



# Grafika Komputerowa (GRK)

## Laboratorium

### Ćwiczenie 5

### Realizm obrazów 3D



## 1. Wstęp

Proces tworzenia realistycznego obrazu 3D, przeznaczonego do wyświetlenia na płaskiej powierzchni ekranu komputera, składa się z dwóch podstawowych etapów:

- **modelowania**, realizowanego w formule opisu sceny, która definiuje kształty i położenia obiektów przeznaczonych do wizualizacji oraz sposoby ich oświetlenia i obserwacji;
- **renderingu**, mającego na celu wygenerowanie dwuwymiarowego obrazu sceny.

Techniki modelowania zostały przedstawione w ramach ćwiczenia nr 4, natomiast niniejsze laboratorium poświęcone jest zagadnieniom związanym z renderingiem i realizmem obrazów 3D. Temat zajęć jest ściśle związany z zagadnieniami omawianymi w trakcie wykładów 5 i 6 z GRK, stąd zalecane jest wcześniejsze zapoznanie się z ich treścią.

W części praktycznej zajęć używany będzie Blender (<https://www.blender.org>), darmowy program służący do tworzenia obrazów oraz animacji trójwymiarowych. W trakcie zajęć zaprezentowane zostaną podstawy obsługi Blendera, jednak należy mieć na uwadze, że jest on niezwykle rozbudowanym narzędziem, a w pełni świadome korzystanie z jego możliwości wymaga sporej praktyki. Dlatego też przed laboratorium warto obejrzeć jeden z wielu dostępnych w sieci kursów wideo poświęconych tej aplikacji, np. <https://www.youtube.com/watch?v=TPrnSACiJ4>.

## 2. Realizm obrazów 3D

Realizm obrazów 3D jest zagadnieniem trudnym i wieloaspektowym. Pojęcie to może odnosić się już do samych modeli umieszczanych w obrębie tworzonej scenie: obiekty odwzorowywane za pomocą gładkich krzywych lub siatek wielokątów o dużej szczegółowości zazwyczaj będą cechować się większym stopniem realizmu. Jednak podstawowe znaczenie dla uzyskanych efektów ma wykorzystanie odpowiednich modeli oświetlenia oraz realistycznych definicji właściwości materiałów, z których zbudowane są obiekty.

### 2.1. Model oświetlenia i właściwości optyczne materiałów

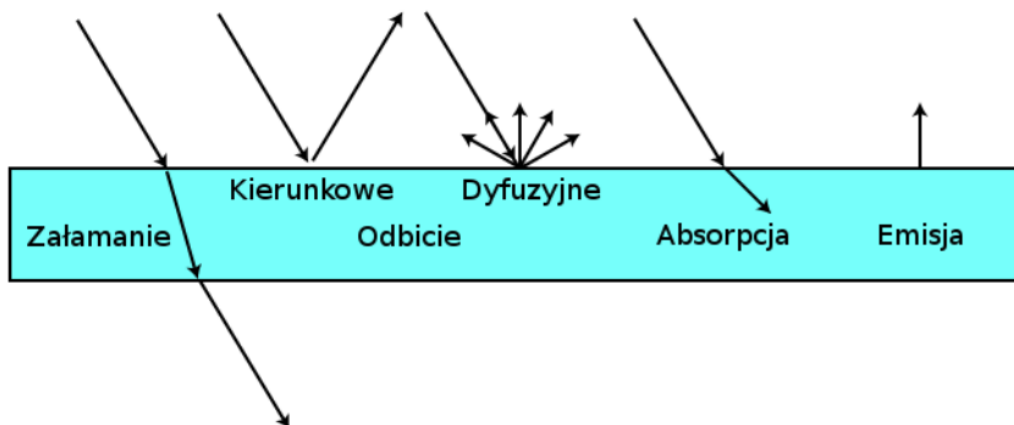
O oświetleniu, szczególnie w odniesieniu do właściwości różnych typów źródeł światła, wspomniano już w poprzednim ćwiczeniu. Warto jednak podkreślić istotne znaczenie światła w kontekście realizmu obrazu 3D, jak również przyjrzeć się bliżej modelom oświetlenia wykorzystywanym podczas renderingu.

Modelowanie sposobu rozchodzenia się światła w obrębie sceny jest jednym z najbardziej złożonych i kosztownych obliczeniowo zagadnień związanych z renderingiem, gdyż wymaga uwzględnienia nie tylko jego źródeł, ale także – a może przede wszystkim – interakcji światła z obiektami i ich otoczeniem. Innymi słowy, do prawidłowej symulacji zjawisk świetlnych konieczne są zarówno ilościowy i jakościowy opis źródeł światła, jak i znajomość właściwości optycznych materiałów, z których tworzone są obiekty.

Analizując zachowanie światła w ramach sceny możemy uwzględniać sytuacje, w których dociera ono do powierzchni obiektów bezpośrednio od źródła, jak również takie, w których światło dociera do obiektów pośrednio, po wcześniejszych odbiciach. W związku z powyższym algorytmy renderingu możemy podzielić na dwie kategorie: rozwiązujące **problem oświetlenia lokalnego** (ang. *local illumination*) lub **oświetlenia globalnego** (ang. *global illumination*). Algorytmy należące do pierwszej grupy uwzględniają jedynie światło docierające bezpośrednio od źródeł oraz lokalne właściwości poszczególnych obiektów. Charakteryzują się one relatywnie niskim kosztem obliczeniowym, dzięki czemu dobrze sprawdzają się w zastosowaniach wymagających generowania grafiki na bieżąco, np. w grach komputerowych. Natomiast

do tworzenia fotorealistycznych obrazów i animacji konieczne jest stosowanie złożonych obliczeniowo metod o charakterze globalnym, które biorą pod uwagę oddziaływania pomiędzy wszystkimi obiektami sceny i modelują zarówno zachowanie światła padającego bezpośrednio, jak i pośrednio. Do tej kategorii należą algorytmy używane przez wszystkie zaawansowane programy służące do generacji grafiki 3D, w tym również przez Blendera.

Prawidłowe działanie algorytmów renderingu wymaga użycia właściwego **modelu oświetlenia**, czyli pewnej funkcji matematycznej pozwalającej wyznaczyć natężenie i charakterystykę światła odbitego od dowolnego punktu na powierzchni obiektu przy zadanych źródłach oświetlenia oraz właściwościach materiału. W idealnym przypadku model ten powinien dokładnie odwzorowywać wszystkie zjawiska zachodzące podczas interakcji światła z materiałem, w tym: odbicie, załamanie oraz pochłanianie (rysunek 1).

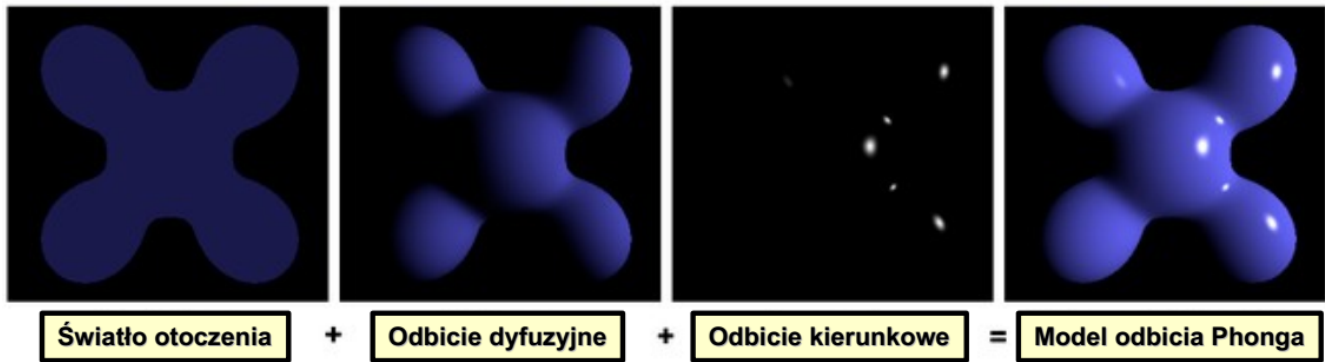


**Rys. 1.** Podstawowe zjawiska fizyczne zachodzące podczas interakcji światła z materiałem [wikipedia.org]

Udział poszczególnych zjawisk zależy od właściwości materiału oświetlanego obiektu. Przykładowo dla materiałów przezroczystych podstawowe znaczenie ma załamanie światła na powierzchni. Natomiast w przypadku powierzchni nieprzezroczystych i matowych część światła jest pochłaniana, a część ulega odbiciu, przy czym powierzchnie ciemne odznaczają się większą absorpcją. Światło odbijające się od tego typu powierzchni rozchodzi się tak samo we wszystkich kierunkach, co jest określane mianem **odbicia rozproszonego lub dyfuzyjnego** (ang. *diffuse reflection*). Jeszcze inaczej wygląda odbicie od gładkich powierzchni nieprzezroczystych: następuje ono w jednym kierunku i nazywane jest **zwierciadlanym lub kierunkowym** (ang. *specular reflection*). W rzeczywistości dla większości materiałów jednocześnie zachodzi wiele rodzajów odbicia. Wówczas odbicie zwierciadlane jest obserwowane jako miejscowe rozświetlenie, podczas gdy pozostałe światło pochodzące od obiektu jest efektem odbicia rozproszonego.

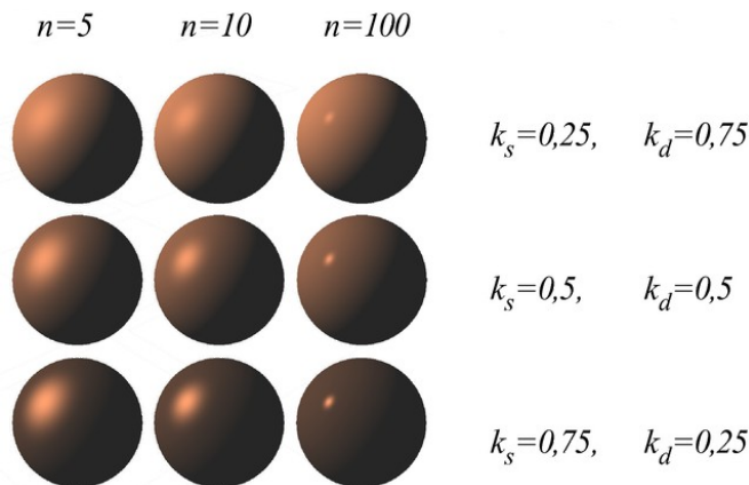
Model oświetlenia pozwalający osiągnąć pełen realizm powinien bazować na dokładnym fizycznym opisie oddziaływania światła z materiałem. Byłby on jednak niezwykle złożony i kosztowny obliczeniowo, dlatego zwykle stosuje się uproszczone zależności empiryczne, które jedynie przybliżają obserwowane zjawiska. Przykładem prostego, jednak często stosowanego opisu zjawiska odbicia jest model Phong'a, który zakłada, że na całkowitą intensywność promieniowania odbitego składają się trzy elementy (rysunek 2):

- odbicie dyfuzyjne, zależne jedynie od kąta padania światła na powierzchnię;
- odbicie zwierciadlane, o wartości zależnej zarówno od kąta padania światła, jak i kierunku obserwacji;
- światło otoczenia (ang. *ambient light*) padające w bezkierunkowy sposób na każdą powierzchnię.



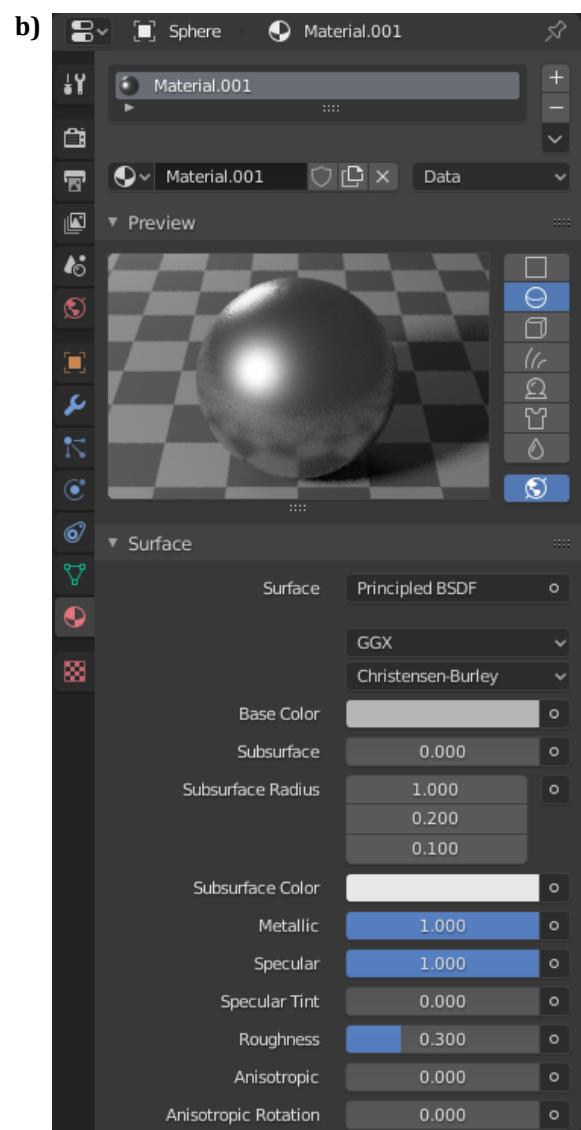
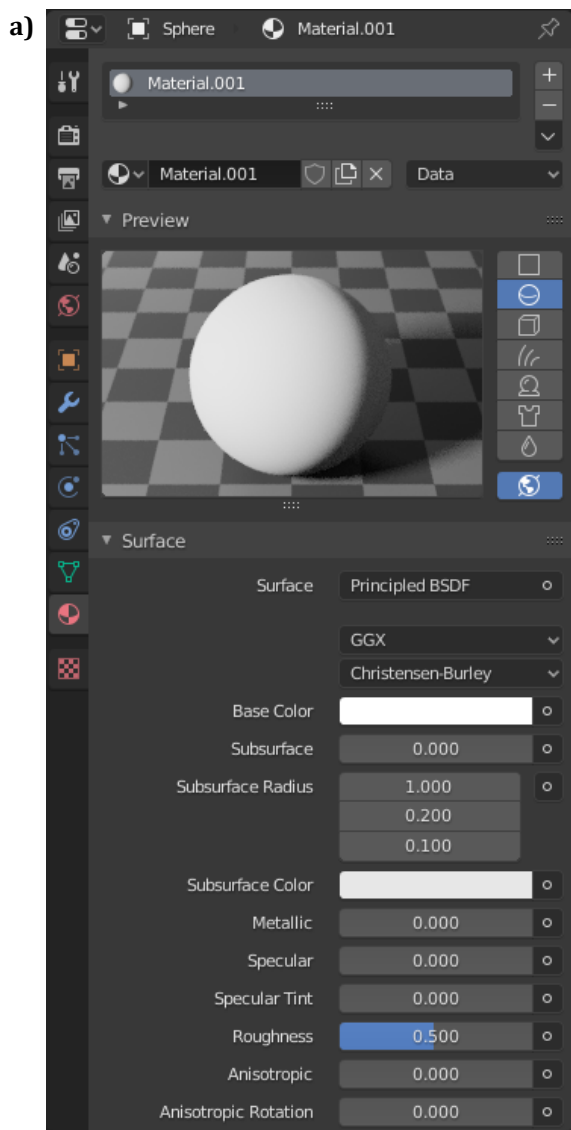
Rys. 2. Ilustracja elementów składowych modelu oświetlenia Phong [wikipedia.org]

Z każdym składnikiem modelu powiązany jest odpowiedni **współczynnik odbicia**, który determinuje jego udział w ostatecznej jasności rozpatrywanego punktu na powierzchni obiektu. Odpowiedni dobór wartości współczynników odbicia oraz **współczynnika chropowatości** (ang. *roughness*) i podstawowego koloru pozwala modelować właściwości różnorodnych materiałów (rysunek 3).



Rys. 3. Wygląd materiału w zależności od wartości współczynników odbicia dyfuzyjnego ( $k_d$ ) i kierunkowego ( $k_s$ ) oraz chropowatości ( $n$ ) [Dariusz Sawicki: *Grafika komputerowa i wizualizacja*, Politechnika Warszawska]

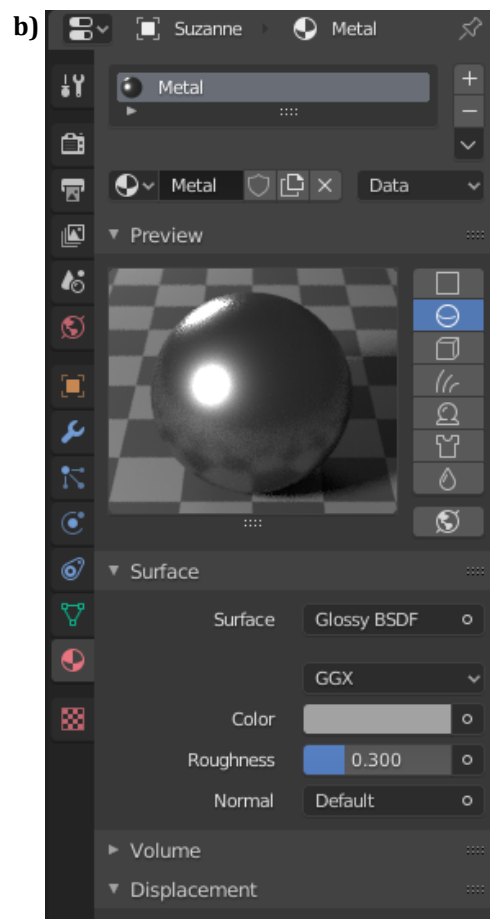
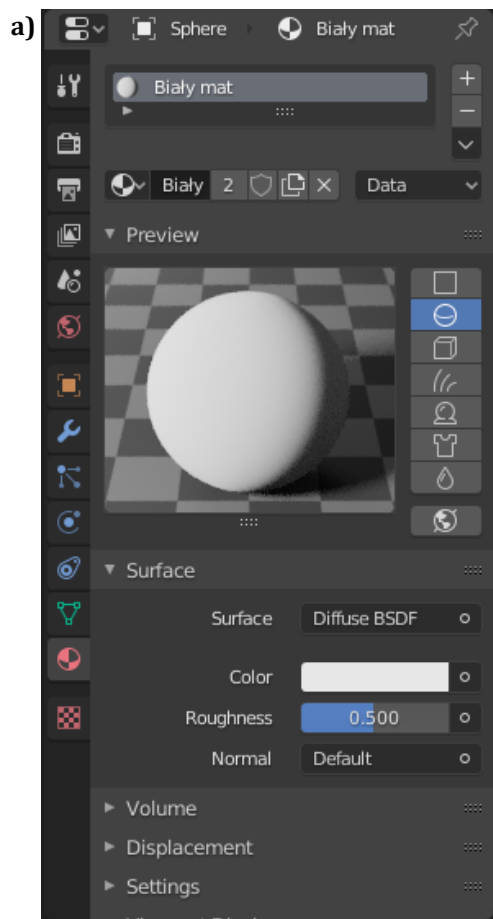
Używany w trakcie laboratorium program Blender opisuje materiały za pośrednictwem **shaderów**. W pewnym uproszczeniu można powiedzieć, że są to zaimplementowane w specjalizowanym języku programowania procedury, które definiują modele oświetlenia opisujące interakcję światła z różnymi materiałami. Domyślnym shaderem jest **Principled BSDF** – uniwersalne narzędzie, udostępniające użytkownikowi zestaw współczynników niezwykle złożonego modelu oświetlenia (rysunek 4). Poprzez odpowiedni dobór ich wartości możliwe staje się utworzenie materiału o praktycznie dowolnych właściwościach optycznych. Dokładny opis shadera Principled BSDF znajduje się pod adresem [https://docs.blender.org/manual/en/latest/render/shader\\_nodes/shader/principled.html](https://docs.blender.org/manual/en/latest/render/shader_nodes/shader/principled.html), jednak na potrzeby laboratorium można ograniczyć się do wymienienia jego głównych parametrów: **Specular** (o podobnym znaczeniu, jak współczynnik odbicia kierunkowego modelu Phong), **Roughness** (współczynnik chropowatości), **Metallic** (decyduje o stopniu refleksyjności powierzchni), **Transmission** i **Transmission Roughness** (związane z przezroczystością materiału) oraz **Emission** (powala modelować emisję światła).



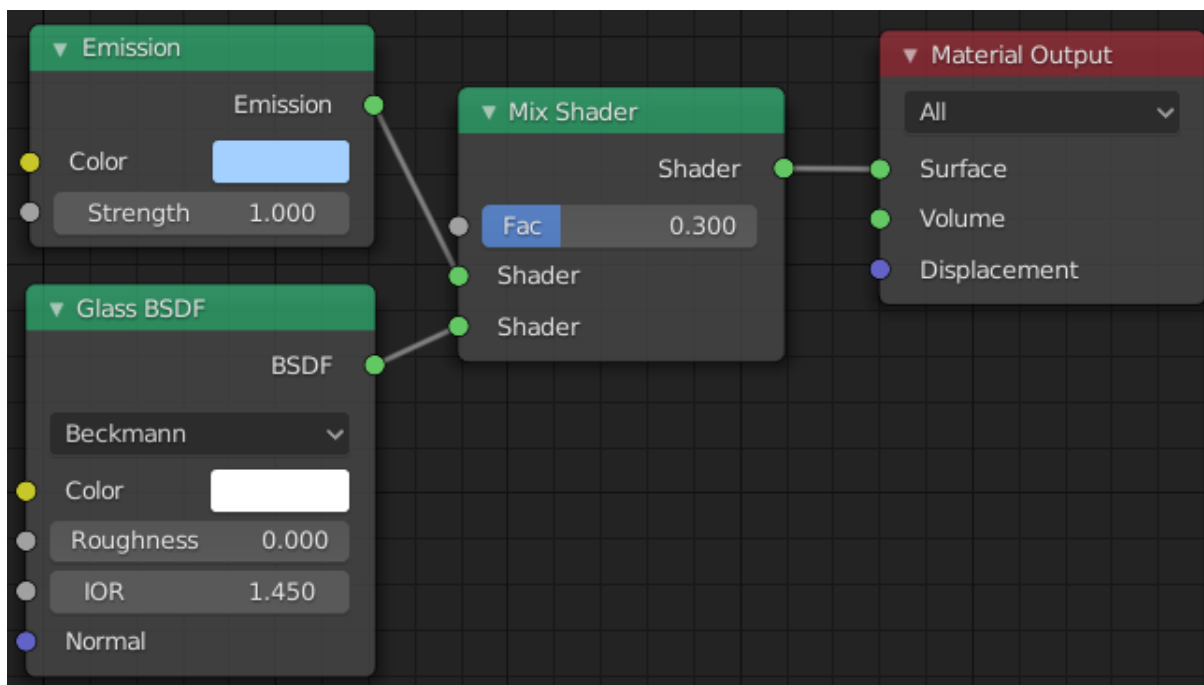
**Rys. 4.** Przykłady definiowania materiałów za pomocą parametrów shadera Principled BSDF

Oprócz wspomnianego wyżej shadera uniwersalnego, Blender zawiera też cały zestaw specjalizowanych procedur, które modelują konkretne rodzaje materiałów (rysunek 5). Przykładowo shader **Glossy BSDF** modeluje powierzchnie błyszczące, w tym lustrzane, zaś **Diffuse BSDF** – matowe. W obu przypadkach użytkownik może wpływać na kolor powierzchni oraz jej gładkość. Innymi często stosowanymi shaderami są **Glass BSDF** (szkło) i **Emmision** (powierzchnia emitująca światło). Więcej informacji o rodzajach shaderów znajduje się w dokumentacji programu Blender oraz m.in. na stronach <https://graphicway.pl/kompendium-shaderow> i <https://www.blenderguru.com/articles/cycles-shader-encyclopedia>.

Dodatkowe możliwości oferuje panel **edytora shaderów** (*Shader Editor*), w którym materiał definiowany jest przez łączenie wejść i wyjść węzłów (ang. *nodes*), reprezentujących zarówno shadery, jak i inne bloki funkcjonalne. Przykładowo, chcąc utworzyć szklaną powierzchnię emitującą światło należy połączyć wyjścia shaderów **Glass BSDF** i **Emmision** z wejściami bloku **Mix shader**, służącego do mieszania w zadanych proporcjach właściwości różnych materiałów (rysunek 6).



Rys. 5. Specjalizowane shadery w programie Blender: Diffuse BSDF (a) oraz Glossy BSDF (b)

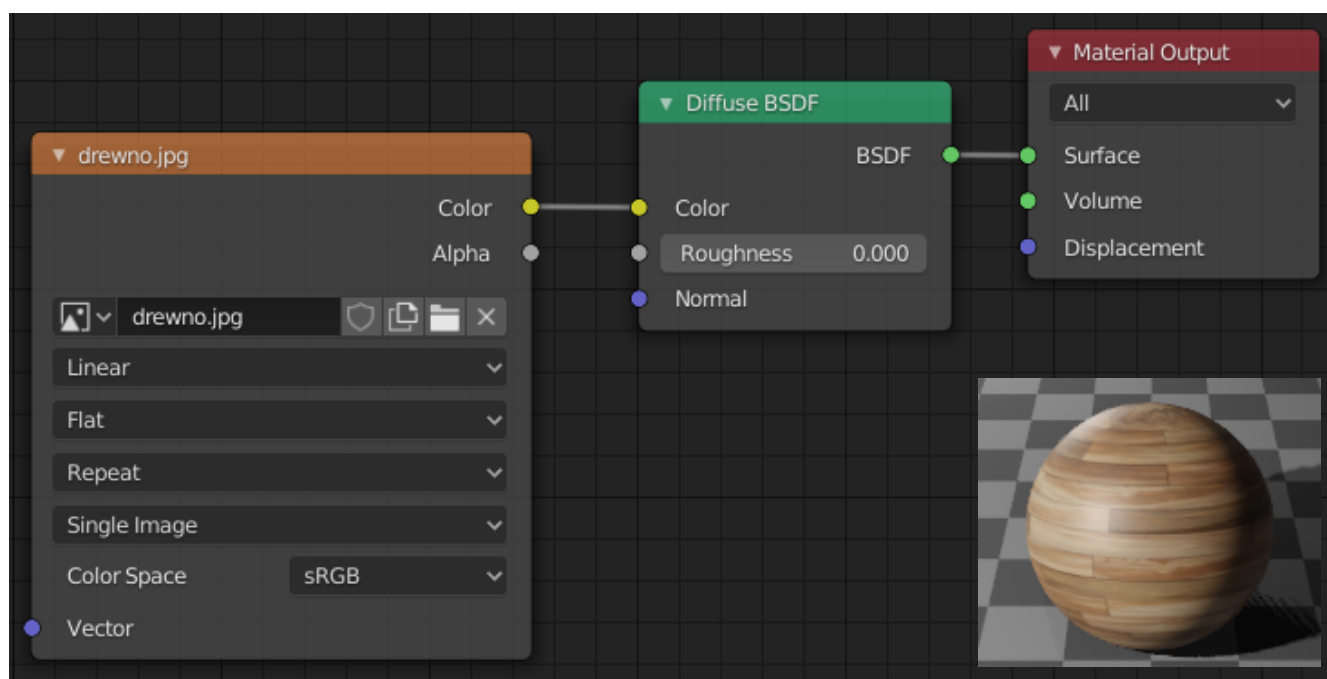


Rys. 6. Przykład definiowania materiału za pomocą edytora shaderów

## 2.2. Tekstury

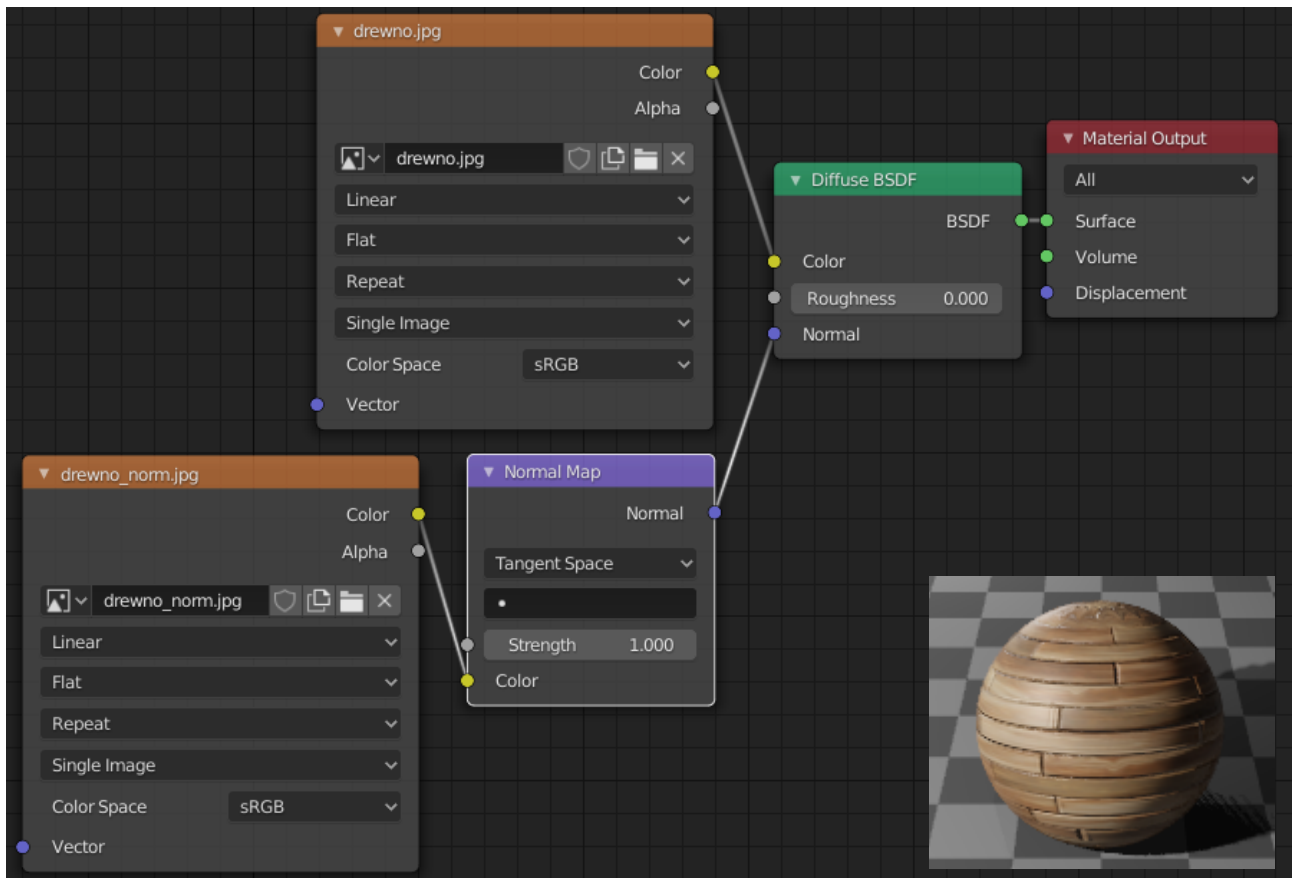
Jeśli przyjrzymy się uważnie rzeczywistym obiektom, to bardzo szybko zauważymy, że różne materiały, takie jak choćby tynk czy drewno, mają nie tylko określoną barwę i właściwości optyczne, ale również charakterystyczną fakturę. Proces odwzorowywania tego typu detali określa się mianem **teksturowania**.

Najprostszą formą teksturowania jest wykorzystanie dwuwymiarowego obrazu do zdefiniowania kolorów poszczególnych obszarów powierzchni obiektu. Blender pozwala dodać tego typu **tekstury bitmapowe** bezpośrednio do opisu materiału – w edytorze shaderów służy do tego blok **Image Texture**, który wyjście można połączyć z wejściem koloru dowolnego shadera (np. Principled BSDF). W efekcie jednolity kolor powierzchni zostanie zastąpiony kolorami pobranymi z obrazu wczytanego do bloku Image Texture, co zostało pokazane na rysunku 7. Podobną rolę spełniają **tekstury proceduralne**, jednak w ich przypadku kolory wyznaczone są przy użyciu funkcji matematycznych. Bardzo prostym przykładem tego typu tekstury jest dostępny w edytorze shaderów węzeł **Checker Texture**, który tworzy wzorek szachownicy.



Rys. 7. Przykład definiowania materiału z teksturą za pomocą edytora shaderów

Aby dodatkowo poprawić realizm obrazu, tekstury łączy się funkcjami wektorowymi dwóch zmiennych, opisującymi lokalne właściwości materiału. Dobrym przykładem takiego podejścia jest **mapowanie wypukłości** (ang. *bump mapping*), pozwalające symulować wszelkiego rodzaju niewielkie nierówności, które trudno byłoby dokładnie oddać na etapie tworzenia modelu trójwymiarowego. Klasyczna **mapa wypukłości** jest obrazem w skali szarości o takich samych rozmiarach jak tekstura, w którym jasności pikseli reprezentują stopień wklęsłości lub wypukłości punktów na powierzchni obiektu. W edytorze shaderów Blendera tego typu mapowanie wypukłości realizowane jest przez blok **Bump**. Natomiast blok **Normal Map** pozwala dokonać **mapowania kierunków wektorów normalnych** (ang. *normal mapping*), co zwykle daje lepsze efekty niż samo przesunięcie wysokości danego punktu względem powierzchni bez zmiany kierunku związanego z nim wektora normalnego. W obu przypadkach za wczytanie mapy – która jest odpowiednio przygotowanym obrazem bitmapowym – odpowiada blok Image Texture. Rysunek 8 przedstawia przykład materiału zdefiniowanego przy użyciu tekstury bitmapowej i mapy wypukłości.



**Rys. 8.** Przykład definiowania materiału z teksturą bitmapową i mapą wypukłości za pomocą edytora shaderów



### 3. Zadania do wykonania


W trakcie zajęć należy przygotować w edytorze tekstu (np. Writer z LibreOffice) dokument zawierający sprawozdanie. Potrzebne do sprawozdania obrazy należy wygenerować używając silnika renderującego **Cycles** (utworzenie renderingu odbywa się przez naciśnięcie klawisza **F12**). Obrazy można zapisywać w Blenderze (opcja **Image** → **Save** w oknie renderingu), a potem importować do sprawozdania (opcja **Wstaw** → **Obraz**), lub też pozyskiwać jako zrzuty ekranu (kombinacja **Win+Shift+S**).

#### Podstawy renderingu w programie Blender

**Zad. 1.** Wykonać wspólnie z prowadzącym zestaw zadań mających na celu przybliżenie podstawowych zagadnień związanych z tworzeniem i renderingiem fotorealistycznych obrazów w programie Blender.

**UWAGA:** w prezentacji *Lab5\_materialy.pdf* dostępne są dodatkowe informacje o definiowaniu materiałów, które mogą być przydatne podczas realizacji dalszych zadań.

#### Właściwości materiałów (2 punkty)


**Zad. 2.1.** Otworzyć scenę z pliku *scena2\_1* i obejrzyć ją, najlepiej w trybie podglądu renderingu (klawisz **Z** i wybranie opcji **Rendered**). Trzy obiekty w obrębie sceny mają już zdefiniowane materiały: czerwona kulka jest idealnie matowa (shader **Diffuse BSDF**), małpka Suzie odbija światło w sposób imitujący metal (**Glossy BSDF**), zaś podłoga jest powierzchnią półmatową, częściowo odbijającą (**Diffuse BSDF** oraz **Glossy BSDF** połączone przy użyciu **Mix Shader**). Ponadto oświetlenie sceny pochodzi od powierzchni emisyjnej (**Emission**). Wybierać kolejne obiekty przyjrzeć się definicjom ich materiałów (zakładka z ikoną  w panelu po prawej stronie ekranu), zwracając też uwagę na sposób, w jaki są one reprezentowane w **edytorze shaderów** (panel na dole ekranu). Dla czerwonej kulki oraz Suzie dokonać zmian wartości współczynnika chropowatości (parametr **Roughness** obu shaderów) i sprawdzając jak wpływają one na wygląd obiektów.

*W sprawozdaniu umieścić dwa renderingu wykonane przy skrajnych ustawieniach parametru **Roughness** obu obiektów. Skomentować jego wpływ na właściwości powierzchni matowych i połyskujących. Szczególną uwagę zwrócić na stopień odbijania innych obiektów sceny oraz kształt i rozmiar rozjaśnień będących efektem odbić kierunkowych. Który shader i z jaką wartością **Roughness** może służyć do imitowania powierzchni lustrzanych?*

**Zad. 2.2.** Zduplikować obiekt małpki Suzie (kombinacja klawiszy **Shift+D**) i przenieść jego kopię w inne miejsce sceny. Usunąć z kopii obiektu obecny materiał i dodać nowy – domyślnie zostanie mu przypisany uniwersalny shader **Principled BSDF**. Zadanie polega na takim doborze jego parametrów, aby uzyskać materiał jak najbardziej przypominający ten użyty w oryginalnym obiekcie.

**UWAGA:** oryginalny materiał jest połyskujący i metaliczny, to oznacza, że w shaderze **Principled BSDF** trzeba ustawić wysokie wartości parametrów **Spacular** oraz **Metallic**. Znaczenie mają również kolor oraz chropowatość powierzchni (**Roughness**).

*W sprawozdaniu umieścić rendering zmodyfikowanej sceny oraz informacje o ustawieniach **Principled BSDF** (te ostatnie mogą mieć postać odpowiednio skadrowanego zrzutu ekranu z edytora shaderów).*

**Zad. 2.3.** Ponownie wczytać plik *scena2\_1*. Zdefiniować materiały dla płaszczyzn będących ścianami: jedna powinna być idealnym lustrem, a dwie pozostałe – jasnymi matowymi powierzchniami o niewielkiej chropowatości. W zadaniu można korzystać z panelu właściwości materiałów (zakładka z ikoną ) lub też z edytora shaderów. W pierwszym przypadku shader wybierany jest za pośrednictwem listy **Surface**, natomiast w edytorze należy dodać odpowiedni węzeł (kombinacja **Shift+A** i wybór pozycji z podmenu **Shader**) i połączyć jego wyjście z wejściem **Surface** węzła **Material Output**. Dopuszczalne jest użycie dwóch różnych shaderów dla obu typów materiału (**Glossy BSDF** dla połyskujących i **Diffuse BSDF** dla matowych) lub też uniwersalnego shadera **Principled BSDF** z właściwie dobranymi parametrami.

*W sprawozdaniu umieścić rendering zmodyfikowanej sceny, informacje o rodzajach i parametrach shaderów (mogą mieć postać skadrowanych zrzutów ekranu z edytora shaderów) oraz uzasadnienie ich wyboru.*

**Zad. 2.4.** Dodać materiały do pozostałych obiektów, w taki sposób, aby jedna kulka była z matowego szkła (np. **Glass BSDF**), druga połyskująca i jednocześnie emitująca światło (np. połączenie **Emission** z **Glossy BSDF**), a sześcian był idealnie matowy (np. **Diffuse BSDF** lub **Velvet BSDF**). Podobnie jak w poprzednim zadaniu, można też użyć dla każdego obiektu uniwersalnego shadera **Principled BSDF** z odpowiednio ustawionymi wartościami parametrów.

*W sprawozdaniu umieścić rendering zmodyfikowanej sceny, informacje o rodzajach i parametrach shaderów (mogą mieć postać skadrowanych zrzutów ekranu z edytora shaderów) oraz uzasadnienie ich wyboru.*

## Tekstury (1 punkt)

**Zad. 3.1.** Otworzyć plik *scena3\_1*. Zmienić materiał podłogi przez dodanie do niego bitmapowej tekstury drewna: w edytorze shaderów dodać węzeł typu **Image Texture** (kombinacja **Shift+A** i wybór pozycji z podmenu **Texture**), wczytać do węzła teksturę z pliku *drewno.jpg* (przycisk **Open**), a następnie połączyć jego wyjście **Color** z wejściem koloru węzła materiału podłogi (w zależności od rodzaju shadera jest to wejście **Color** lub **Base Color**). W analogiczny sposób zmodyfikować materiał ścian, tym razem jednak wybierając teksturę z pliku *tynk.jpg*. W ostatnim kroku należy dodać do materiału sześcianu dowolną teksturę proceduralną. Dodaje się je podobnie jak tekstury bitmapowe, tylko należy utworzyć inny rodzaj węzła, np. **Checker Texture** generuje wzorek szachownicy.

*W sprawozdaniu umieścić efekt renderingu sceny oraz skadrowane zrzuty ekranu z edytora shaderów.*

**Zad. 3.2.** Uzupełnić tekstury bitmapowe z poprzedniego zadania o wypukłości. Przykładowo dla podłogi trzeba w pierwszej kolejności dodać do definicji materiału drugi węzeł **Image Texture**, wczytać do niego obraz z mapą wektorów normalnych z pliku *drewno\_norm.jpg* i wybrać opcję **Non-Color** z listy **Color Space**. Następnie należy utworzyć węzeł **Vector → Normal Map** i połączyć go z pozostałymi: wejście **Color** z wyjściem **Color** węzła **Image Texture**, a wyjście **Normal** z wejściem **Normal** shadera materiału. Stopień wypukłości regulowany jest za pomocą parametru **Strenght** w bloku **Normal Map**. W taki sam sposób można zmapować wypukłości ścian (mapa zapisana jest w pliku *tynk\_norm.jpg*)

*W sprawozdaniu umieścić efekt renderingu sceny oraz skadrowane zrzuty ekranu z edytora shaderów. Skomentować różnicę wobec obrazu uzyskanego w zadaniu 3.1.*

### **Realizm złożonych modeli (1 punkt)**

**Zad. 4.1.** Celem zadania jest dodanie materiałów i tekstur do bardziej złożonego modelu, którego rodzaj można wybrać samodzielnie, po uzgodnieniu z prowadzącym. W przypadku braku własnych pomysłów można skorzystać z modelu zadanego przez prowadzącego lub z poleconego przez niego tutorialu.

*W sprawozdaniu umieścić efekt renderingu obiektu. Zapisać scenę do pliku L5\_Imię nazwisko.blend i oddać prowadzącemu razem ze sprawozdaniem.*



**Fundusze Europejskie**

Wiedza Edukacja Rozwój

**Politechnika  
Warszawska**

**Unia Europejska**  
Europejski Fundusz Społeczny



Materiały opracowane w ramach zadania 15 „Modyfikacja międzywydziałowych studiów I stopnia na kierunku Inżynieria Biomedyczna” projektu „NERW PW. Nauka – Edukacja – Rozwój – Współpraca”, współfinansowanego jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego