Strona pierwsza:

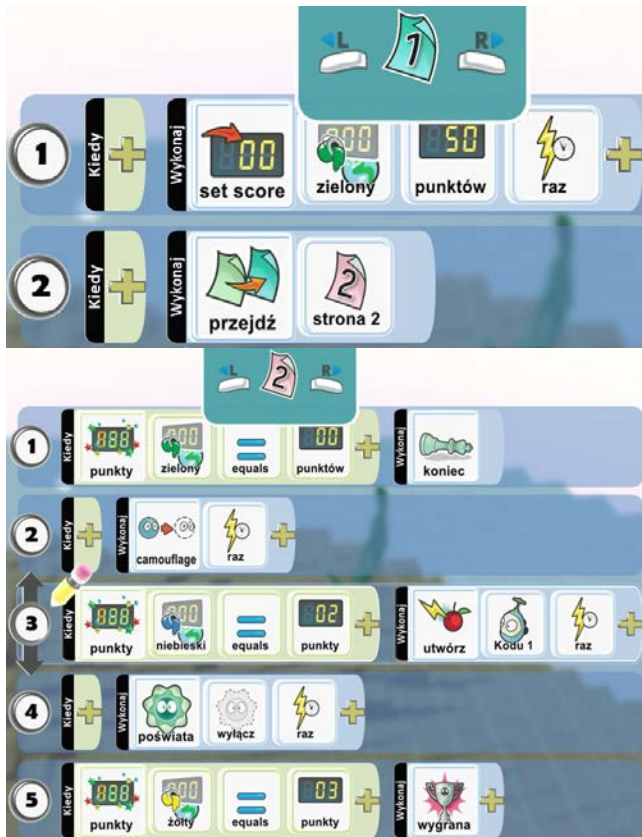
- 1) Wykonaj: camouflage + raz
- 2) Kiedy: punkty + niebieski + equals + 1 punkt; Wykonaj: utwórz + Kodu 1 + raz
- 3) Wykonaj: poświęta + wyłącz + raz
- 4) Kiedy: punkty + żółty + equals + 2 punkt; Wykonaj: przejdź + strona 2

Strona druga:

- 1) Wykonaj: poświęta + raz + losowy
- 2) Wykonaj: uncamouflage
- 3) Kiedy: zderzenie + kodu; Wykonaj: zanikanie + jego
- 4) ORAZ Wykonaj: +punkt + niebieski + 1 punkt



Druga strona skryptu zupełnie nie ulega zmianie względem poprzedniej wiązki. Różnica pojawia się tylko na pierwszej. Tworzenie kodu ma miejsce tylko wtedy, gdy niebieski wynik osiągnie wartość "1" (czyli jest to wymuszone akcją poprzedniej wiązki) oraz zmiana strony następuje po zdobyciu dwóch monet.



Skrypt trzeciej wiązki:

Strona pierwsza:

- 1) Wykonaj: set score + zielony + 50 punktów + raz
- 2) Wykonaj: przejdź + strona 2

Strona druga:

- 1) Kiedy: punkty + zielony + equals + 0 punktów;
Wykonaj: koniec
- 2) Wykonaj: camouflage + raz
- 3) Kiedy: punkty + niebieski + equals + 2 punkty;
Wykonaj: utwórz + Kodu 1 + raz
- 4) Wykonaj: poświęta + wyłącz + raz
- 5) Kiedy: punkty + żółty + equals + 3 punkty;
Wykonaj: wygrana

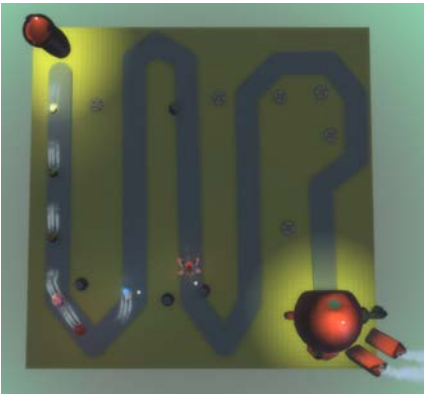
Skrypt ostatniej wiązki różni się względem pozostałych. Pierwsza strona inicjalizuje początkową wartość życia bohatera i natychmiast przerywa program na drugą stronę.

Druga strona sprawdza warunki zwycięstwa: w pierwszej linii porażkę (spadek punktów życia Kodu do 0) i w ostatniej linii zwycięstwa (zdobycie wszystkich monet - zlicza je wynik żółty). Linia druga i czwarta narzuca

parametry początkowe wiązki, zaś trzecia sprawia, że kodu zostanie stworzony, kiedy poprzednia wiązka przyzna niebieski punkt.

Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 9 - Obrona Fabryki



Cel

Realizując ten projekt uczniowie poznają w jaki sposób wykorzystać zagnieźdżanie komend, aby budować wielopoziomowe warunki.

Efekt

Uczniowie przygotowują projekt gry, w trakcie której gracz musi zaplanować obronę przed nadciągającą hordą przeciwników.

Konstrukcja poziomu

Krok 1

Należy przygotować rozległy, płaski obszar, na którym toczyć się będzie rozgrywka. Kształt planszy może pozostać dowolny - nada się również szeroki tunel o dużym zagęszczeniu zakrętów.



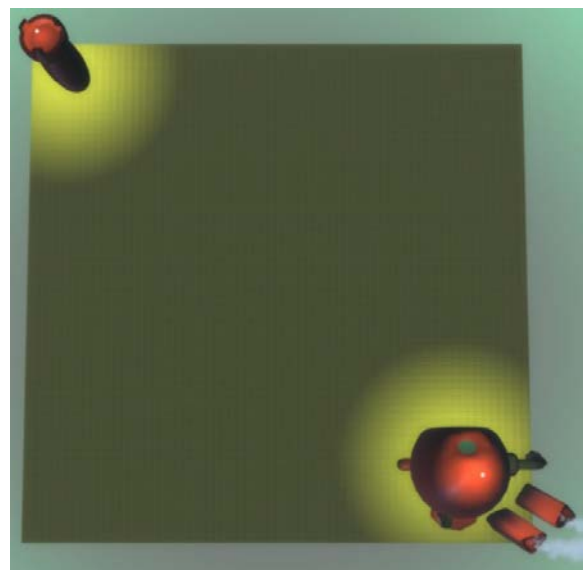
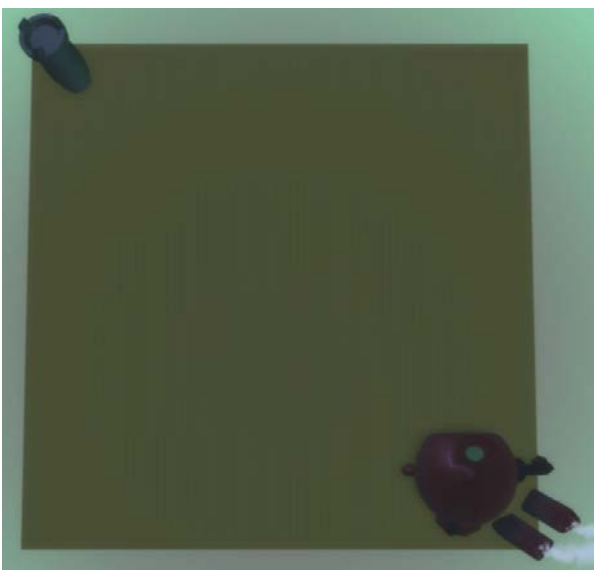
Krok 2

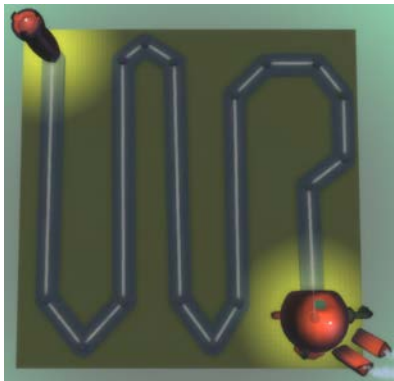
Spośród ustawień świata należy zmienić następujące opcje: pracę kamery (pozycja stała), niebo (ciemne), oświetlenie (ciemne) oraz widoczność wyników żółty (ich etykietę należy opisać jako: "Monety").



Krok 3

Po przeciwnych stronach planszy należy umieścić dwa duże obiekty: zamek oraz fabrykę. Dodatkowo można umieścić ponad nimi obiekty światło, aby wyodrębnić ich sylwetki z ciemności.





Krok 4

Wzdłuż całej planszy należy poprowadzić ścieżkę łączącą postawione wcześniej budynki. Trasa powinna się wić, a między kolejnymi ulicami należy zachować szeroki odstęp, który utworzy chodnik.

Krok 5

Należy przygotować pięć obiektów, które zostaną wykorzystane w grze. Cztery z nich muszą mieć zaznaczoną opcję "Do utworzenia" w swoich ustawieniach. Są to: kodu, działo, patyk oraz samolot. Piąty obiekt - krążek - należy ustawić na takiej wysokości, aby leżał niemal na ziemi.

Każdemu z dodanych obiektów należy zmienić trzy parametry:

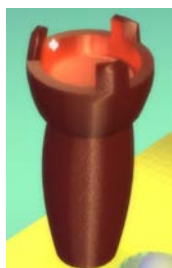
- Siła świecenia - 10,0
- Siła światła świecącego - 5,0
- Emisja światła - 10,0

Zabieg ten sprawi, że obiekty będą dobrze widoczne w ciemności.



Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.



Zamek - baza przeciwnika

Budynek będzie tworzył w równych odstępach czasu przeciwników szturmujących bazę gracza. Należy obrócić go w taki sposób, aby produkowani przeciwnicy lądowali od strony ścieżki.

Skrypt zawiera trzy strony.

Pierwsza strona:

- 1) Kiedy: czas + 3 sek; Wykonaj: utwórz + Kodu 1
- 2) Kiedy: czas + 5 sek + 10 sek + 3 sek; Wykonaj: utwórz + Samolot 1
- 3) Kiedy: czas + 30 sek + 5 sek + 1 sek; Wykonaj: przejdź + strona 2



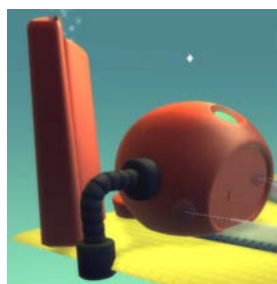
Druga strona:

- 1) Kiedy: czas + 2 sek; Wykonaj: utwórz + Kodu 1
- 2) Kiedy: czas + 5 sek + 10 sek; Wykonaj: utwórz + Samolot 1
- 3) Kiedy: czas + 30 sek + 5 sek + 10 sek; Wykonaj: przejdź + strona 3

Trzecia strona:

- 1) Kiedy: czas + 1 sek; Wykonaj: utwórz + Kodu 1
- 2) Kiedy: czas + 10 sek; Wykonaj: utwórz + Samolot 1
- 3) Kiedy: czas + 60 sek; Wykonaj: wygrana

Każda ze stron skryptu działa w podobny sposób. Cyklicznie, w krótkich odstępach czasu (3, 2 i 1 sekunda) tworzone jest Kodu 1, a interwał zmniejsza się na każdej kolejnej stronie. Podobnie powstaje Samolot 1, lecz tutaj opóźnienie jest zdecydowanie dłuższe - są to kolejno 18, 15 i 10 sekund. Trzeci wiersz każdej strony wywołany jest po dłuższej chwili i albo zmienia stronę na kolejną, albo jak ma to miejsce na ostatniej z nich, ogłasza zwycięstwo gracza.



Fabryka - baza gracza

Budynek pełni bierną rolę na końcu ścieżki. Zadaniem gracza jest uchronienie go przed nadciągającymi przeciwnikami.

Skrypt wygląda następująco:

- 1) Wykonaj: set score + żółty + 20 punktów + raz
- 2) Kiedy: zderzenie + cokolwiek; Wykonaj: uszkodz + 10 punktów + ja
- 3) Kiedy: życie + mniejsze lub równe + 0 punktów; Wykonaj: koniec



Wraz z uruchomieniem gry fabryka jednokrotnie ustawia liczbę monet na 20 (wynik żółty). Przez resztę gry sprawdza dwa warunki. Jeżeli dojdzie między nim, a jakimś obiektem do kolizji, to ulega uszkodzeniu. Jeśli punkty życia spadną do wartości "0", gra się kończy.



Samolot i kodu - przeciwnicy

Dwa obiekty są tworzone przez zamek przeciwnikami, którzy podróżują po szlaku, aby dotrzeć do fabryki. Kodu pojawia się częściej, ale porusza się wolniej, a samolot odwrotnie.

Skrypty postaci nie różnią się dużo od siebie i wyglądają następująco:

Skrypt samolotu:

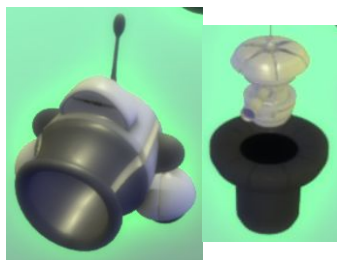
- 1) Wykonaj: poświata + losowy + raz
- 2) Wykonaj: kolor + losowy + raz
- 3) Wykonaj: ruch + na ścieżce

- 4) Kiedy: zderzenie + fabryka; Wykonaj: wybuch + ja
- 5) Kiedy: życie + mniej lub równo + 0 punktów; Wykonaj: + punkt + żółty + 5 punktów

Skrypt kodu:

- 1) Wykonaj: poświata + losowy + raz
- 2) Wykonaj: kolor + losowy + raz
- 3) Wykonaj: ruch + na ścieżce
- 4) Kiedy: zderzenie + fabryka; Wykonaj: wybuch + ja
- 5) Kiedy: życie + mniej lub równo + 0 punktów; Wykonaj: + punkt + żółty 1 punkt

Samolot i kodu przy powstaniu obierają losowy kolor i poświatę oraz zawsze podążają za ścieżką. Gdy zderzą się z fabryką, wybuchają, a przy samozniszczeniu dodają monety graczowi - 5 za samolot i 1 za kodu.



Patek i działo - obrońcy

Dwa obiekty są tworzone przez gracza w wybranych miejscach. Ostrzeliwiają maszerujących przeciwników, aby zatrzymać ich atak na fabrykę.

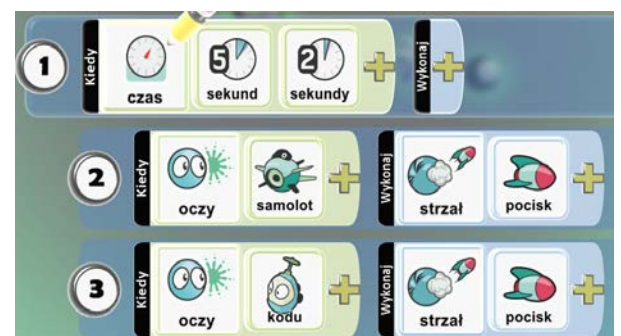
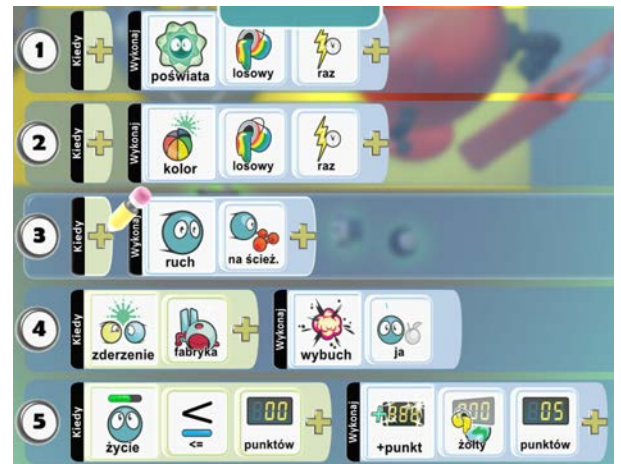
Skrypt pływaka:

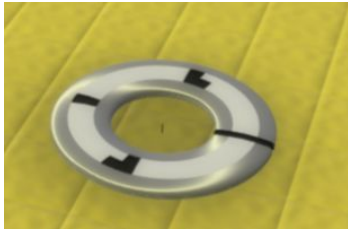
- 1) Kiedy: czas + .25 sek + .25 sek
- 2) ORAZ Kiedy: oczy + kodu + blisko; Wykonaj: strzał + punkt
- 3) ORAZ Kiedy: oczy + samolot + blisko; Wykonaj: strzał + punkt

Skrypt działa:

- 1) Kiedy: czas + 5 sek + 2 sek
- 2) ORAZ Kiedy: oczy + samolot + blisko; Wykonaj: strzał + pocisk
- 3) ORAZ Kiedy: oczy + kodu + blisko; Wykonaj: strzał + pocisk

Patek i działo funkcjonują w podobny sposób. Czekają pewien czas, a następnie, gdy dostrzegą cel, strzelają do niego. Różnica polega na tym, że dla pływaka interwały wynoszą pół sekundy, strzela słabymi punktami i celuje przede wszystkim do kodu, gdy działo wystrzeliwuje potężne pociski co 7 sekund i celuje przede wszystkim do samolotów.





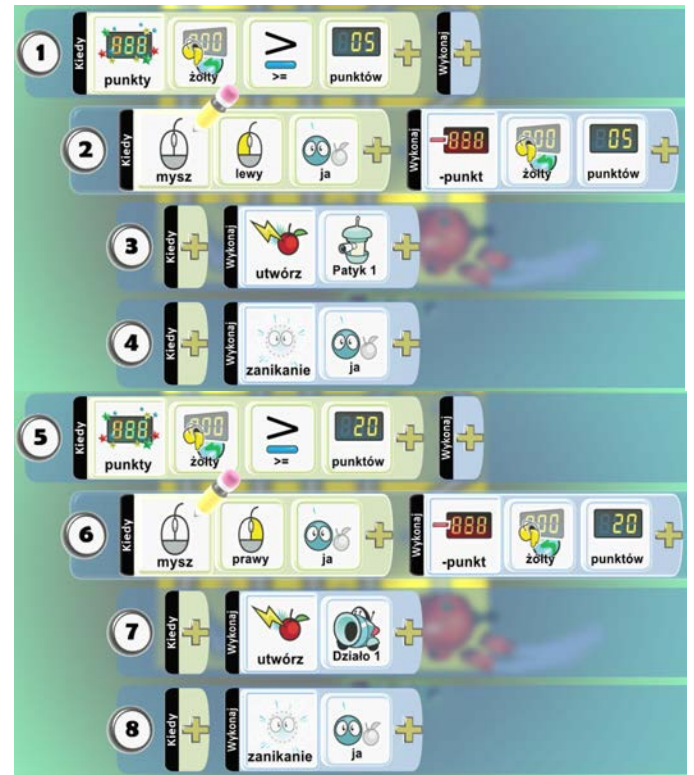
Krażek - miejsce ustawienia obrońcy

Krażki należy umieścić w liczbie 10 pomiędzy ulicami przygotowanej wcześniej trasy. Ze względu na długi kod, należy przygotować pierwszy obiekt, a następnie go skopiować. W miejscu krążka gracz będzie mógł stawiać obrońców.

Skrypt wygląda następująco:

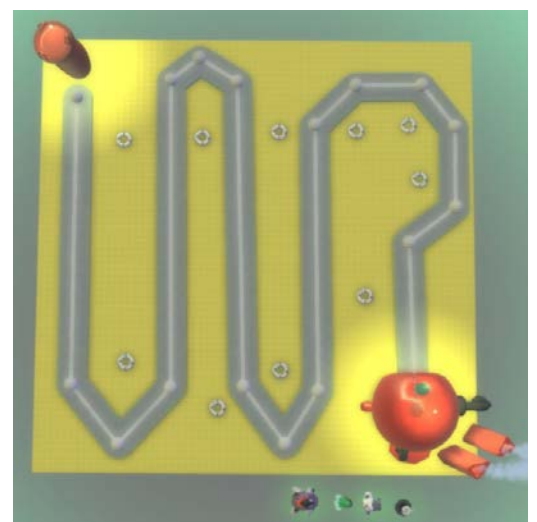
- 1) Kiedy: punkty + żółty + większy lub równy + 5 punktów
- 2) ORAZ Kiedy: mysz + lewy + ja; Wykonaj: -punkt + żółty + 5 punktów
- 3) ORAZ Wykonaj: utwórz + Patyk 1
- 4) ORAZ Wykonaj: zanikanie + ja
- 5) Kiedy: punkty + żółty + większy lub równy + 20 punktów
- 6) ORAZ Kiedy: mysz + prawy + ja; Wykonaj: -punkt + żółty + 20 punktów
- 7) ORAZ Wykonaj: utwórz + Działo 1
- 8) ORAZ Wykonaj: zanikanie + ja

Krażek leży pośród ulicami i czeka aż gracz wybierze go lewym lub prawym przyciskiem myszy. Aby ten pierwszy mógł zadziałać, licznik żółty musi wynosić przynajmniej 5. Wtedy odejmowane są monety, tworzony patyk, a krążek znika. W przypadku prawego przycisku, na koncie gracza musi znajdować się przynajmniej 20 monet. Po użyciu tego klawisza monety są odliczane i powstaje działo.

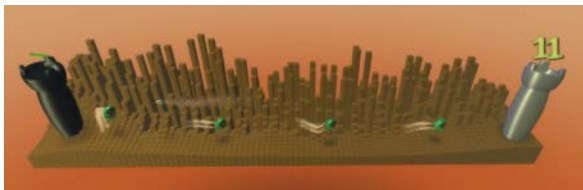


Gotowy obiekt kopiujemy i rozmieszczamy na planszy

Program można przetestować wybierając narzędzie *Graj grę*.



Zajęcia 10 - Dwie Wieże



Cel

Podczas realizacji projektu uczniowie będą mogli wykorzystać zmienne w postaci “flagi”, która zmienia sytuacyjnie parametry obiektu.

Efekt

Uczniowie przygotują projekt gry polegającej na utrzymaniu wieży i zburzeniu przeciwnika.

Konstrukcja poziomu

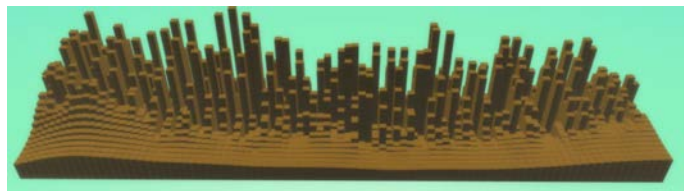


Krok 1

W pierwszym kroku należy przygotować długą i wąską planszę.

Krok 2

Przy pomocy odpowiednich narzędzi należy wykreować wzniesienia, które utworzą “mur” wzdłuż jednego z długich boków planszy

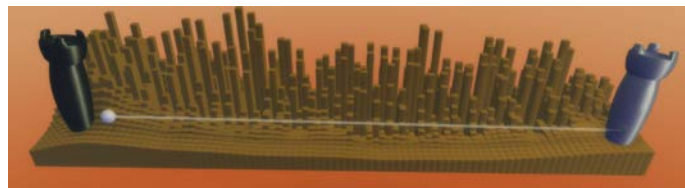


Krok 3

Wśród ustawień świata należy zmienić szereg właściwości. Modyfikacja “Nieba” jest opcjonalna. Pracę kamery należy ustawić na stałą tak, aby ta obejmowała kadrem całą planszę. Spośród wyników należy zmienić widoczność żółtego (etykieta: “Monety”) oraz niebieskiego (zupełnie wyłączona, niedostępna dla gracza). Trzeba też nadać opis na niebieski przycisk (etykieta: “Pospiech”).

Krok 4

Na planszy należy umieścić dwie wieże - czarną oraz białą. Pomiędzy nimi należy poprowadzić ścieżkę.



Krok 5

W ostatnim etapie trzeba umieścić jeden obiekt kodu i włączyć jego właściwość “Do utworzenia”.



Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.



Zamek - baza przeciwnika

Budynek będzie tworzył w równych odstępach czasu przeciwników szturmujących zamek gracza. Należy obrócić go w taki sposób, aby produkowani przeciwnicy lądowali od strony ścieżki.

Skrypt wygląda następująco:

- 1) Kiedy: czas + 5 sek; Wykonaj: utwórz + Kodu 1
- 2) Kiedy: czas + 10 sek + 5 sek; Wykonaj: utwórz + Kodu 1
- 3) ORAZ utwórz + Kodu 1
- 4) ORAZ utwórz + Kodu 1
- 5) Kiedy: życie + mniejsze lub większe + 0 punktów; Wykonaj: wygrana

Kod zamku sprawia, że ten co 5 sekund tworzy pojedynczego kodu, a co 15 dodatkowych 3. Kiedy życie konstrukcji spadnie do 0, następuje wygrana gracza.



Kodu - poplecznik przeciwnika

Kodu ma proste zadanie - dotrzeć do bazy gracza i zniszczyć ją.

Skrypt wygląda następująco:

- 1) Wykonaj: ruch + na ścieżce
- 2) Kiedy: zderzenie + zamek + biały; Wykonaj: uszkodz + 10 punktów
- 3) Kiedy: zderzenie + zamek + biały; Wykonaj: wybuch + ja



Kodu wędruje po ścieżce, na końcu której znajduje się biały zamek. Gdy się z nim zderzy, uszkodza go oraz dokonuje samodestrukcji.



Zamek - baza gracza

Obiekt reprezentuje bastion gracza, który należy uchronić przed nadciągającymi przeciwnikami. W celu zbalansowania rozgrywki należy zmienić mu następujące cechy:

- Siła punktu - 10
- Czas przeładowania punktu - 0,50
- Siła pocisku - 5
- Czas przeładowania pocisku - 0,50

Skrypt wygląda następująco:

- 1) Kiedy: mysz + lewy + kodu; Wykonaj: strzał + punkt
- 2) Kiedy: mysz + lewy + blue (przycisk)
- 3) ORAZ Kiedy: punkty + żółty + większe lub równe + 10 punktów; Wykonaj: -punkt + żółty + 10 punktów
- 4) ORAZ Wykonaj: czas przeładowania punktu - 0,05





5) ORAZ Wykonaj: ustaw wynik + niebieski + 1 punkt

6) Kiedy: punkty + żółty + większy lub równy + 5 punktów

7) ORAZ: Kiedy: mysz + lewy + zamek + czarny + strzał + pocisk + raz

8) ORAZ Wykonaj: -punkt + żółty + 5 punktów + raz

9) Kiedy: punkty + niebieski + równy + 1 punkt

10) ORAZ Kiedy: czas + 5 sek; Wykonaj: czas przeładowania punktu - 0,50

11) ORAZ Wykonaj: ustaw wynik + niebieski + zero punktów

12) Kiedy: czas + 1 sek; Wykonaj: +punkt + żółty + 1 punkt

13) Kiedy: życie + mniejsze lub równe + 0 punktów; Wykonaj: Koniec

Zamek nie porusza się w trakcie rozgrywki, a wszystkie rozkazy są wydawane przez gracza za pośrednictwem kursora myszy. Kiedy zostanie wskazany kodu, zamek strzeli do niego. Gdy zostanie wybrany niebieski przycisk i gracz posiada 10 monet, skróci się czas przeładowania punktów. Nie jest to efekt stały - wynik niebieski ustawiony na wartość "1" uruchamia linię dziewiątą, a ta z kolei uruchamia zegar. Po 5 sekundach szybkość strzelania jest przywracana do normy, a wynik niebieski zerowany.

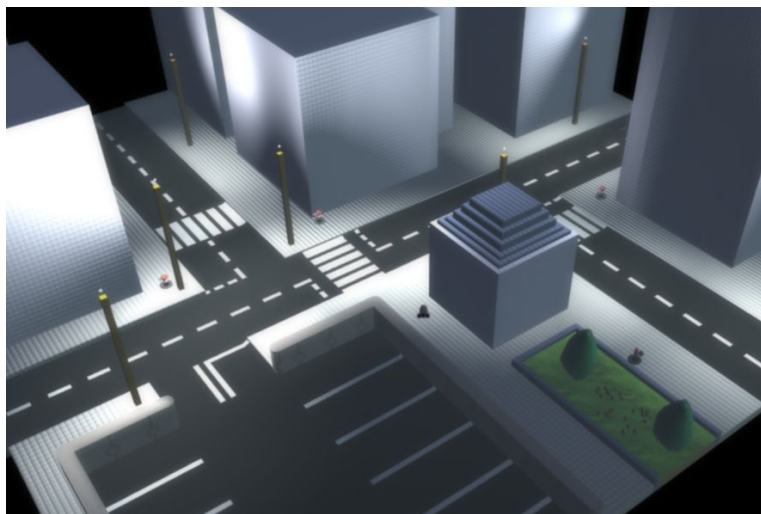
Linia szоста opisuje możliwość strzału do zamku przeciwnika. Jeśli na koncie gracza znajduje się 5 monet, ten może odpalić w jej kierunku pocisk (oczywiście płacąc za niego).

Dwie ostatnie linie programu dodają globalne warunki. Wiersz 11 sprawia, że co sekundę gracz otrzymuje

jedną monetę, a 12 informuje, że gra kończy się po zniszczeniu przez kodu białej wieży.

Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 11 - Kodu City - tworzenie świata



Cel

Podczas realizacji tego projektu uczniowie poznają zaawansowane techniki budowania przestrzeni przeznaczonej do rozgrywki.

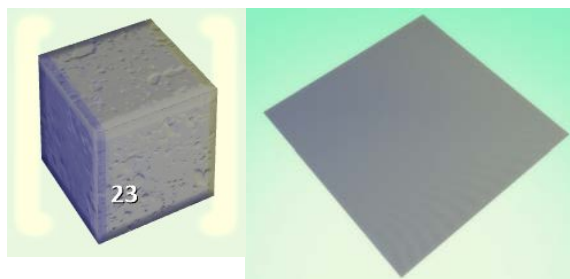
Efekt

Uczniowie przygotują świat z zaprogramowanym głównym bohaterem, którym będą mogli eksplorować planszę.

Konstrukcja ulic

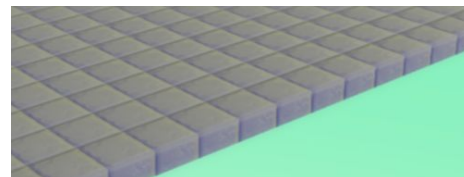
Krok 1

Wykorzystując narzędzie *Rysuj, dodaj lub usuń podłoże* należy stworzyć przestrzeń, na której toczyć się będzie rozgrywka. Jej kształt może być dowolny, a powierzchnia na tyle duża, aby zmieściła na sobie budynki i ulice. Do zbudowania placu najlepiej wykorzystać teksturę o numerze 23.



Krok 2

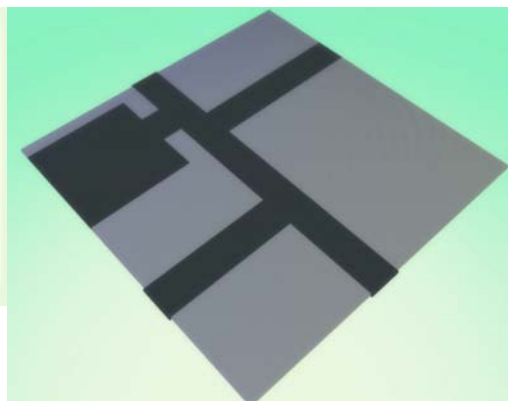
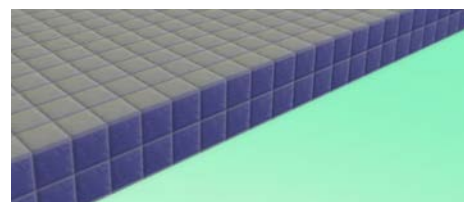
Przy pomocy narzędzia *Dodaj lub obniż podłoże* przy wybranym "magicznym pędzlu" należy podnieść plac na niewielką wysokość.



Krok 3

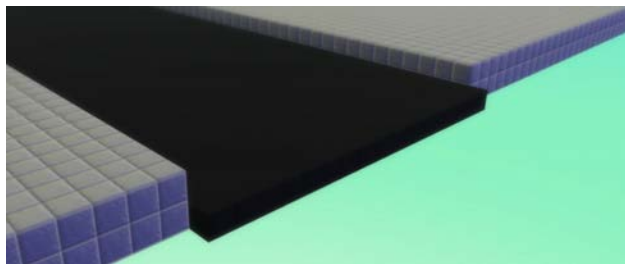
Wybierając teksturę o numerze 45 należy przygotować układ ulic oraz/lub placów parkingowych.

Przy konstruowaniu dróg najlepiej skorzystać z "przeciąganego, kwadratowego pędzla". W przypadku parkingu wystarczy standardowy "kwadratowy".



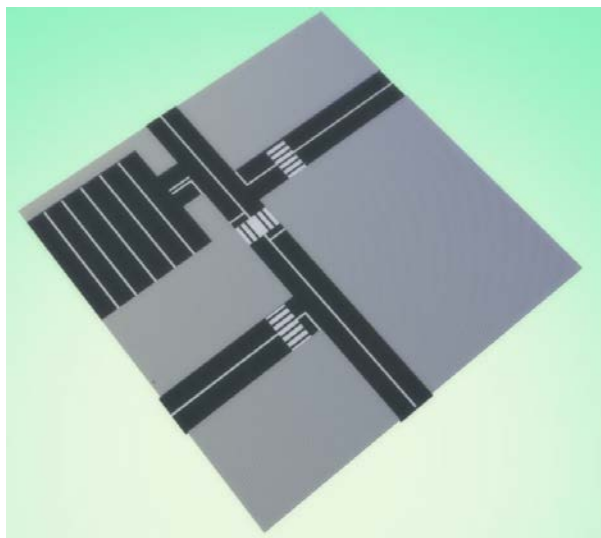
Krok 4

Posiłkując się “magicznym pędzlem” oraz narzędziem obniżającym podłoże należy opuścić ulicę. W ten sposób uzyskuje się również krawężnik.

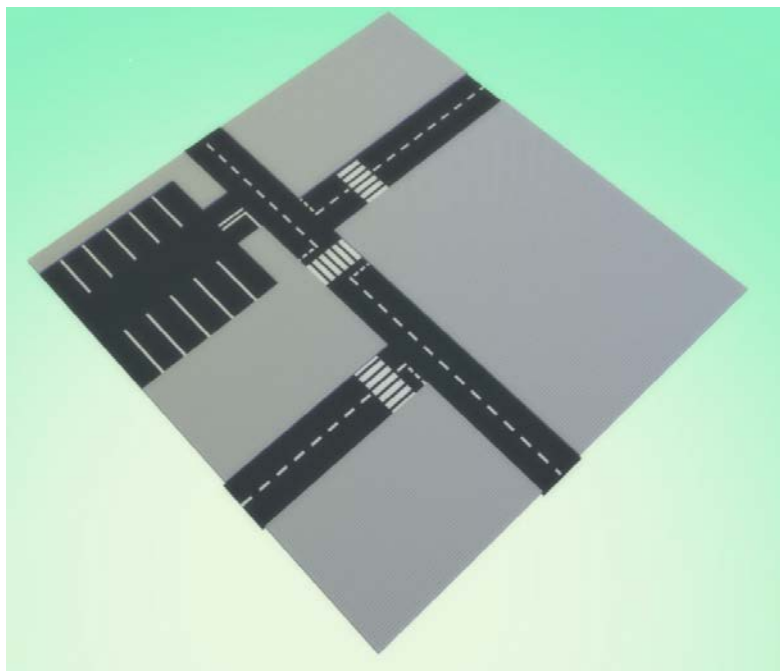


Krok 5

Należy wybrać nową teksturę o numerze 114 i z jej pomocą nanieść na drogę pasy - zarówno przejścia dla pieszych, jak i te, które oddzielają pasy jezdni. Do konstrukcji tego etapu warto wykorzystać “przeciągany pędzel”.

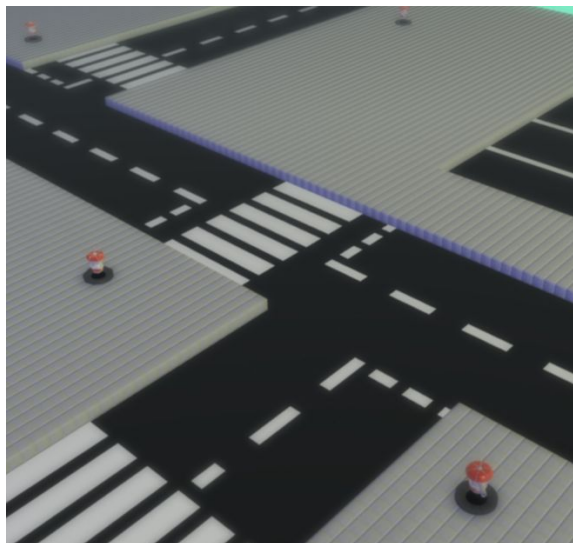


Następnie, ponownie wykorzystując teksturę 45 należy “obrobić” namalowane linie - czy to poprawić niedoskonałości, czy rozdzielić pasy.



Krok 6

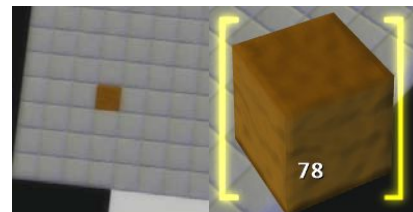
Ostatni etap przygotowania ulic polega na rozmieszczeniu kilku hydrantów. W tym celu należy umieścić kilka obiektów o nazwie “patyk” w czerwonym kolorze wzdłuż drogi.



Konstrukcja oświetlenia i ścieżek

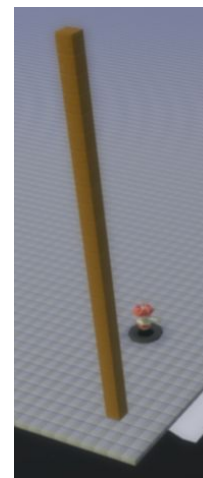
Krok 1

Wykorzystując teksturę o numerze 78 należy rozmieścić kilka słupów, na których znajdzie się oświetlenie. Belkę przygotowuje się rysując jeden kwadrat w miejscu, gdzie ta będzie miała stać.



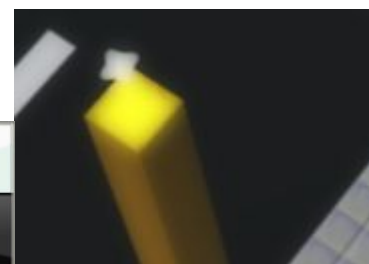
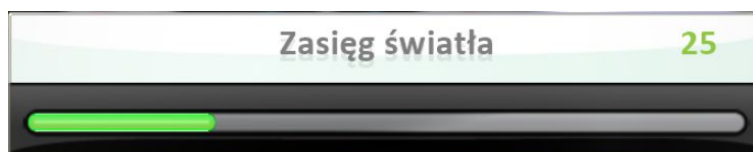
Krok 2

Przy pomocy narzędzia do podnoszenia terenu należy wynieść wcześniej postawiony brązowy kwadrat. Czynność najprościej wykonać, gdy kamera podglądu jest skierowana od góry bezpośrednio nad podnoszonym punktem. Wszystkie słupy należy podnieść do podobnego poziomu.



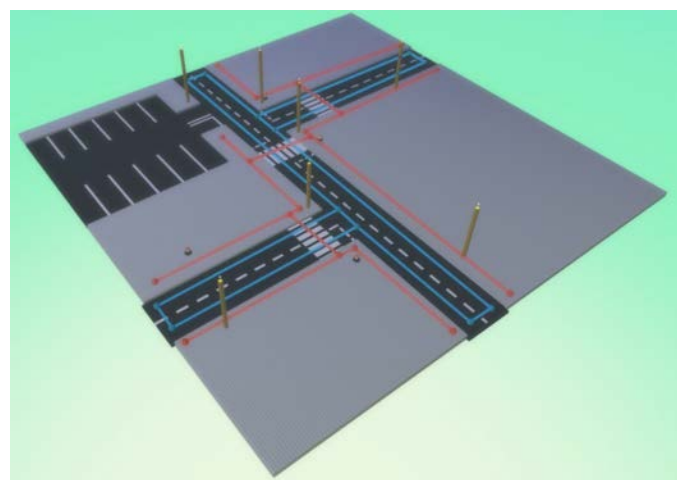
Krok 3

Na szczycie każdego słupa należy umieścić obiekt światło. Należy zmienić jego parametr zasięg światła do poziomu zbliżonego do wartości 25.



Krok 4

Wykorzystując narzędzie, które umieszcza ścieżki należy nanieść sieć dwóch tras o odmiennych kolorach. Pierwsza ma być poprowadzona wzdłuż ulic, druga wzdłuż chodników i przejść dla pieszych.



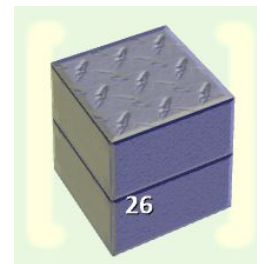
Konstrukcja budynków

Krok 1

Budowę struktur należy rozpocząć od naniesienia ich zarysu w obrębie planszy. Można wykorzystać do tego teksturę o numerze 26. Kształty budynków mogą być dowolne.

Krok 2

Wykorzystując “magiczny pędzel” oraz narzędzie do podnoszenia podłoża należy wypiętrzyć zarysy wszystkich struktur. Modyfikując jego rozmiary można unosić tylko część konstrukcji uzyskując w ten sposób gzymsy, dachy.



Konstrukcja dodatkowych elementów

Wykorzystując narzędzie do tworzenia ścieżek można umieścić na planszy mury oraz kwiaty. Używając odpowiednich struktur oraz nieznacznie zmieniając wysokość podłoża istnieje możliwość zainstalowania przestrzeni zielonej. Tworząc nierówności w jezdni i uzupełniając je wodą można uzyskać kałuże.

Ustawienia parametrów świata

Należy zmienić następujące ustawienia świata:

- Niebo: 2
- Oświetlenie: ciemność

Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.

Kodu - gracz

Obiekt jest postacią sterowaną przez gracza. Z uwagi na przyszłe zmiany w programie należy zmienić mu następujące ustawienia: zaznaczyć cechę "Do utworzenia" oraz zmienić "Maks utworzenia" na wartość 1.

Skrypt wygląda następująco:

- 1) Kiedy: klawiatura + WASD; Wykonaj: ruch
- 2) Wykonaj: podąż (kamera)
- 3) Kiedy: klawiatura + spacja; Wykonaj: skocz
- 4) Kiedy: klawiatura + E; Wykonaj: 1 osoba (kamera)

Kodu jest sterowany przez gracza z pomocą klawiatury. Może się przemieszczać i skakać, a kiedy wciskany jest klawisz "E", kamera ukazuje podgląd pierwszej osoby. W każdej innej sytuacji widok będzie ustalony zza pleców obiektu.

Patyk - hydrant

Jeden z hydrantów należy zaprogramować tak, aby jednorazowo utworzył kodu, aby ten mógł znaleźć się w rozgrywce.

Skrypt jest następujący:

- 1) Wykonaj: utwórz + Kodu 1 + raz

Program można przetestować wybierając narzędzie *Graj grę*.

Zajęcia 12 - Kodu City - programowanie obiektów

Cel

Podczas realizacji tego projektu uczniowie poznają metody unikania kolizji między obiektami oraz tworzenia zamkniętych pętli.

Efekt

Uczniowie ukończą świat Kodu City i wypełnią go dynamicznymi obiektami.

Konstrukcja lądowiska

Krok 1

Na dachu jednego z budynków należy utworzyć lądowisko. Z pomocą podstawowych narzędzi rysowania terenu należy namalować prosty, płaski wzór.

Krok 2

Dookoła lądowiska należy umieścić obiekty typu wiązka, a w jego centrum samolot.

Programowanie obiektów

Przy aktywnym narzędziu *Dodaj i programuj obiekty* należy wskazać kursorem myszy obiekt, który ma zostać zaprogramowany. Gdy nad postacią/przedmiotem pojawi się chmurka z komunikatem, należy wcisnąć prawy przycisk myszy. To sprawi, że otworzy się edytor skryptu.

Kodu - pieszy

W rozgrywce występuje wiele takich obiektów, więc należy przygotować jednego z nich, a następnie go powielić. Kodu - przechodzeń pełni rolę biernego świadka wydarzeń w grze.

Skrypt wygląda następująco:

- 1) Wykonaj: ruch + na ścieżce + czerwony
- 2) Kiedy: oczy + kodu + blisko; Wykonaj: obrót + w przód
- 3) Kiedy: oczy + kodu + blisko; Wykonaj: ruch + unikaj
- 4) Kiedy: oczy + rower + blisko; Wykonaj: ruch + unikaj

Kodu porusza się wzdłuż chodników po przygotowanej wcześniej ścieżce. Kiedy dostrzeże innego kodu, obraca się w jego kierunku, lecz kiedy staną sobie na drodze, wymijają się. Gdy przed obiektem pojawi się rower, ten również go ominie.

Rower - pojazd na drodze

W rozgrywce występuje wiele takich obiektów, więc należy przygotować jednego z nich, a następnie go powielić. Rower jest uczestnikiem ulicznego ruchu.

Skrypt wygląda następująco:

- 1) Kiedy: oczy + kodu + in front + blisko + nie; Wykonaj: ruch + na ścieżce + niebieski
- 2) Kiedy: oczy + rower + in front + blisko; Wykonaj: ruch + unik

Rower porusza się po ulicy tak długo, jak nie pojawi się przed nim kodu. Oznacza to, że zawsze przepuści obiekt na pasach. Kiedy pojawi się przed nim inny pojazd, próbuje go ominąć.

Rower - pojazd gracza

W rozgrywce występuje kilka takich obiektów, więc należy przygotować jednego z nich, a następnie go powielić i umieścić na parkingu lub wzdłuż drogi. Rower pozwala na szybsze poruszanie się gracza.

Skrypt zawiera się na dwóch stronach.

Pierwsza wygląda następująco:

- 1) Kiedy: zderzenie + kodu + fioletowy; Wykonaj: zanikanie + jego
- 2) ORAZ Wykonaj: przejdź + strona 2

Rower w trakcie wykonywania skryptu z pierwszej strony oczekuje jedynie na zderzenie z graczem. Gdy to nastąpi, kodu znika, a skrypt przechodzi na stronę drugą - od tej pory rower będzie sterowany z pomocą klawiatury.

Druga strona wygląda następująco:

- 1) Kiedy: klawiatura + spacja; Wykonaj: wystrzel + kodu 1
- 2) ORAZ Wykonaj: ignoruj
- 3) ORAZ Wykonaj: przejdź + strona 1
- 4) Kiedy: klawiatura + WASD; Wykonaj: ruch
- 5) Wykonaj: podąż (kamera)

Wysiadanie z roweru następuje poprzez wciśnięcie klawisza spacji. Powstaje wtedy kodu, kamera porzuca obecny obiekt, a skrypt wraca na stronę 1. Przez resztę czasu działania skryptu, gracz steruje pojazdem z pomocą klawiszy W, A, S oraz D, a kamera podąża za rowerem.

Samolot - pojazd gracza

Samolot pozwala na szybsze poruszanie się gracza. Należy go umiejscowić na przygotowanym wcześniej lądowisku.

Skrypt zawiera się na dwóch stronach.

Pierwsza wygląda następująco:

- 1) Kiedy: zderzenie + kodu + fioletowy; Wykonaj: zanikanie + jego
- 2) ORAZ Wykonaj: przejdź + strona 2
- 3) Wykonaj: ruch + w dół

Samolot w trakcie wykonywania skryptu z pierwszej strony oczekuje na zderzenie z graczem. Gdy to nastąpi, kodu znika, a skrypt przechodzi na stronę drugą - od tej pory samolot będzie sterowany z pomocą klawiatury. Bez pasażera w środku, obiekt opada na dół.

Druga strona wygląda następująco:

- 1) Kiedy: klawiatura + spacja; Wykonaj: wystrzel + kodu 1
- 2) ORAZ Wykonaj: ignoruj
- 3) ORAZ Wykonaj: przejdź + strona 1
- 4) Kiedy: klawiatura + WASD; Wykonaj: ruch
- 5) Wykonaj: podąż (kamera)
- 6) Kiedy: klawiatura + E; Wykonaj: ruch + w górę
- 7) Kiedy: klawiatura + Q; Wykonaj: ruch + w dół

Wysiadanie z samolotu następuje poprzez wciśnięcie klawisza spacji. Powstaje wtedy kodu, kamera porzuca obecny obiekt, a skrypt wraca na stronę 1. Przez resztę czasu działania

skryptu, gracz steruje pojazdem z pomocą klawiszy W, A, S oraz D, a kamera podąża za samolotem. Wznoszenie i opadanie realizowane jest z pomocą klawiszy E oraz Q.

Chaty - system wind

Chaty należy umieścić w dwóch odległych miejscach np. jedną na chodniku, a drugą na dachu budynku obok lądowiska. Kiedy gracz zderzy się z jedną z nich, to druga go

wystrzeliwuje.

Skrypty pierwszej windy wygląda następująco:

- 1) Kiedy: zderzenie + kodu + fioletowy;
Wykonaj: ustaw wynik + wynik A + 1 punkt
- 2) Kiedy: punkty + wyniku A + nie jest równy + 0 punktów
- 3) ORAZ Kiedy: zderzenie + kodu + fioletowy;
Wykonaj: znikanie + jego
- 4) Kiedy: punkty + wynik B + równa się + 1 punkt; Wykonaj: wystrzel + Kodu 1
- 5) ORAZ Wykonaj: ustaw wynik + wynik B + 0 punktów

Skrypty drugiej windy wygląda następująco:

- 1) Kiedy: zderzenie + kodu + fioletowy;
Wykonaj: ustaw wynik + wynik B + 1 punkt
- 2) Kiedy: punkty + wyniku B + nie jest równy + 0 punktów
- 3) ORAZ Kiedy: zderzenie + kodu + fioletowy;
Wykonaj: znikanie + jego
- 4) Kiedy: punkty + wynik A + równa się + 1 punkt; Wykonaj: wystrzel + Kodu 1
- 5) ORAZ Wykonaj: ustaw wynik + wynik A + 0 punktów

Gdy kodu gracza zderzy się z windą, zmienna globalna jest ustawiana do wartości 1. To sprawia, że druga winda wystrzeliwuje nowego bohatera obok siebie i odblokowana jest możliwość usunięcia starego. Powstanie nowego kodu zeruje zmienną, która odpowiada za wystrzeliwanie go.

Program można przetestować wybierając narzędzie *Graj grę*.