

Programowanie gier w środowisku Unity z zastosowaniem Blendera

Adam Szmagliński, p. F215
<http://fizyka.szmaglinski.eu>

Instytut Fizyki PK

Kraków, 27.09.2023

Środowisko Unity

stosuje się w wielu *dziedzinach*:

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna
- filmy i animacje

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna
- filmy i animacje
- edukacja techniczna

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna
- filmy i animacje
- edukacja techniczna
- motoryzacja

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna
- filmy i animacje
- edukacja techniczna
- motoryzacja
- lotnictwo

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna
- filmy i animacje
- edukacja techniczna
- motoryzacja
- lotnictwo
- architektura, inżynieria i budownictwo

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna
- filmy i animacje
- edukacja techniczna
- motoryzacja
- lotnictwo
- architektura, inżynieria i budownictwo
- sprzedaż detaliczna

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna
- filmy i animacje
- edukacja techniczna
- motoryzacja
- lotnictwo
- architektura, inżynieria i budownictwo
- sprzedaż detaliczna
- rozrywka na żywo

Środowisko Unity

stosuje się w wielu *dziedzinach*:

- gry i rzeczywistość wirtualna
- filmy i animacje
- edukacja techniczna
- motoryzacja
- lotnictwo
- architektura, inżynieria i budownictwo
- sprzedaż detaliczna
- rozrywka na żywo
- dobre samopoczucie i wpływ społeczny

Instalacja Unity

Unity Hub

Najpierw instalujemy Unity Hub: <https://unity.com/download>
Unity może poprosić o zaakceptowanie licencji oraz o utworzenie konta użytkownika Unity.

Następnie poprzez ten Hub możemy zainstalować wiele wersji silnika Unity, co zajmuje znaczną część pamięci. Dla danej wersji można doinstalować dodatkowe moduły również w późniejszym czasie. Moduł dokumentacji przydaje się do pracy offline.

Instalacja Unity

Unity Hub

Najpierw instalujemy Unity Hub: <https://unity.com/download>
Unity może poprosić o zaakceptowanie licencji oraz o utworzenie konta użytkownika Unity.

Następnie poprzez ten Hub możemy zainstalować wiele wersji silnika Unity, co zajmuje znaczną część pamięci. Dla danej wersji można doinstalować dodatkowe moduły również w późniejszym czasie. Moduł dokumentacji przydaje się do pracy offline.

Edytor kodu

można powiązać z Unity.

Microsoft Visual Studio <https://visualstudio.microsoft.com/pl/vs/>
lub *Microsoft Visual Studio Code* <https://code.visualstudio.com>
wyróżniają specjalne symbole i słowa oraz wspomagają pracę z kodem.

Wersja edytora

do budowania projektów Unity zalecana jest najnowsza spośród LTS. Long Term Support oznaczają stabilne wersje wspierane przez dwa lata od ich wydania.

Każda wersja edytora numerowana rokiem i następną liczbą (niezależnie od kolejnego oznaczenia po kropce np. 2023.3.____), opatrzona tagiem LTS, powinna współpracować z projektem zbudowanym na danym oznaczeniu edytora (tym samym roku i następnej liczbie).

Szablony nowych projektów gier

dostępne są z poziomu Unity Hub w karcie *Projects*.
Można je podzielić na kategorie:

Szablony nowych projektów gier

dostępne są z poziomu Unity Hub w karcie *Projects*.

Można je podzielić na kategorie:

- **Core:** podstawowe

Szablony nowych projektów gier

dostępne są z poziomu Unity Hub w karcie *Projects*.

Można je podzielić na kategorie:

- **Core:** podstawowe
- **URP:** Universal Render Pipeline

Szablony nowych projektów gier

dostępne są z poziomu Unity Hub w karcie *Projects*.

Można je podzielić na kategorie:

- **Core**: podstawowe
- **URP**: Universal Render Pipeline
- **HDRP**: High Definition Render Pipeline

Szablony nowych projektów gier

dostępne są z poziomu Unity Hub w karcie *Projects*.

Można je podzielić na kategorie:

- **Core:** podstawowe
- **URP:** Universal Render Pipeline
- **HDRP:** High Definition Render Pipeline
- **Sample:** *Creator Kit* – przykładowe sceny

Szablony nowych projektów gier

dostępne są z poziomu Unity Hub w karcie *Projects*.

Można je podzielić na kategorie:

- **Core:** podstawowe
- **URP:** Universal Render Pipeline
- **HDRP:** High Definition Render Pipeline
- **Sample:** *Creator Kit* – przykładowe sceny
- **Learning:** *Microgame* – gry wraz z tutorialiem

Szablony nowych projektów gier

dostępne są z poziomu Unity Hub w karcie *Projects*.

Można je podzielić na kategorie:

- **Core:** podstawowe
- **URP:** Universal Render Pipeline
- **HDRP:** High Definition Render Pipeline
- **Sample:** *Creator Kit* – przykładowe sceny
- **Learning:** *Microgame* – gry wraz z tutorialiem

Aktywa

są wykorzystywane w Unity do budowy scen, postaci itd. Aktywami mogą być np. tekstury, modele, animacje, dźwięki, skrypty.

Przydatne ustawienia edytora

Okno: Edit -> Preferences

Przydatne ustawienia edytora

Okno: Edit -> Preferences

- motyw edytora: General -> Editor Theme

Przydatne ustawienia edytora

Okno: Edit -> Preferences

- motyw edytora: General -> Editor Theme
- kolor trybu uruchomionej gry: Colors -> Playmode tint (A: 255 - przezroczysty)

Przydatne ustawienia edytora

Okno: Edit -> Preferences

- motyw edytora: General -> Editor Theme
- kolor trybu uruchomionej gry: Colors -> Playmode tint (A: 255 - przezroczysty)
- wybrany edytor kodu: External Tools -> External Script Editor

Przydatne ustawienia edytora

Okno: Edit → Preferences

- motyw edytora: General → Editor Theme
- kolor trybu uruchomionej gry: Colors → Playmode tint (A: 255 - przezroczysty)
- wybrany edytor kodu: External Tools → External Script Editor

Inne okna

Przydatne ustawienia edytora

Okno: Edit → Preferences

- motyw edytora: General → Editor Theme
- kolor trybu uruchomionej gry: Colors → Playmode tint (A: 255 - przezroczysty)
- wybrany edytor kodu: External Tools → External Script Editor

Inne okna

- doinstalowywanie pakietów: Window → Package Manager

Edytor Visual Studio

potrzebuje dodatkowych składników wspierających Unity:

Edytor Visual Studio

potrzebuje dodatkowych składników wspierających Unity:

- 1 wybieramy z menu: Narzędzia → Pobierz narzędzia i funkcje...
(otwiera się Visual Studio Installer)

Edytor Visual Studio

potrzebuje dodatkowych składników wspierających Unity:

- 1 wybieramy z menu: Narzędzia → Pobierz narzędzia i funkcje...
(otwiera się Visual Studio Installer)
- 2 wybieramy pakiet: Opracowywanie gier za pomocą aparatu Unity

Edytor Visual Studio

potrzebuje dodatkowych składników wspierających Unity:

- 1 wybieramy z menu: Narzędzia → Pobierz narzędzia i funkcje...
(otwiera się Visual Studio Installer)
- 2 wybieramy pakiet: Opracowywanie gier za pomocą aparatu Unity

Zapisany plik skryptu zostaje uruchomiony gdy jest składnikiem obiektu okna *Hierarchy*. Dodać jako składnik obiektu można poprzez przeciągnięcie skryptu na obiekt w oknie *Hierarchy* lub na okno *Inspector* wybranego obiektu lub w oknie *Inspector* wybieramy poprzez przycisk "Add Component".

Interfejs Środowiska Unity

w swoim podstawowym układzie składa się z okien:

Interfejs Środowiska Unity

w swoim podstawowym układzie składa się z okien:

- *Hierarchy*, zawierająca wszystkie elementy bieżącej sceny,

Interfejs Środowiska Unity

w swoim podstawowym układzie składa się z okien:

- *Hierarchy*, zawierająca wszystkie elementy bieżącej sceny,
- *Scene* i *Game* - domyślnie ustawione centralnie, widok w *Scene* sami ustawiamy, a w *Game* to obraz z głównej kamery,

Interfejs Środowiska Unity

w swoim podstawowym układzie składa się z okien:

- *Hierarchy*, zawierająca wszystkie elementy bieżącej sceny,
- *Scene* i *Game* - domyślnie ustawione centralnie, widok w *Scene* sami ustawiamy, a w *Game* to obraz z głównej kamery,
- *Inspector* - przedstawia składniki wybranego obiektu,

Interfejs Środowiska Unity

w swoim podstawowym układzie składa się z okien:

- *Hierarchy*, zawierająca wszystkie elementy bieżącej sceny,
- *Scene* i *Game* - domyślnie ustawione centralnie, widok w *Scene* sami ustawiamy, a w *Game* to obraz z głównej kamery,
- *Inspector* - przedstawia składniki wybranego obiektu,
- *Project* - zawiera pliki wchodzące w skład projektu,

Interfejs Środowiska Unity

w swoim podstawowym układzie składa się z okien:

- *Hierarchy*, zawierająca wszystkie elementy bieżącej sceny,
- *Scene* i *Game* - domyślnie ustawione centralnie, widok w *Scene* sami ustawiamy, a w *Game* to obraz z głównej kamery,
- *Inspector* - przedstawia składniki wybranego obiektu,
- *Project* - zawiera pliki wchodzące w skład projektu,
- *Console* - wyświetla informacje generowane przez Środowisko Unity.

Interfejs Środowiska Unity

w swoim podstawowym układzie składa się z okien:

- *Hierarchy*, zawierająca wszystkie elementy bieżącej sceny,
- *Scene* i *Game* - domyślnie ustawione centralnie, widok w *Scene* sami ustawiamy, a w *Game* to obraz z głównej kamery,
- *Inspector* - przedstawia składniki wybranego obiektu,
- *Project* - zawiera pliki wchodzące w skład projektu,
- *Console* - wyświetla informacje generowane przez Środowisko Unity.

Szablony układu okien można zapisywać i zmieniać przy pomocy przycisku *Layout* w prawym górnym rogu edytora.

Składniki

są elementami funkcjonalności gry, dołączanymi do obiektu *GameObject*. W silniku Unity występuje wiele różnych składników, spośród których składnik *Transform* (określa położenie, obrót i rozmiar) zawarty jest w każdym *GameObject*. Obiekt *Main Camera* domyślnie zawiera składnik *Camera*, rysujący scenę na ekranie gracza. Podobnie *Directional Light* zawiera składnik *Light*, do rzucania światła.

Składniki

są elementami funkcjonalności gry, dołączanymi do obiektu *GameObject*. W silniku Unity występuje wiele różnych składników, spośród których składnik *Transform* (określa położenie, obrót i rozmiar) zawarty jest w każdym *GameObject*. Obiekt *Main Camera* domyślnie zawiera składnik *Camera*, rysujący scenę na ekranie gracza. Podobnie *Directional Light* zawiera składnik *Light*, do rzucania światła.

Pod nagłówkiem składnika są różnego rodzaju pola, pozwalające na zmianę właściwości składnika.

Składnik można włączyć lub wyłączyć w *GameObject* za pomocą pola wyboru. Ikonka z symbolem ? po prawej stronie nagłówka każdego składnika łączy z odpowiednią dokumentacją.

Składniki

są elementami funkcjonalności gry, dołączanymi do obiektu *GameObject*. W silniku Unity występuje wiele różnych składników, spośród których składnik *Transform* (określa położenie, obrót i rozmiar) zawarty jest w każdym *GameObject*. Obiekt *Main Camera* domyślnie zawiera składnik *Camera*, rysujący scenę na ekranie gracza. Podobnie *Directional Light* zawiera składnik *Light*, do rzucania światła.

Pod nagłówkiem składnika są różnego rodzaju pola, pozwalające na zmianę właściwości składnika.

Składnik można włączyć lub wyłączyć w *GameObject* za pomocą pola wyboru. Ikonka z symbolem ? po prawej stronie nagłówka każdego składnika łączy z odpowiednią dokumentacją.

Kod źródłowy zapisany jako *Script*, również dołączany jest jako składnik *GameObject*. Można dołączać dowolną kombinację składników typu *Script* do danego *GameObject*, aby połączyć różne funkcje w danym obiekcie.

Inne składniki

Mesh Filter – (*mesh* – siatka, model trójwymiarowy) filtr siatki, zawiera siatkę obiektu, którą można dalej wyświetlić,

Mesh Renderer odtwarzacz siatki – pozwala na rysowanie modelu trójwymiarowego poprzez składnik *Camera*.

Elementy GameObject

Do modelowania, animacji i edycji obiektów trójwymiarowych i obrazów może posłużyć np. Blender: <https://www.blender.org>

Utworzone w ten sposób obiekty można zaimportować do Unity, zapisując w folderze projektu i następnie włączać do scen.

Elementy GameObject

Do modelowania, animacji i edycji obiektów trójwymiarowych i obrazów może posłużyć np. Blender: <https://www.blender.org>

Utworzone w ten sposób obiekty można zaimportować do Unity, zapisując w folderze projektu i następnie włączać do scen.

Obiekty w scenie można również dołączać wybierając w menu GameObject lub klikając prawym przyciskiem myszy w oknie *Hierarchy* i wybierając np. pozycję *3D Object*, dającej dostęp do obiektów o podstawowych kształtach.

Elementy GameObject

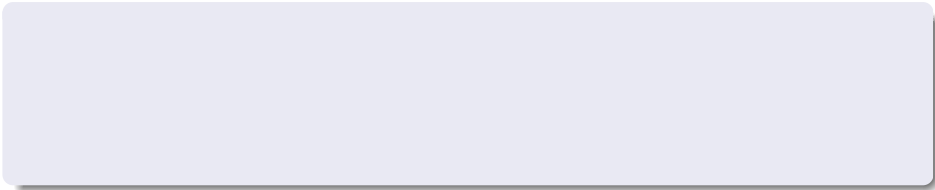
Do modelowania, animacji i edycji obiektów trójwymiarowych i obrazów może posłużyć np. Blender: <https://www.blender.org>

Utworzone w ten sposób obiekty można zaimportować do Unity, zapisując w folderze projektu i następnie włączać do scen.

Obiekty w scenie można również dołączać wybierając w menu GameObject lub klikając prawym przyciskiem myszy w oknie *Hierarchy* i wybierając np. pozycję *3D Object*, dającej dostęp do obiektów o podstawowych kształtach.

Dany obiekt na scenie można znaleźć np. poprzez dwukrotne kliknięcie jego nazwy w oknie Hierarchy lub po zaznaczeniu w oknie Hierarchy i umieszczeniu kursora myszy nad oknem Scene oraz przyciśnięcie klawisza F.

Nawigacja w oknie *Scene*



Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

podczas wciśniętego prawego przycisku myszy:

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

podczas wciśniętego prawego przycisku myszy:

- *obracanie się* - poprzez ruch myszy,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

podczas wciśniętego prawego przycisku myszy:

- *obracanie się* - poprzez ruch myszy,
- *do przodu* - poprzez wciśnięty klawisz *W*,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

podczas wciśniętego prawego przycisku myszy:

- *obracanie się* - poprzez ruch myszy,
- *do przodu* - poprzez wciśnięty klawisz *W*,
- *do tyłu* - klawisz *S*,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

podczas wciśniętego prawego przycisku myszy:

- *obracanie się* - poprzez ruch myszy,
- *do przodu* - poprzez wciśnięty klawisz *W*,
- *do tyłu* - klawisz *S*,
- *w lewo* - klawisz *A*,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

podczas wciśniętego prawego przycisku myszy:

- *obracanie się* - poprzez ruch myszy,
- *do przodu* - poprzez wciśnięty klawisz *W*,
- *do tyłu* - klawisz *S*,
- *w lewo* - klawisz *A*,
- *w prawo* - klawisz *D*,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

podczas wciśniętego prawego przycisku myszy:

- *obracanie się* - poprzez ruch myszy,
- *do przodu* - poprzez wciśnięty klawisz *W*,
- *do tyłu* - klawisz *S*,
- *w lewo* - klawisz *A*,
- *w prawo* - klawisz *D*,
- *w dół* - klawisz *Q*,

Nawigacja w oknie *Scene*

- *do przodu* - strzałka w górę lub kółko myszy do przodu,
- *do tyłu* - strzałka w dół lub kółko myszy do tyłu,
- *w lewo* - strzałka w lewo,
- *w prawo* - strzałka w prawo.

podczas wciśniętego prawego przycisku myszy:

- *obracanie się* - poprzez ruch myszy,
- *do przodu* - poprzez wciśnięty klawisz *W*,
- *do tyłu* - klawisz *S*,
- *w lewo* - klawisz *A*,
- *w prawo* - klawisz *D*,
- *w dół* - klawisz *Q*,
- *w górę* - klawisz *E*.

Przesuwanie sceny środkowym przyciskiem myszy albo lewym z użyciem narzędzia *View Tool*, aktywowanego przyciskiem na pasku narzędziowym lub klawiszem *Q*. Z jednocześnie przyciśniętym klawiszem *Shift* przesuwanie następuje szybciej.

Prędkość przesuwania klawiszami można regulować kółkiem myszy z jednocześnie wciśniętym jej prawym przyciskiem.

Przesuwanie sceny środkowym przyciskiem myszy albo lewym z użyciem narzędzia *View Tool*, aktywowanego przyciskiem na pasku narzędziowym lub klawiszem *Q*. Z jednocześnie przyciśniętym klawiszem *Shift* przesuwanie następuje szybciej.

Prędkość przesuwania klawiszami można regulować kółkiem myszy z jednocześnie wciśniętym jej prawym przyciskiem.

podczas wciśniętego lewego klawisza *Alt* na klawiaturze:

Przesuwanie sceny środkowym przyciskiem myszy albo lewym z użyciem narzędzia *View Tool*, aktywowanego przyciskiem na pasku narzędziowym lub klawiszem *Q*. Z jednocześnie przyciśniętym klawiszem *Shift* przesuwanie następuje szybciej.

Prędkość przesuwania klawiszami można regulować kółkiem myszy z jednocześnie wciśniętym jej prawym przyciskiem.

podczas wciśniętego lewego klawisza *Alt* na klawiaturze:

- *obracanie sceną wokół punktu obrotu wybranego obiektu* - poprzez ruch myszy z wciśniętym lewym przyciskiem,

Przesuwanie sceny środkowym przyciskiem myszy albo lewym z użyciem narzędzia *View Tool*, aktywowanego przyciskiem na pasku narzędziowym lub klawiszem *Q*. Z jednocześnie przyciśniętym klawiszem Shift przesuwanie następuje szybciej.

Prędkość przesuwania klawiszami można regulować kółkiem myszy z jednocześnie wciśniętym jej prawym przyciskiem.

podczas wciśniętego lewego klawisza Alt na klawiaturze:

- *obracanie sceną wokół punktu obrotu wybranego obiektu* - poprzez ruch myszy z wciśniętym lewym przyciskiem,
- *do przodu* - ruch myszy w prawo lub w dół z wciśniętym prawym przyciskiem,

Przesuwanie sceny środkowym przyciskiem myszy albo lewym z użyciem narzędzia *View Tool*, aktywowanego przyciskiem na pasku narzędziowym lub klawiszem *Q*. Z jednocześnie przyciśniętym klawiszem *Shift* przesuwanie następuje szybciej.

Prędkość przesuwania klawiszami można regulować kółkiem myszy z jednocześnie wciśniętym jej prawym przyciskiem.

podczas wciśniętego lewego klawisza *Alt* na klawiaturze:

- *obracanie sceną wokół punktu obrotu wybranego obiektu* - poprzez ruch myszy z wciśniętym lewym przyciskiem,
- *do przodu* - ruch myszy w prawo lub w dół z wciśniętym prawym przyciskiem,
- *do tyłu* - ruch myszy w lewo lub w górę z wciśniętym prawym przyciskiem.

Centrum kontroli widoku Sceny "gizmo"

znajduje się w prawym górnym rogu widoku okna Sceny. Odzwierciedla bieżącą orientację widoku na Scenę oraz określa tryb jego rzutowania. Składa się z sześciangu, z którego wychodzą trzy stożkowe ramiona opisane przez X, Y, Z. Wybranie danego ramienia powoduje natychmiastowe przełączenie widoku Sceny. Przyciśnięcie prawym przyciskiem myszy sześciangu, powoduje wyświetlenie listy wyboru kątów widoku Sceny oraz możliwość przejścia z powrotem do domyślnego widoku poprzez wybór opcji "Free".

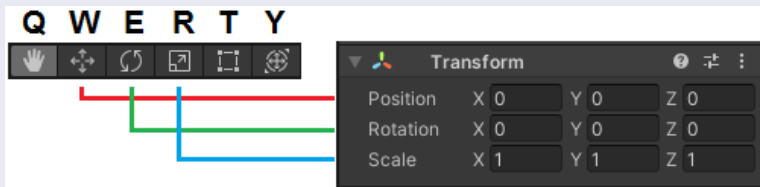
Centrum kontroli widoku Sceny "gizmo"

znajduje się w prawym górnym rogu widoku okna Sceny. Odzwierciedla bieżącą orientację widoku na Scenę oraz określa tryb jego rzutowania. Składa się z sześciangu, z którego wychodzą trzy stożkowe ramiona opisane przez X, Y, Z. Wybranie danego ramienia powoduje natychmiastowe przełączenie widoku Sceny. Przyciśnięcie prawym przyciskiem myszy sześciangu, powoduje wyświetlenie listy wyboru kątów widoku Sceny oraz możliwość przejścia z powrotem do domyślnego widoku poprzez wybór opcji "Free".

Przyciśnięcie tekstu "Persp" lub "Iso" poniżej kontrolki "gizmo", odzwierciedla bieżący tryb rzutowania widoku. Jego przyciśnięcie powoduje przejście do trybu widoku prostopadłego lub perspektywicznego.

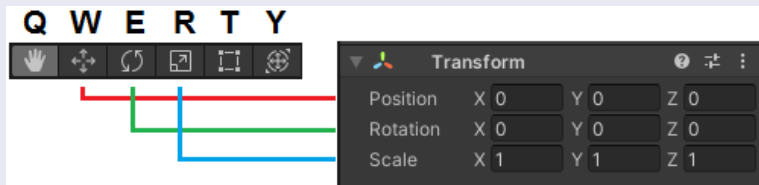
Klawiaturowe narzędzia przekształcania obiektów

powiązane są z odpowiednimi klawiszami oraz ikonami narzędzi (Tool):
View, Move, Rotate, Scale, Rect i Transform.



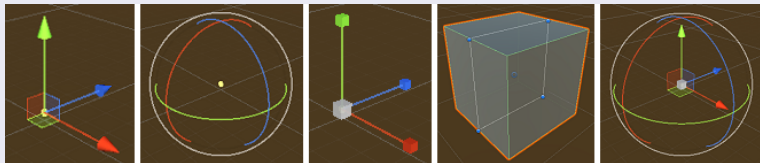
Klawiaturowe narzędzia przekształcania obiektów

powiązane są z odpowiednimi klawiszami oraz ikonami narzędzi (Tool):
View, Move, Rotate, Scale, Rect i Transform.



Narzędzia przekształcania obiektów "Gizmo"

umożliwiają za pomocą myszki dokonywać zmiany w składniku *Transform*:



Przytrzymanie przycisku myszy pomiędzy liniami tych narzędzi pozwala na jednoczesne operacje na tych liniach.

Przekształcanie przy wciśniętym klawiszu Ctrl pozwala na zmiany o ustalone wartości.

Przytrzymanie przycisku myszy pomiędzy liniami tych narzędzi pozwala na jednoczesne operacje na tych liniach.

Przekształcanie przy wciśniętym klawiszu Ctrl pozwala na zmiany o ustalone wartości.

Przekształceń można dokonać wpisując odpowiednie wartości w składniku *Transform* lub przesuwając myszką z wciśniętym lewym przyciskiem nad odpowiednimi współrzędnymi XYZ.

- *współrzędna X* – oś czerwona - określa położenie na prawo i lewo
- *współrzędna Y* – oś zielona - w górę i w dół
- *współrzędna Z* – oś niebieska - w przód i w tył

Hierarchia obiektów

Elementy GameObject można jako podrzędne dołączyć do jednego nadrzędnego. Można je wtedy przenosić, obracać i skalować razem z GameObject nadrzędnym. Stanowią one "bryłę sztywną".
Zmiana rozmiarów GameObject nadrzędnego powoduje proporcjonalne zmiany odległości całej tej hierarchii.

Hierarchia obiektów

Elementy GameObject można jako podrzędne dołączyć do jednego nadrzędnego. Można je wtedy przenosić, obracać i skalować razem z GameObject nadrzędnym. Stanowią one "bryłę sztywną". Zmiana rozmiarów GameObject nadrzędnego powoduje proporcjonalne zmiany odległości całej tej hierarchii.

Punkt obrotu

jest punktem odniesienia GameObject jeśli chodzi o położenie i obrót. Do ustawienia *punktu obrotu* przydatny jest pusty GameObject. Dla GameObject nadrzędnych można wybrać położenie gadżetu narzędzia Gizmos na **Center** lub **Pivot** przy pomocy przycisku "Toggle Tool Handle Position" lub klawisza "Z". *Center* ustawia położenie Gizmos w środkowym punkcie składników hierarchii, a *Pivot* w środkowym punkcie GameObject nadrzędnego.

Dla GameObject podrzędnych rozróżniamy *położenie w świecie* i *położenie lokalne* względem GameObject nadrzędnego.

Współrzędne GameObject podrzędnych w składniku Transform określają ich położenie względem nadrzędnego GameObject. Zmiana współrzędnych *Transform* obiektów podrzędnych w hierarchii nie wpływa na współrzędne nadrzędnego. Skala i położenie w świecie obiektów podrzędnych wynika z iloczynu skali wszystkich kolejnych nadrzędnych.



Prefabrykaty

Prefabrykaty są aktywem projektu działającym jak do szablonu dla obiektów. Po przygotowaniu prefabrykatu, można tworzyć jego wystąpienia w dowolnych scenach.

Jego zmiana automatycznie wpływa na wszystkie jego wystąpienia.

Prefabrykaty

Prefabrykaty są aktywnym projektem działającym jak do szablonu dla obiektów. Po przygotowaniu prefabrykatu, można tworzyć jego wystąpienia w dowolnych scenach.

Jego zmiana automatycznie wpływa na wszystkie jego wystąpienia.

Nowy prefabrykat

można utworzyć z `GameObject` poprzez przeciągnięcie go z panelu Hierarchy do panelu Project.

Nastąpi automatyczne skojarzenie `GameObject` z prefabrykatem.

Aby umieścić na scenie kolejne wystąpienia prefabrykatu, przeciągamy prefabrykat na panel Hierarchy.

Edycja prefabrykatu

dostępna jest po podwójnym kliknięciu jego nazwy w oknie Project lub dla zaznaczonego prefabrykatu kliknięciu przycisku *Open* w panelu Inspector. Na scenie pojawi się jako jedyny element wystąpienie prefabrykatu. Podobnie w panelu Hierarchy pojawią się jako jedyne składniki edytowanego prefabrykatu.

Edycja prefabrykatu

dostępna jest po podwójnym kliknięciu jego nazwy w oknie Project lub dla zaznaczonego prefabrykatu kliknięciu przycisku *Open* w panelu Inspector. Na scenie pojawi się jako jedyny element wystąpienie prefabrykatu. Podobnie w panelu Hierarchy pojawią się jako jedyne składniki edytowanego prefabrykatu.

Edytujemy zmieniając w tym trybie wybrany prefabrykat. Wychodzimy z trybu edycji klikając w przycisk "<" w panelu Hierarchy lub przycisk *Scenes* obok nazwy prefabrykatu na samej górze okna Scene.

Przedefiniowywanie wartości

następuje gdy dokonujemy zmian na wystąpieniu prefabrykatu na scenie. Tak wprowadzone zmiany zostają zachowane podczas zmian w oryginalnym prefabrykacie.

Nazwy i wartości przedefiniowanych składników są wyróżnione np. przez pogrubienie. Położenie i obrócenie składnika nadrzędnego dla wszystkich innych wystąpienia prefabrykatu jest domyślnie traktowane jako przedefiniowane.

Przedefiniowywanie wartości

następuje gdy dokonujemy zmian na wystąpieniu prefabrykatu na scenie. Tak wprowadzone zmiany zostają zachowane podczas zmian w oryginalnym prefabrykacie.

Nazwy i wartości przedefiniowanych składników są wyróżnione np. przez pogrubienie. Położenie i obrócenie składnika nadrzędnego dla wszystkich innych wystąpienia prefabrykatu jest domyślnie traktowane jako przedefiniowane.

Zmiany w wystąpieniu prefabrykatu

można zapisywać na 4 sposoby:

Przedefiniowywanie wartości

następuje gdy dokonujemy zmian na wystąpieniu prefabrykatu na scenie. Tak wprowadzone zmiany zostają zachowane podczas zmian w oryginalnym prefabrykacie.

Nazwy i wartości przedefiniowanych składników są wyróżnione np. przez pogrubienie. Położenie i obrócenie składnika nadrzędnego dla wszystkich innych wystąpienia prefabrykatu jest domyślnie traktowane jako przedefiniowane.

Zmiany w wystąpieniu prefabrykatu

można zapisywać na 4 sposoby:

- *w bieżącym wystąpieniu* - dla wybranego obiektu

Przedefiniowywanie wartości

następuje gdy dokonujemy zmian na wystąpieniu prefabrykatu na scenie. Tak wprowadzone zmiany zostają zachowane podczas zmian w oryginalnym prefabrykacie.

Nazwy i wartości przeddefiniowanych składników są wyróżnione np. przez pogrubienie. Położenie i obrócenie składnika nadrzędnego dla wszystkich innych wystąpienia prefabrykatu jest domyślnie traktowane jako przeddefiniowane.

Zmiany w wystąpieniu prefabrykatu

można zapisywać na 4 sposoby:

- *w bieżącym wystąpieniu* - dla wybranego obiektu
- *w oryginalnym prefabrykacie* - przesuając to wystąpienie z panelu Hierarchy na edytowany prefabrykat w panelu Project

Przedefiniowywanie wartości

następuje gdy dokonujemy zmian na wystąpieniu prefabrykatu na scenie. Tak wprowadzone zmiany zostają zachowane podczas zmian w oryginalnym prefabrykacie.

Nazwy i wartości przedefiniowanych składników są wyróżnione np. przez pogrubienie. Położenie i obrócenie składnika nadrzędnego dla wszystkich innych wystąpienia prefabrykatu jest domyślnie traktowane jako przedefiniowane.

Zmiany w wystąpieniu prefabrykatu

można zapisywać na 4 sposoby:

- *w bieżącym wystąpieniu* - dla wybranego obiektu
- *w oryginalnym prefabrykacie* - przesuając to wystąpienie z panelu Hierarchy na edytowany prefabrykat w panelu Project
- *jako nowy prefabrykat* - przesuując to wystąpienie z panelu Hierarchy na panel Project i zatwierdzając wybór

Przedefiniowywanie wartości

następuje gdy dokonujemy zmian na wystąpieniu prefabrykatu na scenie. Tak wprowadzone zmiany zostają zachowane podczas zmian w oryginalnym prefabrykacie.

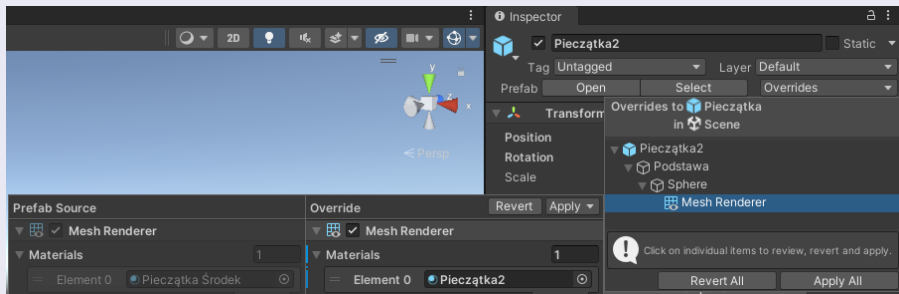
Nazwy i wartości przedefiniowanych składników są wyróżnione np. przez pogrubienie. Położenie i obrócenie składnika nadrzędnego dla wszystkich innych wystąpienia prefabrykatu jest domyślnie traktowane jako przedefiniowane.

Zmiany w wystąpieniu prefabrykatu

można zapisywać na 4 sposoby:

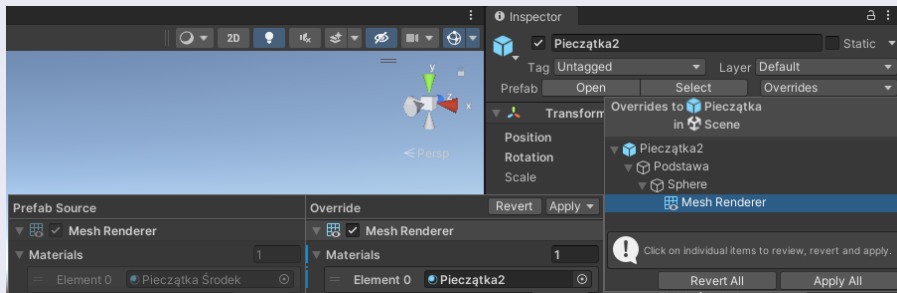
- *w bieżącym wystąpieniu* - dla wybranego obiektu
- *w oryginalnym prefabrykacie* - przesuając to wystąpienie z panelu Hierarchy na edytowany prefabrykat w panelu Project
- *jako nowy prefabrykat* - przesuując to wystąpienie z panelu Hierarchy na panel Project i zatwierdzając wybór
- *wariant prefabrykatu* - podobnie jak powyżej

Po zaznaczeniu głównego obiektu wystąpienia prefabrykatu, w panelu Inspector pojawią się przyciski dotyczące zastosowanych przeddefiniowań:



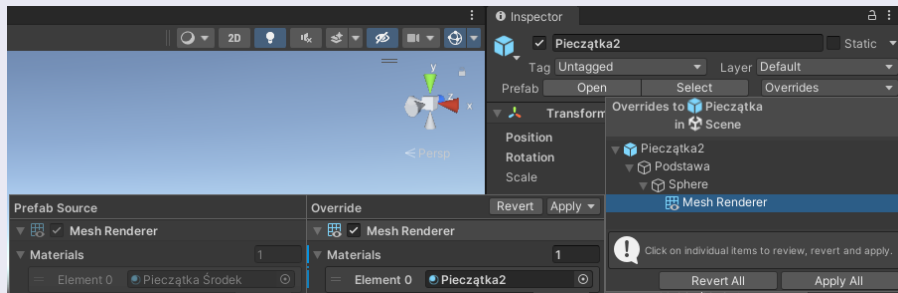
Po zaznaczeniu głównego obiektu wystąpienia prefabrykatu, w panelu Inspector pojawią się przyciski dotyczące zastosowanych przeddefiniowań:

- *Open* - otworzy powiązany prefabrykat do edycji



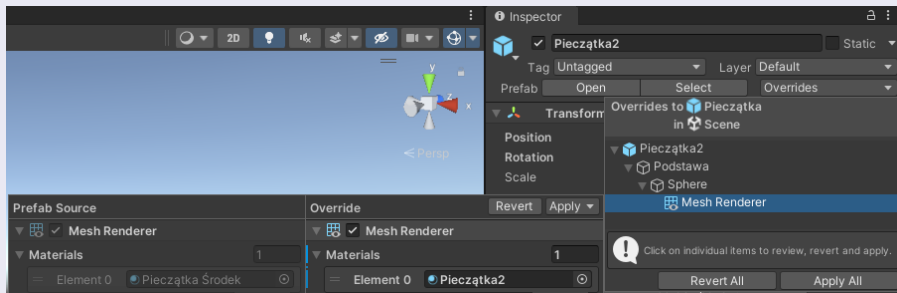
Po zaznaczeniu głównego obiektu wystąpienia prefabrykatu, w panelu Inspector pojawią się przyciski dotyczące zastosowanych przeddefiniowań:

- *Open* - otworzy powiązany prefabrykat do edycji
- *Select* - zaznaczy prefabrykat w panelu Project



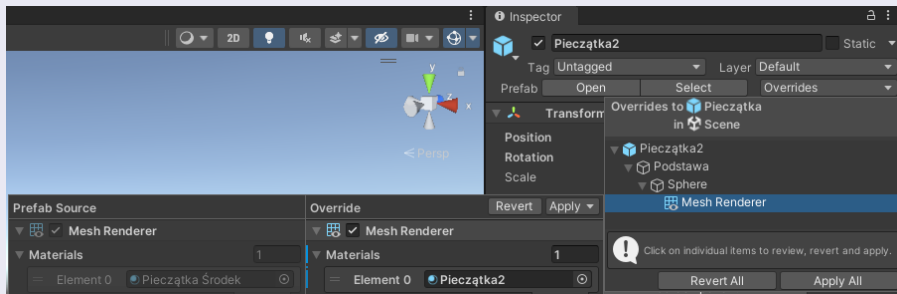
Po zaznaczeniu głównego obiektu wystąpienia prefabrykatu, w panelu Inspector pojawią się przyciski dotyczące zastosowanych przeddefiniowań:

- *Open* - otworzy powiązany prefabrykat do edycji
- *Select* - zaznaczy prefabrykat w panelu Project
- *Overrides* - pokaże hierarchię elementów prefabrykatu z zaznaczeniem zmian



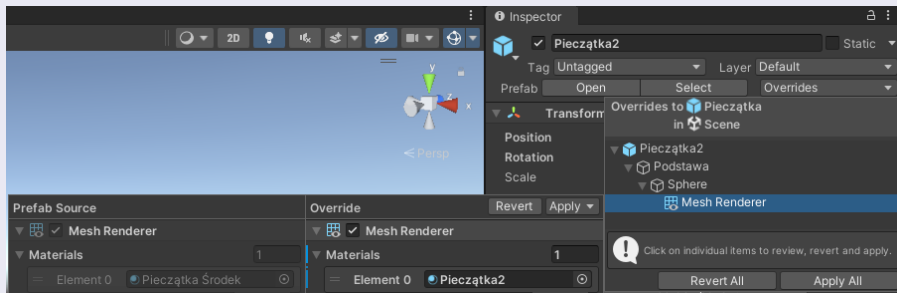
Po zaznaczeniu głównego obiektu wystąpienia prefabrykatu, w panelu Inspector pojawią się przyciski dotyczące zastosowanych przeddefiniowań:

- *Open* - otworzy powiązany prefabrykat do edycji
- *Select* - zaznaczy prefabrykat w panelu Project
- *Overrides* - pokaże hierarchię elementów prefabrykatu z zaznaczeniem zmian
- *Revert All* - odwraca wszystkie przeddefiniowania w danym wystąpieniu



Po zaznaczeniu głównego obiektu wystąpienia prefabrykatu, w panelu Inspector pojawią się przyciski dotyczące zastosowanych przeddefiniowań:

- *Open* - otworzy powiązany prefabrykat do edycji
- *Select* - zaznaczy prefabrykat w panelu Project
- *Overrides* - pokaże hierarchię elementów prefabrykatu z zaznaczeniem zmian
- *Revert All* - odwraca wszystkie przeddefiniowania w danym wystąpieniu
- *Apply All* - zastosują wszystkie przeddefiniowania do prefabrykatu



W przypadku zaznaczenia składnika hierarchii zawierającego przedefiniowania, wyświetlają się dodatkowe dwa okna "Prefab Source" i "Override". W **Override** można edytować wybrany składnik i resetować przyciskiem "Revert" lub zastosować zmiany "Apply".
W ramach przedefiniowania elementu GameObject nie można usunąć wystąpienia prefabrykatu. Można go wyłączyć przyciskiem CheckBox.

W przypadku zaznaczenia składnika hierarchii zawierającego przedefiniowania, wyświetlają się dodatkowe dwa okna "Prefab Source" i "Override". W **Override** można edytować wybrany składnik i resetować przyciskiem "Revert" lub zastosować zmiany "Apply". W ramach przedefiniowania elementu GameObject nie można usunąć wystąpienia prefabrykatu. Można go wyłączyć przyciskiem CheckBox.

Zagnieżdzone prefabrykaty

są częścią hierarchii innych prefabrykatów. Zmiany w zagnieżdżonym prefabrykacie automatycznie są stosowane w jego zagnieżdżonych wystąpieniach.

Wariant prefabrykatu

przedefiniowuje prefabrykat bazowy. Przedefiniowane elementy wariantu odporne są na zmiany w prefabrykacie bazowym.

Wariant prefabrykatu

przedefiniowuje prefabrykat bazowy. Przedefiniowane elementy wariantu odporne są na zmiany w prefabrykacie bazowym.

Wariant prefabrykatu można utworzyć klikając prawym przyciskiem myszy prefabrykat w panelu Project i wybierając Create → Prefab Variant
Można tworzyć warianty dla których bazą jest inny wariant prefabrykatu.

Skrypty

Skrypty są plikami o rozszerzeniu *.cs zawierającymi kod w języku C#.

Skrypty

Skrypty są plikami o rozszerzeniu *.cs zawierającymi kod w języku C#.

Po jego zapisaniu na dysku środowisko Unity automatycznie go kompiluje. Panel *Console* wypisuje ewentualną listę błędów kompilacji.

Panel Project

zawiera pliki wchodzące w skład projektu. Katalog **Assets** czyli aktywa zawiera składowe naszego projektu oraz katalog **Scenes** i w nim znajduje się plik *SampleScene*, definiujący podstawową scenę.

Panel Project

zawiera pliki wchodzące w skład projektu. Katalog **Assets** czyli aktywa zawiera składowe naszego projektu oraz katalog **Scenes** i w nim znajduje się plik *SampleScene*, definiujący podstawową scenę.

Skrypt

możemy utworzyć w folderze *Assets* w panelu *Project*, klikając prawym przyciskiem myszy na wewnętrznym polu katalogu i wybierając z listy: Create → C# Script. Po utworzeniu wpisujemy nazwę. Podgląd jego zawartości widoczny jest w oknie *Inspector*. Poprzez dwukrotne kliknięcie Skrypt jest otwierany w skojarzonym edytorze.

Kod Skryptu

zaczyna się od dołączenia trzech przestrzeni nazw:

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;
```

UnityEngine zawiera klasy definiujące główne składniki obiektów.

Kod Skryptu

zaczyna się od dołączenia trzech przestrzeni nazw:

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;
```

UnityEngine zawiera klasy definiujące główne składniki obiektów.

Klasa o nazwie zgodnej z nazwą skryptu

dziedziczy po klasie *MonoBehaviour* i domyślnie zawiera dwie funkcje:

- *Start()* – wykonuje się jeden raz przed pierwszym wyświetleniem klatki
- *Update()* – wykonywana jest jeden raz z każdą klatką

Kod Skryptu

zaczyna się od dołączenia trzech przestrzeni nazw:

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;
```

UnityEngine zawiera klasy definiujące główne składniki obiektów.

Klasa o nazwie zgodnej z nazwą skryptu

dziedziczy po klasie *MonoBehaviour* i domyślnie zawiera dwie funkcje:

- *Start()* – wykonuje się jeden raz przed pierwszym wyświetleniem klatki
- *Update()* – wykonywana jest jeden raz z każdą klatką

Funkcja *Awake()* wywoływana jest od razu po zainicjowaniu skryptu, nawet gdy jest on wyłączony. Funkcja *Start()* wywoływana jest zawsze po *Awake()* przy pierwszej klatce włączonego skryptu.

Skrypt wykonywany jest poprzez powiązanie go z danym obiektem gry. Powiązać można np. poprzez przeciągnięcie go na dany obiekt lub przeciągnięcie na okno *Inspector* wybranego obiektu. Można go też dodać jako składnik do obiektu, poprzez kliknięcie przycisku **Add Component** na oknie *Inspector* i wyszukanie go z listy dostępnych składników. Wtedy po zaznaczeniu obiektu, w oknie *Inspector* znajdzie się na liście składników wybranego obiektu.

Skrypt wykonywany jest poprzez powiązanie go z danym obiektem gry. Powiązać można np. poprzez przeciągnięcie go na dany obiekt lub przeciągnięcie na okno *Inspector* wybranego obiektu. Można go też dodać jako składnik do obiektu, poprzez kliknięcie przycisku **Add Component** na oknie *Inspector* i wyszukanie go z listy dostępnych składników. Wtedy po zaznaczeniu obiektu, w oknie *Inspector* znajdzie się na liście składników wybranego obiektu.

Składnik klasy z trybem dostępu *public* jest dostępny w oknie *Inspector* danego *GameObject*, aby dostosowywać jego wartości. Można edytować jego wartości podczas odtwarzania gry, testując różne ustawienia. Zmiany wpisane podczas odtwarzania gry nie są zachowywane. Zdefiniowany typ *Vector3* daje możliwość wpisania trzech wartości liczbowych.

Niektóre funkcje skryptów

- `Debug.Log("komunikat")` – wypisuje komunikaty na konsoli
- `Input.GetKeyDown(KeyCode.RightArrow)` – zwraca `true` tylko raz po naciśnięciu klawisza
- `Transform.Rotate(wektor * Time.deltaTime);` – obraca obiekt o podaną ilość stopni wokół odpowiednich osi. `deltaTime` to czas w sekundach, który upłynął od poprzedniej klatki.

Niektóre funkcje skryptów

- `Debug.Log("komunikat")` – wypisuje komunikaty na konsoli
- `Input.GetKeyDown(KeyCode.RightArrow)` – zwraca `true` tylko raz po naciśnięciu klawisza
- `Transform.Rotate(wektor * Time.deltaTime);` – obraca obiekt o podaną ilość stopni wokół odpowiednich osi. `deltaTime` to czas w sekundach, który upłynął od poprzedniej klatki.

Atrybuty związane z parametrami w oknie Inspector

wpisujemy w nawiasach prostokątnych przed definicją zmiennej:

- `[HideInInspector]` – ukrywa publiczną zmienną w oknie Inspector, umożliwiając jednocześnie innym skryptom do niej dostęp
- `[Header("Parametry:")]` – umieszcza pogrubiony nagłówek ponad zmienną w oknie Inspector
- `[Tooltip("Wskazówka")]` – ukazuje się po najechaniu myszką na zmienną w oknie Inspector

Deklaracja zmiennej typu składnika obiektu

ustawiana jest poprzez panel Inspector. Nie potrzeba wtedy kodu do odnalezienia tego składnika.

Deklaracja zmiennej typu składnika obiektu

ustawiana jest poprzez panel Inspector. Nie potrzeba wtedy kodu do odnalezienia tego składnika.

Przykładem składnika może być *Transform* czy *CharacterController*, zaprojektowany do zapewnienia postaci możliwości poruszania się i wykrywania kolizji.

Deklaracja zmiennej typu składnika obiektu

ustawiana jest poprzez panel Inspector. Nie potrzeba wtedy kodu do odnalezienia tego składnika.

Przykładem składnika może być *Transform* czy *CharacterController*, zaprojektowany do zapewnienia postaci możliwości poruszania się i wykrywania kolizji.

Odwołania do składników skryptu

tworzymy ustawiając składnik w danym polu poprzez:

- kliknięcie w panelu Inspector na małą okrągłą ikonę po prawej stronie pola zmiennej i wybranie `GameObject`, posiadający ten składnik
- przeciągnięcie składnika w panelu Inspector na dane pole tego panelu
- przeciągnięcie z panelu Hierarchy `GameObject` na dane pole panelu Inspector

Wyłączenie GameObject

zatrzymuje działanie wszystkich jego składników oraz obiektów podrzędnych. Aktywacja obiektu nadrzędnego nie powoduje aktywacji niezależnie dezaktywowanego obiektu podrzędnego. Nazwy wyłączonych obiektów w panelu Hierarchy wyświetlane są szarym tekstem.

Wyłączenie GameObject

zatrzymuje działanie wszystkich jego składników oraz obiektów podrzędnych. Aktywacja obiektu nadrzędnego nie powoduje aktywacji niezależnie dezaktywowanego obiektu podrzędnego. Nazwy wyłączonych obiektów w panelu Hierarchy wyświetlane są szarym tekstem.

Stan aktywacji GameObject opisują zmienne:

- *activeSelf* – indywidualny stan GameObject. Jest on aktywny gdy również wszystkie nadrzędne przyjmują wartość *true*
- *active* – przyjmuje wartość *true* gdy elementy nadrzędne GameObject i sam obiekt są aktywne

Wykrywanie kolizji

Z kolizjami w Unity związane są 3 pojęcia: bryły ograniczające, bryły sztywne i warstwy.

Wykrywanie kolizji

Z kolizjami w Unity związane są 3 pojęcia: bryły ograniczające, bryły sztywne i warstwy.

Bryły ograniczające (Colliders)

są składnikami GameObject o niewidocznym w czasie gry kształcie. Oddziałują z innymi bryłami ograniczającymi na scenie.

Gdy jest oznaczona w polu wyboru jako Trigger, nie działa jak obiekt fizyczny i inne obiekty mogą przechodzić. Jednak wysyłają komunikaty zdarzeń.

Bryła sztywna (Rigidbody)

Dodaje fizyczne zachowania do GameObject. Kolizje przedstawiane są z użyciem praw fizyki. Dodaje grawitację do obiektu. Umożliwia dodanie siły i momentu obrotowego przez skrypty.

Składnik *Rigidbody* potrzebuje do wykrywania kolizji *bryły ograniczającej*, definiującej kształt obiektu na potrzeby silnika fizycznego.

Bryła sztywna (Rigidbody)

dodaje fizyczne zachowania do `GameObject`. Kolizje przedstawiane są z użyciem praw fizyki. Dodaje grawitację do obiektu. Umożliwia dodanie siły i momentu obrotowego przez skrypty.

Składnik *Rigidbody* potrzebuje do wykrywania kolizji *bryły ograniczającej*, definiującej kształt obiektu na potrzeby silnika fizycznego.

Bezpośrednia zmiana składnika *Transform* (ruch kinematyczny) z pominięciem istniejącego *Rigidbody*, może powodować niepożądane efekty. Silnik Unity sam symuluje prawa fizyki.

Poruszamy się wtedy dodając siłę i moment obrotowy poprzez odwołanie do składnika *Rigidbody*.

Jeśli obiekt jest *kinematyczny*, poruszamy go poprzez składnik *Transform*, jednak nadal uwzględniane są kolizje.

Korzystamy z Rigidbody

gdy obiekt porusza się. Gdy jest nieruchomy, to nie korzystamy.
Gdy reagujemy na kolizje pomiędzy dwoma obiektami, przynajmniej jeden z nich musi posiadać składnik *Rigidbody*.

Korzystamy z Rigidbody

gdy obiekt porusza się. Gdy jest nieruchomy, to nie korzystamy. Gdy reagujemy na kolizje pomiędzy dwoma obiektami, przynajmniej jeden z nich musi posiadać składnik *Rigidbody*.

Oznaczamy Rigidbody jako kinematyczny

gdy obiekt jest poruszany przez skrypt lub animacje. Gdy porusza się jak obiekt fizyczny poprzez siły i moment obrotowy, nie oznaczamy go jako kinematyczny.

Korzystamy z Rigidbody

gdy obiekt porusza się. Gdy jest nieruchomy, to nie korzystamy. Gdy reagujemy na kolizje pomiędzy dwoma obiektami, przynajmniej jeden z nich musi posiadać składnik *Rigidbody*.

Oznaczamy Rigidbody jako kinematyczny

gdy obiekt jest poruszany przez skrypt lub animacje. Gdy porusza się jak obiekt fizyczny poprzez siły i moment obrotowy, nie oznaczamy go jako kinematyczny.

Składnik CharacterController

zachowuje się jak *Rigidbody*.

Warstwy

są zbiorami GameObject które można wybiórczo spośród innych warstw renderować lub umożliwiać czy wykluczać kolizje z obiektami wybranych warstw.

Warstwy

są zbiorami `GameObject` które można wybiórczo spośród innych warstw renderować lub umożliwiać czy wykluczać kolizje z obiektami wybranych warstw.

Każdy element `GameObject` jest przyporządkowany do jednej z warstw. Do nowych elementów `GameObject` przypisywana jest warstwa *Default*. Warstwę (*Layer*) można wybrać w panelu Inspector nagłówka `GameObject` lub dodać własną wybierając *Add Layer...*

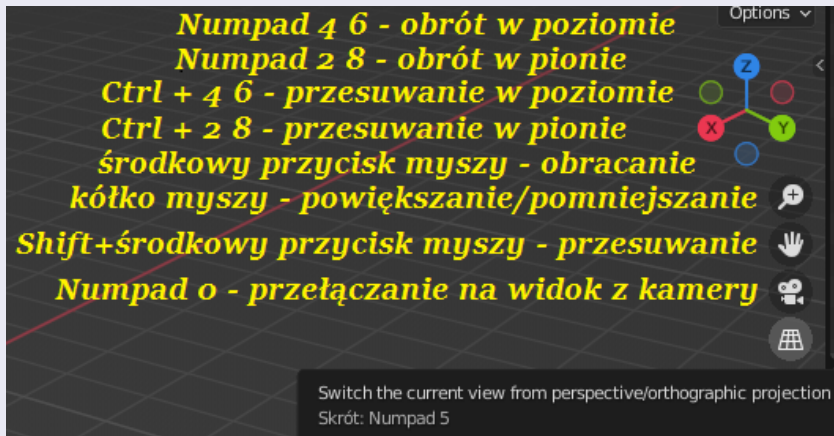
Do zmiany występowania kolizji pomiędzy obiektami poszczególnych warstw służy *Layer Collision Matrix*. Znajdziemy tę macierz u dołu panelu *Project Settings* po wyborze z lewej strony *Physics*. Panel ten wybieramy z menu `Edit -> Project Settings`.

Układ programu graficznego Blender

podzielony jest na panele. Ich zawartość można zmieniać przy pomocy rozwinięcia ikonki *Typ edytora* umieszczonej w lewym górnym rogu każdego panelu. Rozmiar paneli można zmieniać poprzez przesuwanie myszką łączących ich krawędzi. Kliknięcie prawym przyciskiem myszki na wybranej krawędzi rozwija listę opcji dotyczących podziału, zamiany i łączenia. Dla poszczególnych etapów tworzenia grafik przygotowane są zestawy układów paneli w zakładkach przy górnym menu.

Nawigacja w panelu 3D Viewport

opiera się na środkowym przycisku myszki, klawiaturze numerycznej oraz ikonkach pokazanych poniżej:



The image shows a screenshot of a 3D Viewport navigation controls panel. The panel has a dark background with a grid pattern. On the right side, there are several icons: a 3D coordinate system with X (red), Y (green), and Z (blue) axes; a plus sign in a speech bubble; a hand icon; a camera icon; and a grid icon. Below these icons, there is a text box that reads: "Switch the current view from perspective/orthographic projection" and "Skrót: Numpad 5".

Numpad 4 6 - obrót w poziomie
Numpad 2 8 - obrót w pionie
Ctrl + 4 6 - przesuwanie w poziomie
Ctrl + 2 8 - przesuwanie w pionie
środkowy przycisk myszy - obracanie
kółko myszy - powiększanie/pomniejszanie
Shift+środkowy przycisk myszy - przesuwanie
Numpad 0 - przełączanie na widok z kamery

Options ▾

Switch the current view from perspective/orthographic projection
Skrót: Numpad 5

Przekształcenia obiektów

w panelu 3D Viewport można wykonywać przy pomocy okna po prawej stronie *Transformuj* wpisując wartości numeryczne – wywoływane przy pomocy klawisza N. Graficzne narzędzia przekształceń uruchamiamy przyciskiem T.

Nowe obiekty możemy dodawać np. w wewnętrznym menu panelu 3D Viewport, wybierając *Add*.

Przekształcenia obiektów

w panelu 3D Viewport można wykonywać przy pomocy okna po prawej stronie *Transformuj* wpisując wartości numeryczne – wywoływane przy pomocy klawisza N. Graficzne narzędzia przekształceń uruchamiamy przyciskiem T.

Nowe obiekty możemy dodawać np. w wewnętrznym menu panelu 3D Viewport, wybierając *Add*.

4 tryby wyświetlania obiektów

można wybrać w prawym górnym rogu panelu 3D Viewport:

- *wire edges* – widoczne są tylko krawędzie obiektów
- *solid* – zewnętrzne kształty obiektów (tryb domyślny)
- *material preview* – do pracy z materiałami
- *render preview* – wygląd po ukończeniu pracy

Materiały

są widoczne w obiektach wyświetlanych w trybie material lub render. W panelu *Właściwości* obiektu druga od dołu zakładka dotyczy właściwości materiału. Można tu dodać materiał i zmieniać jego właściwości.

Materiały

są widoczne w obiektach wyświetlanych w trybie material lub render. W panelu *Właściwości* obiektu druga od dołu zakładka dotyczy właściwości materiału. Można tu dodać materiał i zmieniać jego właściwości.

Właściwość *Base Color* umożliwia dobór koloru. *Metallic* odpowiada za metaliczny odcień. *Roughness* jako stopień rozpraszania światła na powierzchni oznacza chropowatość.

Tryb edycji

dostępny jest w menu rozwijanym umieszczonym w lewym górnym rogu panelu 3D Viewport oraz poprzez skrót klawiaturowy Tab - naprzemiennie z innymi trybami edycji.

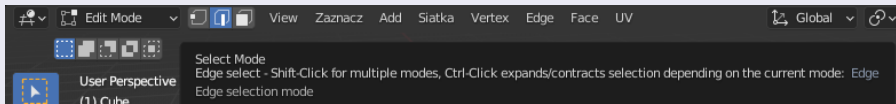
Tryb edycji

dostępny jest w menu rozwijanym umieszczonym w lewym górnym rogu panelu 3D Viewport oraz poprzez skrót klawiaturowy Tab - naprzemiennie z innymi trybami edycji.

Można zaznaczać w zależności od trybu: wierzchołki, krawędzie lub ściany – wybierając myszką lub na klawiaturze odpowiednio 1, 2 lub 3.

Zaznaczenia można rozszerzać przy wciśniętym klawiszu *Shift* lub *Ctrl*.

Można zaznaczyć wszystkie elementy obiektu wciskając klawisz A.



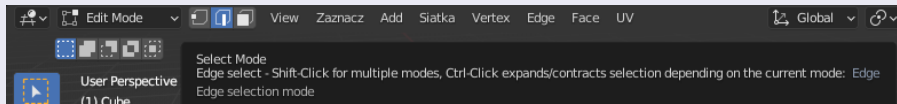
Tryb edycji

dostępny jest w menu rozwijanym umieszczonym w lewym górnym rogu panelu 3D Viewport oraz poprzez skrót klawiaturowy Tab - naprzemiennie z innymi trybami edycji.

Można zaznaczać w zależności od trybu: wierzchołki, krawędzie lub ściany – wybierając myszką lub na klawiaturze odpowiednio 1, 2 lub 3.

Zaznaczenia można rozszerzać przy wciśniętym klawiszu *Shift* lub *Ctrl*.

Można zaznaczyć wszystkie elementy obiektu wciskając klawisz A.



Zaznaczone elementy obiektu w połączeniu z narzędziami przesunięcia, obrotu i skali pozwalają na odpowiednie przekształcenia wybranych fragmentów obiektu.

Podczas skalowania można połączyć blisko leżące wierzchołki wybierając 0 na klawiaturze.

W trybie edycji można po zaznaczeniu całego obiektu przesunąć go względem jego środka obrotu. Potem w trybie obiektu można względem tego punktu obracać obiekt i go skalować.

Podczas skalowania można połączyć blisko leżące wierzchołki wybierając 0 na klawiaturze.

W trybie edycji można po zaznaczeniu całego obiektu przesunąć go względem jego środka obrotu. Potem w trybie obiektu można względem tego punktu obracać obiekt i go skalować.

Aby użyć obiekt w środowisku Unity, można skasować z kolekcji kamerę i światło. Obiekt można zapisać w domyślnym formacie *blend*. Następnie taki obiekt można zaimportować do projektu Unity.







Konstruktorowa lista inicjalizacyjna składników klasy

służy do inicjalizacji składników klasy (w tym składników stałych).
W definicji konstruktora przed ciałem klasy, zaraz za nawiasem zamykającym listę jego argumentów stawiamy dwukropek, po którym oddzielone przecinkami wypisywane są nazwy składników klasy, a po nich w nawiasach klamrowych wartości inicjalizujące, np.

Konstruktorowa lista inicjalizacyjna składników klasy

służy do inicjalizacji składników klasy (w tym składników stałych).
W definicji konstruktora przed ciałem klasy, zaraz za nawiasem zamykającym listę jego argumentów stawiamy dwukropek, po którym oddzielone przecinkami wypisywane są nazwy składników klasy, a po nich w nawiasach klamrowych wartości inicjalizujące, np.

```
class K{
    const int n;
    int p;
    char c = 't';
    double &d;
public:
    K(int a, char z, double &ref) : n{a}, p{8}, d{ref} {
        c = z;
    }
};
```

Literatura



<https://learn.unity.com>



Casey Hardman, Programowanie gier przy użyciu Unity i C#, APress, Warszawa (2020)



Mike Geig, Unity. Przewodnik projektanta gier, Wydanie III, Helion, (2020)