



2021-2-IT02-KA210-SCH-000048086

Erasmus+



PLAY

PROTECT, LEARN, ENGAGE AND ENJOY

Toolkit



A.M.E.F.E

asociación malagueña de
educación y formación europea



Moduł 1

Część 1 – Opis i cele projektów	1
Część 2 – Ramy kompetencji cyfrowych dla nauczycieli (DigCompEdu)	5
Część 3 – Grywalizacja	9
Część 4 - Wartość edukacyjna stosowania podejścia skoncentrowanego na technologii.	13
Część 5 – Wyniki projektu	15

Moduł 2

Część 1 - Platforma Unity 3D: zastosowania i możliwości	17
Część 2 - Interfejs	20
Część 3 - Jak tworzyć kody przy użyciu języka programowania C#	56
Część 4 - Wejście, fizyka, interfejs użytkownika, sceny i renderowanie	71
Część 5 - Praktyczny przewodnik z przykładami kodów	83



Moduł 1

Część 1 – Opis i cele projektów

Baw się – chroń, ucz się, angażuj i ciesz się to projekt finansowany przez Unię Europejską w ramach programu Erasmus+, unijnego programu w dziedzinie Edukacji, Szkoleń, Młodzieży i Sportu. Partnerstwo tworzą włoska i polska szkoła, Hiszpańskie Stowarzyszenie Europejskiej Edukacji i Szkoleń oraz Paidea, włoska firma edtech. Pomysł zrodził się z chęci uczynienia ze szkoły środowiska, w którym uczniowie mogą swobodnie eksperymentować i poszerzać swoją wiedzę w praktyczny sposób, przy wsparciu technologii. Przedmioty nauczane w szkole powinny być bezpośrednio związane ze światem, w którym żyjemy i powinny sprzyjać rozwojowi umiejętności cyfrowych, które są niezbędne w technologicznym i stale ewoluującym społeczeństwie, zgodnie z wytycznymi Digcomp.

Projekt PLAY ma na celu promowanie umiejętności cyfrowych uczniów w wieku od 11 do 14 lat oraz podnoszenie świadomości znaczenia ochrony naszej planety poprzez bezpośrednie angażowanie ich w tworzenie gry wideo za pośrednictwem platformy Unity 3D oraz zachęcanie ich do nauki język programowania C#.

Jeden z Celów Zrównoważonego Rozwoju przewidzianych w Agendzie 2030 podkreśla potrzebę podjęcia pilnych działań w celu przeciwdziałania zmianom klimatycznym i ich skutkom. PLAY wykorzystuje podejście ludyczne, aby zwiększyć świadomość uczniów na temat znaczenia modyfikowania naszego stylu życia i wspierać rozwój kompetencji cyfrowych, które stanowią ważny zestaw umiejętności w świecie pracy. Po specjalistycznym szkoleniu uczniowie opracowali grę wideo 3D, w której można wchodzić w interakcje z różnymi środowiskami, aby zrozumieć główne przyczyny marnotrawstwa zasobów i zanieczyszczeń, a także poznać zrównoważone alternatywy, które przyczynią się do zmniejszenia naszego wpływu na środowisko. uderzenie.



Unity to wieloplatformowy silnik gier używany głównie do tworzenia gier wideo i symulacji na komputery PC, konsole, urządzenia mobilne i strony internetowe. Możliwe jest programowanie w C#, Javie, UnityScript lub Boo, języku podobnym do Pythona. W projekcie wybrano język programowania C#, ponieważ jest bardzo łatwy w użyciu, więc jest odpowiedni dla początkujących programistów.

Opracowali grę wideo pod okiem doświadczonych programistów i trenerów. Mówiąc dokładniej, w ramach działania przewidziano następujące etapy:

- Trening i praktyka. W pierwszym etapie uczniowie zostali przeszkoleni z obsługi Unity oraz programowania w języku C# (wspólnie z nauczycielami); po otrzymaniu podstaw teoretycznych przećwiczyli pisanie wydajnego kodu.
- Plan rozwoju. Zorganizowano kilka sesji burzy mózgów, podczas których uczniowie mogli przedyskutować swoje pomysły i współpracować. Następnie musieli przeprowadzić badania nad głównymi przyczynami zmian klimatu i nad bardziej ekologicznymi rozwiązaniami. Doprowadziło to do wyboru sześciu ustawień, które reprezentują interaktywne tło i poziomy gry.
- Produkcja. To jest etap faktycznego rozwoju gry.
- Testowanie i walidacja. Po ukończeniu produkcji ostateczna wersja została przetestowana przez uczniów szkół partnerskich, którzy nie brali udziału w projekcie.



Nauczyciele aktywnie uczestniczyli we wszystkich etapach produkcji tego działania, co pozwoliło im poznać innowacyjne metody nauczania, a także rozwinąć nowe umiejętności.

Główne cele projektu można podsumować w następujący sposób:

1. promowanie rozwoju umiejętności cyfrowych i programistycznych u młodych uczniów, które uczynią ich konkurencyjnymi na rynku pracy;
2. podnoszenie świadomości wśród uczniów i ogółu społeczeństwa na temat przyczyn globalnego ocieplenia i znaczenia walki ze zmianami klimatycznymi poprzez zmianę naszych nawyków;
3. promowanie rozwoju innowacyjnych i cyfrowych metod nauczania poprzez dostarczanie nauczycielom niezbędnej wiedzy i kompetencji, w celu zwiększenia poziomu motywacji uczniów i ich wyników w nauce.

Tworząc grę, uczniowie mieli okazję praktycznie poeksperymentować z programowaniem i nabyli kompetencje cyfrowe, które przydadzą im się w przyszłym życiu zawodowym. Programowanie komputerowe odgrywa integralną rolę w naszym świecie. Nauka podstaw programowania może zapewnić studentom przewagę konkurencyjną w tym napędzanym technologią świecie. Umiejętności programistyczne są również ważne dla uczenia się innowacji, tworzenia przyjaznych dla środowiska rozwiązań globalnych problemów. Studenci jednocześnie poszerzyli swoją wiedzę na tak ważny temat i stali się odpowiedzialnymi osobami, które wezmą aktywny udział w walce ze zmianami klimatycznymi. Uświadomienie sobie takiego zagrożenia i pomoc innym w zrozumieniu konsekwencji naszych działań miało istotny wpływ na życie uczestników, uczynienie ich bardziej odpowiedzialnymi i świadomymi środowiska i ich roli w społeczeństwie. Przewidziane działania miały również na celu promowanie integracji i pracy zespołowej. Oczywiście niektórzy uczniowie byli bardziej biegli w korzystaniu z technologii cyfrowych, a inni byli bardziej zainteresowani środowiskiem i wrażliwi na ten temat. Pracując razem, mieli możliwość uczenia się od siebie nawzajem.



A.M.E.F.E.
asociación magisteria de
educación y formación europea

Paidea



 **Erasmus+**
Enriching lives, opening minds.

W nowych programach nauczania konieczne jest zastosowanie odpowiednich metod, np. grywalizacji lub uczenia się opartego na grach, aby zaangażować uczniów, pobudzić ich kreatywność, dostosować programy nauczania i uatrakcyjnić treści. Każdy z tych aspektów stanowi kluczową zmienną, którą nauczyciele powinni wziąć pod uwagę, aby osiągnąć cele pedagogiczne i edukacyjne. Wymaga to, aby nauczyciele posiadali znaczące kompetencje cyfrowe nauczycieli: umożliwia to nauczycielowi korzystanie z technologii nie tylko jako wsparcia dla ich istniejących praktyk, ale także do wykorzystania jej do odbudowy placówek edukacyjnych i szkoleniowych.



Część 2 – Ramy kompetencji cyfrowych dla nauczycieli (DigCompEdu)

Europejskie ramy kompetencji cyfrowych nauczycieli (DigCompEdu) to naukowo uzasadnione ramy opisujące, co dla nauczycieli oznacza posiadanie kompetencji cyfrowych. Zapewnia ogólne ramy odniesienia wspierające rozwój kompetencji cyfrowych nauczycieli w Europie. DigCompEdu jest skierowany do edukatorów na wszystkich poziomach edukacji, od wczesnego dzieciństwa po szkolnictwo wyższe i dorosłych, w tym kształcenie i szkolenie ogólne i zawodowe, edukację ze specjalnymi potrzebami i konteksty uczenia się pozaformalnego.

Kompetencje cyfrowe są niezbędne w nauce i pracy oraz sprawiają, że jesteśmy aktywnymi członkami społeczeństwa. W odniesieniu do edukacji równie ważna jak zrozumienie samych kompetencji jest wiedza o tym, jak je rozwijać.

Czym są kompetencje cyfrowe?

Kompetencje cyfrowe są jednym z elementów ośmiu kompetencji kluczowych i odnoszą się do świadomego i krytycznego korzystania z całego spektrum technologii cyfrowych umożliwiających zdobywanie informacji, komunikację i podstawowe rozwiązywanie problemów we wszystkich aspektach życia.

Czym jest framework DigCompEdu?

Ramka wyróżnia 6 obszarów krycia:

1. Zaangażowanie zawodowe - wykorzystanie efektywnej technologii komunikacyjnej, współpraca i rozwój komunikacji.
2. Zasoby cyfrowe – wykorzystanie i użytkowanie zasobów, tworzenie, ocena i udostępnianie zasobów.
3. Nauczanie i uczenie się – stosowanie różnych metod i narzędzi w nauczaniu i uczeniu się.
4. Ewaluacja - kontrola różnych metod i narzędzi do rozważenia i uzyskania informacji zwrotnej.
5. Wspieranie uczenia się – wykorzystanie technologii produkcyjnych w celu wzmocnienia i uczynienia środowisk edukacyjnych bardziej dostępnymi, ekskluzywnymi, integracyjnymi i ukierunkowanymi na naukę.

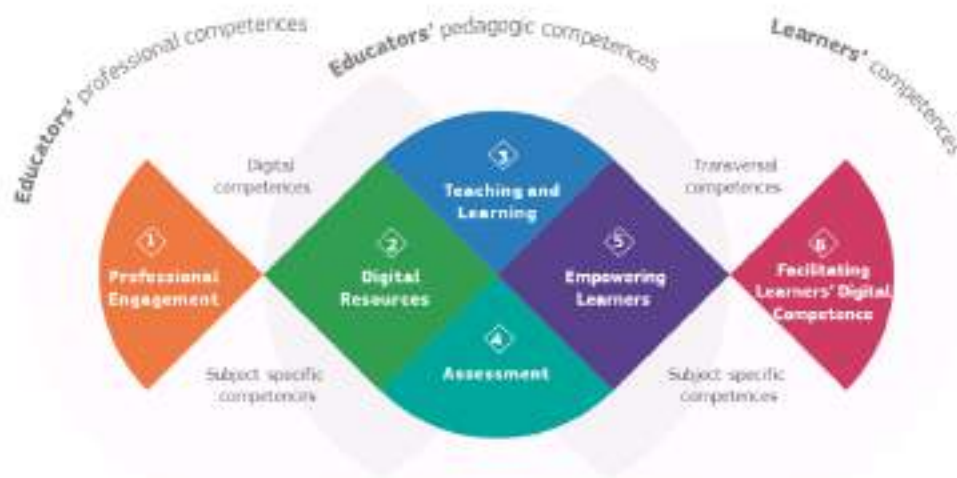


6. Wspieranie kompetencji osób uczących się – wspieranie i ułatwianie rozwoju ich kompetencji.



Europejskie ramy kompetencji cyfrowych nauczycieli to ustandaryzowane ramy, które stanowią punkt odniesienia i przewodnik dotyczący pracy z kompetencjami cyfrowymi nauczycieli. Określa 6 przydatnych kategorii cyfrowych działań edukacyjnych, które są podzielone na określone kompetencje.

Ramy mają na celu pokazanie, w jaki sposób technologie cyfrowe mogą być wykorzystywane do wspierania rozwoju zawodowego w kształceniu i szkoleniu. Nacisk kładziony jest nie tylko na umiejętności techniczne, ale przede wszystkim na kontekst nauczania i uczenia się. Same ramy mają na celu „zapewnienie wspólnego języka i logiki jako punktu wyjścia do opracowywania, porównywania i omawiania różnych instrumentów rozwoju kompetencji cyfrowych nauczycieli na poziomie krajowym, regionalnym lub lokalnym”.



Aby lepiej zrozumieć naturę kompetencji cyfrowych, Komisja Europejska przygotowała Europejskie Ramy Kompetencji Cyfrowych dla Obywateli (DigComp), które zostały podzielone na pięć obszarów: analiza informacji i danych; komunikacja i współpraca; tworzenie treści cyfrowych; bezpieczeństwo; i rozwiązywanie problemów. W sumie obejmują one 21 kompetencji.



Ciągły rozwój kompetencji cyfrowych dotyczy zarówno dzieci, jak i nauczycieli. Każdy, niezależnie od wieku, jeśli tylko nabeździe umiejętności posługiwania się nowymi technologiami, ułatwi mu funkcjonowanie we współczesnym społeczeństwie. Konieczne jest edukowanie dzieci w zakresie ich odpowiedzialności za wdrażanie i przygotowanie do życia we współczesnym świecie, w którym głównym przesłaniem



jest szybki rozwój mediów i technologii. Niezbędne jest również doskonalenie przygotowania nauczycieli w zakresie nauczania z wykorzystaniem nowoczesnych technologii w taki sposób, aby celowo wspierali jeden prowadzony proces dydaktyczny, wspierając wzrost efektywności nauczania i uczenia się. Zalecenia Parlamentu Europejskiego i Rady oraz wiele aktów prawnych dotyczących dowodów edukacyjnych dla działalności edukacyjnej koncentrowały się na rozwoju naukowców studiujących je do życia we współczesnym społeczeństwie. Jedną z takich kompetencji są kompetencje cyfrowe.



Aby uzyskać dodatkowe informacje, odwiedź następującą stronę internetową: https://joint-research-centre.ec.europa.eu/digcompedu_en



Część 3 – Grywalizacja

Istnieje wiele definicji grywalizacji. Według Briana Burke'a (wiceprezes ds. badań w firmie technologicznej Gartner i ekspert w dziedzinie architektury korporacyjnej), „Gamifikacja to wykorzystanie mechaniki gry i doświadczonego projektanta do cyfrowego angażowania i motywowania graczy do osiągnięcia ich celów”. Słownik oksfordzki definiuje grywalizację jako „zastosowanie typowych elementów gry (np. zdobywanie punktów, współzawodnictwo z innymi, zasady gry) w innych obszarach aktywności, [...] aby zachęcić do zaangażowania”. Według słownika Cambridge gamifikacja to „praktyka polegająca na upodabnianiu czynności do gier, aby uczynić je ciekawszymi lub przyjemniejszymi”. Według Yu-kai Chou (autora i międzynarodowego głównego mówcy na temat grywalizacji i projektowania behawioralnego) istnieje 8 głównych napędów, które sprawiają, że gry są atrakcyjne dla ludzi:

1. **Epickie znaczenie i powołanie** → Jesteś częścią czegoś większego niż ty sam.
2. **Rozwój i realizacja** → Jesteś zmotywowany, ponieważ czujesz, że się poprawiasz i osiągasz mistrzostwo.
3. **Wzmacnianie kreatywności i informacji zwrotnych** → Dawanie ludziom nawet prostych elementów, takich jak klocki LEGO® i pozwolenie im na znalezienie własnego sposobu na ich łączenie, to bardzo angażujący proces. Tak jak zrobili to nasi uczniowie, gdy poproszono ich o stworzenie „lalek” z odpadów plastikowych i papierowych (zdjęcia).



A.M.E.F.E.
asociación municipal de
educación y temas de Europa

Paidea



Erasmus+
Enriching lives, opening minds.



4. **Własność i posiadanie** → Kiedy czujesz, że coś posiadasz, chcesz to ulepszyć, chcesz to chronić i chcesz zdobyć więcej.
5. **Wpływ społeczny i pokrewieństwo** → Wpływa na nas to, co robią ludzie wokół nas.
6. **Niedobór i niecierpliwość** → Wiąże się to z faktem, że czasami chcesz czegoś tylko dlatego, że nie możesz tego mieć.
7. **Nieprzewidywalność i ciekawość** → Jeśli nie wiesz, co będzie dalej, zawsze o tym myślisz.
8. **Strata i unikanie** → Jeśli robisz coś, chcesz uniknąć straty, ponieważ nikt nie chce, aby stało się coś złego.

Zdaniem Yu-kai Chou wszystko, co robimy, opiera się na jednym lub kilku z tych podstawowych napędów. Według innych badań gry sprawiają, że dzieci są mądrzejsze, ponieważ gry wideo zasadniczo przedstawiają użytkownikom ciągły proces uczenia się. Cały czas się rozwijają i idą do przodu. Gry są zaprogramowane tak, aby wywołać określoną reakcję u ludzi. Gra stanowi wyzwanie, a kiedy je pokonasz, w twoim mózgu uwalniana jest dopamina, więc mamy poprawę uczenia się, wielozadaniowość zwiększa połączenie i silną pętlę dopaminy w mózgu, co powoduje wewnętrzne wzmocnienie.



A.M.E.F.E.
asociación municipal de
educación y temas de Europa

Paidea



Erasmus+
Enriching lives, opening minds.



Od 2010 roku grywalizacja stała się metodologią w edukacji na całym świecie. Badania pokazują, że przekształcenie celów edukacyjnych poprzez ekscytujące wyzwania, wykorzystanie modeli gier wideo, na przykład przy użyciu odznak postępów lub dających możliwość zobaczenia wykresów wyników, zaspokajają potrzebę ludzi w zakresie kompetencji w określonej czynności i zwiększa postrzeganą wartość zadania. Grywalizacja nie oznacza zamiany lekcji w grę, ale wprowadzenie elementu gier wideo do edukacji. Gry wideo mogą stymulować naszą motywację, zainteresowanie, kreatywność, poczucie przynależności i szczęścia u ludzi. Wszystkie te uczucia są zasobami, które możemy spożytkować w naszych codziennych czynnościach.



A.M.E.F.E
asociación malagueña de
educación y temáticas europeas

Paidea



 **Erasmus+**
Enriching lives, opening minds.



Jeśli więc wprowadzimy zasady i schematy gier wideo do wszystkich zadań, które mamy do wykonania w życiu codziennym, moglibyśmy je wykonywać z większą motywacją i satysfakcją, osiągając lepsze wyniki.



Część 4 - Wartość edukacyjna stosowania podejścia skoncentrowanego na technologii.

Włączenie zasobów technologii informacyjnej do procesu uczenia się ma wiele zalet: w szczególności zapewnia alternatywne sposoby uczenia się, umożliwia nabywanie umiejętności poznawczych i wiedzy, które są ważne dla uczenia się przez całe życie, a przede wszystkim sprzyja motywacji. Według Międzynarodowego Towarzystwa Technologii w Edukacji (ISTE) w ciągu ostatniej dekady powstało wiele dzisiejszych zawodów o dużym popycie. Ponieważ postęp technologiczny napędza globalizację i transformację cyfrową, nauczyciele mogą pomóc uczniom w zdobyciu umiejętności niezbędnych do odniesienia sukcesu w karierze zawodowej w przyszłości. Pandemia COVID-19 szybko pokazuje, dlaczego edukacja online powinna być istotną częścią nauczania i uczenia się. Dzięki włączeniu technologii do istniejących programów nauczania, w przeciwieństwie do używania jej wyłącznie jako narzędzia zarządzania kryzysowego, nauczyciele mogą tworzyć wciągające i wciągające doświadczenia edukacyjne. Efektywne wykorzystanie cyfrowych narzędzi do nauki w klasach może zwiększyć zaangażowanie uczniów, pomóc nauczycielom w ulepszeniu planów lekcji i ułatwić spersonalizowane nauczanie. Technologia zapewnia uczniom łatwy dostęp do informacji, przyspieszoną naukę i przyjemne możliwości praktycznego wykorzystania zdobytej wiedzy. Umożliwia uczniom odkrywanie nowych przedmiotów i pogłębianie zrozumienia trudnych pojęć, szczególnie w dyscyplinach STEM.



A.M.E.F.E.
asociación municipal de
educación y temas de Europa

Paidea



Erasmus+
Enriching lives, opening minds.



Główne korzyści płynące z wykorzystania technologii w edukacji można podsumować w następujący sposób:

- Lepsza współpraca i komunikacja. Technologia edukacyjna może sprzyjać współpracy. Nauczyciele mogą nie tylko kontaktować się z uczniami podczas lekcji, ale także uczniowie mogą komunikować się ze sobą. Dzięki lekcjom online i grom edukacyjnym uczniowie mogą wspólnie rozwiązywać problemy. Podczas wspólnych działań uczniowie mogą dzielić się swoimi przemyśleniami i pomysłami oraz wspierać się nawzajem.
- Spersonalizowane możliwości uczenia się. Technologia umożliwia całodobowy dostęp do zasobów edukacyjnych. Można go używać do dostosowywania planów nauki dla każdego ucznia. Nauczyciele mogą tworzyć lekcje w oparciu o zainteresowania i mocne strony uczniów. Dodatkową korzyścią jest to, że uczniowie mogą uczyć się we własnym tempie i przeglądać materiał, aby lepiej zrozumieć podstawowe pojęcia.
- Ciekawość napędzana angażującymi treściami. Dzięki angażującym i edukacyjnym treściom nauczyciele mogą rozbudzać w uczniach dociekliwość i pobudzać ich ciekawość, co według badań ma związek z sukcesami w nauce. Tworzenie angażujących treści może wiązać się z wykorzystaniem AR, wideo, podcastów i oczywiście gier.



Wprowadzenie elementów technologicznych i gier wideo do procesów uczenia się w środowiskach edukacyjnych jest coraz częściej obecne w debatach na temat sposobów poprawy doświadczeń naszych młodych ludzi w kontekście, w którym technologia cyfrowa jest włączana do większej liczby dziedzin naszego życia. Pedagodzy używają gier do nauczania w ten sposób co najmniej od 1971 r., kiedy trzech uczniów-studentów stworzyło klasyczną grę edukacyjną The Oregon Trail do wykorzystania na kursie historii Stanów Zjednoczonych. Wczesne argumenty za tym podejściem często koncentrowały się na idei, że „gry wideo są zabawne i jako takie zapewniają intrygującą perspektywę zmuszania niektórych dzieci do nauki” (Silvern 10). Fakt, że gry skutecznie motywują graczy, pozostaje podstawowym uzasadnieniem nauczania za pomocą gier zorientowanych na treść. Prensky opisuje uczniów, którzy osiągnęli pełnoletność w erze internetu, jako „cyfrowych tubylców”, którzy w naturalny sposób uczą się lepiej, korzystając z technologii (Prensky, „Listen to the Natives” 9-13). Badania potwierdzają pogląd, że młodzież XXI wieku ma silne powinowactwo do technologii, zwłaszcza gier wideo i sieci komunikacyjnych (Jones i in. 6, Lenhart i in. 2-3). Inni autorzy przypisują atrakcyjność gier wciągającej mieszance czynników, takich jak wyzwanie, fantazja i interaktywność (Malone 162; Owston 978). Gry wideo mogą być bardzo stymulującym narzędziem edukacyjnym dla młodych ludzi w szkołach. Oprócz zajmowania się niektórymi treściami programowymi mogą wspierać rozwój pewnych umiejętności niezbędnych do promowania dobrych nawyków. Taki był dokładny cel projektu:

Część 5 – Wyniki projektu

Studenci zaangażowani w projekt podeszli do języka programowania, używając Scratch , a następnie Unity do stworzenia gry wideo. Pomogło im to w podniesieniu umiejętności cyfrowych; zwiększyli swoje zainteresowanie i poszerzyli swoją wiedzę na temat globalnego ocieplenia, zanieczyszczenia i eko- zrównoważenia. Zachęcono ich do znalezienia bardziej ekologicznych i zrównoważonych rozwiązań w walce z globalnym ociepleniem i zanieczyszczeniem. Stworzenie gry wideo, która obejmowała takie tematy badawcze, jak globalne ocieplenie, zrównoważony rozwój i zanieczyszczenie środowiska, wpisuje się w strategię znaną jako grywalizacja. Jest



to strategia stosowana nie tylko w szkole, ale także w środowisku pracy, aby stymulować ludzi do znajdowania kreatywnych rozwiązań problemów i zwiększać motywację uczniów lub pracowników do osiągnięcia określonego celu.

Podczas wyjazdów studenci doskonalili znajomość języka obcego (angielski), poznali zwyczaje i kulturę innego kraju zwiększając wzajemne zrozumienie, empatię i akceptację odmienności. Uczniowie i nauczyciele odwiedzając różne krajobrazy wymieniali się wiedzą i kompetencjami oraz doskonalili umiejętność pracy w zespole i współpracy. Projekt ten pomógł uczniom i nauczycielom rozwinąć sieć relacji z lokalnymi instytucjami edukacyjnymi i szkołami w innych krajach, która będzie trwała po zakończeniu projektu.



Moduł 2

Część 1 - Platforma Unity 3D: zastosowania i możliwości

WPROWADZANIE.

Postęp technologii jest niesamowity. Kto mógł sobie wyobrazić 100 lat temu, co jesteśmy w stanie zrobić dzisiaj przy wsparciu narzędzi technologicznych.

Jednym z takich narzędzi jest Unity 3D, które daje nam możliwość tworzenia rzeczywistych przestrzeni. Pragniemy podkreślić, że po zapoznaniu się z tym zestawem narzędzi użytkownicy końcowi będą mogli zbudować własną wirtualną wycieczkę po wybranym przez siebie scenariuszu, pod warunkiem, że puszczą wodze fantazji i w ten sposób nadadzą pożądany artystyczny akcent projekt.

Projekt przedstawiony w tym zestawie narzędzi został stworzony z myślą o osobach, które nie mają żadnej wiedzy na temat Unity 3D, więc jeśli chcesz nauczyć się go od podstaw to dobrze trafiłeś, zaczniemy od podstawowej konfiguracji środowiska programistycznego w Unity 3D, czym jest każdy przycisk i każde menu aplikacji, zapewniając wstępną wiedzę na temat tych narzędzi przed rozpoczęciem tworzenia wirtualnej wycieczki. Zostanie to wyjaśnione w obsłudze wymiarów w Unity 3D, ponieważ jak sama nazwa wskazuje, obsługuje 3 wymiary (x, y, z) i gdy ktoś nie ma praktyki w obsłudze programu, może to być coś nużącego, ponieważ nauczenie się otwierania na te wymiary jest kwestią przyzwyczajenia. Mając wcześniejszą wiedzę na temat narzędzia, zaczniemy od stworzenia naszego projektu i jeśli nadal nie jesteśmy pewni niektórych przycisków, wątpliwości będą wyjaśniane w miarę postępów, praktyki i zdobywania doświadczenia w obsłudze Unity 3D. Po zakończeniu tworzenia pierwszej przestrzeni lub wirtualnego spaceru uczniowie będą mogli stworzyć własne środowisko. Nie można sobie wyobrazić, jak potężne jest to narzędzie i jak łatwe jest w użyciu, nie mówiąc już o tym, jak łatwo jest stworzyć wirtualną wycieczkę za pomocą Unity 3D, dlatego zachęcamy do opracowania tej wycieczki krok po kroku, aby osiągnąć zamierzone cele .

JEDNOŚĆ 3D.



Unity 3D to narzędzie do tworzenia głównie gier wideo, niewielu wie, jak go używać i nie trzeba być ekspertem w programowaniu, aby móc go używać, ponieważ przy kilku drobnych pojęciach wymaganego języka można ustrukturyzować dowolną grę interaktywną. „Unity 3D, silnik graficzny 3D dla komputerów PC i Mac, jest używany do tworzenia gier, aplikacji interaktywnych, wizualizacji i animacji 3D. Unity obsługuje platformy takie jak PC, Mac, Nintendo, Wii, iPhone, Android i Internet za pomocą swojego Wtyczka „Unity web player”. Unity to aplikacja stworzona przez Unity Technologies, która „została założona w 2004 roku przez Davida Helgasona (CEO), Nicholasa Francisa (CCO) i Joachima Ante (CTO) w Kopenhadze w Danii po ich pierwszej grze, GooBall , nie powiodło się. Ta trójka uznała wartość silnika i narzędzi programistycznych i postanowiła stworzyć silnik, z którego każdy mógłby korzystać w przystępnej cenie. Unity Technologies otrzymało fundusze od takich firm jak Sequoia Capital, Capital WestSummit i iGlobe Partners. Sukces Unity wynika częściowo z koncentracji na potrzebach niezależnych programistów, którzy nie mogą ani stworzyć własnego silnika gry, ani niezbędnych narzędzi ani licencji, aby w pełni wykorzystać dostępne opcje. Firma koncentruje się na „demokratyzacji tworzenia gier” i uczynieniu tworzenia interaktywnych treści 2D i 3D możliwie jak najbardziej dostępnym dla jak największej liczby ludzi na całym świecie”. Wraz z pojawieniem się iPhone'a w 2008 roku, Unity było jednym z pierwszych silników, które zaczęły obsługiwać tę platformę, a dziś Unity jest używany przez 53. 1% deweloperów według ankiety przeprowadzonej przez firmę Game Developer ds. technologii mobilnych i społecznościowych, do tworzenia setek gier na urządzenia z systemem Android i iOS. Unity ma edytor wizualny, aby móc w nim tworzyć gry, ponieważ cała zawartość gry jest zbudowana z tego edytora, a sposób zachowania obiektów jest programowany za pomocą języka skryptowego (JavaScript); powyższe daje nam do zrozumienia, że nie trzeba być ekspertem w językach takich jak C++, aby móc stworzyć grę lub animację w Unity 3D. Struktura Unity polega na zarządzaniu i tworzeniu scen służących do rozwoju pożądanej aplikacji, scena może być dowolną częścią gry lub animacją, na poziomie gry lub określonym obszarze. Zaczynasz od pustej przestrzeni, w której możesz kształtować wszystko, co chcesz, używając narzędzi Unity. Ten silnik unity zawiera również edytor terenu, w którym możesz rzeźbić kształt terenu za pomocą narzędzi wizualnych oferowanych



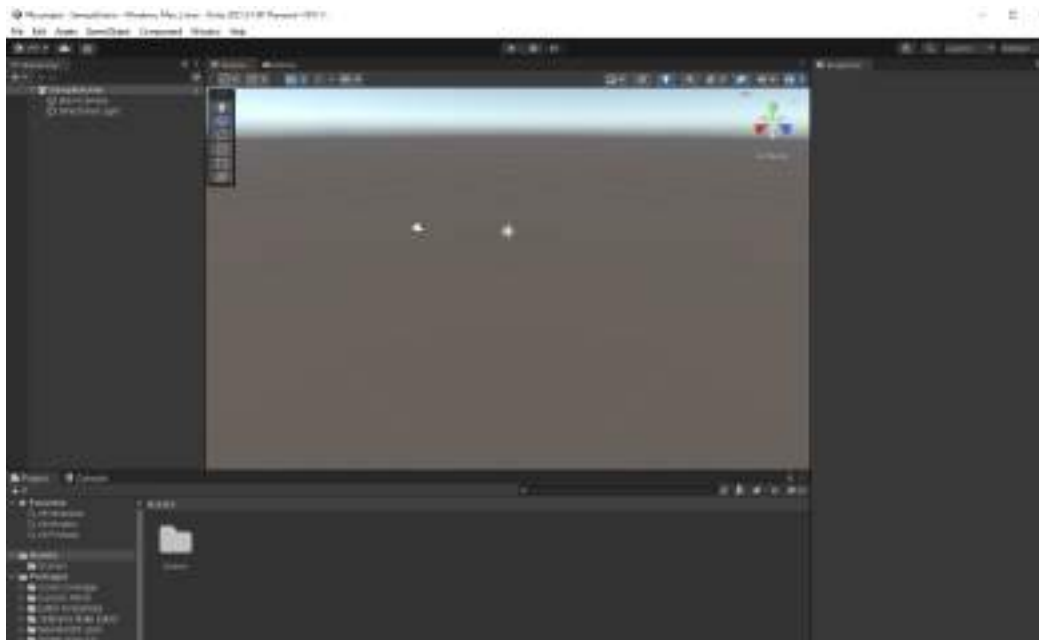
przez unity, możesz malować, tekstuować, dodawać trawę, umieszczać drzewa itp., A nawet importować inne materiały z innych silników programistycznych. Unity jest dostępne dla każdego rodzaju odbiorców, ponieważ jest rozwijane w kilku wersjach, darmowej i profesjonalnej, obie mają ogromne zalety przy opracowywaniu tego, co jest wymagane, jednak najbardziej kompletna wersja jest wersją profesjonalną, ale należy wyjaśnić, że ta darmowa wersja ma koszt, na który nie każdy może sobie pozwolić, a jeśli jesteś kimś, kto dopiero zapoznaje się z narzędziem, darmowa wersja na razie jest więcej niż wystarczająca. Oprócz Unity 3D na rynku dostępne są inne narzędzia, które mogą być nawet bardziej znane, takie jak UDK, Epic Games czy CryEngine. Jednakże, Unity 3D ma nad nimi wielką przewagę i polega na tym, że nie jesteśmy zobowiązani do programowania na systemach operacyjnych Windows, ponieważ Unity 3D ma również wersję na systemy operacyjne Mac. Unity zapewnia również możliwość importowania nie tylko terenów, ale także modeli 3D, tekstur, dźwięków itp. Za pomocą zaledwie kilku kliknięć, z których można korzystać w dowolnym momencie podczas tworzenia.



Część 2 - Interfejs

1. PIERWSZY DOSTĘP

Istnieją dwie wersje Unity, darmowa i profesjonalna. Niniejsza instrukcja będzie oparta na funkcjonalnościach dostępnych dla początkujących w darmowej wersji. Bezpłatną wersję można znaleźć na stronie internetowej Unity. Po zainstalowaniu aplikacji możemy uzyskać dostęp do wersji demonstracyjnej, w której możemy zobaczyć możliwości Unity i pozwolić nowym użytkownikom odkryć oferowane przez nią funkcjonalności, badając dzieła jej programistów.



2. INTERFEJS UŻYTKOWNIKA

W tej części instrukcji przyjrzymy się, z czego składa się interfejs użytkownika Unity. Istnieje głównie 5 obszarów interfejsu Unity, wymienionych na poniższym obrazku:

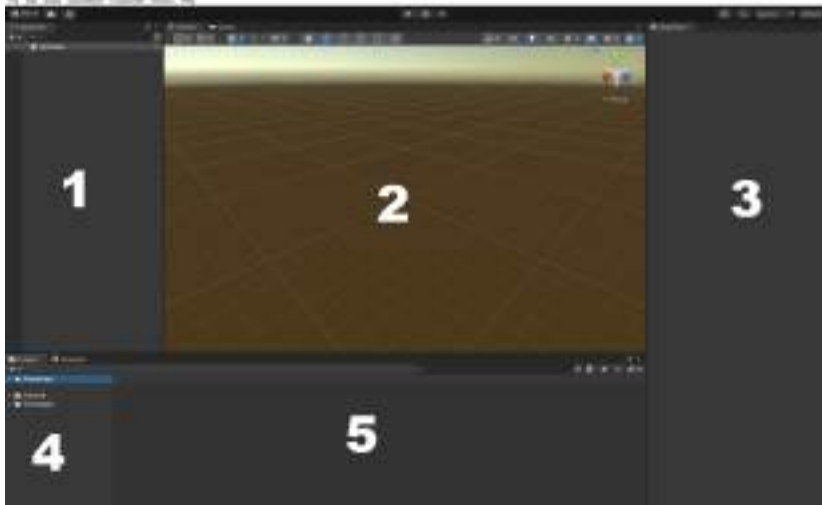


A.M.E.F.E.
asociación municipal de
educación y temas de Europa

Tpaidea



Erasmus+
Enriching lives, opening minds.



- Widok hierarchii. Gdzie będą miały wszystkie obiekty w bieżącej scenie.
- Widok sceny Jest to obszar, w którym wizualnie konstruujemy każdą scenę. Najłatwiejszym sposobem użycia widoku sceny byłoby przeciągnięcie obiektów z widoku zarządzania obiektami do widoku sceny, co spowoduje umieszczenie obiektu w scenie; następnie można go pozycjonować, skalować i obracać bez opuszczania widoku sceny. Widok sceny to także miejsce, w którym edytujesz tereny (rzeźbiąc je, malując tekstury i umieszczając elementy), umieszczasz światła i kamery oraz inne obiekty.
- Widok inspektora. Ten widok ma kilka funkcji w zależności od tego, co wybierze użytkownik, czy chce uzyskać charakterystykę obiektu, czy konfigurację terenu. Jeśli wybierzesz „aparat”, zostaną wyświetlone właściwości, w których możesz dostosować funkcje, takie jak obrót, pozycja lub skala.
- Widok projektu lub zarządzanie obiektami. To jest biblioteka naszego projektu, podobna do biblioteki narzędzi i/lub obiektów. Możesz importować obiekty 3D z różnych aplikacji do biblioteki, możesz importować tekstury i tworzyć inne obiekty, które możesz przechowywać do wykorzystania w swoim projekcie. Normalny projekt będzie zawierał kilka scen i dużą ilość zasobów, dlatego konieczne jest uporządkowanie biblioteki w różnych folderach, co ułatwi organizację zasobów i ułatwi korzystanie z nich.



- Widok wyszukiwania. Służy do lokalizowania obiektów, które potrzebujemy wykorzystać w realizacji projektu.

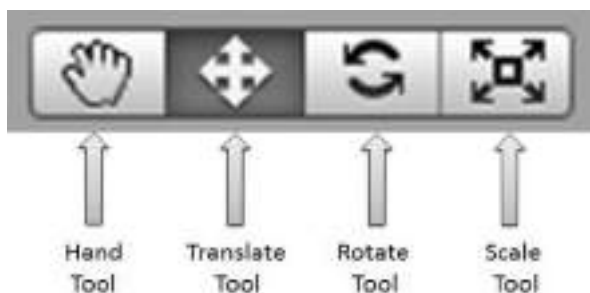
3. MENU APLIKACJI

Poniżej pokazano menu opcji Unity w lewym górnym rogu ekranu. W całej instrukcji zostaną pokazane narzędzia każdej z sekcji tego menu.



3.1 PRZYCISKI STERUJĄCE

Poniżej opcji wyświetlania zobaczysz rząd z 4 przyciskami, których możesz użyć Q, W, E, R, aby przełączać się między poszczególnymi kontrolkami, które są szczegółowo opisane poniżej:



- Hand Tool (Q): Ta kontrolka pozwala nam poruszać się w widoku sceny. o ALT pozwala nam obracać. o CTRL pozwala nam przybliżyć i oddalać SHIFT zwiększa szybkość ruchu podczas używania narzędzia.

- Translate Tool (W): Pozwala nam przesuwając dowolny wybrany obiekt w scenie w osiach X, Y i Z.

- Narzędzie Obróć (E): Pozwala nam obrócić dowolny wybrany obiekt w scenie.

- Scale Tool (R): Pozwala nam skalować dowolny wybrany obiekt w scenie.

Podczas korzystania z narzędzi translacji, obracania lub skalowania zobaczymy metauchwyty wokół wybranego obiektu z liniami dla każdej z osi. Możemy użyć tego gadżetu do wykonywania operacji translacji, obracania i skalowania.

3.2 PRZYCISKI ODTWARZANIA



A.M.E.F.E.
asociación matemática de
educación y formación europea



Erasmus+
Enriching lives, opening minds.

W Unity możesz uruchomić swoją grę bez wychodzenia z edytora, co jest dobrodziejstwem dla projektantów budujących poziomy i programistów dodających nowe mechaniki gry



Obraz przedstawia elementy sterujące odtwarzaniem, które znajdują się w górnej części edytora. Możesz wejść do widoku gry w dowolnym momencie, naciskając przycisk odtwarzania, aby zobaczyć stan projektu (pierwszy od lewej), wstrzymać, używając przycisku pauzy (w środku) lub przeskoczyć do przodu, używając prawego przycisku. Możesz grać z widoku gry lub rozszerzyć go do pełnego.

3.3 SKRYPTY

Skrypty pozwalają na zdefiniowanie logiki gry. W całym samouczku zobaczysz kilka prostych skryptów, takich jak definiowanie ruchu, włączanie i wyłączanie świateł. Operacja jest bardzo prosta, najpierw definiujesz pożądane zachowanie w skrypcie, a to jest dodawane jako komponent do obiektu, dla którego chcesz przypisać to zachowanie, możesz nawet dodać skrypt do pustego obiektu. Istnieją 3 podstawowe funkcje skryptów:

- Funkcja `Update()`: funkcja, która definiuje główną pętlę gry, więc jest wykonywana raz dla każdej renderowanej klatki.
- Funkcja `Awake()`: Zmienne zdefiniowane w ramach tej funkcji zostaną zainicjowane tuż przed rozpoczęciem gry. Ta funkcja zostanie wykonana tylko raz na czas trwania instancji skryptu.
- Funkcja `Start()`: Ta funkcja jest bardzo podobna do funkcji „`Awake()`”, ale ma zasadniczą różnicę; zostanie wykonany tylko wtedy, gdy instancja skryptu jest włączona. Pozwoli nam to opóźnić inicjalizację niektórych zmiennych stanu gry. Musimy również pamiętać, że funkcja „`Start()`” będzie zawsze wykonywana po zakończeniu funkcji „`Awake()`”.

Unity 3D ma również domyślny edytor skryptów o nazwie `uniscite`.



A.M.E.F.E.
asociación magisterfa de
educación y formación europea

Paidea

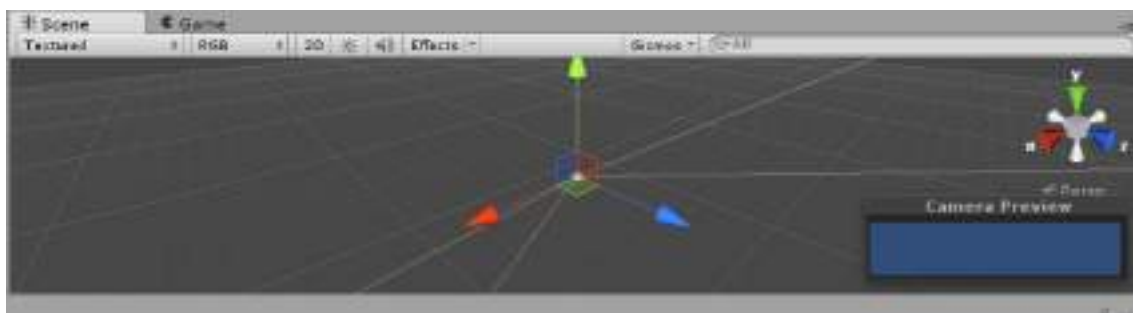


Erasmus+
Enriching lives, opening minds.

```
rotation.js - Unity3D
File Edit Search View Options Language Buffer Help
rotation.js
var rotationSpeed: float = 2.0;
function Update () {
    transform.Rotate(Vector3(0,rotationSpeed,0));
}
```

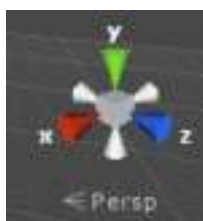
4. URUCHOMIENIE UNITY 3D

Pierwszą rzeczą do zrobienia jest uruchomienie Unity 3D. Gdy to zrobisz, załaduje się pusty projekt. Jeśli chcesz inną scenę, zajrzyj do menu: „Plik -> Otwórz scenę” i zajrzyj do katalogu projektu („Assets/scenes”). Tryby wyświetlania domyślnie widok sceny ma perspektywę 3D sceny.



Perspektywiczny

Możesz zmienić widok sceny na kilka widoków: z góry na dół, z boku i z przodu. Po prawej stronie widoku sceny zobaczysz „Gizmo”, które wygląda jak pudełko z wychodzącymi stożkami.



- Za pomocą tego Gizmo możemy zmienić perspektywę widoku.
- Kliknięcie na pole przeniesie Cię do trybu perspektywy.
- Kliknięcie na zielony stożek (y) przeniesie Cię do trybu Top-Down.



- Kliknięcie na czerwony stożek (x) przeniesie Cię do trybu bocznego (po prawej).
- Kliknięcie niebieskiej ikony (z) przeniesie Cię do trybu Przód.
- Mamy też 3 szare pachołki, które przeniosą nas do następujących trybów:
- „Tył, lewy i dolny”.

4.1 WIDOK HIERARCHICZNY

Widok hierarchii zawiera wszystkie elementy, które składają się na scenę gry. Możesz manipulować wszystkimi obiektami w scenie z tego widoku, za każdym razem, gdy wprowadzasz element do sceny, wpis dla tego elementu jest dodawany w tym widoku. Wybranie obiektu w tym widoku powoduje również zaznaczenie go w scenie w inspektorze. Pozwala to i ułatwia przesuwanie, skalowanie, obracanie oraz usuwanie obiektu czy edytowanie jego parametrów. Wszystko to bez dotykania ani jednej linijki kodu.

Aby wstawić nowy obiekt do sceny, kliknij „UTWÓRZ”, a wszystkie obiekty, które chcesz dodać, zostaną wyświetlone.



4.2 WIDOK SCENY

Widok sceny jest częścią środowiska graficznego 3D do tworzenia każdej sceny. Aby pracować w widoku sceny, najłatwiej jest przeciągnąć obiekt z widoku projektu do



widoku sceny, co spowoduje umieszczenie obiektu w scenie, a tym samym umieszczenie go tam, gdzie chcesz.

Widok sceny to także miejsce, w którym edytujesz tereny (rzeźbiąc je, malując tekstury i umieszczając elementy), umieszczasz światła i kamery oraz inne obiekty. Poniżej wyjaśniono opcje widoku sceny:

Po lewej stronie widoku sceny znajdziesz zestaw przycisków służących do zmiany ogólnych ustawień wyświetlacza. Przyjrzyjmy się każdemu z nich, od lewej do prawej:



Tryb renderowania

Wartość domyślna to „Teksturowana”. Po kliknięciu pojawi się lista rozwijana z wieloma różnymi opcjami renderowania:

- Teksturowane: Tekstury są renderowane w widoku.
- Wireframe: Powierzchnie nie są renderowane, widzimy tylko siatkę.
- Textured Wire: Tekstury są renderowane, ale widzimy też siatkę.
- Ścieżki renderowania: zmieniaj reprezentacje i charakterystyki obiektów zgodnie z tworzonymi projektami.
- Rozdzielczość mapy świetlnej: aby nadać bardziej optymalną rozdzielczość obiektom tworzącym scenę.



Tryb koloru, wyświetlany jako „RGB”.

Jeśli klikniesz na ten przycisk, pojawi się rozwijana lista z dostępnymi trybami kolorów i będziesz mógł zobaczyć różne sposoby wyświetlania sceny:



A.M.E.F.E.
asociación matemática de
educación y formación europea

Paidea

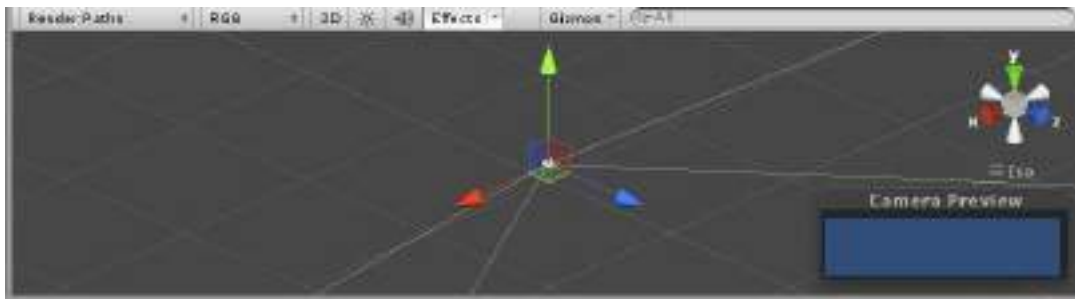


Erasmus+
Enriching lives, opening minds.

- RGB: Renderowane są wszystkie kolory.
- Alpha: Tryb zostaje zmieniony na „Alpha”.
- Overdraw: Tryb zostaje zmieniony na „Overdraw”.
- Mipmapy: Tryb zostaje zmieniony na „Mipmapy”.

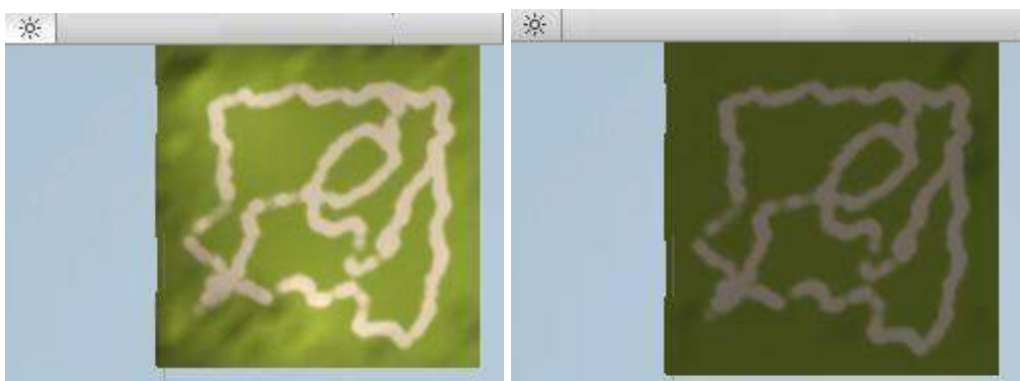
Opcja 3D-2D

Gdzie zmieniamy wyświetlanie sceny z płaszczyzny X, Y, Z na płaszczyznę oglądania X, Y.



Przełącznik światła

Następny przycisk włącza lub wyłącza oświetlenie sceniczne.





A.M.E.F.E.
asociación matemática de
educación y formación europea



Erasmus+
Enriching lives, opening minds.

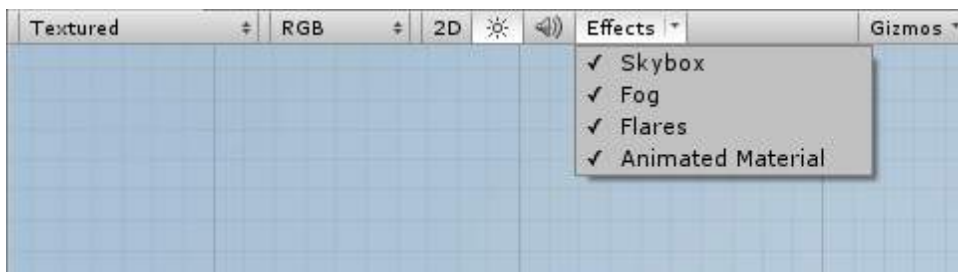
Audio

Następny przycisk włącza lub wyłącza dźwięk sceny.



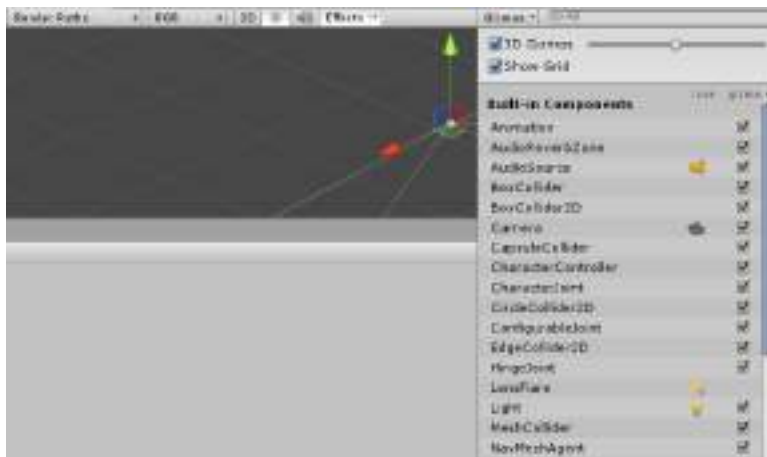
Efekty

Przycisk efektów służy do wyłączania efektów wideo w celu poprawy wydajności podczas projektowania projektu.



Gadżety

Pomagają udoskonalić grafikę sceny nawet przy użyciu języków programowania.



4.3 WIDOK GRY

Widok gry umożliwia podgląd gry w widoku takim, jaki jest lub poprzez zmaksymalizowanie jej do pełnego ekranu. W tym widoku możesz zobaczyć grę w ruchu, aby sprawdzić, czy gra działa poprawnie.



A.M.E.F.E.
asociación matemática de
educación y formación europea



Erasmus+
Enriching lives, opening minds.

Widok gry jest bardzo przydatny, ale ma tę wadę, że podczas testowania projektu na komputerze porusza się za pomocą klawiszy lub myszki, co nie pomaga, jeśli projekt jest rozwijany na inną platformę, np. urządzenie mobilne. W takim przypadku testy należy wykonać za pomocą emulatora lub skompilować wersję mobilną.



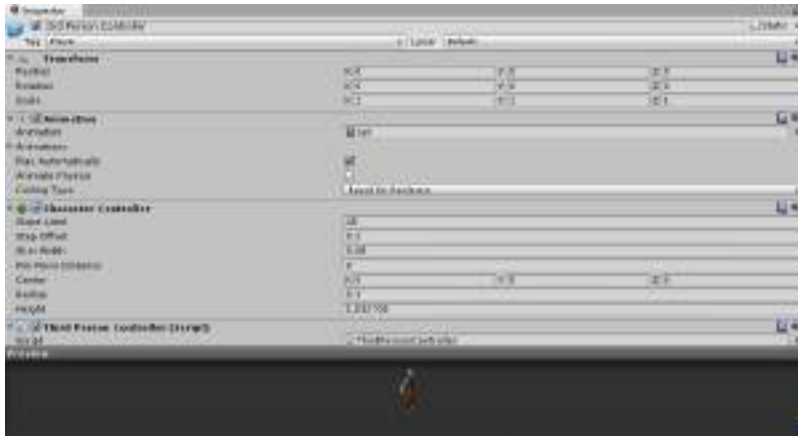
- Przycisk „Maksymalizuj podczas gry”: pozwala zmaksymalizować widok gry na pełnym ekranie.
- Przycisk „Statystyki”:
- Przycisk „Gizmo”:
- Menu rozwijane Dowolny aspekt:

4.4 WIDOK INSPEKTORA

W widoku inspektora wyświetlane są parametry obiektów wybranych w widoku sceny lub w widoku hierarchii. W ten sposób po wybraniu obiektu w scenie możesz zmodyfikować jego parametry w inspektorze (pozycja, obrót, przesunięcie itp.). Inspektor służy również jako panel narzędzi dla niektórych elementów, takich jak tereny, umożliwiając ich rzeźbienie, dodawanie tekstur i nie tylko.

W widoku inspektora można zobaczyć elementy, które definiują zachowanie obiektów, elementy te nazywane są komponentami. Ponadto można tutaj aktywować lub dezaktywować obiekty i powiązane z nimi komponenty. Komponentami mogą być skrypty, animacje, zderzacze...itp.

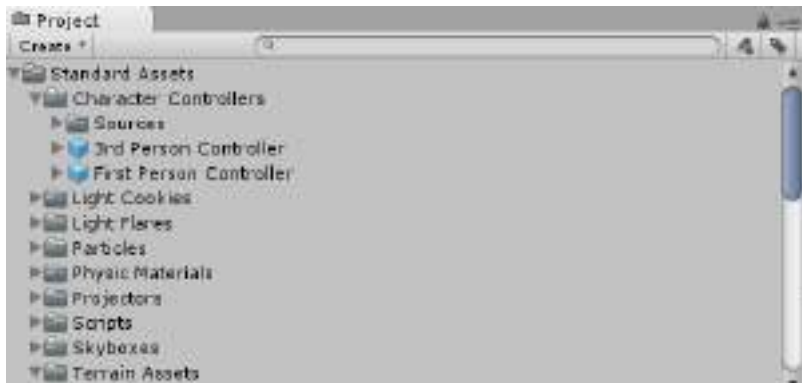
Na koniec inspektor zmienia się w zależności od wybranego elementu.



4.5 WIDOK PROJEKTU

Widok projektu to biblioteka zasobów do stworzenia gry. W tym widoku przechowywane są wszystkie elementy, które są tworzone i importowane do wykorzystania w grze. Elementami tymi mogą być obiekty 3D, tekstury, dźwięki lub skrypty.

W tym oknie można mieć wiele zasobów, ale nie oznacza to wykorzystania wszystkich tych zasobów do stworzenia gry wideo.





Ponieważ ten widok zawiera zarówno elementy używane w scenie, jak i elementy nieużywane w scenie, może zawierać dużą ilość danych. Oznacza to, że należy zachować dobrą strukturę, aby ułatwić pracę użytkownikowi. Organizację tego widoku można wykonać, tworząc hierarchię folderów. Menu rozwijane „utwórz” zawiera opcje umożliwiające dobrą organizację widoku.

5. STWÓRZ NOWY PROJEKT

Tworzenie nowego projektu polega na przejściu do górnego paska Unity i kliknięciu Plik -> Nowy projekt, aby otworzyć okno dialogowe „Utwórz nowy projekt”.

Najpierw wybierz folder, w którym chcesz zapisać projekt, klikając „Przeglądaj...”. Aby utworzyć nowy projekt, należy zaimportować niezbędne pakiety dla projektu.

Pakiety to zestawy zasobów, które można importować podczas tworzenia projektu lub w dowolnym momencie później.

Importowane pakiety mogą być predefiniowane w Unity, inne tworzone przez użytkownika lub pobierane z sieci, niektóre z nich można pobrać bezpłatnie, a inne są płatne.

Z listy, która pojawi się na stronie, wybierz zasoby i kliknij „pobierz”.

Z zasobów znalezionych w programie wybieramy te, które chcemy i klikamy „Utwórz”. Podstawowy pakiet do zaimportowania to standardowy zestaw aktywów.



Kiedy tworzony jest nowy projekt, Unity 3D jest ponownie uruchamiany przez utworzenie struktury projektu w określonym folderze. Otworzy się ekran startowy Unity 3D, pokazujący zaimportowane zasoby w widoku projektu oraz kamerę w widoku sceny i hierarchii.

Niektóre z najważniejszych pakietów to:

- Kontroler postaci: Umożliwia tworzenie skryptów do poruszania postaciami.
- Cząsteczki: Umożliwia tworzenie cząsteczek, takich jak ogień, dym, wodospady i eksplozje.
- Skyboxes: Umożliwia tworzenie nieba.
- Teren: Umożliwia tworzenie i rzeźbienie terenu.
- Woda: Stwórz wodę (morze, rzeki itp.).
- Drzewo: Kreator do tworzenia drzew.

5.1 STWÓRZ NOWĄ SCENĘ

Unity 3D działa scenami i kiedy stworzymy nowy projekt automatycznie tworzy nową scenę, jeśli chcemy stworzyć kolejną scenę, przechodzimy do Plik _ Nowa scena i zapisujemy to automatycznie w Plik _ Zapisz scenę. Scena jest zapisywana w folderze zasobów projektu i pojawi się w widoku projektu (foldery można znaleźć w Documents/New Unity Project lub jakkolwiek nazwiemy scenę).

Po zapisaniu sceny możemy stworzyć wszystkie elementy, z których będzie się ona składać, i mamy różne sposoby manipulowania sceną.

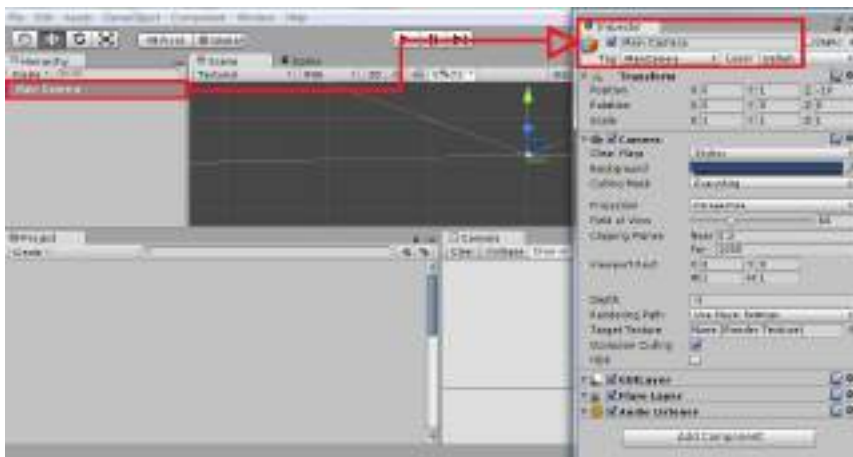
Kilka sztuczek do manipulacji sceną:

- Jeśli chcemy zobaczyć scenę z różnych kątów, możemy ją obrócić za pomocą ALT i poruszając myszką.
- Jeśli chcesz powiększyć lub pomniejszyć widok, możesz użyć kółka myszy.
- Strzałki na klawiaturze umożliwiają przesuwanie sceny w lewo, w prawo, w górę i w dół. To samo można zrobić, przytrzymując mysz i przesuując ją w jednym lub drugim kierunku.
- Aby zwiększyć prędkość poruszania się sceny, przytrzymaj SHIFT.



Podczas tworzenia sceny w naszym świecie domyślnie pojawia się kamera. Jeśli klikniemy na „Kamera główna”, charakterystyka kamery może zostać wyświetlona w inspektorze i może być modyfikowana (pozycje, obrót, skala).

Kamera określa punkt widzenia gracza, a kilka kamer można połączyć w tej samej scenie.

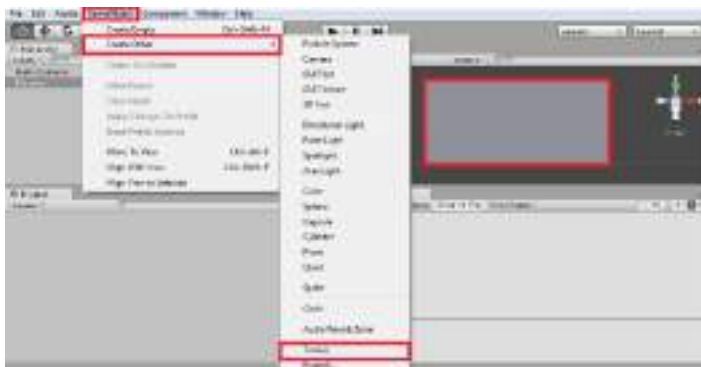


5.2 TWORZENIE PODSTAWY

Unity 3D ma narzędzie, za pomocą którego możesz zdefiniować dowolny teren, zwykle obsługuje tereny jako płaską siatkę, ale możesz teksturować i rzeźbić bez opuszczania edytora. Z menu możesz stworzyć teren klikając na:

- „Teren -> Utwórz teren”.

Otrzymujesz płaski teren, jeśli teren nie jest widoczny, musisz wyłączyć światła w widoku sceny.





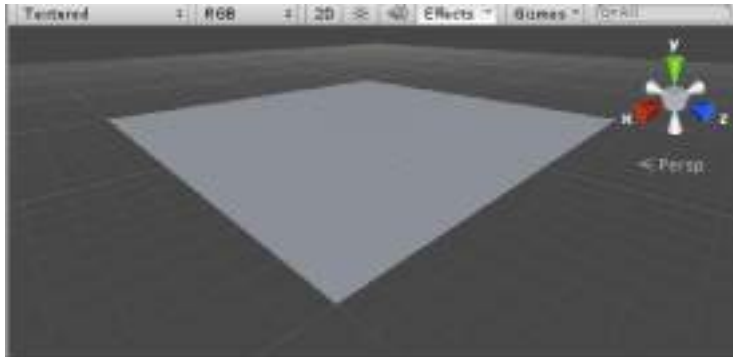
A.M.E.F.E.
asociación magisterfa de
educación y formación europea

Tpaidea



Erasmus+
Enriching lives, opening minds.

Po utworzeniu terenu uzyskuje się następujący wynik:



Jeśli chcemy dostosować teren i uczynić go bardziej realistycznym, musimy skorzystać z narzędzia terenu. Teren jest wybrany w widoku sceny lub w widoku hierarchii, dzięki czemu narzędzie do edycji terenu pojawia się w widoku inspektora.



To narzędzie jest podzielone na 3 części:

- Transformacja: Pozwala przesuwać, obracać i skalować teren na osiach x,y,z.
- Skrypt terenu: zawiera kilka narzędzi i właściwości terenu, które zostaną wyjaśnione później.
- Terrain Collider: Zawiera właściwości kolizji dla terenu.
- Zamierzamy przejść do szczegółów panelu „Terrain Script” i zobaczymy rząd przycisków. Te przyciski edytora terenu umożliwiają wykonywanie różnych zadań. Opis tego, co umożliwiają przyciski od lewej do prawej:
- Podnieś / opuść: umożliwia podnoszenie i opuszczanie geometrii terenu za pomocą pędzla.



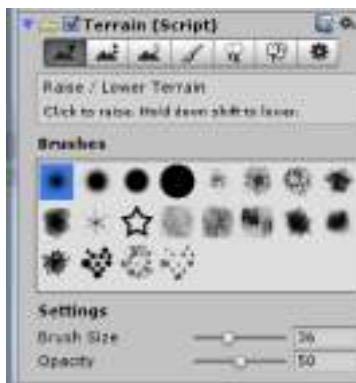
A.M.E.F.E.
asociación municipal de
educación y formación europea

Paidea

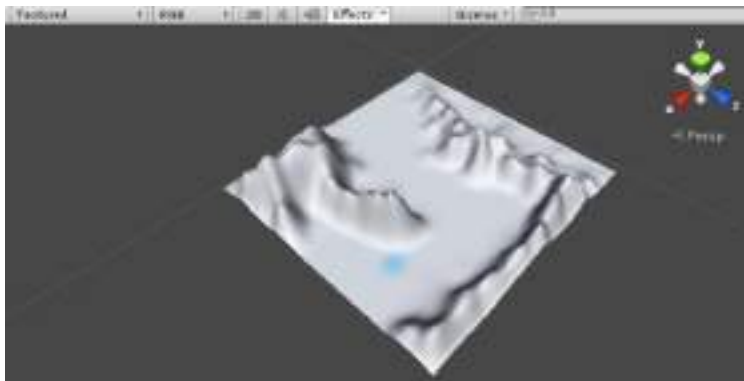


Erasmus+
Enriching lives, opening minds.

- Ustaw wysokość: malujemy teren z ograniczeniem wysokości.
- Gładki: umożliwia wygładzenie terenu w celu wyeliminowania zakrętów.
- Paint Texture: pozwala malować tekstury na powierzchni terenu.
- Umieść drzewa: umożliwia umieszczanie drzew.
- Maluj detale: umożliwia rysowanie detali terenu, takich jak trawa.
- Ustawienia terenu: Dostęp do właściwości terenu, w których możemy zmienić kilka właściwości.



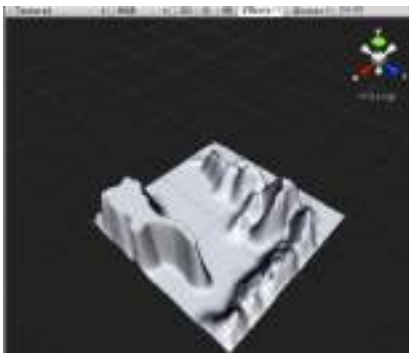
Najpierw określasz wysokość terenu, klikając miejsca, w których chcesz umieścić górę, im dłużej przytrzymasz myszkę, tym góra będzie większa. Teren można przekształcić w następujący obraz:



Druga opcja „Ustaw wysokość” umożliwia ustawienie maksymalnej wysokości w polu „Wysokość”.

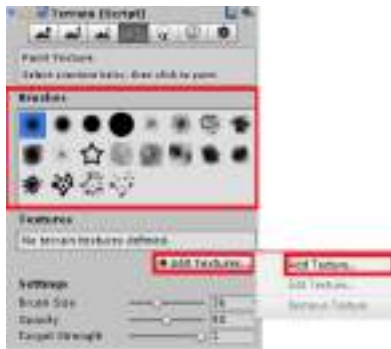


Ustawiając maksymalną wysokość, teren można dalej kształtować, aby nadać mu odpowiednią geometrię. Po osiągnięciu maksymalnej wysokości teren zostaje spłaszczony. Trzecia opcja „Wygladzanie” pozwala na wygładzenie szczytów terenu. Wskazane jest rzeźbienie etapami, najpierw duże części, a następnie dopracowanie mniejszych detali. Na koniec otrzymasz teren jak na obrazku:

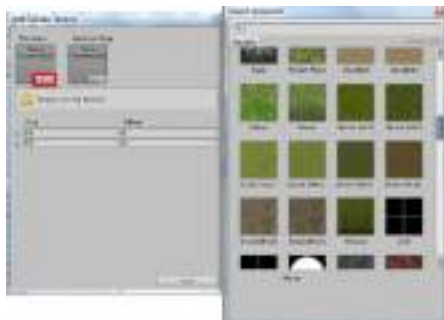


Po uzyskaniu terenu o pożądanym wyglądzie można zastosować kilka tekstur. Pierwsza tekstura jest uważana za teksturę bazową i jest nakładana na cały teren. Pozostałe tekstury zostaną nałożone w wyższych warstwach.

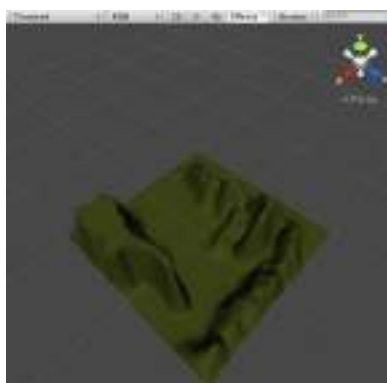
Czwarta opcja „Pomaluj teksturę terenu” pozwala na dodanie tekstury do terenu. Najpierw wybierz typ pędzla, którego chcesz użyć w „Pędzlach”, a następnie dodaj wybraną teksturę w „Teksturach”, klikając „Edytuj teksturę > Dodaj teksturę”.



Pojawi się okno „Dodaj teksturę terenu”, kliknij podświetlone na czerwono kółko, aby wyświetlić okno „Wybierz teksturę 2D”, możesz wybrać teksturę, która pojawi się w polu „Ilustracja” oraz w polu „Tekstura” w inspektorze . Możesz także wybrać szerokość i długość tekstury.



Po nałożeniu tekstur pojawią się one na terenie, który będzie wyglądał tak, jak pokazano na poniższym obrazku:



Unity 3D oferuje specjalne wsparcie przy wprowadzaniu drzew. Możesz dodać dowolną liczbę drzew, korzystając z piątej opcji „Drzewo miejsc”. Najpierw korzystasz



z opcji „Edytuj drzewa”, aby wybrać drzewo, które chcesz dodać do swojego świata, a następnie dostosowujesz drzewo, definiując kilka parametrów, rozmiar pędzla, którym ma zostać umieszczone drzewo, gęstość drzew (określa liczbę drzew do posadzenia), zmienność koloru (różnica koloru między jednym drzewem a drugim), wysokość i szerokość drzewa.

Na koniec kliknij dowolne miejsce na terenie, aby dodać drzewo w wybranym miejscu. Aby usunąć drzewa, przytrzymaj klawisz Shift i kliknij teren.



Możesz także dodać kilka drzew do swojego świata zbiorczo, aby stworzyć las za jednym razem. Kliknij „Terrain->MassPlace Trees”, a pojawi się okno dialogowe, w którym możesz określić liczbę drzew, które chcesz wprowadzić do gry.

Drzewo można wybrać, przechodząc do folderu „Standardowe zasoby > Kreator drzew > Tekstury”, klikając i przeciągając je na teren.



5.3 OŚWIETLENIE

Jeśli chcesz oświetlić scenę z taką samą intensywnością, po prostu dodaj światło punktowe i umieść żądane światła na scenie: create>point light



Jeśli pożądane jest oświetlenie wysokiej jakości, Unity integruje technologię Beast, czyli oprogramowanie firmy Autodesk, które generuje wysokiej jakości odwzorowania oświetlenia.

„Utwórz mapę światła” służy do różnych kreacji świetlnych, które umożliwiają zróżnicowane oświetlenie dla różnych tekstur terenu.

Wejść do Windows> Lightmapping i zobacz:





A.M.E.F.E.
asociación matemática de
educación y formación europea



Erasmus+
Enriching lives, opening minds.

Aby kontrolować rozdzielczość lightmap, wartość rozdzielczości jest ustawiana w panelu Bake i stosowana automatycznie.



Lightmapping poprawia definicję terenu jako obszaru drzew i ich cieni.

5.4 ŚRODOWISKO LĄDOWE

Środowisko terenowe pomoże użytkownikowi dostosować środowisko, w którym znajduje się budowana przez niego scena, na przykład dodając niebo lub wodę.

5.4.1. PODLEWANIE WODY DO SCENY

Unity zawiera kilka sposobów dodawania wody do sceny, z których jednym jest to, że użytkownik może użyć bezpośrednio. Dwa z nich bezpośrednio to „Bake de lightmapping” i „Nighttime Simple Water”.



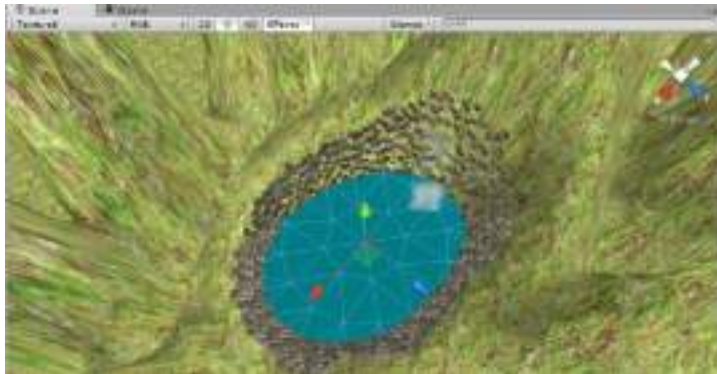


A.M.E.F.E.
asociación municipal de
educación y temas de Europa



Erasmus+
Enriching lives, opening minds.

Jako przykład dodamy „Daylight Simple Water” wewnątrz terenu. Przejdź do „Standardowy zasób > Woda”, wybierz „Daylight Simple Water” i przeciągnij go na scenę.



Charakterystykę wody można zobaczyć w inspektorze:



5.4.2 DODAJ NIEBO

Niebo można dodać na dwa sposoby, podłączając skybox do aparatu lub bezpośrednio do sceny. Przejdź do „Edycja > Ustawienia renderowania”, aby aktywować pola w inspektorze w następujący sposób:



To menu umożliwia skonfigurowanie nieba sceny, efektów mgły i niektórych globalnych efektów świetlnych. Możesz zaimportować własne tekstury lub użyć domyślnych z zasobów unity.

Istnieją 2 opcje zastosowania nieba do sceny:

Opcja 1: przejdź do menu Ustawienia renderowania, kliknij małą ikonę dysku po prawej stronie pola Skybox Material, otworzy się nowe okno ze wszystkimi zasobami materiałowymi załadowanymi do projektu.

Opcja 2: w widoku projektu pod „Standardowy zasób > Skyboxy” wybierz niebo odpowiednie dla swojej sceny i przeciągnij je do „Materiału Sybox”. Do tego celu wybrano niebo „Sunny 1 Skybox”.



Unity 3D oferuje kilka opcji dostosowywania środowiska sceny, oto niektóre z nich:

- Mgła: umożliwia dodanie mgły do sceny, scena staje się mglista.
- Kolor mgły: kolor mgły, domyślny kolor to szary.



- Tryb mgły: liniowy, wykładniczy lub wykładniczy do kwadratu. To kontroluje sposób zanikania mgły.
- Gęstość mgły: gęstość mgły, używana tylko dla wykładniczego i wykładniczego kwadratu. a. Większa gęstość zmniejszy widoczność, a niższa gęstość zwiększy widoczność.
- Liniowa mgła Początek/Koniec: początek i koniec odległości mgły, używany tylko wtedy, gdy opcja liniowej mgły jest aktywna.
- Światło otoczenia: domyślnie jest to kolor szary. Jeśli chcemy nadać scenie inny schemat oświetlenia, dobrym początkiem jest zmiana oświetlenia otoczenia.
- Skybox Material: tekstura nieba sceny.
- Siła halo: Rozmiar wszystkich jasnych halo w stosunku do ich zasięgu.
- Siła rozbłysków: Intensywność wszystkich rozbłysków w scenie.
- Tekstura aureoli: odniesienie do tekstury, która będzie wyglądać jak poświata wszystkich aureoli w światłach.
- Spot Cookie: Odniesienie do Texture2D, które pojawi się jako maska cookie dla wszystkich reflektorów.

5.4.3 RUCH

Unity zawiera kontroler pierwszoosobowy, który ułatwia tworzenie gier z widokiem pierwszoosobowym. Uzyskując dostęp do „Zasoby standardowe > Kontrolery postaci”, wybierz „Kontroler pierwszej osoby” i przeciągnij go do miejsca, w którym chcesz go umieścić w scenie.

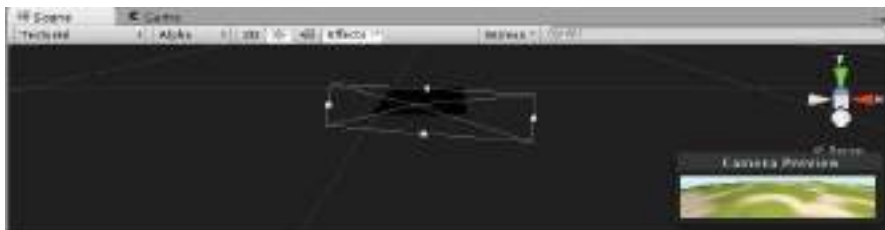




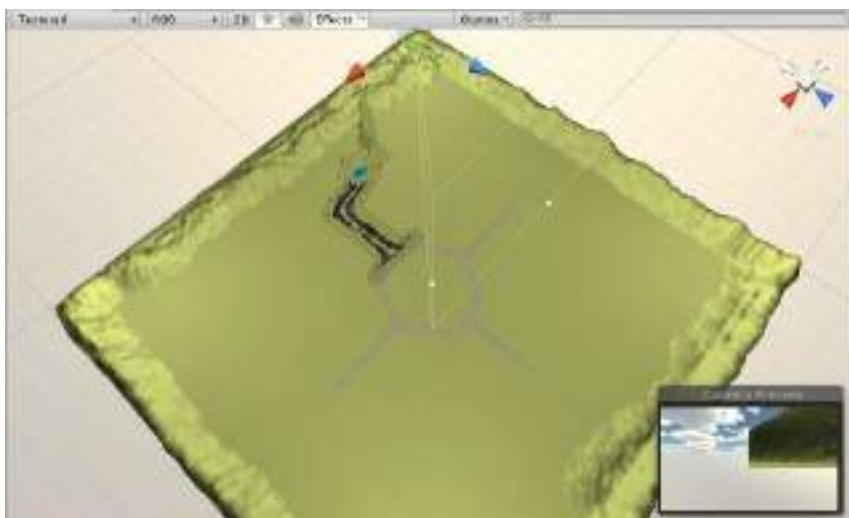
W poprzednich wersjach Unity można było zaprogramować ruch tylko za pomocą małego skryptu pokazanego poniżej:

```
var velocidad=8;
public var ptosalida:GameObject;
function update () {
    var direccion=transform.TransformDirection( Vector3( 0, 0, 0.1 ));
    if( Input.GetKey("up")){
        //transform.Translate( 0, 0, 0.1 );
        rigidbody.AddForce( direccion*velocidad );
    }
    if( Input.GetKey("left"))
        transform.Rotate( 0, -1, 0 );
    if( Input.GetKey("down")){
        //transform.Translate( 0, 0, -0.1 );
        rigidbody.AddForce( direccion*velocidad*-1 );
    }
    if( Input.GetKey("right"))
        transform.Rotate( 0, 1, 0 );
}
```

Możesz zobaczyć kamerę, która została dodana do twojej sceny, z czterech punktów kamery możesz zwiększyć lub zmniejszyć ostrość, jaką będzie miała kamera.



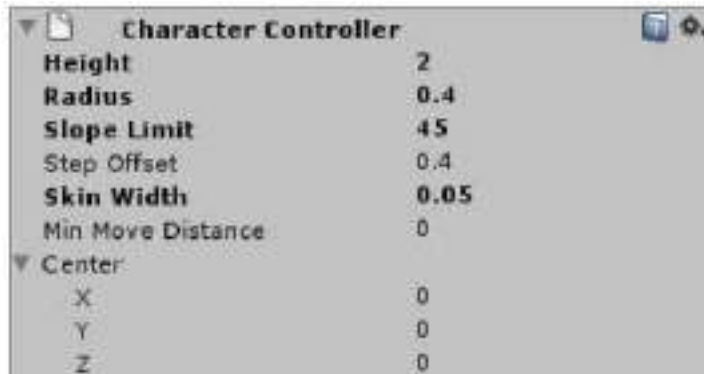
Kiedy już mamy gotowy aparat, ruch można wykonać za pomocą kilku opcji: Ruch: za pomocą strzałek lub klawiszy (W: do przodu, A: w lewo, D: w prawo i S: do tyłu). Kamerą steruje się myszką.





Skok z klawiszem spacji.

„Kontroler pierwszej osoby” można dostosować w taki sam sposób, jak inne elementy inspektora. W tym celu jest on wybierany w widoku hierarchii, a jego charakterystyka pojawia się w inspektorze.



Wyjaśnienie kontrolerów:

- Wysokość: wysokość kontrolera
- Promień: kontroler radiowy
- Limit nachylenia: maksymalna wartość nachylenia, na jakie może wspiąć się kontroler.
- Przesunięcie kroku: ta wartość wynosi od 0,1 do 0,4 dla człowieka o wzroście 2 metrów.
- Szerokość skóry: pozwala kontrolować zderzacze postaci, tak aby postać nie była blokowana.
- Minimalna odległość ruchu: Jeśli postać spróbuje przesunąć się poniżej wskazanej wartości, w ogóle się nie poruszy. Można to wykorzystać do zmniejszenia wibracji. W większości sytuacji wartość tę należy pozostawić na poziomie 0.
- Środek: położenie kontrolera, jego współrzędne x, y, z.

5.4.4 DŹWIĘK NA MIEJSCU



A.M.E.F.E.
asociación municipal de
educación y temas de Europa

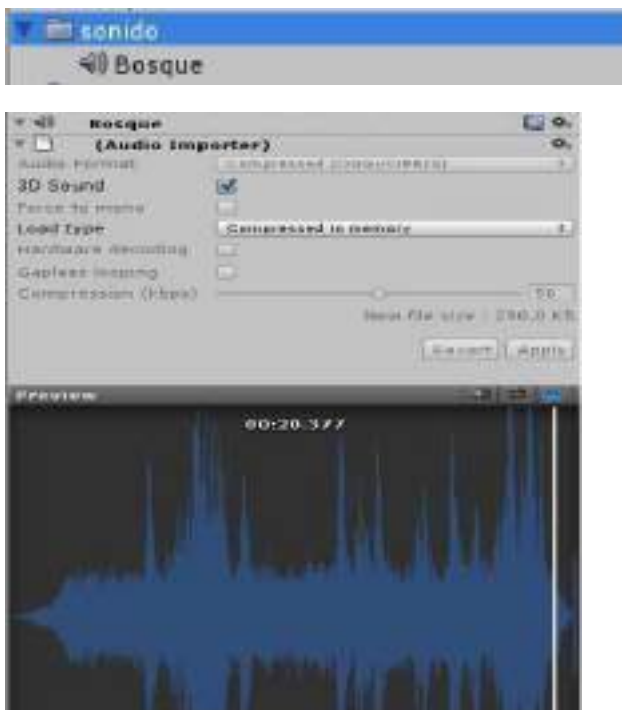


Erasmus+
Enriching lives, opening minds.

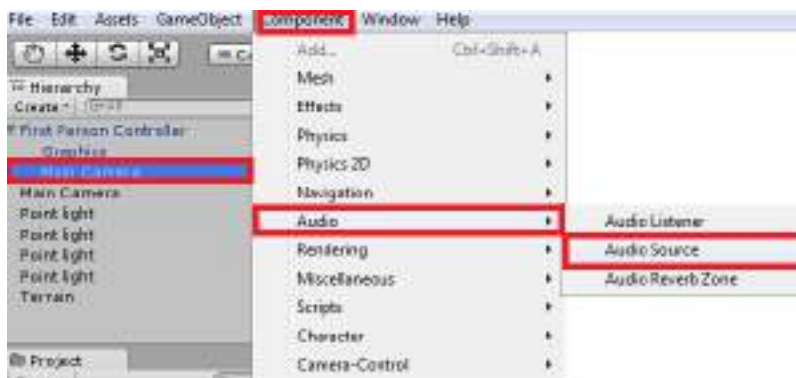
W możemy wstawić dźwięk do gry. Istnieją dwa różne rodzaje dźwięku, dźwięk 3D i dźwięk 2D.

Dźwięk 3D jest słyszalny w jednym miejscu i słabnie w miarę oddalania się od tego miejsca. Dźwięk 2D jest słyszalny równomiernie w całej scenie.

Najpierw tworzysz folder o nazwie „dźwięk”, w którym zapisujesz zaimportowany dźwięk, na przykład dźwięk reprezentujący dźwięki otoczenia dżungli lub lasu.



Wybierz „Main Camera” i przejdź do Component> Audio> Audio Source.





Zaimportowany dźwięk jest następnie przeciągany do pola „Klip audio” w widoku inspektora, a dźwięk jest konfigurowany:

Pole wyboru „Play On Awake” jest zaznaczone, dzięki czemu dźwięk rozpoczyna się na początku projektu.

Pole wyboru „Pętla” jest aktywne, dzięki czemu muzyka jest zapętлана i powtarzana tyle razy, ile to konieczne.

Na wykresie „Listner” w komponencie wybierz zielony punkt w najwyższym punkcie i przeciągnij go do środka współrzędnych (0,0).

Minimalna odległość 0,1 jest zdefiniowana w „Min Distance”.



5.4.5 WŁĄCZANIE I WYŁĄCZANIE ŚWIATŁA SCENICZNEGO

W tej wersji Unity włączanie i wyłączenie światła na scenie odbywa się poprzez kliknięcie przycisku światła znajdującego się w opcjach widoku sceny, ale jeśli chcesz w pewnym momencie wyłączyć światło na scenie, którą tworzysz, możesz napisać prosty skrypt: pokazane poniżej:

```

var myLight : Light;
var myLight1 : Light;

-function Start(){
  myLight = Light.FindObjectOfType(typeof(Light));
}

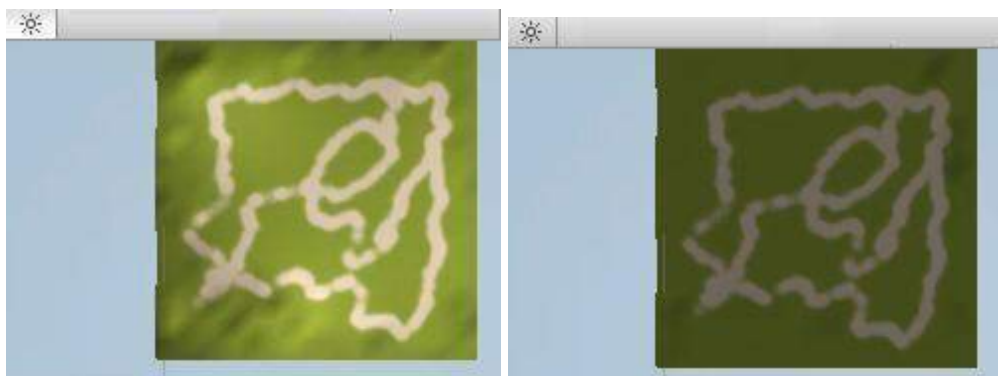
-function Update () {
-  if(Input.GetButtonDown("Fire1")){
    myLight.enabled = !myLight.enabled;
    myLight1.enabled = !myLight.enabled;
  }
}

```

Wyjaśnienie skryptu:

- `myLight = Light.FindObjectOfType(typeof(light));` _ przeszukuje scenę w poszukiwaniu obiektu typu „light” (światło w scenie).
- `if(Input.GetButtonDown("Fire1"))` po naciśnięciu lewego przycisku myszy.
- `mojeŚwiatło.włączone = !mojeŚwiatło.włączone;` _ zmienia stan światła, jeśli jest włączone, to jest wyłączone i odwrotnie.
- `myLight1.enabled = !myLight.enabled` _ ta linia jest zduplikowana, ponieważ w naszej scenie są 2 światła.

Po utworzeniu skryptu jest on dodawany do obiektu i w parametrach, które pojawiają się w widoku inspektora, przeciągnij każde z 2 światel, aby wyglądało to tak:

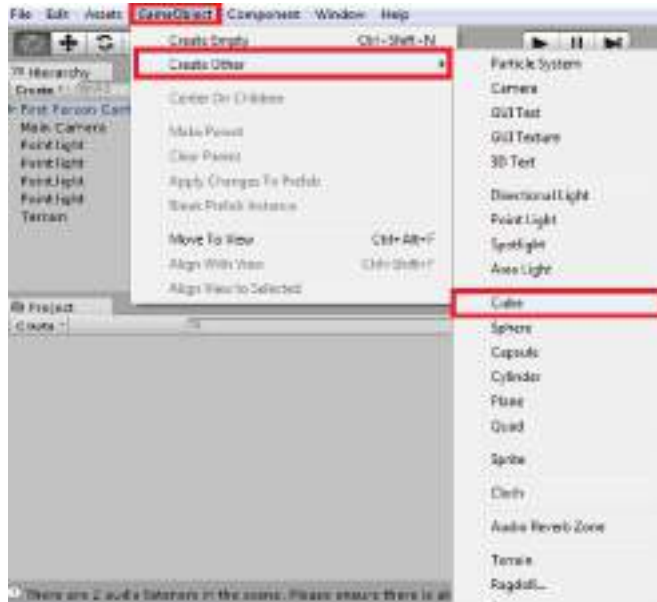


5.4.6. TWORZENIE OBIEKTU

Wyjaśnimy, jak stworzyć element, w tym przypadku sześciąt.



Aby utworzyć predefiniowane obiekty w Unity 3D, przejdź do GameObject > Utwórz inny i zostanie wyświetlona lista elementów.



Niektóre ważne elementy GameObject to:

- Kamera: umożliwia wstawienie kamery do sceny, chociaż podczas tworzenia projektu Unity domyślnie umieszcza kamerę.
- Światło kierunkowe: wprowadza jednolite światło kierunkowe, które pozwala zobaczyć scenę.
- Sześcian: umieść sześcian na scenie.
- Kula: włóż kulę
- Kapsułka: włożyć kapsułkę
- Cylinder: umieścić cylinder
- Samolot: wprowadza samolot do sceny. Po wybraniu jednego z „GameObjects” pojawia się on w widoku sceny, widoku gry i widoku hierarchii. Dodatkowo w widoku inspektora można zobaczyć atrybuty obiektu do parametryzacji obiektu.



A.M.E.F.E.
asociación matemática de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.



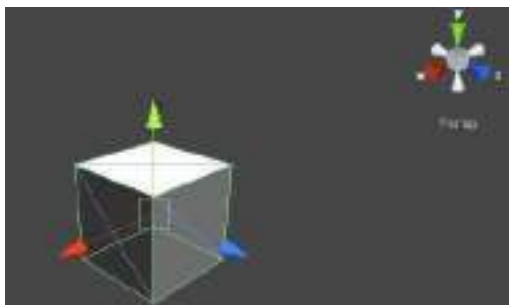
5.4.7 MANIPULACJA PRZEDMIOTEM

Po utworzeniu obiektu, w tym przypadku sześcianu, możemy go zmodyfikować, aby miał niezbędne wymiary, położenie i kształt, aby osiągnąć pożądane cechy.

Położenie kostki można zmienić za pomocą drugiego przycisku sterującego:



Jak widać na poniższym obrazku, możemy przesuwać sześcian wzdłuż osi X (czerwony), Y (zielony) i Z (niebieski). Aby przesunąć kostkę, po prostu kliknij oś, na której chcesz ją przesunąć, i przeciągnij ją w nowe miejsce.



Trzeci przycisk sterujący umożliwia obracanie kostki.



Porównując poprzedni obraz, na tym widzimy, jak sześcian się obraca.

Różne okręgi pojawiające się wokół sześcianu po wybraniu opcji obracania pozwalają użytkownikowi wybrać oś, na której chce wykonać tę operację. Aby

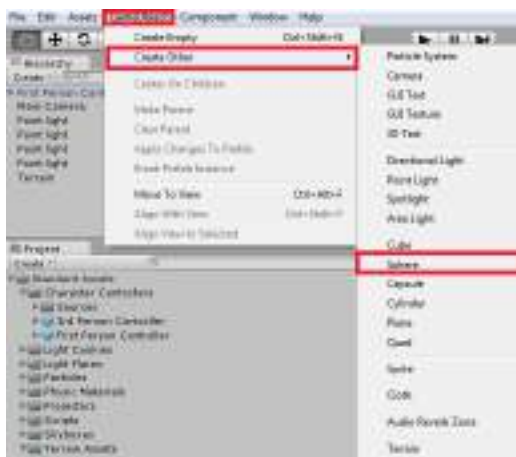


obrócić element, kliknij element oznaczony kolorowymi kółkami i przesuń mysz w wybranym kierunku.



5.4.7 TWORZENIE ROBAKA

Ponieważ wiemy już, jak stworzyć obiekt, stworzymy prostą postać, robaka opartego na kulach. Pierwsza sfera jest tworzona w „GameObject > Create Other > Sphere”:



Umieść kulę na terenie w żądanej pozycji. Aby utworzyć inne identyczne kule, naciśnij „CTRL + D”, ta nowa kula pojawi się na wierzchu starej i można ją przesuwać. Powtarzaj tę samą czynność, aż uzyskasz 4 kule. Możesz zobaczyć, że sfery pojawiają się w widoku hierarchii, w którym możesz zmienić nazwę pierwszej sfery na „głowa”, a pozostałych na „ciało”. Aby zmienić nazwę obiektu gry, wybierz go w widoku hierarchii, kliknij prawym przyciskiem myszy i „Zmień nazwę”. Zmodyfikuj kule reprezentujące ciało, zmniejszając je za pomocą inspektora, aby uzyskać większą precyzję.

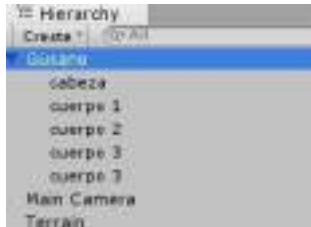


A.M.E.F.E.
asociación municipal de
educación y temas de Europa



Erasmus+
Enriching lives, opening minds.

Ponadto, aby zoptymalizować organizację naszego projektu, tworzony jest pusty obiekt w „GameObject > Create Empty”, przemianowany na robaka, a głowa i ciało są przeciągane na robaka.



Następnie tekstura jest importowana, nakładana na postać i uzyskuje się następujący wynik:



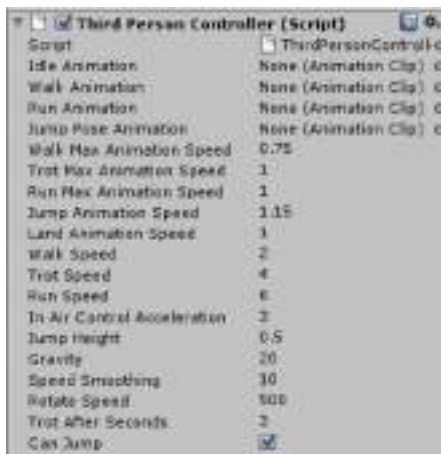
Na koniec, aby wszystkie elementy tworzące ciało robaka podążały za głową, zostanie użyty komponent „Smooth follow”, który zostanie wyjaśniony w następnej sekcji.

5.4.8 MOBILNOŚĆ OBIEKTU

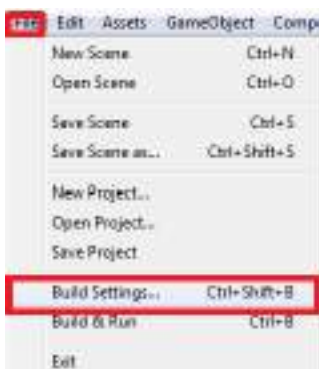
Wybierz głowę w widoku hierarchii, przejdź do Komponent> Skrypty> Kontroler trzeciej osoby. W ten sposób można nadać głowie ruchliwość i sterować nią w trzeciej osobie. Jeśli chcesz przenieść całego węża, musisz zastosować „Kontroler trzeciej osoby” do całego robaka.



I możemy zobaczyć w widoku inspektora, jak wygląda nowy komponent ze wszystkimi parametrami, które można skonfigurować.



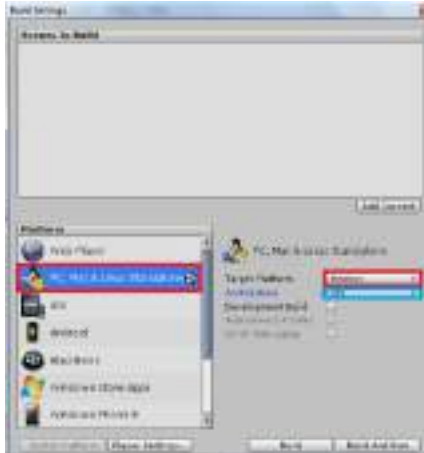
Aby sfinalizować projekt możemy wyeksportować uruchomienie projektu:



Ty wybierasz platformę i architekturę na jaką chcesz wyeksportować projekt, możesz wyeksportować na platformy WINDOWS, MAC, LINUX, ANDROID, IOS może z



darmową licencją jeśli chcesz wyeksportować na konsole do gier takie jak XBOX, WII i PLAY STATION by trzeba zapłacić za licencję.



Na koniec kliknij Build and Run, a projekt rozpocznie eksport, nadaj mu nazwę i zapisz go w wybranym folderze, kończąc na pliku wykonywalnym wybranej platformy, biorąc pod uwagę architekturę komputera, na którym realizowany jest projekt.



Na zakończenie projektu można stwierdzić, że silniki gier wideo są bardzo przydatnymi narzędziami, które ułatwiają pracę twórcom, pozwalając im skupić się na aspektach kreatywnych, a nie na aspektach strukturalnych, w celu zaspokojenia wymagań konsumentów, zamiast dostosowywać się do wymagających konsumentów.

Unity 3D jest jednym z najnowocześniejszych silników do projektowania gier wideo, jest bardzo kompletnym i bardzo wydajnym silnikiem. Ponadto każda wersja została dostosowana do nowych trendów, takich jak projektowanie wieloplatformowe. Ta cecha daje Unity3D przewagę nad niektórymi silnikami gier wideo.

Do realizacji tego projektu wykorzystaliśmy darmową wersję Unity 3D, ale ponieważ jest darmowa, nie jest mniej wydajna. Unity oferuje użytkownikom proste środowisko, które pozwala na bardzo szybkie stworzenie projektu, a ponieważ nie jest wymagana



A.M.E.F.E.
asociación matemática de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.

głęboka wiedza z zakresu programowania, każdy może z niego korzystać z kilkoma podstawowymi pojęciami.

Musimy pamiętać, że Unity nie tylko umożliwia tworzenie gier wideo czy wirtualnych wycieczek, ale jest również bardzo popularnym narzędziem wśród architektów i projektantów, ponieważ pozwala w prosty i szybki sposób tworzyć środowiska 3D.



Część 3 - Jak tworzyć kody przy użyciu języka programowania C#

WSTĘP.

W 2002 roku Microsoft ogłosił wydanie języka C#, podobnego do C++ i Javy, ale ulepszego (w muzyce symbol # oznacza „pół tonu wyżej”). Ten rozwój był ważną częścią inicjatywy Microsoft „dot net”, którą opiszemy poniżej. Ze względu na podobieństwo między C# i jego poprzednikami, nauczenie się go sprawi, że języki C, C++ i Java będą bardziej zrozumiałe, jeśli kiedykolwiek będziesz ich potrzebować.

- C# to język zorientowany obiektowo, wywodzący się z C++ i Javy.
- Microsoft .NET Framework jest ważnym produktem, który obejmuje języki C#, C++ i Visual Basic.
- Programy to listy instrukcji, które komputery wykonują automatycznie.
- Obecnie głównym nurtem w praktyce programistycznej jest Object Oriented Methodology (OOP), a C# w pełni ją wspiera.

1. TWORZENIE PROJEKTU W C#

W tym podręczniku będziemy używać akronimu „IDE” (Integrated Development Environment) w odniesieniu do zintegrowanego środowiska programistycznego C#. Podobnie jak edytory tekstu, programy ułatwiające tworzenie dokumentów, IDE posiada narzędzia do tworzenia (rozwoju) programów. Wszystkie narzędzia, których potrzebujemy do tego celu, są zintegrowane, dzięki czemu możemy wykonywać całą naszą pracę w środowisku IDE zamiast konieczności korzystania z innych programów. Innymi słowy, IDE zapewnia nam kompletne środowisko programistyczne.

1.2 INSTALACJA I KONFIGURACJA

Aby wykonać ćwiczenia, musisz pobrać Microsoft Visual C# 2008 Express Edition. Postępuj zgodnie z instrukcjami instalacji i zarejestruj produkt, jeśli zostaniesz o to poproszony.

Wyjaśnimy teraz, jak pogrupować programy utworzone w języku C# w określonym folderze.

Podczas tworzenia programu język C# generuje kilka plików i umieszcza je w nowym folderze. Wynikowy zestaw plików nazywany jest „projektem”. Dobrym pomysłem jest utworzenie folderu najwyższego poziomu do przechowywania wszystkich projektów. Domyślny folder używany przez język C# nosi nazwę Projekty i znajduje się w folderze Moje dokumenty w programie Visual Studio 2008 na dysku C. Prawie



wszyscy użytkownicy uważają to za właściwe, ale jeśli z jakiegoś powodu musisz zmienić tę domyślną lokalizację, wykonaj następujące kroki:

1. Kliknij menu Narzędzia u góry ekranu. Wybierz Opcje... Upewnij się, że wybrana jest opcja Pokaż wszystkie ustawienia.
2. Kliknij opcję Projekty i rozwiązania, a następnie przycisk ... po prawej stronie sekcji Lokalizacja projektu.
3. Następnie otworzy się okno lokalizacji projektu, w którym możesz wybrać istniejący folder lub utworzyć nowy. Kliknij przycisk OK, gdy skończysz.

Od teraz C# będzie zapisywał całą Twoją pracę w domyślnym folderze projektu lub w specjalnie wybranym przez Ciebie folderze. W każdym razie przy pierwszym zapisywaniu projektu pojawi się okno dialogowe z przyciskiem Przeglądaj, na wypadek gdybyś potrzebował użyć innego folderu niż ten, który wybrałeś.

1.3 TWORZENIE PIERWSZEGO PROGRAMU

Program, który stworzymy w następnej kolejności, wyświetli komunikat *Hello World* na ekranie Twojego komputera. Niezależnie od powyższego, będziesz musiał wykonać te ogólne kroki, aby utworzyć dowolny program.



Rysunek 2.1 Strona główna



Rysunek 2.2 Okno Nowy projekt

- Otwórz IDE. Pojawi się strona główna, jak pokazano na rysunku 2.1.



- **Kliknij** łącze **Utwórz: Projekt...** Rysunek 2.2 przedstawia okno **Nowy projekt** , które pojawi się w odpowiedzi.
- Upewnij się, że wybrano szablon **aplikacji Windows Forms** . Wybierz nazwę dla swojego projektu, która stanie się również identyfikatorem folderu. Zalecamy używanie tylko liter, cyfr i spacji. W naszym przypadku użyliśmy nazwy **First Hello** . Kliknij **OK** . Pojawi się obszar układu podobny (ale niekoniecznie identyczny) do pokazanego na rysunku 2.4.
- Aby ułatwić sobie zapoznanie się z językiem C#, zestaw narzędzi powinien być zawsze widoczny. Kliknij **menu Widok** i wybierz **Przybornik** . Teraz kliknij symbol „pinezki” w przyborniku, jak pokazano na rysunku 2.3; skrzynka narzędziowa zostanie trwale zamocowana i otwarta. Kliknij **Wszystkie formularze systemu Windows** , aby wyświetlić listę dostępnych narzędzi. Twój ekran będzie teraz wyglądał prawie identycznie jak na rysunku 2.4.

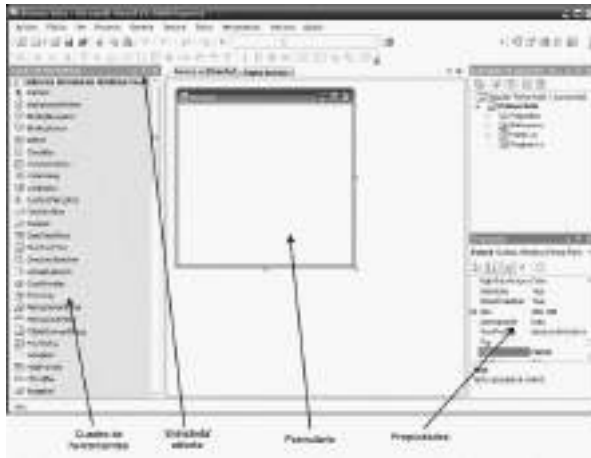
Jak wcześniej wspomniano, powyższe kroki są wspólne w realizacji każdego projektu. Wykonajmy teraz konkretne zadanie dla tego konkretnego projektu.

Zwróć uwagę na formularz i zestaw narzędzi przedstawiony na rysunku 2.4. Teraz wybierzemy formant z przybornika i umieścimy go w formularzu. Oto kroki, które należy wykonać:

- Znajdź przybornik i kliknij kontrolkę **Etykieta** .
- Przesuń wskaźnik myszy na formularz. Kliknij i nie zwalniając przycisku przeciągnij, aby utworzyć etykietę, jak na rysunku 2.5.
- Teraz ustawimy niektóre właściwości etykiety: kliknij etykietę prawym przyciskiem myszy i wybierz **Właściwości** . Przesuń mysz do listy Właściwości w prawym dolnym rogu ekranu, jak pokazano na rysunku 2.6.



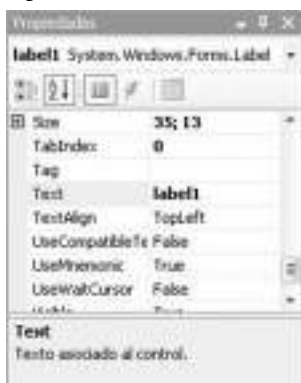
Rysunek 2.3 Zestaw narzędzi jest zamocowany i otwarty.



Rysunek 2.4 IDE w czasie projektowania.



Rysunek 2.5 Do formularza dodano etykietę.



Rysunek 2.6 Właściwości etykiety.



Rysunek 2.7 Kliknij strzałkę, aby uruchomić program.

- Przewiń w dół do właściwości **Text** i zastąp **etykietę tekstową1** wartością **Hello World** .



- Teraz uruchommy program, klikając strzałkę na górze IDE (rysunek 2.7); jest to przycisk **Rozpocznij debugowanie**.

Pojawi się nowe okno, jak pokazano na rysunku 2.8. To jest program, który stworzyłeś. Pokazuje tylko trochę tekstu, ale jest to prawdziwe okno, w którym można je przesuwając, zmieniać jego rozmiar i zamykać, klikając ikonę **X** w prawym górnym rogu. Poeksperymentuj z tym oknem, a następnie zamknij je.

Aby zapisać program do późniejszego wykorzystania:

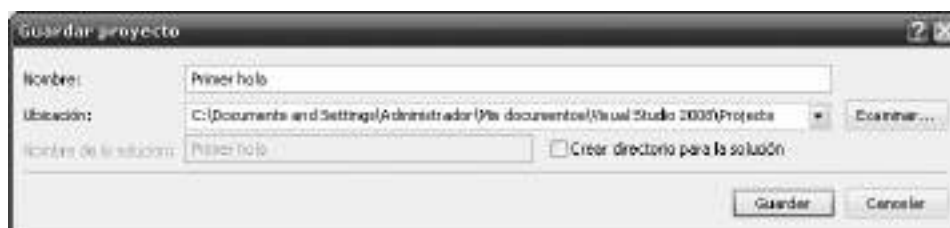
- Przejdź do menu **Plik** i wybierz opcję **Zapisz wszystko** (dalej, aby wskazać tego typu działania, użyjemy krótkiej formy: **Plik | Zapisz wszystko**).
- Następnie pojawi się okno **Zapisz projekt** , jak pokazano na rysunku 2.9. Upewnij się, że opcja **Utwórz katalog dla rozwiązania** nie jest zaznaczona. Pozostałe opcje pozostaw bez zmian i kliknij przycisk **Zapisz** . Gdy ponownie zapiszesz projekt, te same opcje zostaną automatycznie użyte, a okno **Zapisz projekt** nie będzie już wyświetlane.

Możesz teraz użyć **Plik | Wyjdź** polecenie, aby zamknąć IDE.

Następnym razem, gdy użyjesz C#, nazwa twojego projektu pojawi się na stronie startowej, więc możesz go otworzyć jednym kliknięciem, bez konieczności powtarzania pracy, którą wykonaliśmy, aby skonfigurować projekt.



Rysunek 2.8 Uruchomiony program First Hello .



Rysunek 2.9 Okno **Zapisz projekt** z wyłączoną opcją **Utwórz katalog** .



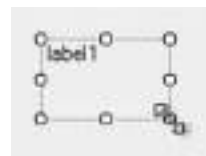
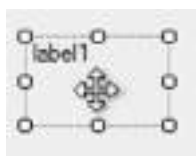
1.4 KONTROLA CZASU PROJEKTOWANIA

W naszym programie First Hello umieściliśmy etykietę na formularzu i zmodyfikowaliśmy wyświetlany tekst. Głównym celem ćwiczenia było omówienie kroków związanych z tworzeniem projektu. Następnie nauczymy się podstaw kontrolek i właściwości.

Co to jest kontrola? Jest to „urządzenie”, które pojawia się na ekranie w celu wyświetlania informacji, umożliwienia użytkownikowi interakcji z aplikacją lub obu. Samo środowisko IDE języka C# korzysta z wielu kontrolek. Na przykład w poprzednim ćwiczeniu używałeś menu rozwijanych do zapisywania projektów i przycisku OK do potwierdzania działań. W przypadku aplikacji systemu Windows zestaw narzędzi zawiera około 70 elementów sterujących; część zadania programistycznego polega na wybraniu odpowiednich kontrolek, umieszczeniu ich w formularzu i ustawieniu ich właściwości. W przeciwieństwie do „czasu działania”, ta faza jest związana z użyciem elementów sterujących i nazywana jest „czasem projektowania”. Gdy program działa, użytkownik wchodzi w interakcję z elementami sterującymi. Zadaniem programisty jest stworzenie graficznego interfejsu użytkownika (GUI) w celu ułatwienia tej interakcji. Przyjrzyjmy się teraz, jak można manipulować kontrolkami w czasie projektowania.

- Możliwe jest wybranie kontrolki z przybornika i umieszczenie jej na formularzu. Przypisana mu pozycja początkowa nie ma znaczenia, ponieważ możemy ją łatwo zmienić.
- Możliwe jest również przesunięcie sterowania. Jeśli umieścisz wskaźnik myszy nad kontrolką, obok niej pojawi się strzałka z czterema grotami, jak pokazano na rysunku 2.10. Następnie możesz przeciągnąć kontrolkę za pomocą myszy.
- Możesz zmienić rozmiar kontrolki. Gdy klikniesz kontrolkę, na jej krawędziach pojawi się kilka małych ramek (uchwytów rozmiaru). Umieść kursor myszy nad jednym z tych małych pudełek, a pojawi się dwukierunkowa strzałka, jak pokazano na rysunku 2.11. Przeciągnij krawędź lub róg, aby zmienić rozmiar prostokąta.

W rzeczywistości metoda zmiany rozmiaru zależy od samej kontrolki. Na przykład formant etykiety ma właściwość o nazwie `AutoSize` (automatyczne dopasowywanie rozmiaru zgodnie z wymiarami wprowadzonego tekstu), którą ustawiamy na



Rysunek 2.10 Kontrolki można przesuwac... **Rysunek 2.11** ... i odpowiednio zmieniać ich rozmiar.



Jeśli pozostawimy właściwość AutoSize ustawioną na True, szerokość etykiety zostanie określona na podstawie szerokości wyświetlanego tekstu, a wysokość zostanie określona na podstawie rozmiaru czcionki tekstu. Inne elementy sterujące umożliwiają zmianę szerokości poprzez przeciąganie uchwytów myszą, ale nie wysokość (która jest ustawiana zgodnie z czcionkami). Niektóre elementy sterujące — takie jak przyciski — umożliwiają zmianę ich rozmiaru w dowolnym kierunku. Ogólnie rzecz biorąc, rozmiar kontrolki zależy od ich przeznaczenia, dlatego zalecamy eksperymentowanie z nimi. Ponieważ jednak etykiety są tak powszechne, nie powinniśmy zapominać o ich właściwości AutoSize.

Następnie przyjrzymy się właściwościom. Oto analogia: telewizory mają takie właściwości, jak kolor obudowy, rozmiar ekranu, kanał, który w danej chwili pokazuje, cena i marka.

Podobnie każda kontrolka ma zestaw właściwości, które można dostosować w czasie projektowania, aby spełnić nasze wymagania. Później zobaczymy, jak zmienić właściwość w czasie wykonywania.

Po umieszczeniu kontrolki na formularzu, aby wyświetlić jej właściwości, kliknij na nią prawym przyciskiem myszy i wybierz Właściwości; Spowoduje to wyświetlenie okna właściwości wybranej kontrolki. Lewa kolumna zawiera nazwy właściwości, a prawa kolumna zawiera ich aktualną wartość. Aby zmienić właściwość, musimy zmodyfikować wartość w prawej kolumnie. W przypadku niektórych właściwości może być konieczne wybranie kilku dodatkowych opcji, takich jak zmiana ustawień kolorów i czcionek tekstu. Czasami, gdy zakres wartości do wyboru jest bardzo szeroki, konieczne jest otwarcie dodatkowego okna.

Innym istotnym aspektem kontrolki jest jej nazwa. To samo w sobie nie jest właściwością, ale dla wygody jest pokazane na liście właściwości jako (Nazwa). Nawiasy wskazują, że w rzeczywistości nie jest to właściwość.

Kiedy umieszczamy wiele etykiet na formularzu, IDE wybiera ich nazwy w następujący sposób:

etykieta 1 etykieta 2 etykieta 3 ...

Te nazwy są na razie akceptowalne, ale w kolejnych rozdziałach przekonamy się, że lepiej jest zmienić domyślne nazwy niektórych kontrolki na bardziej reprezentatywne. Aby zmienić nazwę kontrolki, musisz zmodyfikować tekst po prawej stronie (Nazwa) na liście właściwości.

1.5 ZDARZENIA I KONTROLA PRZYCIISKÓW

Program, który stworzyliśmy w poprzednim ćwiczeniu, nie jest zbyt reprezentatywny, ponieważ zawsze pokazuje te same słowa, a użytkownik nie może z nim wchodzić w



interakcje. Następnie wzbogacimy ten program tak, aby po kliknięciu przycisku przez użytkownika pojawiał się komunikat tekstowy. To jest przykład użycia zdarzenia.

Prawie wszystkie zdarzenia są wyzwalane przez użytkownika i generowane, gdy użytkownik w jakiś sposób manipuluje kontrolką w czasie wykonywania. Każda kontrolka ma kilka zdarzeń, na które może reagować, takich jak kliknięcie przyciskiem myszy, dwukrotne kliknięcie lub umieszczenie wskaźnika myszy nad kontrolką. Inne rodzaje zdarzeń nie pochodzą od użytkownika; na przykład powiadomienie o zakończeniu pobierania strony internetowej.

W kolejnym programie, który stworzymy, wykryjemy zdarzenie (kliknięcie przycisku); następnie sprawimy, że wiadomość tekstowa pojawi się na etykiecie. Oto jak stworzymy interfejs użytkownika:

- Utwórz nowy projekt o nazwie przycisk Witaj.
- Umieść etykietę i przycisk na formularzu. Pozycja, w której to robisz, nie jest ważna.
- Wpisz Kliknij tutaj we właściwości Tekst przycisku.
- Zmodyfikuj właściwość Tekst etykiety, aby wyświetlała się bez zawartości.

Uruchom program, chociaż nie jest jeszcze ukończony. Pamiętaj, że możesz kliknąć przycisk i że nieznacznie zmienia on swój wygląd, aby potwierdzić, że jest naciskany; nic więcej się nie dzieje. Zamknij formularz.

Zobaczmy teraz, jak wykryć zdarzenie kliknięcia. Kliknij dwukrotnie przycisk w formularzu układu. Spowoduje to otwarcie nowego panelu informacyjnego, jak pokazano na rysunku 2.12. Zwróć uwagę na zakładki u góry:

Formularz 1. csStrona główna Form1.cs [Projekt] [Projekt]

Możesz kliknąć te zakładki, aby przełączyć panele. W okienku Form1.cs jest wyświetlany program w języku C#. Jest to znane jako „tekst programu” lub „kod” języka C#. Zmodyfikujmy ten kod za pomocą edytora IDE.



Rysunek 2.12 Kod C# w okienku edycji.



Przewiń kod, aż znajdziesz następną sekcję:

```
private void button1_Click(nadawca obiektu, EventArgs e)
{
}
```

Ta część kodu jest znana jako *metoda* . Nazwa metody to **button1_Click** . Kiedy użytkownik kliknie **przycisk button1** w czasie wykonywania, wykona lub „wykona” dowolną instrukcję, którą umieścimy między dwoma powyższymi wierszami.

W tym konkretnym programie użyjemy instrukcji umieszczenia wiadomości **Hello World** w tekście etykiety **label1 label** . Instrukcja wykonania tej czynności to:

```
label1.Text = "Witaj świecie";
```

Napisz instrukcję dokładnie tak, jak pokazano tutaj, między wierszami { i } . W następnych rozdziałach wyjaśnimy dokładne znaczenie linii takich jak ta powyżej (które implikują „instrukcję przypisania”).

Kolejnym krokiem jest uruchomienie programu. Są na to dwie możliwości:

- Jeśli program działa poprawnie, po kliknięciu przycisku na etykiecie pojawi się komunikat **Hello World** .



Rysunek 2.13 Niewłaściwe działanie. Zaznacz pole wyboru i kliknij **Nie** .

- Inną możliwością jest to, że program nie uruchomi się z powodu błędu. W takim przypadku C# wyświetli komunikat podobny do tego na rysunku 2.13. Zaznacz pole wyboru opcji **Nie pokazuj więcej tego okna dialogowego** i kliknij **Nie** , aby kontynuować. Komunikat nie pojawi się ponownie; odtąd jedynym dowodem błędu będzie podkreślenie w tekście kodu.

W każdym razie błąd musi zostać poprawiony: przejrzyj kod, popraw ewentualne literówki i ponownie uruchom program. Błędy omówimy bardziej szczegółowo później.

Oto nowe funkcje tego programu:



- Może reagować na kliknięcie przycisku. Proces umieszczania kodu w taki sposób, aby po wystąpieniu zdarzenia wykonywana była odpowiednia akcja, nazywany jest „obsługą” zdarzenia.
- Zmodyfikuj wartość właściwości formantu w czasie wykonywania. Wcześniej robiliśmy to tylko w czasie projektowania. Jest to bardzo ważna część programowania, ponieważ często musimy pokazać wyniki, umieszczając je na określonej właściwości kontrolki.

1.6 OTWARCIE ISTNIEJĄCEGO PROJEKTU

Aby ponownie otworzyć wcześniej utworzony projekt, zapisz i zamknij projekt w toku i przejdź do strony startowej. Pokazuje projekty, nad którymi ostatnio pracowałeś; aby otworzyć projekt, wystarczy kliknąć na jego nazwę. Jeśli szukanego projektu nie ma na liście, kliknij link Otwórz: Projekt... i przejdź do odpowiedniego folderu. Wewnątrz folderu znajduje się plik typu .sln (rozwiązanie). Wybierz ten plik i otwórz projekt. Aby wyświetlić formularz, przejdź do Eksploratora rozwiązań, znajdującego się w prawym górnym rogu okna, i kliknij dwukrotnie nazwę formularza.

1.7 DOKUMENTACJA WARTOŚCI NIERUCHOMOŚCI

W tej książce musimy podać wartości właściwości, których będziemy używać w kontrolkach. Gdy liczba właściwości jest mała, możemy je wyjaśnić w kilku stwierdzeniach, ale jeśli liczba jest większa, użyjemy tabeli. Ogólnie mamy kilka kontroli; każdy formant ma kilka właściwości, a każda właściwość ma wartość. Pamiętaj, że istnieje duża liczba właściwości dla każdej kontrolki, ale wymienimy tylko te, które musisz zmodyfikować. Pozostałe zachowają swoją wartość domyślną. Oto przykład:

Kontrol	Nieruchomość	Wartość
a		
etykieta1	Tekst	Witaj świecie
etykieta1	PowrótKolor	Sieć
przycisk 1	Tekst	Kliknij tutaj

Wymienione wartości odpowiadają następującym kontrolom:

- label1 , którego tekst to Hello world, a kolorem tła jest czerwony;
- button1 , którego tekst to Kliknij tutaj.



Zachowaj ostrożność podczas pisania nazw właściwości. Omówimy to bardziej szczegółowo w późniejszych rozdziałach; na razie wystarczy powiedzieć, że należy to robić wielką literą i bez spacji.

1.8 BŁĘDY W PROGRAMACH

W programach często występują błędy; na szczęście IDE może wykryć niektóre z nich dla nas. Oto przykład błędu:

label1.Txt = "Witaj, świecie".

Jeśli wpiszesz ten wiersz i uruchomisz program, IDE go podkreśli. Umieszczenie wskaźnika myszy nad podkreśloną częścią spowoduje wyświetlenie wyjaśnienia błędu. Zwróć uwagę, że wyjaśnienie może być trudne do zrozumienia (nawet dla eksperta) lub że błąd może leżeć gdzie indziej w programie. Jednak pierwszym krokiem powinno być zawsze sprawdzenie podkreślonego obszaru pod kątem błędów pisarskich. Gdy dowiesz się więcej o języku C#, poprawi się Twoja zdolność wykrywania błędów.

Po poprawieniu podświetlonych błędów należy wykonać program. W rzeczywistości, zanim to nastąpi, uruchamiany jest inny program zwany kompilatorem, który sprawdza kod pod kątem być może większej liczby błędów. Twój program może być wykonywany tylko do czasu, gdy wszystkie błędy kompilacji zostaną poprawione.

Błędy kompilacji są wyświetlane w oknie Lista błędów, które otwiera się pod kodem. Klikając dwukrotnie błąd w oknie wyników, IDE przeniesie Cię do miejsca w kodzie, w którym należy poprawić błąd.

Każdy, kto zaczyna pisać programy, napotka wiele błędów kompilacji. Nie zniechęcaj się; jest to część procesu uczenia się.

Po poprawieniu błędów kompilacji program będzie mógł działać. W tym momencie możemy zauważyć, że robi to niepoprawnie, co oznaczałoby „błąd wykonania” lub błąd. Takie błędy są trudniejsze do naprawienia, dlatego konieczne jest debugowanie.

1.9 FUNKCJE REDAKCJI

Edytor nie ogranicza się do wyświetlania tego, co pisze programista; Oto niektóre z jego dodatkowych zalet:

- Oferuje standardowe operacje wycinania, kopiowania i wklejania do schowka, umożliwiając kopiowanie kodu z innych programów Windows.
- Aby pisać większe programy, wymagane są dodatkowe elementy sterujące i zdarzenia. Listy rozwijane u góry edytora umożliwiają wybranie elementów sterujących oraz określonego zdarzenia dla tego elementu sterującego. Edytor jest automatycznie umieszczany w odpowiedniej metodzie.



- Edytor wyświetli kod C# w takiej postaci, w jakiej został wpisany zgodnie z określonymi regułami. Na przykład spacje zostaną automatycznie wstawione wokół symbolu =.
- Niektóre wiersze będą wymagały wcięcia - przesunięcia tekstu w prawo - dla którego należy wstawić dodatkowe białe znaki. Zwykle odbywa się to w odstępach czteroprzestrzennych. Jak zobaczymy w dalszych rozdziałach, kod C# składa się z sekcji w sekcjach, a wcięcia pomagają wskazać, gdzie zaczyna się i kończy każda sekcja.

Proces tworzenia wcięć w programie jest również nazywany formatowaniem. IDE może to zrobić automatycznie za pomocą menu:

Edycja | Zaawansowane | Formatowanie dokumentu Dobrym pomysłem jest częste formatowanie kodu.

- Każdy typ kontrolki ma określony zestaw właściwości. Jeśli wpiszesz nazwę kontrolki z kropką, jak pokazano poniżej: **label1**.

i poczekaj chwilę, edytor wyświetli listę właściwości tagu label1. Jeśli to narzędzie nie jest skonfigurowane w twoim systemie, możesz je aktywować, wybierając opcję:

Narzędzia | Opcje | Edytor tekstu | C#

a następnie zaznacz pole wyboru Automatyczna lista członkostwa.

- Sekcje kodu, nad którymi nie pracujemy, można zwinąć w jedną linię kodu. Aby to zrobić, kliknij mały symbol „-” po lewej stronie edytora. Aby rozwinąć sekcję, kliknij symbol „+”. Znajdziesz te symbole obok linii, jak poniżej:

private void button1_Click (nadawca obiektu, EventArgs e)

Nie musisz pamiętać wszystkich tych narzędzi, ponieważ są one zawsze pod ręką, aby ci pomóc.

Jednym z zadań związanych z układem kodu, które nie jest wykonywane automatycznie, jest dzielenie długich linii. Aby cała linia była widoczna na ekranie, możemy wybrać odpowiednie miejsce do jej podziału i nacisnąć klawisz Enter. IDE wtnie drugą część wiersza o cztery spacje. Oto przykład:

private void button1_Click (nadawca obiektu, EventArgs e)

Jeśli wciśniemy klawisz Enter po przecinku, kod pojawi się w następujący sposób:

private void button1_Click (nadawca obiektu, EventArgs e)

Często IDE tworzy bardzo długie linie, które oczywiście nie powinny zawierać błędów, więc lepiej ich nie dzielić. Jednak w miarę nabierania doświadczenia w posługiwaniu się językiem C# niektóre z tych długich wierszy będą tworzone przez



Ciebie i bardzo wygodnie będzie je podzielić, aby można było zobaczyć cały tekst naraz. Poprawia to również czytelność drukowanego kodu (tzw. listing).

1.10 SKRZYŃKA WIADOMOŚCI

Wcześniej używaliśmy kontrolki etykiety do wyświetlania tekstu na ekranie, ale możemy również użyć do tego celu okna komunikatu. Ta kontrolka nie pojawia się w przyborniku, ponieważ nie zajmuje stałego miejsca w formularzu; pojawia się tylko wtedy, gdy jest to wymagane. Poniżej znajduje się fragment kodu, który wyświetla komunikat Hello World w oknie komunikatu po kliknięciu przycisku:

```
private void button1_Click(nadawca obiektu, EventArgs e)
```

```
{  
MessageBox.Show("Witaj świecie");  
}
```

Rysunek 2.14 pokazuje, co się stanie, gdy uruchomisz program i klikniesz przycisk: zostanie wyświetlone okno komunikatu i musisz kliknąć przycisk OK, aby zniknęło. Ta funkcja oznacza, że okna komunikatów są używane do wyświetlania ważnych komunikatów, których użytkownik nie może zignorować.

Aby użyć okna wiadomości, wpisz wiersz podobny do powyższego, ale użyj własnej wiadomości w cudzysłowach. W tym momencie nie będziemy wyjaśniać celu instrukcji Show ani powodu, dla którego nawiasy są wymagane. Porozmawiamy o tym wszystkim, gdy przestudujemy metody bardziej szczegółowo.



Rysunek 2.14 Okno komunikatu

1.11 POMOC

System pomocy działa w oparciu o dwa możliwe źródła: jedno lokalne, zlokalizowane na własnym komputerze oraz drugie za pośrednictwem Internetu. Źródło internetowe jest często aktualizowane; jednak wersja lokalna jest wystarczająca, gdy zaczynasz pracować z C#. W przypadku braku połączenia z Internetem program automatycznie skorzysta z pomocy lokalnej.

Jeśli chcesz określić, skąd ma pochodzić pomoc, możesz ponownie skonfigurować system w następujący sposób:



1. W głównym oknie C# wybierz **Pomoc | Szukaj...**
2. W nowym oknie pomocy wybierz **Pomoc | Pomoc w pomocy** . Zostanie wyświetlonych kilka opcji, które źródło pomocy będzie miało najwyższy priorytet.

System pomocy języka C# jest rozbudowany, ale początkujący programista może uznać zawarte w nim informacje za skomplikowane. Jest to najbardziej przydatne, gdy masz pewne doświadczenie z językiem C# i potrzebujesz precyzyjnych szczegółów technicznych.

Najbardziej przydatne opcje w menu Pomoc to Indeks i Szukaj, ponieważ umożliwiają wpisanie tekstu, aby system mógł zlokalizować strony, które mogą być przydatne. Różnica polega na tym, że Indeks patrzy tylko na tytuły stron, podczas gdy Wyszukiwanie sprawdza także zawartość strony.

1.11.1 Podstawy programowania

- Formanty można umieszczać na formularzu w czasie projektowania.
- Możliwe jest ustawienie właściwości kontrolki w czasie projektowania.
- Programy mogą modyfikować właściwości w czasie wykonywania.
- Kiedy wystąpi zdarzenie (takie jak kliknięcie przycisku), system C# używa odpowiedniej metody, ale to programista musi umieścić kod w metodzie, aby obsłużyć to zdarzenie.

1.11.2 Typowe błędy programistyczne

- Zapomnij o przerwaniu wykonywania programu przed próbą modyfikacji formularza lub kodu.
- Pomylić formularz czasu projektowania z formularzem czasu wykonywania.

1.11.3 Sekrety kodowania

- W kodzie C#, aby odwołać się do właściwości kontrolki, używamy jej nazwy, po której następuje kropka i nazwa właściwości, jak w:

etykieta1.Tekst

- Do sekcji kodu pomiędzy:

```
private void botonNumero_Click (nadawca obiektu, EventArgs e)
```

```
{  
}
```

jest znany jako metoda.



- Skrzynki wiadomości nie są umieszczane bezpośrednio na formularzach. Aby pojawiły się na ekranie, musimy użyć następującej instrukcji:

MessageBox.Show („Oto tekst, którego szukasz”);

1.11.4 Nowe elementy językowe

Wprowadzenie do właściwości, metod i zdarzeń.

1.11.5 Nowe funkcje IDE

- Programy są zawarte w projektach.
- IDE tworzy folder zawierający pliki potrzebne do projektu.
- jest przesuwanie i zmiana rozmiaru kontrolki na formularzach.
- Zestaw narzędzi zawiera szeroki zakres elementów sterujących.
- Kliknięcie kontrolki prawym przyciskiem myszy pozwala nam wybrać jej właściwości.
- Dwukrotne kliknięcie przycisku w czasie projektowania tworzy metody do obsługi zdarzeń.

1.11.6 Streszczenie

Część zadania programistycznego polega na umieszczeniu formantów na formularzach i ustawieniu ich początkowych właściwości. IDE C# wykonuje to zadanie bezpośrednio, ale musisz poćwiczyć i przeczytać o IDE.



Część 4 - Wejście, fizyka, interfejs użytkownika, sceny i renderowanie

Interakcja z użytkownikiem -Wejście

Aby umożliwić użytkownikowi interakcję z naszymi aplikacjami, Unity automatycznie udostępnia klasę „Input”.

Aby uzyskać więcej informacji, zobacz:

<https://docs.unity3d.com/Manual/class-InputManager.html>

<https://docs.unity3d.com/ScriptReference/Input.html>

Wejście daje możliwość sprawdzenia czy i kiedy fizyczny przycisk (joypad, mysz, klawiatura) jest wciśnięty, uzyskać położenie jednej lub kilku osi fizycznych (np. joystick analogowy, przesunięcie myszki) lub stworzyć wirtualne osie zaczynając od dwa przyciski cyfrowe (np. prędkość - przycisk spowalniający).

Unity może odbierać dane wejściowe z ekranów dotykowych i różnych czujników smartfonów (np. żyroskopu, kamery i akcelerometru)

```
próżniaAktualizacja(){  
    jeśli(Wejście.GetKeyUp(Kod klucza.Space)){           //Wprowadzanie z  
klawiatury lub joypada  
        Odpluskwić.Dziennik("Miejsce zostało zwolnione");  
    }  
}
```

```
próżniaAktualizacja(){  
    pływakmojaVerticalValue =Wejście.GetAxis("Pionowy"); //wejście z  
analogowego joysticka lub wirtualnych osi  
    jeśli(myVerticalValue != 0)  
    {  
        //np. przesunąć gracza  
    }  
}
```



```
próżniaAktualizacja(){
    jeśli(Wejście.GetMouseButton(0)){//do wprowadzania myszy i ekranu
dotykowego
        Odpluskwić.Dziennik("Przycisk myszy 0 został kliknięty");
        PromieńmójRay =Kamera.main.ScreenPointToRay(Wejście.pozycja
myszy);
        jeśli(Fizyka.Raycast(myRay)){
            //Kliknąłem na coś, ponieważ myRay w coś uderzył
        }
    }
}
```

```
próżniaAktualizacja(){
    Wektor3przyspieszenie mojego urządzenia =Wejście.przyspieszenie;//Wejście
z akcelerometru
        jeśli(myDeviceAcceleration.magnitude > 10)
        {
            Odpluskwić.Dziennik("Il urządzenie è stato scosso forte");
        }
}
```

Dobrze jest sprawdzić i zarządzać danymi wejściowymi w funkcji Update (ponieważ funkcja jest wywoływana za każdym razem, gdy tworzona jest ramka), aby mieć responsywność proporcjonalną do szybkości klatek aplikacji.

Interakcja użytkownika — dane wyjściowe

W przypadku wyjścia logu (jako tekst) na konsolę można użyć funkcji:

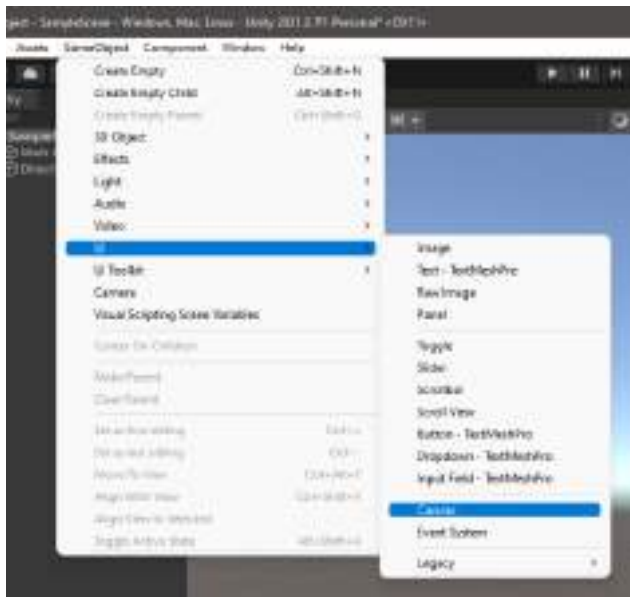
`Odpluskwić.Dziennik("własny tekst");`Większość interakcji między użytkownikiem a aplikacją odbywa się za pośrednictwem interfejsu użytkownika lub interfejsu użytkownika.



Unity pozwala łatwo stworzyć interfejs użytkownika za pomocą GameObject Canvas. Aby uzyskać więcej informacji, zobacz:

<https://docs.unity3d.com/2021.3/Documentation/Manual/UIColorCanvas.html>

<https://docs.unity3d.com/ScriptReference/Canvas.html>



Płótno umożliwia tworzenie programowalnego i konfigurowalnego interfejsu użytkownika przy użyciu różnych elementów.



Istnieje możliwość dodania tekstu i przycisków.

Aby utworzyć kod, który kontroluje tekst za pomocą przycisku, dodaj komponent skryptu do Empty GameObject.

za pomocą `UnityEngine.interfejs użytkownika`



publicznyklasaWyświetlacz zdrowia:JednoZachowanie{

publicznyTekstmój tekst zdrowotny;

intmoje zdrowie = 10;

publicznypróżniafunkcja mojego przycisku(){ //funkcja

do połączenia z przyciskiem

moje zdrowie--;

//Zmniejsz zdrowie

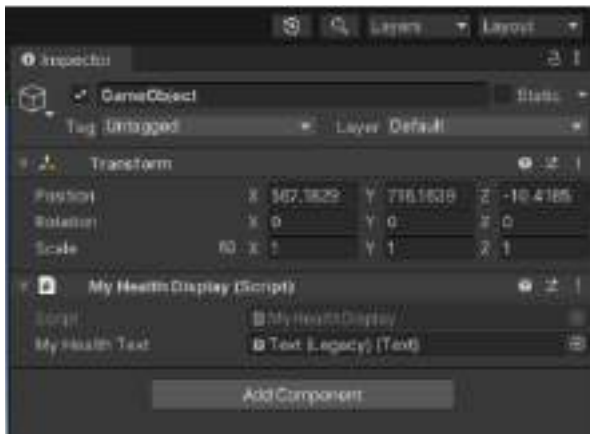
mój tekst zdrowia.tekst= „ZDROWIE:“ + moje zdrowie;

//aktualizuje etykiety

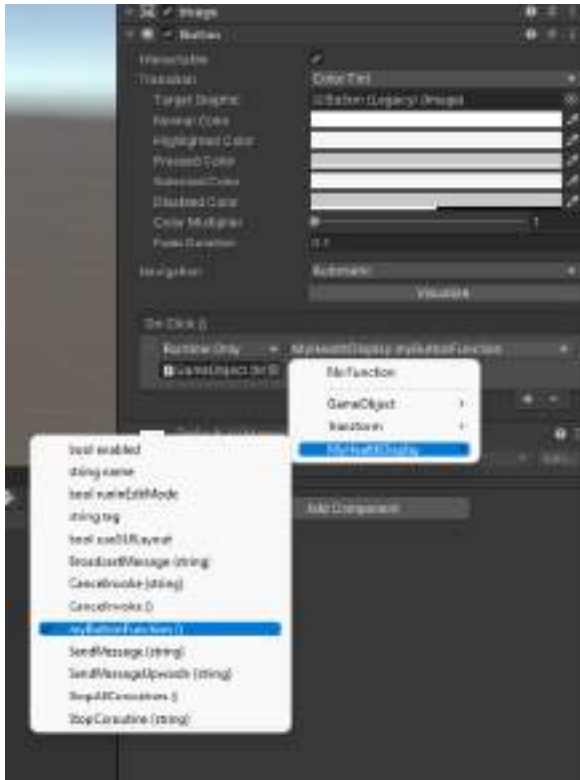
}

}

Połącz tekst ze skryptem, używając atrybutu Tekst publiczny (Tekst mojego zdrowia).



Połącz przycisk ze skryptem Empty GameObject za pomocą funkcji onclick() przycisku, używając funkcji public (próżniafunkcja mojego przycisku()).



Fizyka- Zderzenie

Kolizje w Unity są przetwarzane dzięki komponentom Collider (BoxCollider, SphereCollider, MeshCollider) i zarządzane za pomocą zdarzeń (OnCollisionEnter, OnCollisionExit, OnCollisionStay). Aby uzyskać więcej informacji, zobacz:

<https://docs.unity3d.com/Manual/collision-section.html>

<https://docs.unity3d.com/ScriptReference/Collision.html>

```

próżnia OnCollisionEnter(Kolizja zderzenie){
    jeśli (acollision.relativeVelocity.magnitude >= 10){ //jeśli piłka zderzy się z
gameObject
        Zniszczyć(Obiekt gry); //usuń obiekt gry
        Zniszczyć(acollision.gameObject); //Zniszcz piłkę
    }
}

```



W Edycja > Ustawienia projektu > Znaczniki i warstwy można dodawać nowe warstwy

W Edycja > Ustawienia projektu > Fizyka > Macierz kolizji można określić, między którymi warstwami muszą wystąpić kolizje.

Modyfikacja komponentu transform lub użycie funkcji „transform.Translate lub Transform.Rotate” nie powoduje kolizji.

Aby uniknąć niepotrzebnych obliczeń, wprowadzaj składową kolizji tylko wtedy, gdy jest to konieczne.

Sztywne ciało

Główne aspekty dynamiki są symulowane przez komponent Rigidbody.

<https://docs.unity3d.com/Manual/rigidbody-physics-section.html>

<https://docs.unity3d.com/ScriptReference/Rigidbody.html>

Dzięki zastosowaniu komponentu Rigidbody możesz aplikować **siły i momenty**, możesz ustawić prędkość, siły tarcia i grawitację oraz tworzyć ruchy powodujące kolizje.

```

prywatna pustkaAktualizacja(){
    intsiła skoku = 100;
    intprędkość = 10;
    jeśli(Wejście.PobierzKeyDown(KodKlucza.Spacja))
    obiekt gry. sztywne ciało.Dodaj siłę(Vector3.up * jumpForce, ForceMode.Impulse);
    jeśli(Wejście.PobierzKeyDown(Kod klucza.R))
    gameObject.rigidbody.AddTorque(transform.right * speed, ForceMode.Force);
}

```

Preferowane jest **odłączyć** wykorzystanie symulacji fizycznych z szybkości klatek za pomocą funkcji.

Naprawiono aktualizację, która jest aktualizowana.

```

prywatny booliść naprzód;
prywatna pustkaAktualizacja(){
    goingForward = Input.GetKey(KodKlucza.UpArrow);
}

```



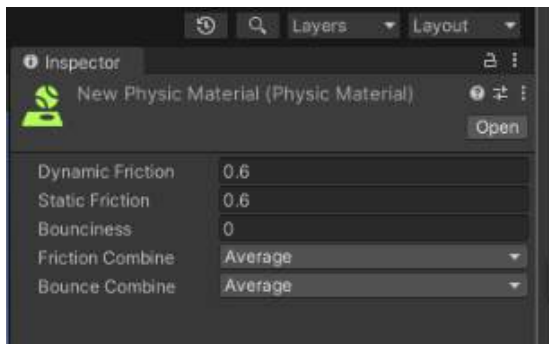
```
prywatna pustkaNaprawiono aktualizację(){  
    if(idzie do przodu){  
        obiekt gry. sztywne ciało.Dodaj siłę(Vector3.Forward * prędkość,  
ForceMode.Force);  
    }  
}
```

Bardzo duże prędkości ruchu mogą nie być zarządzane na czas.

Material fizyczny

Fizyka Materiały zastosowane w GameObecie ze zderzaczem umożliwiają określenie niektórych jego właściwości fizycznych, takich jak powstawanie tarcia statycznego i dynamicznego oraz sprężystość obiektu.

<https://docs.unity3d.com/Manual/class-PhysicMaterial.html>



Stawy

Możliwe jest połączenie dwóch GameObject (z Rigidbody) poprzez wspólny komponent.

<https://docs.unity3d.com/Manual/joints-section.html>

FixedJoint tworzy stałe połączenie między obiektem, do którego ten komponent został dołączony, a innym określonym obiektem. Rezultatem jest stałe połączenie, w którym jeden z obiektów fizycznie podąża za ruchami drugiego (tak samo jak dziecko GameObject podąża za rodzicem GameObject).



Połączenie zawiasowe tworzy połączenie zawiasowe między obiektem, do którego ten komponent został dołączony, a innym obiektem. Rezultatem jest stałe wiązanie między dwoma obiektami, które mają możliwość obracania się tylko wokół osi zawiasu.

SpringJoint tworzy połączenie sprężyste między obiektem, do którego ten komponent został dołączony, a innym obiektem. Rezultatem jest połączenie dwóch obiektów zgodne z fizyką sprężyny.

Ray (przeszkody i kule)

Klasa Ray umożliwia tworzenie promieni w celu zbadania obecności obiektów, zaczynając od określonego punktu i kierunku.

Klasa RaycastHit służy do zapamiętywania kolizji przez promienie.

Funkcja Physics.Raycast(.....) służy do sprawdzania kolizji wzdłuż promienia.

<https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>

PromieńmójRay =NowyPromień(mój punkt początkowy promienia, mój kierunek promienia);

//Promień(transform.pozycja, transform.forward) to promień rozpoczynający się od obiektu i skierowany do przodu

Raycast Hitmoje trafienie;

plywakodległość promienia = 100,0;

jeśli(Fizyka.Raycast(mój Ray,na zewnątrzmoje trafienie, odległość promienia)){

Dziennik debugowania (myHit.zderzak.imię);//możemy uzyskać zderzacz trafionego obiektu i jego nazwę lub inną składniki

}

<https://docs.unity3d.com/Manual/CameraRays.html>

Klasa Physics zapewnia kilka funkcjonalności, które są bardzo przydatne w fizyce.



<https://docs.unity3d.com/ScriptReference/Physics.html>

<https://docs.unity3d.com/Manual/PhysicsSection.html>

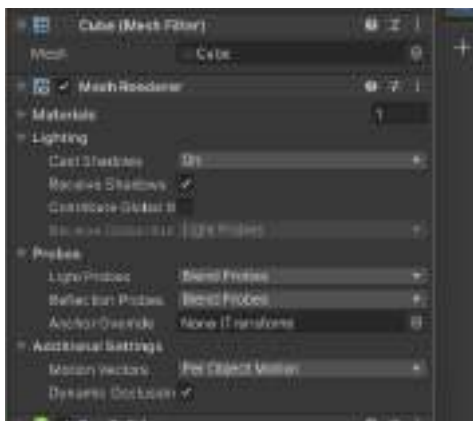
Podstawy renderowania

Istnieją różne komponenty, które charakteryzują grafikę obiektów. Komponent filtra siatkowego określa kształt naszego obiektu GameObject.

<https://docs.unity3d.com/Manual/mesh.html>

Dzięki komponentowi Mesh Render można określić, czy siatki naszego GameObject mają wyświetlać i odbierać cienie oraz jak mają się zachowywać, gdy odbijają światło.

<https://docs.unity3d.com/Manual/class-MeshRenderer.html>



Shadery

Shadery to modele matematyczne używane przez Unity do informowania karty graficznej, jak reprezentować obiekty na ekranie zgodnie ze światłami, cieniami, efektami...

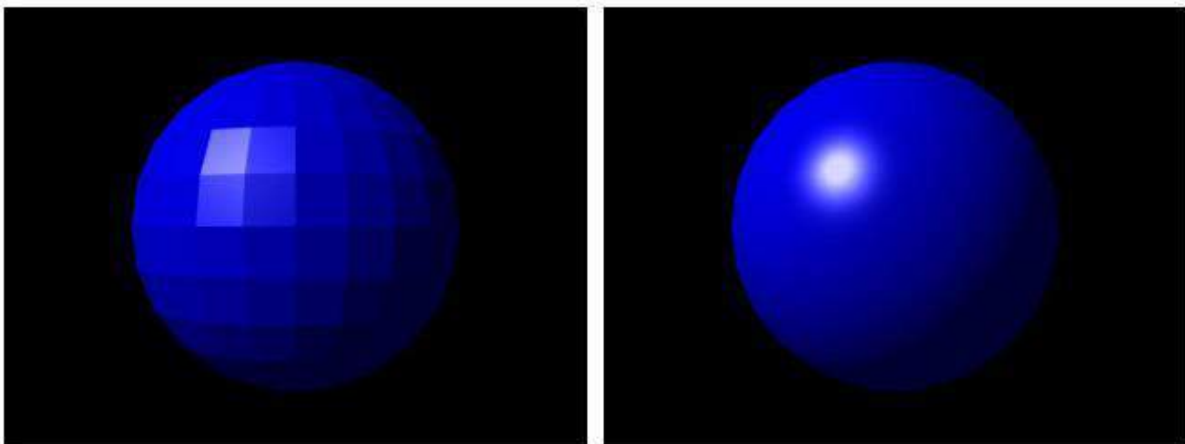
Każdy moduł cieniujący wykonuje inne obliczenia i uzyskuje różne efekty wizualne (na przykład realistyczne cienie, efekt kreskówek, efekt pikseli...)

<https://docs.unity3d.com/Manual/Shader.html>



Moduły cieniujące wierzchołków obliczają odbicia i cienie na powierzchni siatki, uśredniając to, co dzieje się na wierzchołkach powierzchni (szybciej, ale z niższą jakością. Odbicia i cienie podążają za kształtem siatki).

Pixel shadery obliczają światła i cienie, biorąc pod uwagę wszystkie aspekty związane z obiektem ustawionym w jednym pikselu ekranu (wolniej, ale z wyższą jakością, cienie i odbicia są niezależne od kształtu siatki, do której są dołączone).



Materiały

Materiały definiują wartości, których shadery muszą używać podczas obliczeń graficznych.

Za pomocą tego samego shadera można definiować materiały o różnych kolorach, przezroczystościach, jasności, nieprzezroczystości.

Nie wszystkie shadery obsługują wszystkie właściwości (np. przezroczystość).



A.M.E.F.E.
asociación municipal de
educación y temas de Europa

Tpaidea



Erasmus+
Enriching lives, opening minds.



<https://docs.unity3d.com/Manual/Materials.html>

Tekstura

Tekstury mogą definiować kolor i inne właściwości materiału piksel po pikselu.

Importowanie tekstury do Unity jest tak proste, jak przeciągnięcie jej do Zasobów. Aby użyć go w materiale, po prostu przeciągnij go do właściwości materiału, dla którego chcemy go użyć. (np. albedo).

Tekstury o wysokiej rozdzielczości komplikują i wykładniczo wydłużają obliczenia graficzne.

<https://docs.unity3d.com/Manual/Textures.html>

Źródła światła

W Unity są różne źródła światła.

Światło punktowe to światło, które znajduje się w punkcie sceny i emituje światło we wszystkich kierunkach jednakowo

Światło punktowe to światło, które znajduje się w punkcie sceny i emituje światło w kształcie stożka



Światło kierunkowe to światło, które znajduje się nieskończenie daleko i emituje światło tylko w jednym kierunku

Światło obszarowe to światło zdefiniowane przez wielokąt w scenie, które emituje światło we wszystkich kierunkach równomiernie na swojej powierzchni, ale tylko z jednej strony.

Dla każdego źródła światła można określić jego intensywność, kolor, w niektórych przypadkach kierunek, odległość efektu oraz to, czy powoduje to powstawanie cieni.

<https://docs.unity3d.com/Manual/Lighting.html>

<https://docs.unity3d.com/Manual/LightingOverview.html>



Na urządzeniach mobilnych nie zaleca się używania cieni.



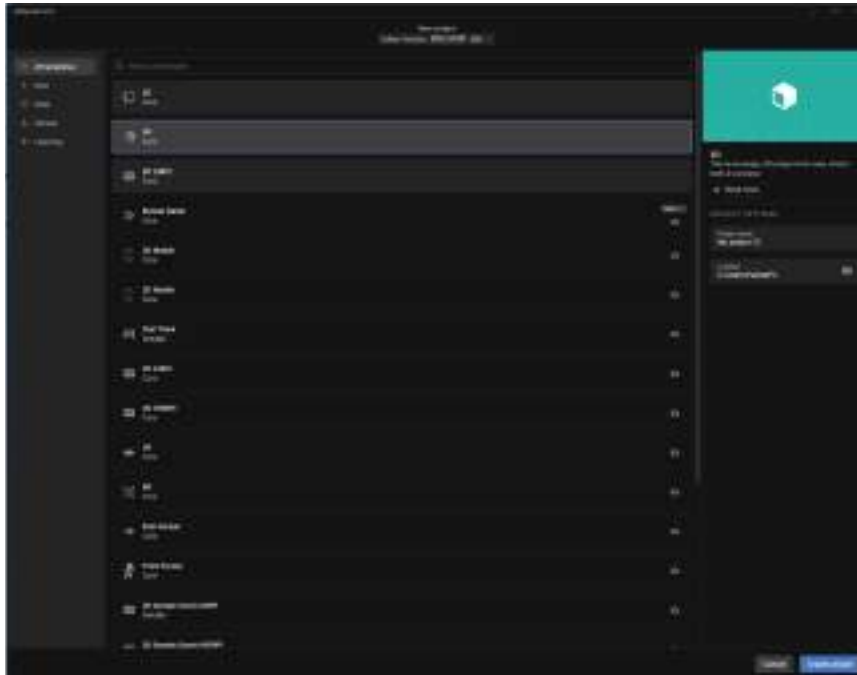
Część 5 - Praktyczny przewodnik z przykładami kodów

w tym rozdział będziemy _ Widzieć jak się zorientować niektóre funkcje gry wideo . _

założmy stworzyć nowy _ projekt , naciskając przycisk *Nowy projekt* przycisk .



Teraz wybierz „3D-core”, ustaw projekt nazwę i wybierz „ Utwórz projekt”.



Raz że projekt _ Jest utworzony użytkownik Unity interfejs będzie pojawić się .

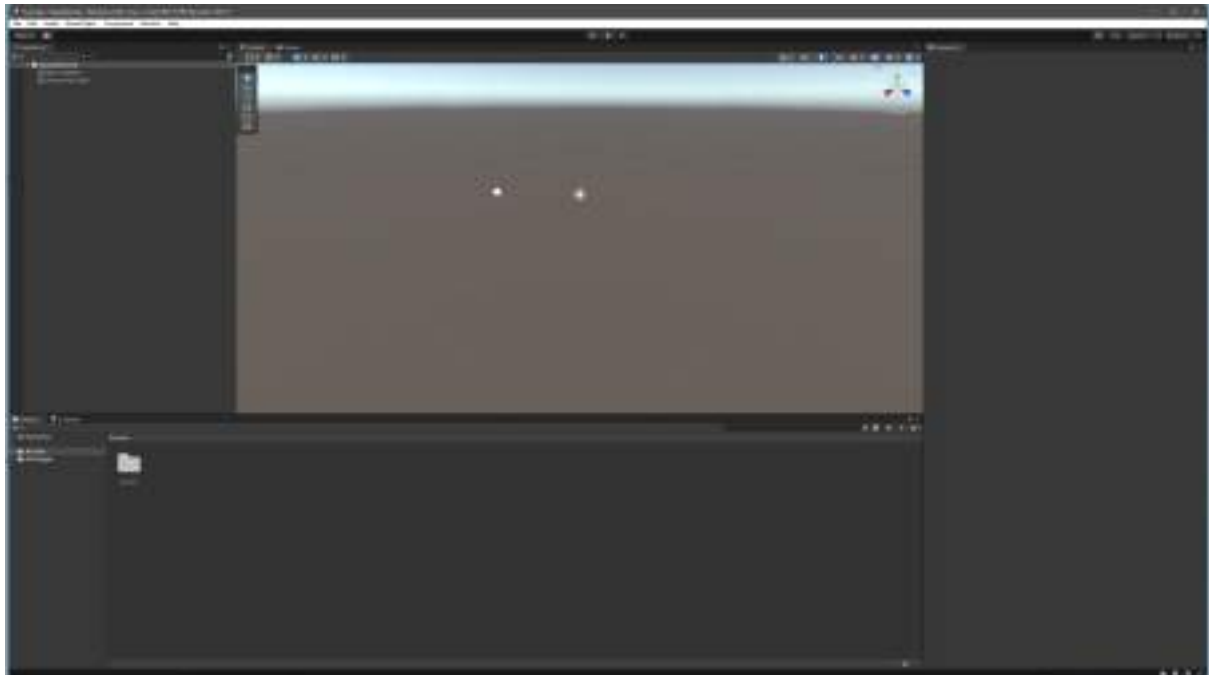


A.M.E.F.E
asociación magisteria de
educación y formación europea

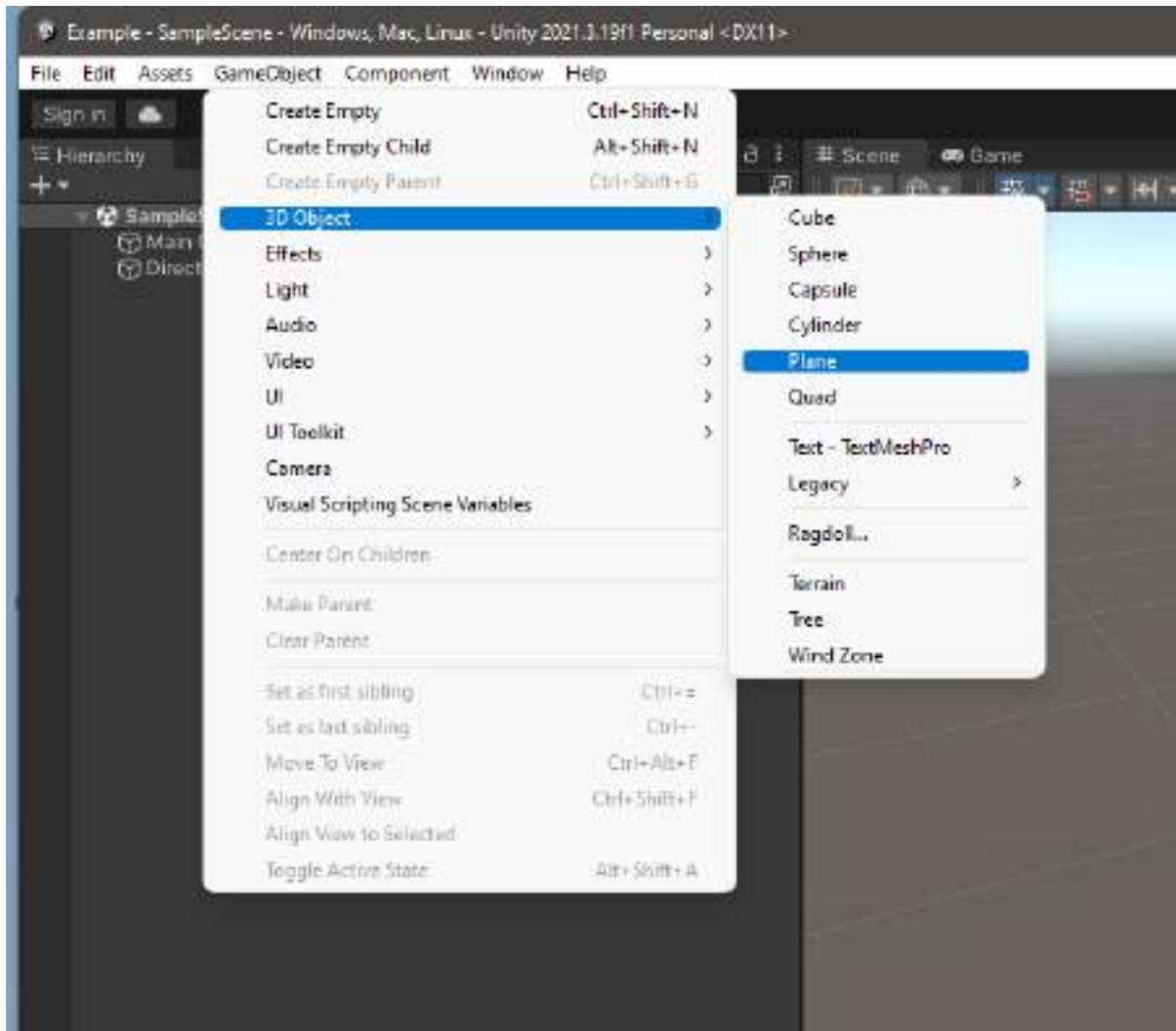
Paidea



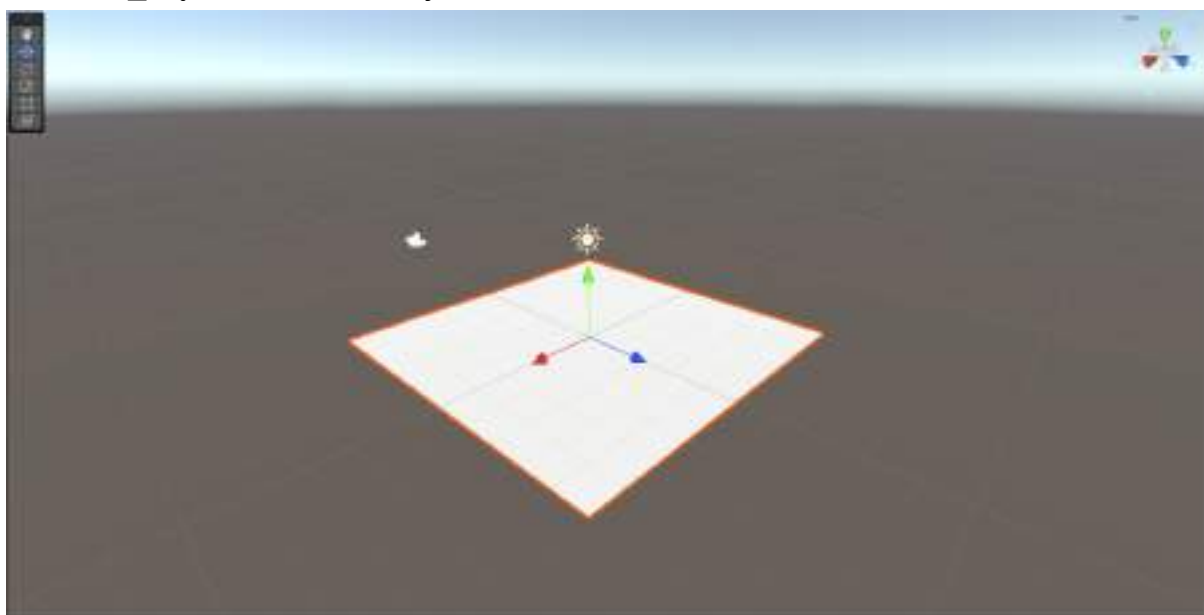
 **Erasmus+**
Enriching lives, opening minds.



začínáme přidat samolot z menu , zgodnie z pisemnym _ intymny stosunek dvojga ludzi ponížej
: GameObject -> Objekt 3D -> Samolot



Samolot _ będzie Patrząc tak jak to :





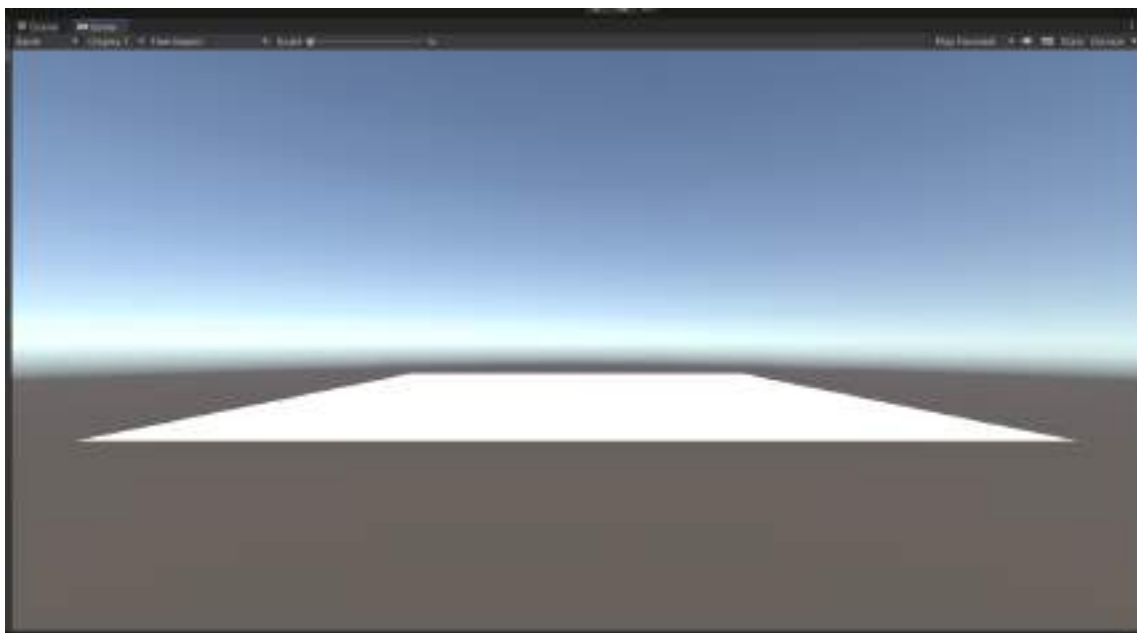
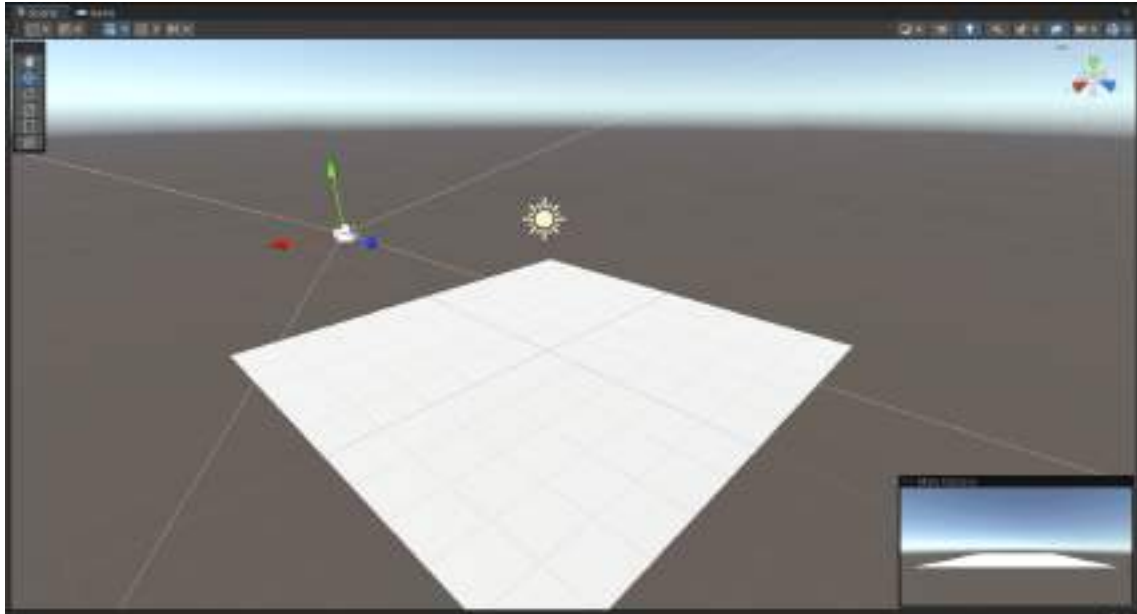
A.M.E.F.E.
asociación municipal de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.

Wybierz aparat i wykonaj pewnie samolot _ Jest oprawiony poprawnie





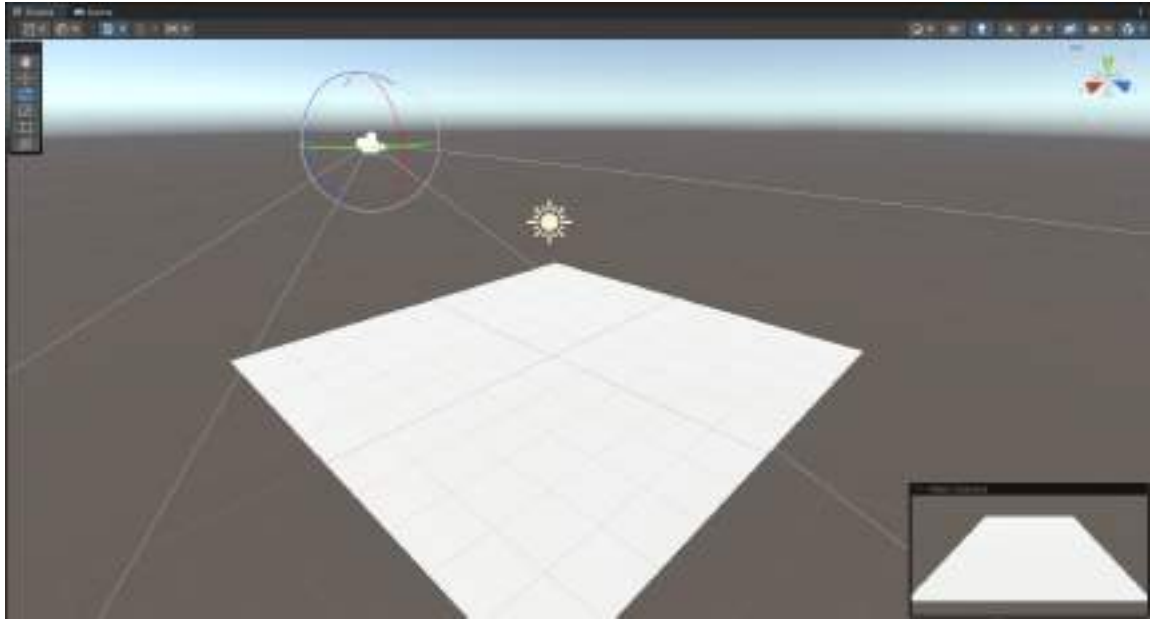
A.M.E.F.E.
asociación municipal de
educación y temas de europa

Paidea



Erasmus+
Enriching lives, opening minds.

Przesuwaj i obracaj aparat za pomocą odpowiednich narzędzi po lewej stronie sceny



_ widok .

załóżmy stworzyć nowy _ materiał do barwienia podłogi _ zielony .

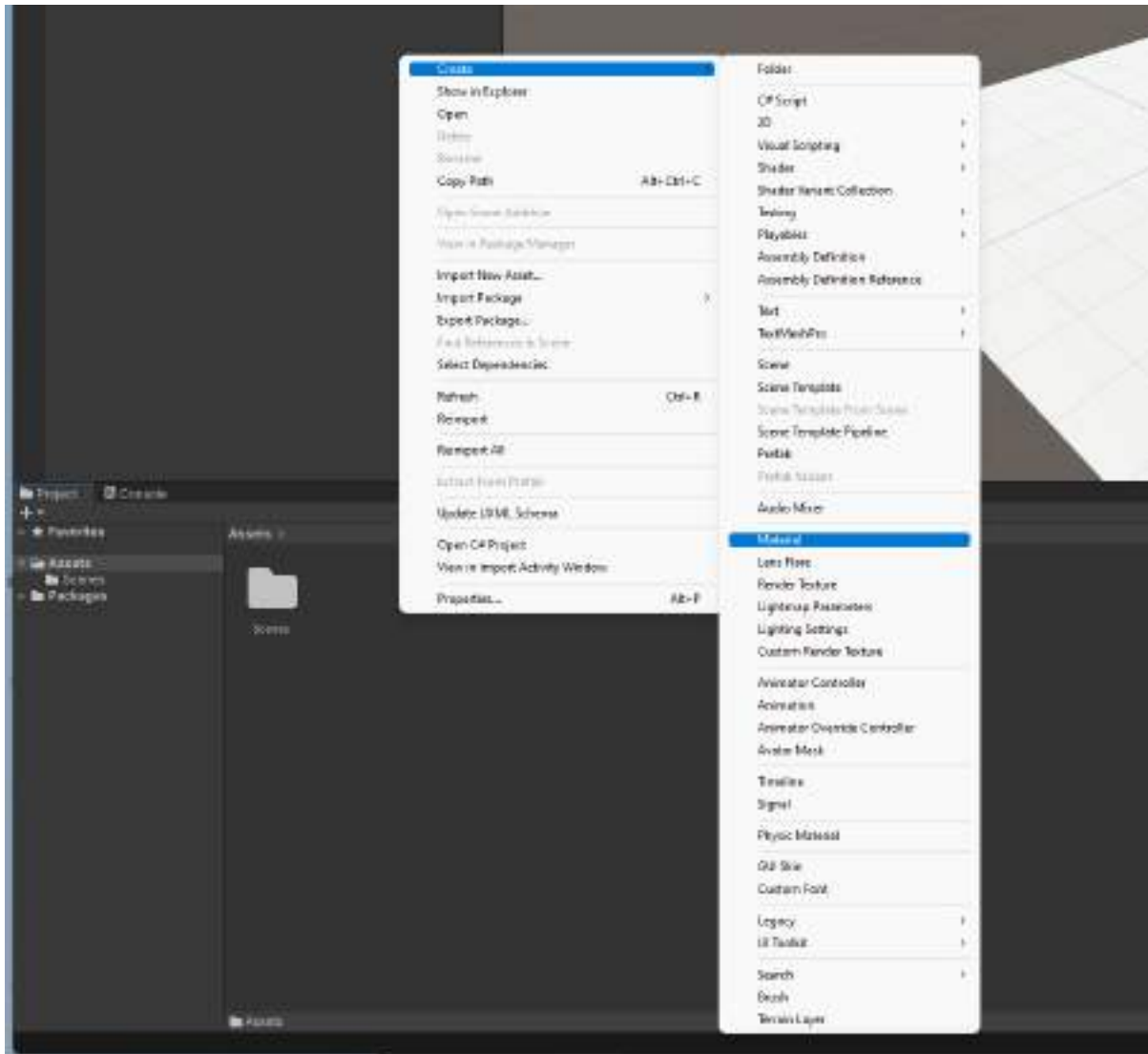


A.M.E.F.E
Asociación Madrileña de
Educación y Tecnología Europea



Erasmus+
Enriching lives, opening minds.

Kliknij lewą _ mysz przycisk na panelu Zasoby , Utwórz -> Materiał .

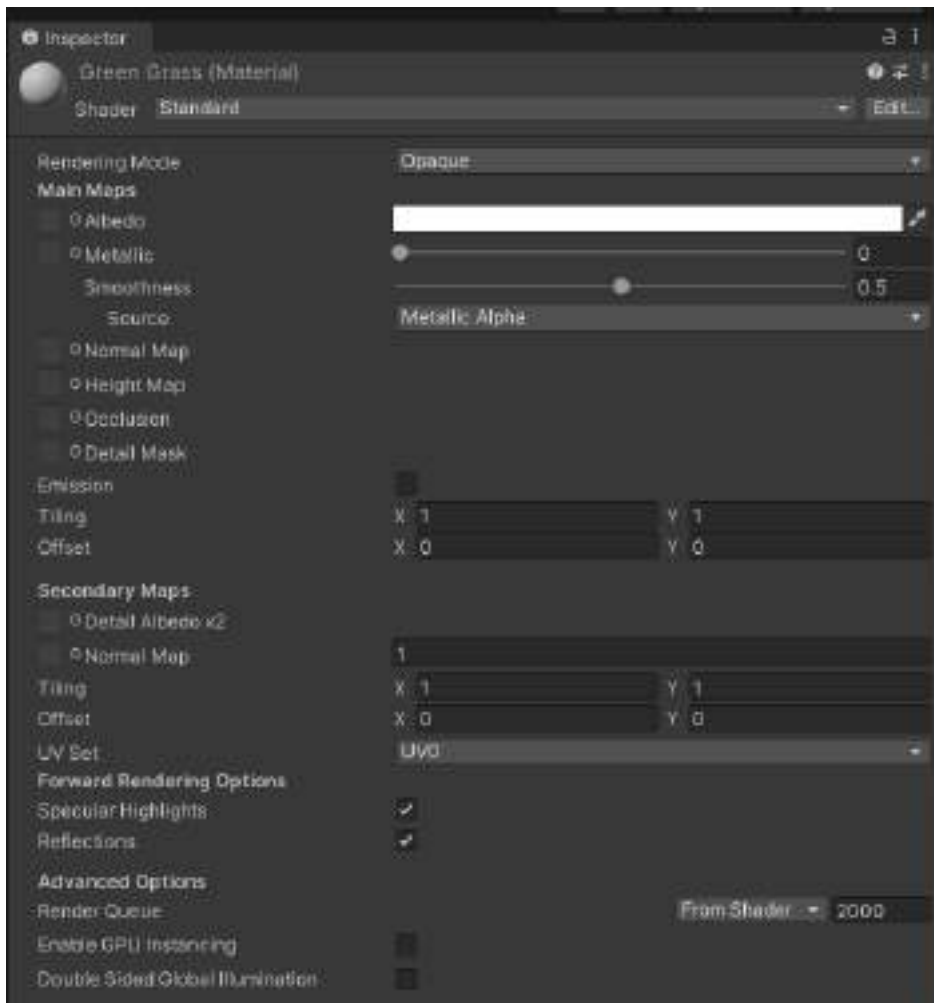


załóżmy nazwać materiał _





Ustaw kolor w panelu „inspektor”, naciskając prostokąt barwione na biało.





Wybierz kolor .



Przecignij materiał z Zasobów i sprowadź samolot _ w scenę _ widok do zmiany jego materiał .



Tą samą metodą utwórz kostkę i pokoloruj _ To światło niebieski .

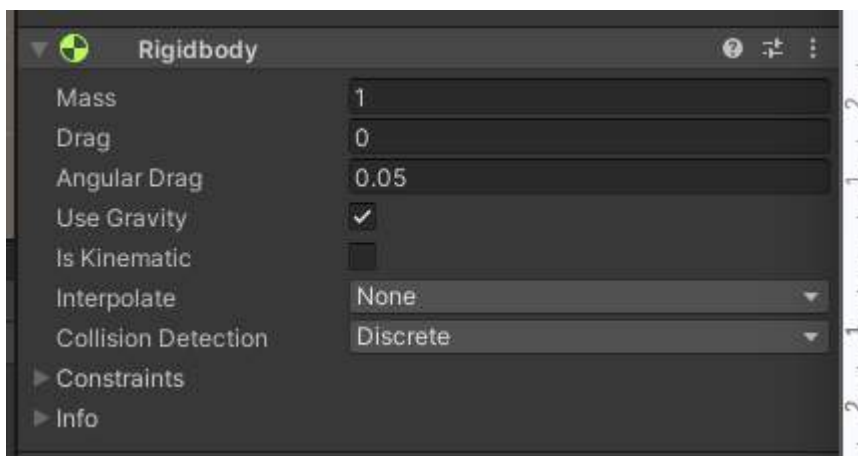
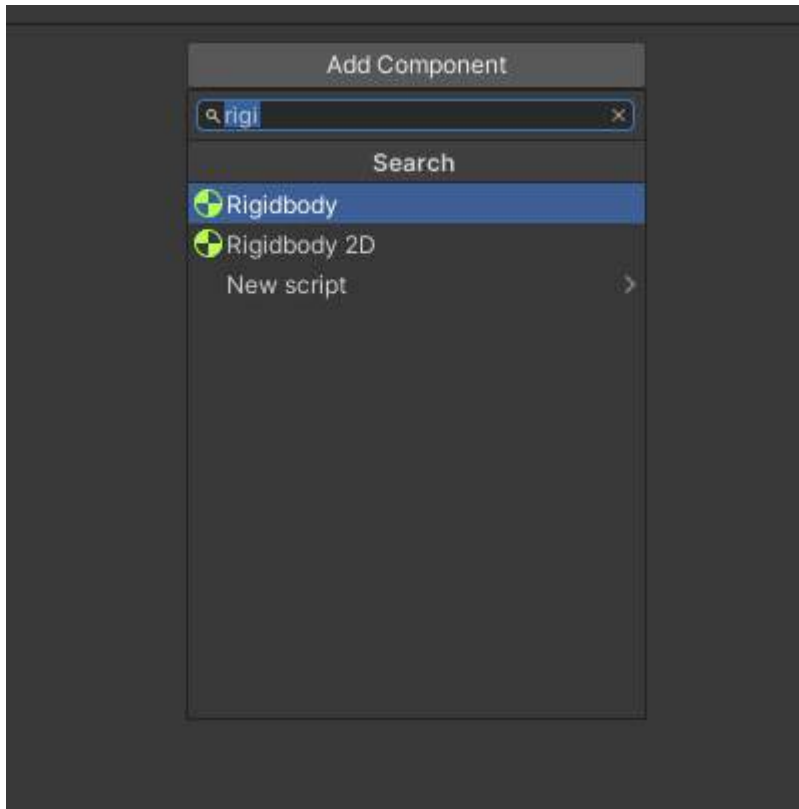


załóżmy dodaj komponent Rigidbody :

Wybierz kostkę , naciśnij Dodaj komponent w panelu inspektora , wyszukaj i wybierz „ Rigidbody ”.



Ten będzie pozwolić kostce być kontrolowane _ za pomocą fizyki i być poddanym grawitacji .





Wybierz Zamroź Obrót (X, Y, Z) jeśli chcesz zapobiec obracaniu się sześcianu __ chwila ruch

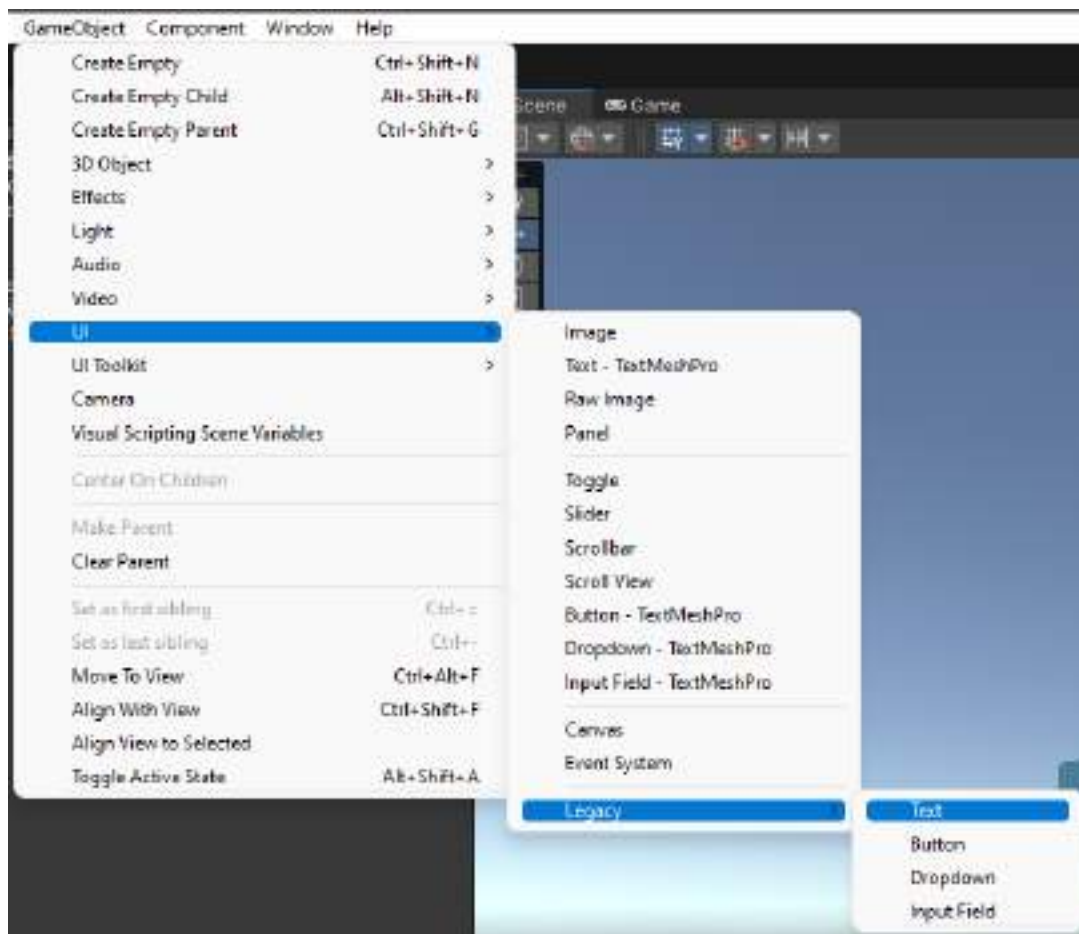


(wskazówka : Zamiast przytrzymywać lewy przycisk mysz przycisk i poruszanie się ; Ty Móc użyj WASD _ umożliwić wybrać __ jakiś obiekt z panelu „Hierarchia” i naciśnij klawisze SHIFT+F, aby ustawić ostrość widoku bezpośrednio na to przedmiot).

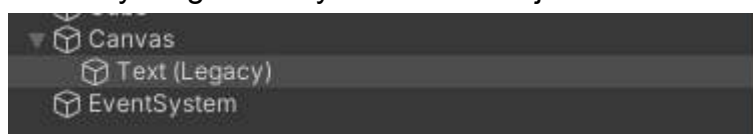
załóżmy dodaj nakładkę _ tekst do komunikacji z użytkownikiem :



GameObject -> Interfejs użytkownika -> Dziedzictwo -> Tekst



A bardzo duże " Płótno " (Płótno) woli pojawić się gdzie możemy _ dodać teksty i inne elementy do grafiki użytkownik interfejs .



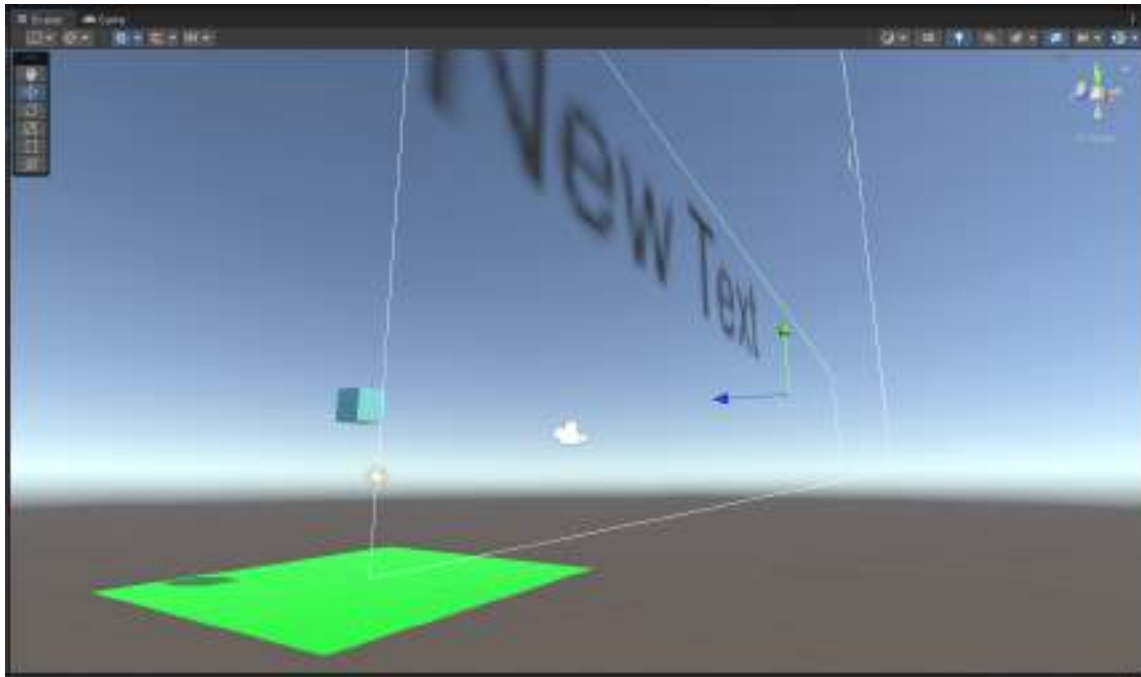


A.M.E.F.E.
asociación municipal de
educación y temas de Europa

Paidea



Erasmus+
Enriching lives, opening minds.

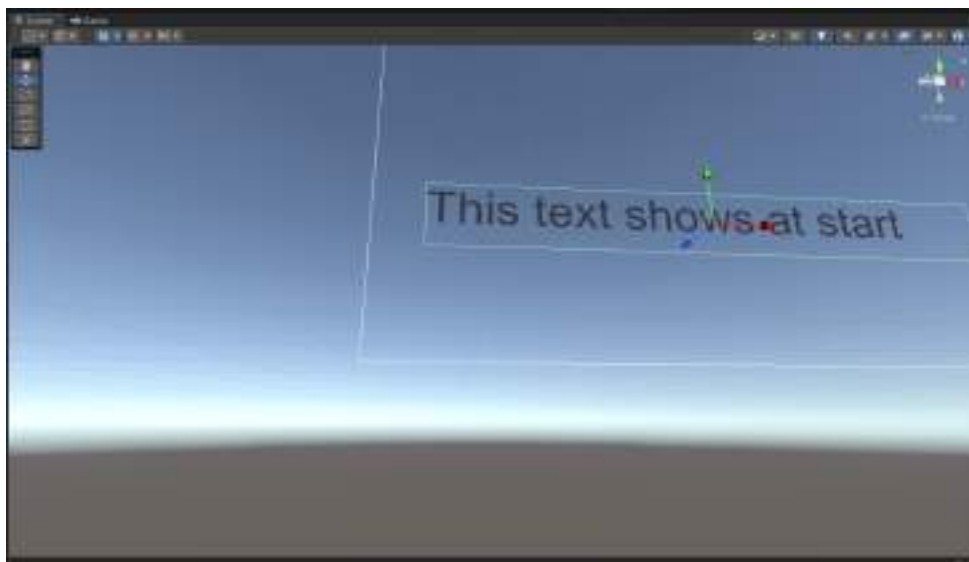
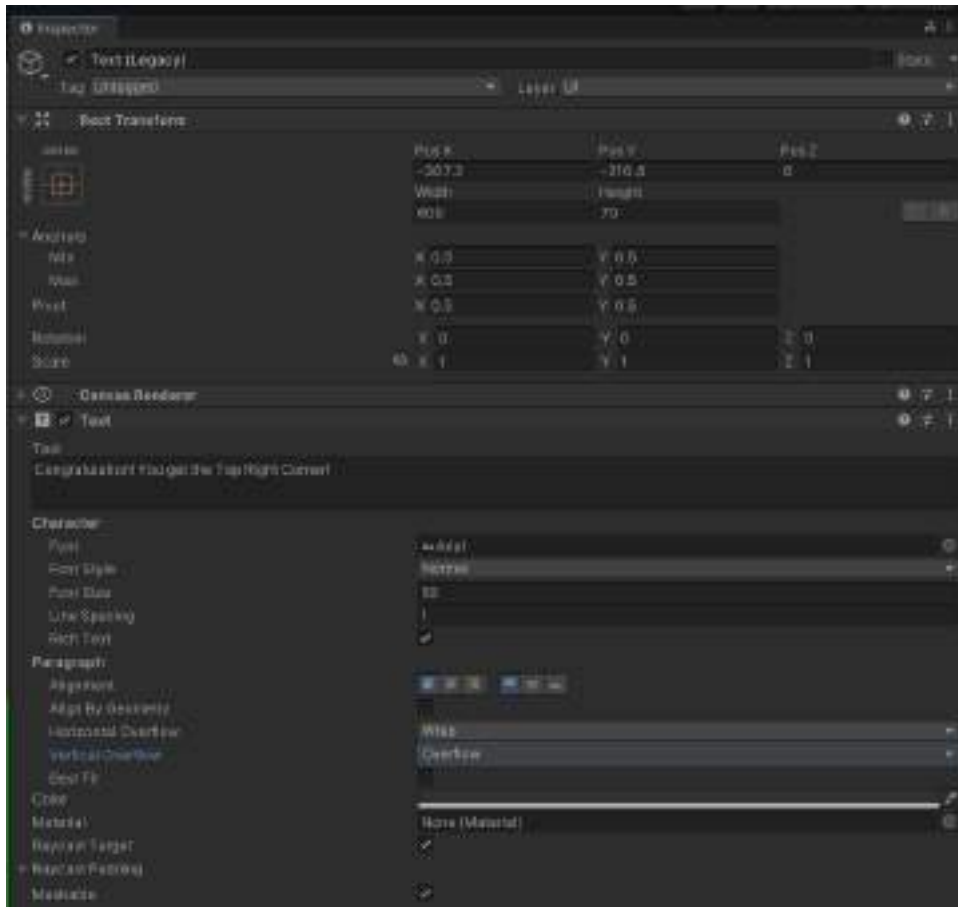


Wybierz Tekst (starszy)

Zmień szerokość , wysokość , tekst i czcionka rozmiar z panelu Inspektora ;



Przenies tekst na pozycję _ To chcesz .





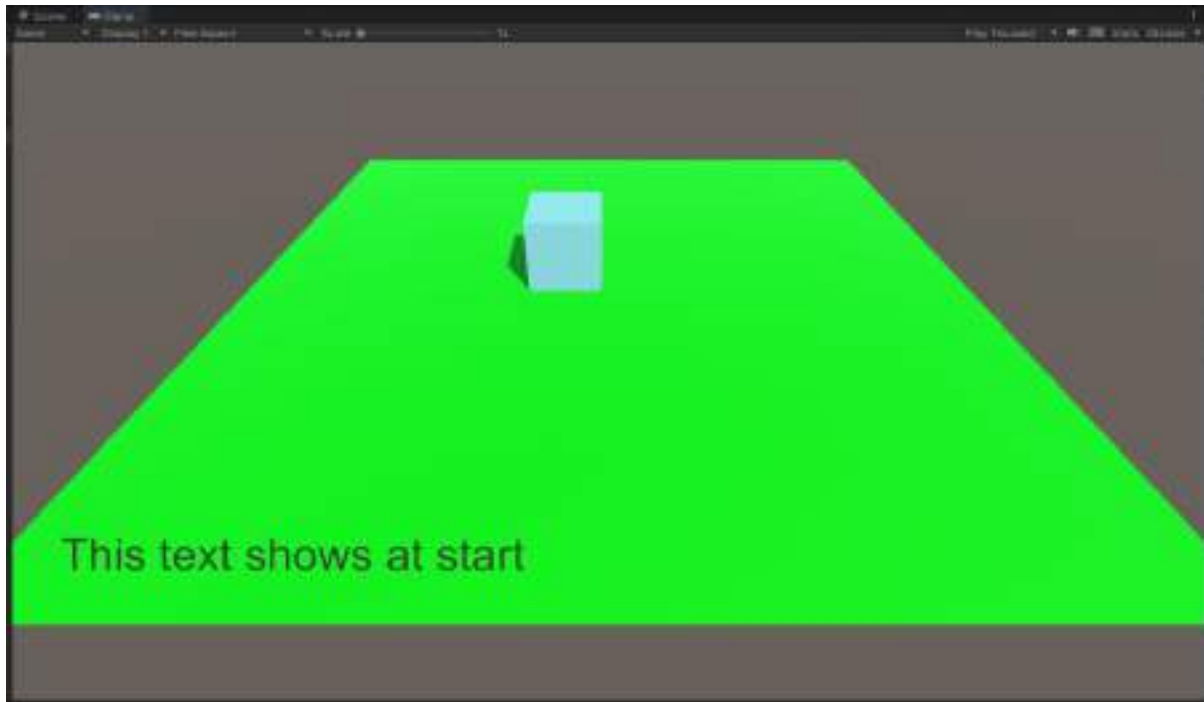
A.M.E.F.E.
asociación matemática de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.

Pozycja tekstu na płótnie _ _ odpowiada pozycji na ekranie _ _ podczas gry .

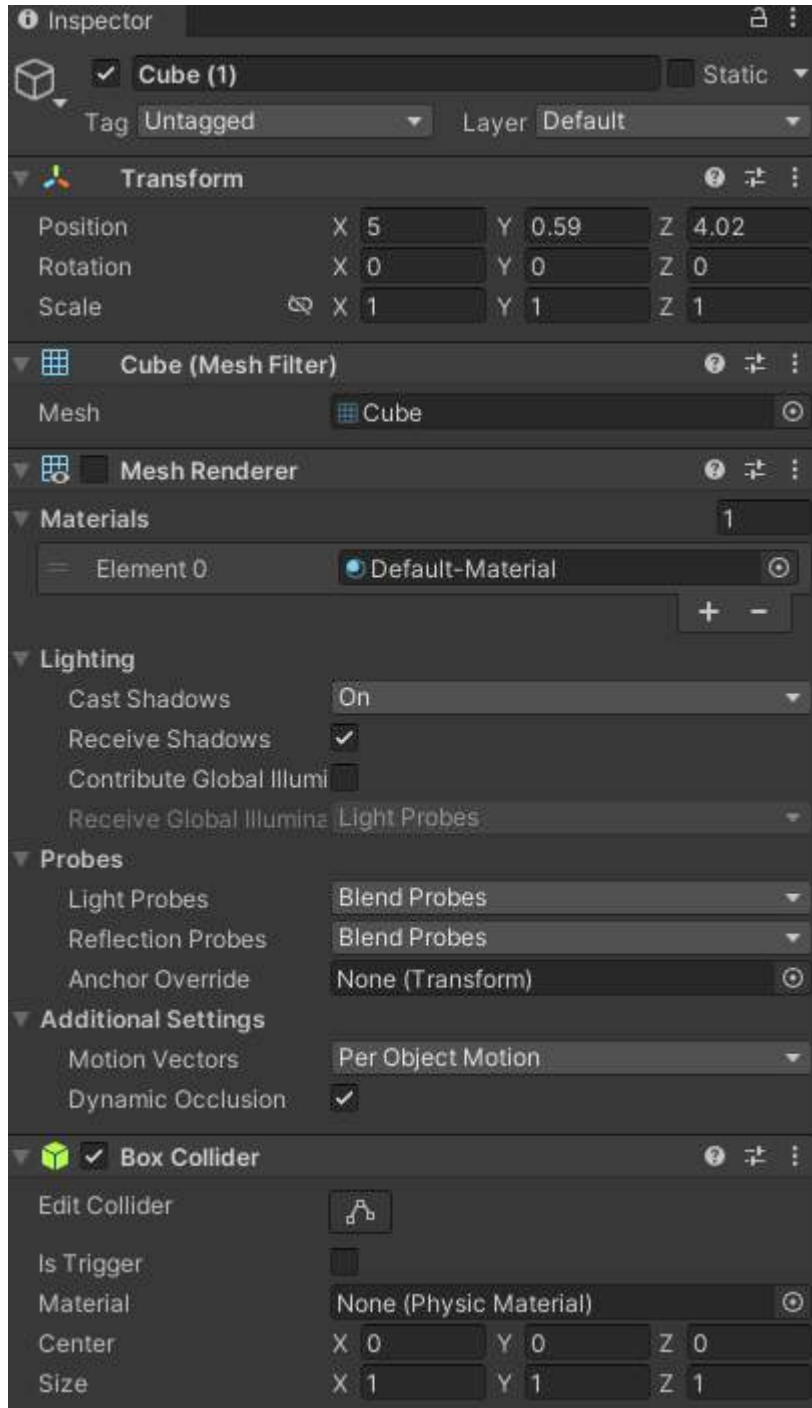


załóżmy tworzyć inny kostka :

w Inspektorze -> Box Collider zaznacz „ jest Wyzwalacz ” , aby przekształcić kostkę w _ _
obszar To zaczyna jakiś wydarzenie .

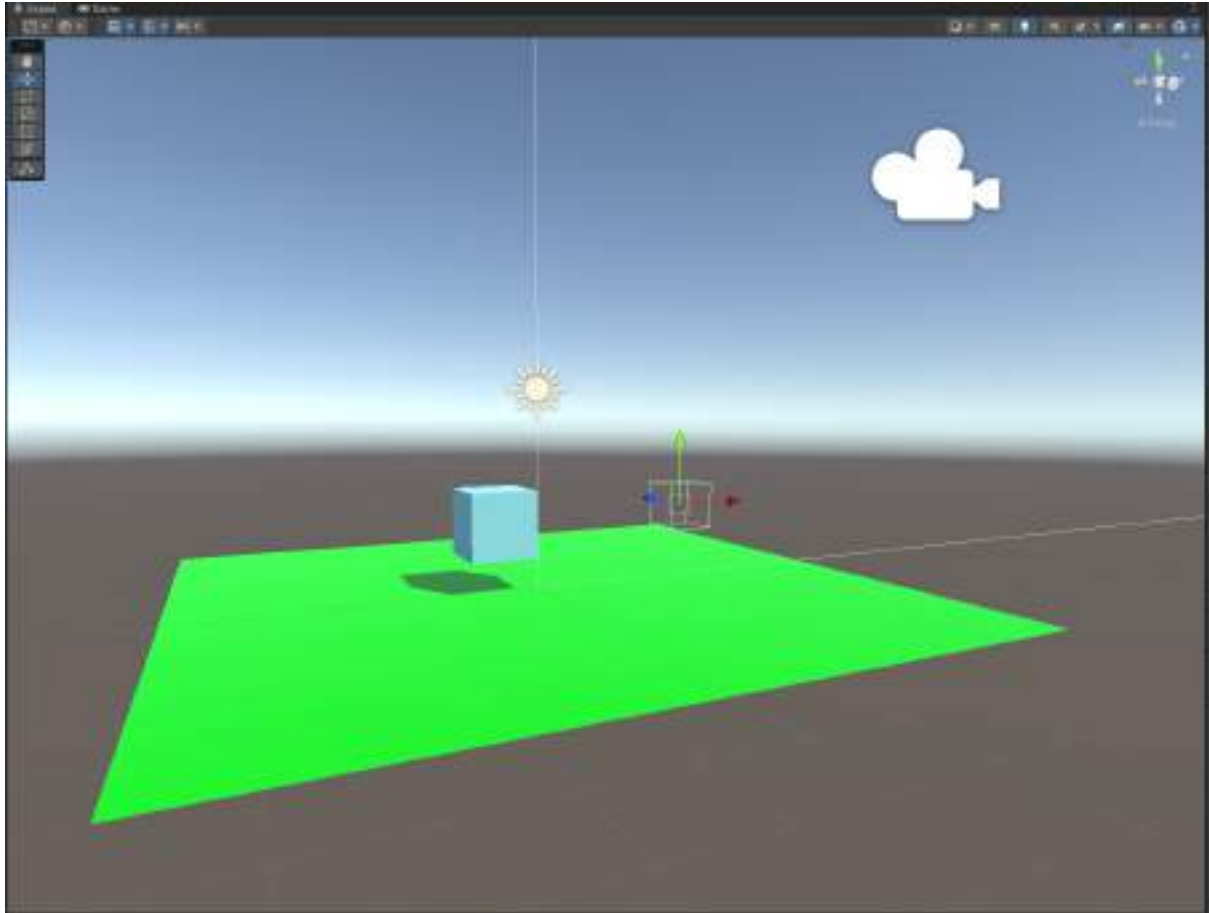


Odznacz siatkę _ Renderowanie pudełko , aby utworzyć obszar To ma zostać utworzony niewidzialny .





Zlokalizuj nowy _ sześcián w prawym górnym rogu rogu powierzchni .



Dodaj składnik skryptu o nazwie TEXT.



Podwójnie kliknij plik, który pojawiają się w dodaj kod _ instrukcja Na linia 8.



aktywach i

Ciąg publiczny textToShow ;



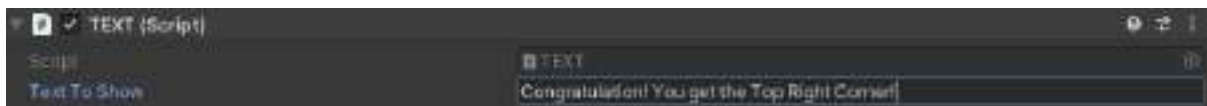
```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5
6  public class TEXT : MonoBehaviour
7  {
8      public string textToShow; //public variables can be used from other scripts
9
10     // Start is called before the first frame update
11     void Start()
12     {
13     }
14
15     // Update is called once per frame
16     void Update()
17     {
18     }
19
20 }
21
22

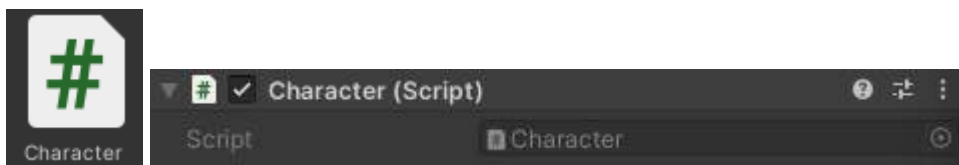
```

Zapisz i zamknij kod _ redaktor .

dodane _ instrukcja pozwala nam użyć i ustawić ciąg textToShow z panelu inspektora .
Zmień tekst , aby pokazać wartość w „ Gratulacje ! Ty zdobądź prawy górny róg !”



Wybierz światło niebieski cube i dodaj do niego również składnik skryptu .
_ Nazwij skrypt „ Postać ” .



Otwórz skrypt i dodaj Ten linia kodu _ po pierwszym _ klamra zanim unieważnienie Start()

plywak publiczny siłaKwota = 0,8f; // pytanie Ryga imposta il waleczność Dela Forza

Dodaj następujące wiersze kodów _ między nimi _ nawiasy klamrowe funkcji Update .

Kierunek Vector3 = nowy Vector3(Input.GetAxis (" Horizontal "), 0, Input.GetAxis (" Vertical ")); // ottiene la direzione utylizacja gli assi „ poziome ” i „ pionowe ” mappati sui smaczne WASD lub sulle poziomica analogiczny

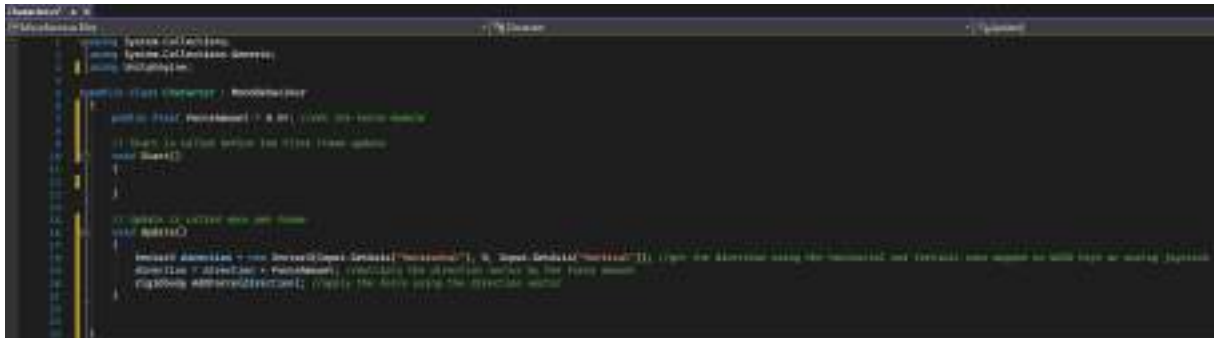
*kierunek = kierunek * siłaKwota ; / moltiplica la direzione per la forzat*



sztynne ciało.AddForce (kierunek); // aplikacja forzqa _ użytkowy il zawetować kierunek

Te instrukcje będzie włączać nam się ruszyć nasz światło niebieski sześcian za pomocą klawiszy WASD na klawiaturze.

Kod _ będzie podobny do tego :



RATOWAĆ

Włącz możliwość korzystania z interfejsu użytkownika _ instrukcje , dodając następujący tekst w wierszu 4

za pomocą UnityEngine.UI ;

Wstaw następujący ciąg znaków po deklaracji „ forceAmount ”

Tekst publiczny tekst do zmiany ;

powyższe _ linia kodu _ będzie umożliwić nam połączyć textToChange z tekstem na płótnie za pomocą panelu inspektora .

Dodać Ten kod przed zamknięciem ostatniego __ klamra .

próżnia OnTriggerEnter (Collider inne)

```

{
    textToChange.GetComponent < Tekst >(). text = Other.gameObject.GetComponent
<TEKST>(). tekst do pokazania ;
}

```



poprzedni _ funkcjonować będzie zmieniń nakładkę _ tekst do tego w niewidzialnym sześcian kod ustawiony jako wyzwalacz obszar .

Kod _ będzie podobny do tego :

```

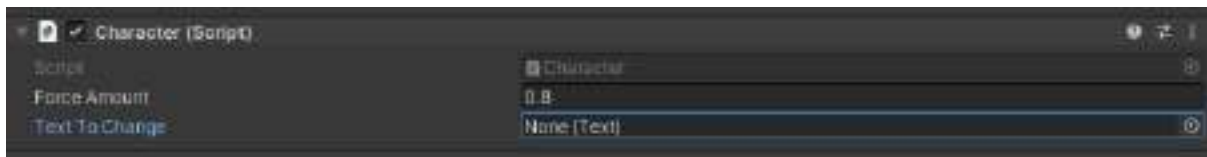
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class Character : MonoBehaviour
7  {
8      public float ForceAmount = 4.5f; //set the force amount
9      public Text textToChange;
10     // Start is called before the first frame update
11     void Start()
12     {
13     }
14
15     // Update is called once per frame
16     void Update()
17     {
18         Vector3 direction = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical")); //get the direction using the key
19         direction = direction * ForceAmount; //multiply the direction vector by the force amount
20         GetComponent()->AddForce(direction); //apply the force using the direction vector and the rigidbody component
21     }
22
23     void OnTriggerEnter(Collider other)
24     {
25         //set the textToChange variable from the UI script component from the trigger that the character has hit and put it in the textToChange.GetComponent<Text>().text = other.gameObject.GetComponent<Text>().textToShow;
26     }
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

_ ZAPISZ i zamknij kod redaktor .

Aby zmienić tekst interfejsu użytkownika , it musi być połączony ze światłem niebieski sześcian za pomocą jego Panel inspektora .

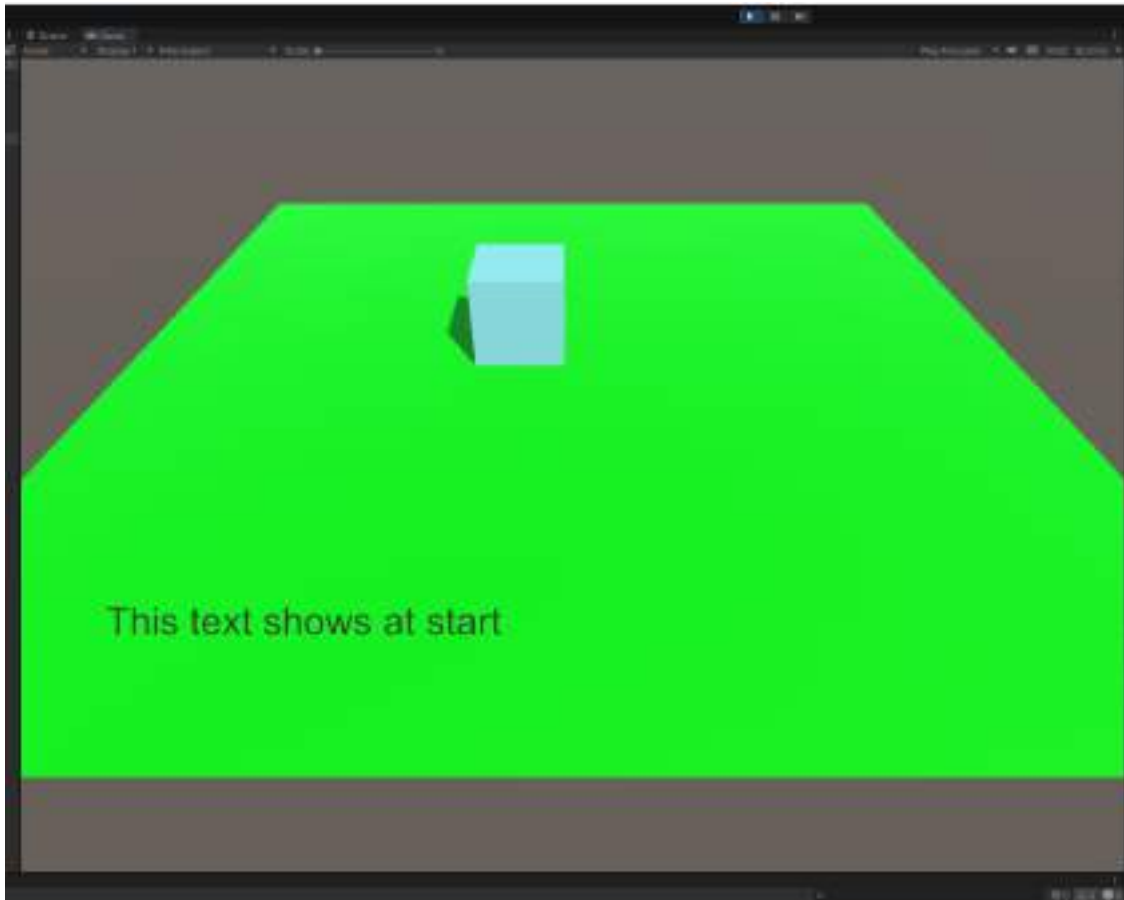
Ustawiamy Znak (skrypt) -> Tekst do zmiany .



Przeciągnij Tekst (starszy) z panelu Hierarchia do Znak (skrypt) -> Tekst do zmiany w panelu inspektora światła niebieski sześcian .



ZAPISZ projekt i naciśnij grać , aby spróbować nasz gra .



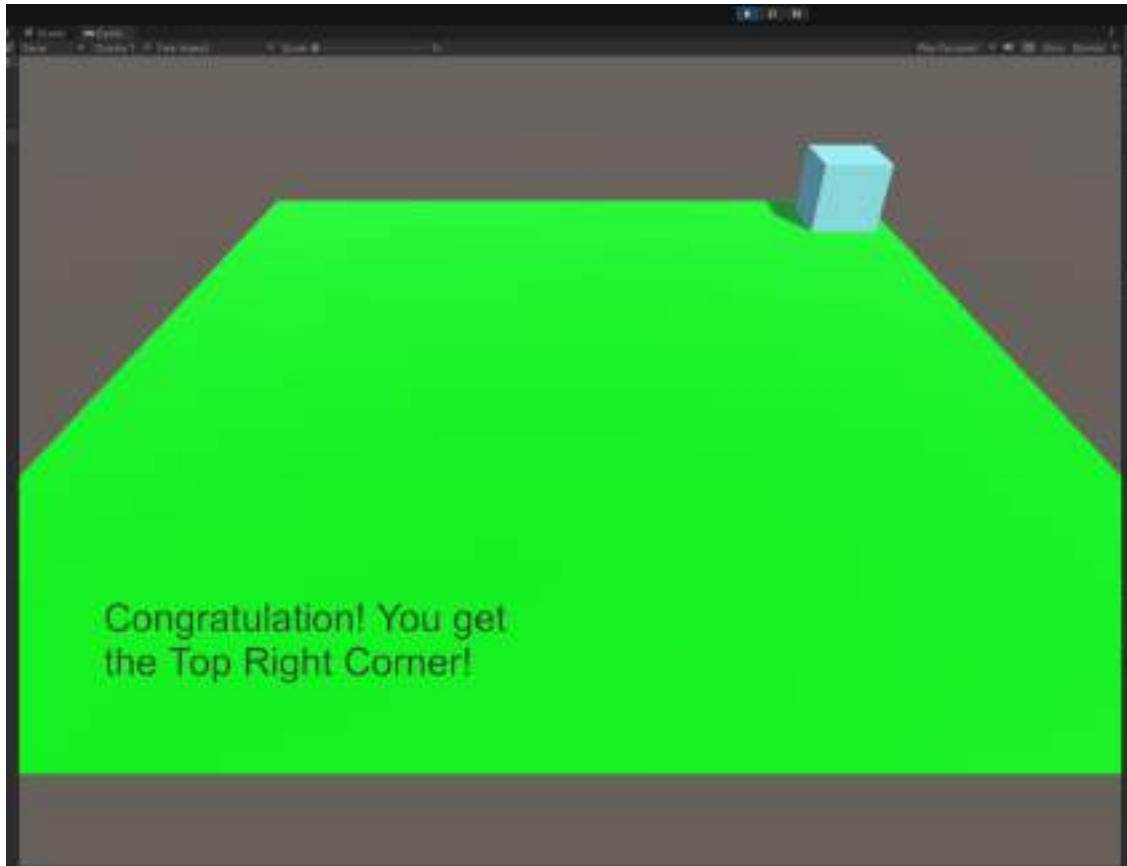


A.M.E.F.E.
asociación magistral de
educación y formación europea

Paidea



Erasmus+
Enriching lives, opening minds.



możemy _ skopiuj i wklej niewidoczne _ sześcian kilka razy tworzyć _ różny obszary które pokazują inaczej teksty .



A.M.E.F.E
Asociación Madrileña de
Educación y Formación Europea

Tpaidea



Erasmus+
Enriching lives, opening minds.

-  Cube (1)
-  Cube (2)
-  Cube (3)
-  Cube (4)
-  Cube (5)

