|  |  |  |
| --- | --- | --- |
| PP-PUT_logo_jasne | POLITECHNIKA POZNAŃSKA  Wydział Informatyki i Telekomunikacji |  |
| PRACA DYPLOMOWA  INŻYNIERSKA | | |

Autor:

Aleksander Pietrzak

Mateusz Starecki

**Portal do wydruku zdjęć sferycznych**

Promotor:

Dr inż. Tomasz Grajek

Poznań, 2023

**Spis treści**

[Streszczenie 3](#_Toc124864923)

[Abstract 3](#_Toc124864924)

[1. Wprowadzenie 3](#_Toc124864925)

[2. Cel pracy 3](#_Toc124864926)

[2.1. System wydruku 3](#_Toc124864927)

[2.2. Założenia 3](#_Toc124864928)

[3. Podział prac 3](#_Toc124864929)

[4. Wstęp teoretyczny 3](#_Toc124864930)

[4.1.1. Bryły platońskie 3](#_Toc124864931)

[4.2. Siatki brył 3](#_Toc124864932)

[4.3. HTML, CSS, Javascript, Node 3](#_Toc124864933)

[4.4. OpenGL 3](#_Toc124864934)

[5. Opracowane rozwiązania 3](#_Toc124864935)

[5.1. Wprowadzenie 3](#_Toc124864936)

[5.1.1. Wybór bryły 3](#_Toc124864937)

[5.1.2. Budowa dwunastościanu foremnego 4](#_Toc124864938)

[5.1.3. Siatka dwunastościanu foremnego 6](#_Toc124864939)

[5.1.4. Składanie modelu 7](#_Toc124864940)

[5.2. System mapowania 7](#_Toc124864941)

[5.2.1. Mapowanie sferyczne 7](#_Toc124864942)

[5.2.2. Poprawa mapowania 7](#_Toc124864943)

[5.3. Aplikacja webowa 7](#_Toc124864944)

[5.3.1. Interfejs 7](#_Toc124864945)

[5.3.2. System płatności 7](#_Toc124864946)

[5.3.3. Animacje 7](#_Toc124864947)

[6. Implementacja systemu 7](#_Toc124864948)

[6.1. Środowisko programistyczne 7](#_Toc124864949)

[6.2. Aplikacja webowa (jak) 7](#_Toc124864950)

[6.3. WebGL (nazwa robocza) 7](#_Toc124864951)

[7. Testy subiektywne 7](#_Toc124864952)

[8. Podsumowanie 7](#_Toc124864953)

[9. Bibliografia 7](#_Toc124864954)

# Streszczenie

# Abstract

# Wprowadzenie

# Cel pracy

## System wydruku

## Założenia

# Podział prac

# Wstęp teoretyczny

### Bryły platońskie

## Siatki brył

## HTML, CSS, Javascript, Node

## OpenGL

# Opracowane rozwiązania

## Wprowadzenie

### Wybór bryły

Zdjęcie sferyczne obejmuje cały widok wokół punktu obserwacji, pozwala na obejrzenie widoku 360 stopni wokół, zarówno w pionie jak i poziomie. Idealną reprezentacją takiego zdjęcia w przestrzeni jest przedstawienie go pod postacią kuli. Jednak zbudowanie jej z zaledwie jednej kartki nie jest możliwe. Dobrą alternatywą są bryły, które swoim kształtem przypominają sferę.

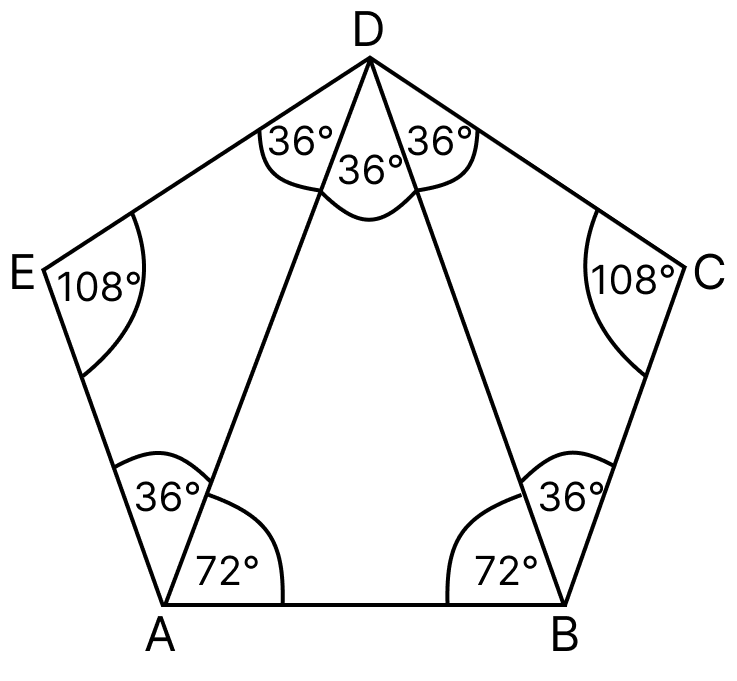
Sferycznych wielościanów znajduję się niezliczona ilość, jednak nie wszystkie nadają się do poprawnej reprezentacji takiego zdjęcia. Przede wszystkim figura musi być jak najłatwiejsza do wycięcia i złożenia. Figury nieregularne zbudowane z dużej ilości niewielkich wielokątów czy nawet regularne, ale zbudowane z figur o dużej liczbie krawędzi są trudne do zbudowania. W gronie takich figur znajduję się jedna grupa, która zawiera bryły mające tylko regularne wielokąty jako boki, a każdy wierzchołek jest otoczony taką samą liczbą krawędzi. Sprawia to dobre wizualnie wrażenie a regularne kształty zapobiegają zniekształceniom obrazu i są łatwiejsze do złożenia. Spośród brył platońskich dwie z nich są łudząco podobne do sfery. Są nimi dwunastościan zbudowany z pięciokątów foremnych oraz dwudziestościan, którego ściany są trójkątami równobocznymi.

Dwudziestościan bardzo dobrze reprezentuje wielościan sferyczny i jest zbudowany z prostych elementów, które nie powinny sprawiać problemów do złożenia. Figura jednak składa się ze stosunkowo dużej liczby ścian, które mimo niezłożonej budowy po przeskalowaniu na kartkę papieru są niewielkich rozmiarów. Sprawia to trudność w zginaniu jak i klejeniu drobnych elementów.

Dwunastościan jest trójwymiarowym wielościanem złożonym z zaledwie dwunastu płaszczyzn. Liczba 20 wierzchołków oraz 30 krawędzi jest znacznie mniejsza od dwudziestościanu, a pięciokąty z których jest zbudowany po nałożeniu na kartkę są znacząco większe co ułatwia jego zbudowanie.

### Budowa dwunastościanu foremnego

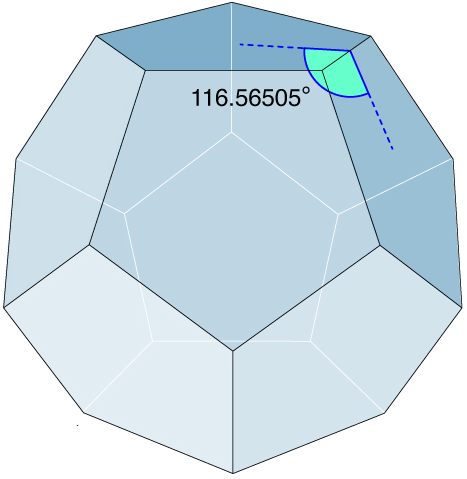
Dwunastościan użyty do reprezentacji zdjęcia sferycznego jest złożony z 12 równych pentagonalnych powierzchni. Każda ściana bryły składa się z trzech trójkątów równoramiennych, jednego centralnego oraz dwóch trójkątów przystających umiejscowionych po jego dłuższych bokach.



Rys. 5.1.1. Ściana dwunastościanu po podziale na trójkąty []

Stosunek długości przeciwległych krawędzi do długości przeciwległych ścian jest taki sam, jak stosunek złotego podziału ϕ = 1 + √5/2 = 1.618033988749....

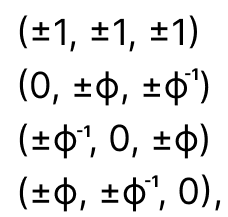
To samo dotyczy kąta pomiędzy sąsiednimi płaszczyznami tzw. kąta dwuściennego figury (rys. 5.1.2.).



Rys. 5.1.2. Kąt dwuścienny w dwunastościanie.

Wierzchołki dwunastościanu w kartezjańskim układzie współrzędnych są reprezentowane przez trzy liczby rzeczywiste x, y, z, które określają położenie danego punktu w przestrzeni trójwymiarowej. Środek bryły znajduje się w centrum układu w punkcie (0, 0, 0), a długość krawędzi wynosi √5-1. Takie położenie figury oraz rozmiar ścian pięciokątów pozwala na proste wyznaczenie współrzędnych, które definiuje poniższy rysunek (rys. 5.1.2.).

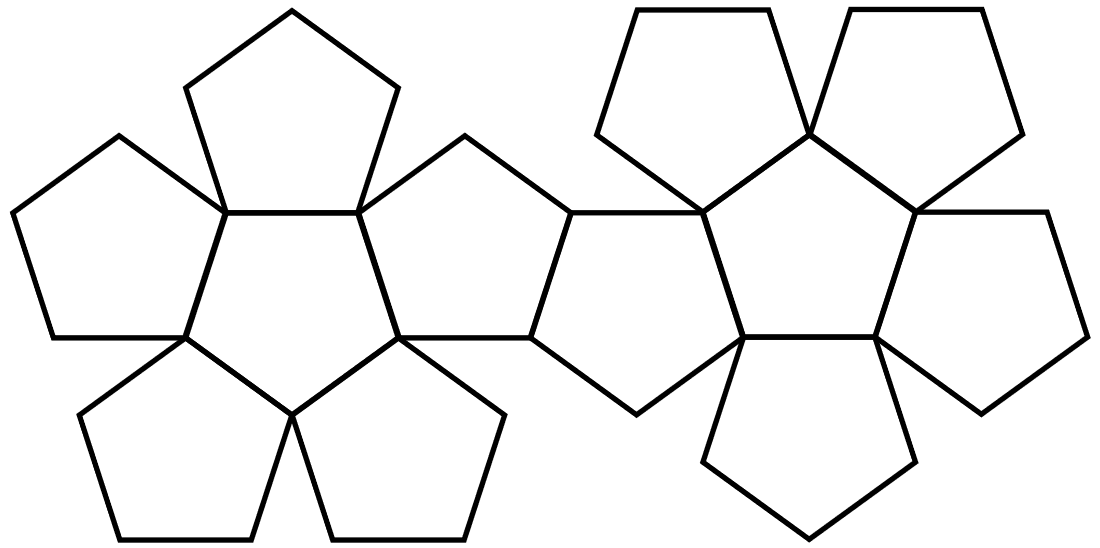
Należy zauważyć, że wszystkie koordynaty oprócz 0 są poprzedzone dwoma znakami przez co mogą przyjąć wartość ujemną lub dodatnią. Daje to łącznie 20 różnych współrzędnyc x, y, z.



Rys. 5.1.2. Współrzędne wierzchołków dwunastościanu foremnego od długości krawędzi √5-1.

### Dwuwymiarowa siatka dwunastościanu

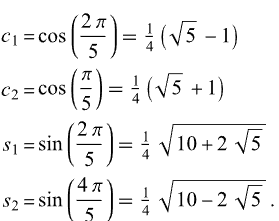
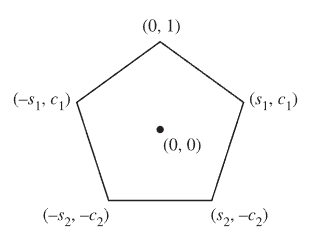
Siatka jest interpretacją przestrzennego modelu bryły na płaszczyźnie. Jest to sposób na rozłożenie trójwymiarowego obiektu na powierzchnie płaską, co pozwala na jego przedstawienie w dwóch wymiarach. Ma za zadanie umożliwić złożenie bryły z kartki papieru. W jej skład wchodzą wszystkie pięciokąty, będące powierzchniami dwunastościanu ze zmienionym kątem pomiędzy ścianami na kąt półpełny.



Rys. 5.1.3. Siatka dwunastościanu.  
Źródło: https://mathworld.wolfram.com/RegularDodecahedron.html

Składa się z dwunastu pięciokątów odpowiadającym powierzchniom dwunastościanu foremnego. Przedstawiona jest w postaci dwóch połączonych ze sobą elementów, które kształtem przypominają kwiat. Każdy element jest zbudowany sześciu pięciokątów foremnych połączonych ze sobą przez krawędzie. Oba elementy są swoją kopią obróconą o 180° wzdłuż osi OX.

Wierzchołki platońskiego dwunastokąta po zrzutowaniu na płaszczyznę mają zmienione współrzędne kartezjańskie (x, y, z) . W celu dostosowania siatki do powierzchni płaskiej wartość współrzędnej (z) jest wyzerowana a pozostałe przesunięte.



Rys. 5.1.4. pięciokąta na płaszczyźnie

<https://mathworld.wolfram.com/RegularPentagon.html>

### Składanie modelu

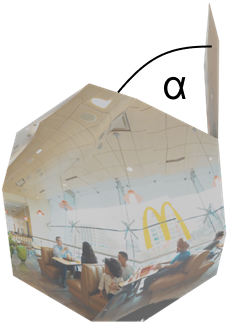
Złożenie modelu polega na przekształceniu dwuwymiarowej siatki dwunastościanu na jego trójwymiarową projekcje. Jest to możliwe gdy wszystkie krawędzie siatki zostaną w odpowiedni sposób złączone. W celu dokonania takiego zabiegu w rzeczywistości potrzebne są elementy w budowie siatki, które pozwolą na połączenie powierzchni nie mających sąsiednich powierzchni z każdej strony. Części te zostały skonstruowane na kształt trapezu równoramiennego, którego dłuższe podstawy są połączone ze ścianami pięciokąta. Mają za zadanie stabilizować powierzchnie w przestrzennej wersji modelu poprzez przyklejenie ich do wewnętrznych części ścian bryły, tak aby krawędzie sąsiadujących pięciokątów były ze sobą połączone. W siatce znajduje się 19 elementów do klejenia, położonych tak, aby każda wspólna krawędź dwunastościanu była połączona jednym elementem.



Rys. 5.1.5. Gotowa siatka dwunastościanu z nałożonym zdjęciem sferycznym.

Transformacja płaskiego na przestrzenny model polega na zmianie kątów sąsiadujących ze sobą ścian ze 180 stopni na wartość liczby φ. Proces ten odbywa się poprzez rotację każdego pięciokąta wzdłuż krawędzi wspólnej dla obu figur. Kąt obrotu jest różnicą kątów między ściennych obu modeli.

α = 180°- φ = 63.43495.

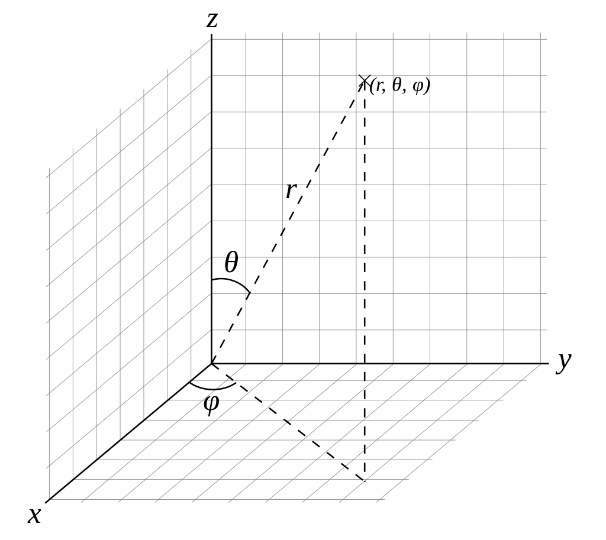


Rys. 5.1.6. Kąt podczas rozkładania dwunastościanu.

## System mapowania

### Mapowanie sferyczne

Mapowanie sferyczne UV jest metodą, która przekształca współrzędne (x, y, z) punktu na sferze na współrzędne (u, v) punktu na płaszczyźnie UV. W procesie mapowania sferycznego powierzchnia sfery jest rozcinana na mniejsze fragmenty, które są następnie rozmieszczane na płaszczyźnie UV. Każdy z nich jest przypisany do unikalnego punktu na mapie UV, który jest odpowiednio przypisany do tekstury.



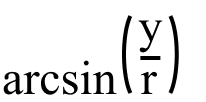
Rys. 5.2.1. Układ współrzędnych sferycznych.  
źródło: https://pl.wikipedia.org/wiki/Układ\_współrzędnych\_sferycznych

Do konwersji współrzędnych wierzchołków z trójwymiarowej powierzchni na koordynaty tekstury należy przekształcić punkty z układu kartezjańskiego na układ sferyczny. Potrzebne są do tego 3 wartości:

- kąt azymutalny *ϕ*, będący kątem podawanym w radianach pomiędzy rzutem prostokątnym wektora wybranego wierzchołka na płaszczyznę OXY, a osią OX. Przekształcenie na układ sferyczny odbywa się za pomocą wzoru podanego poniżej.



- kąt elewacji θ, podawany w radianach jest wektorem wybranego wierzchołka a osią OZ. Przekształcenie na układ sferyczny odbywa się za pomocą wzoru podanego poniżej:



- promień wodzący r, który jest odległością wybranego wierzchołka od początku układu współrzędnych (0, 0, 0). Jego przekształcenie z układu kartezjańskiego odbywa się poprzez spierwiastkowanie sumy kwadratów wartości x, y, z danego wierzchołka.

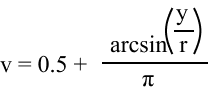
  
https://pl.wikipedia.org/wiki/Uk%C5%82ad\_wsp%C3%B3%C5%82rz%C4%99dnych\_sferycznych

Mając wartości w sferycznym układzie współrzędnych pozostaje przeskalować je do przedziału od 0 do 1, aby wyznaczyć koordynaty tekstury.

Obliczenie ‘u’ polega na podzieleniu kąta azymutalnego przez liczbę 2π, aby uzyskać wynik o przedziale (-0.5, 0.5) po czym dodać liczbę 0.5, która pozwala uzyskać liczbę dodatnią z przedziału (0, 1).



Obliczenie ‘v’ polega na podzieleniu kąta elewacji przez π oraz dodaniu do wyniku wartości 0.5.

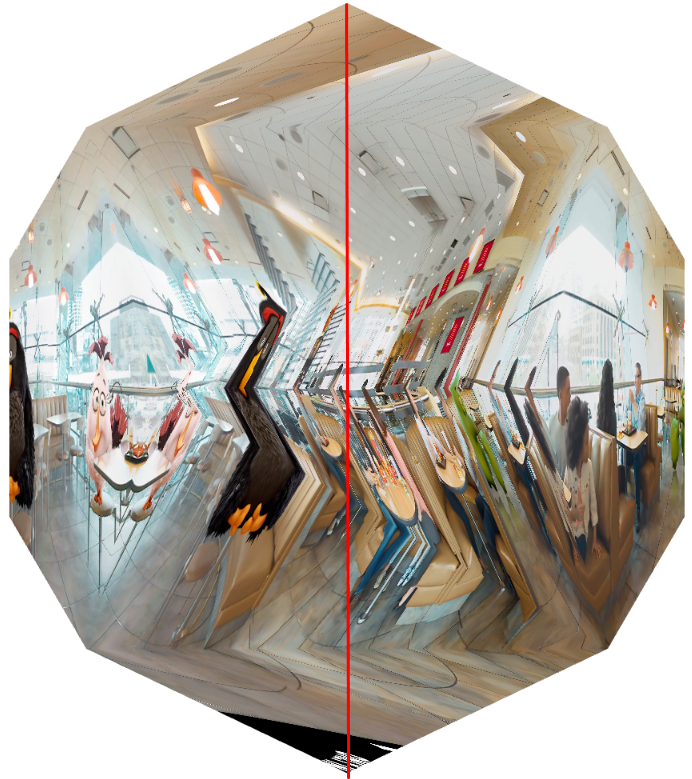


Wynikające z tego koordynaty (u, v) są następnie używane jako koordynaty tekstury dla punktu na sferze, umożliwiając odpowiednie przypisanie tekstury do powierzchni sfery.�=�(�,�,�)=�sin⁡�sin⁡�,

### Poprawa mapowania

Nałożone zdjęcie na dwunastościan metodą mapowania sferycznego nie daje idealnych efektów, ponieważ nie mapujemy obrazu na sferę tylko na siatkę trójkątów ją przypominającą.

Funkcja mapowania UV ma zakres, który pokrywa się z obwodem dwunastościanu co powoduje, że wierzchołki leżące na granicach mogą należeć do kilku trójkątów. Punkty znajdujące się wzdłuż linii geometrii przedstawionej na poniższym obrazie (rys. 5.2.3), mogą współdzielić trójkąty zawierające współrzędne UV początku i końca tekstury. a ponieważ jedynym możliwym przejściem pomiędzy 0 a 1 w układzie współrzędnych teksturowych jest przez cały obraz powoduje to, że cała długość zdjęcia jest mapowana na pojedynczy trójkąt.



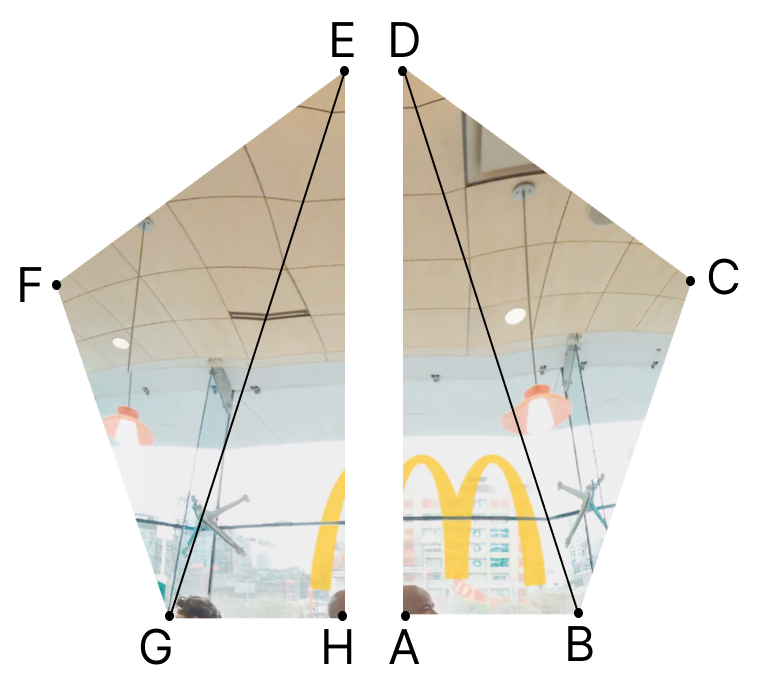
Rys. 5.2.2. Dwunastościan z zaznaczoną linią geometrii

Metodą która zapobiega takiemu zjawisku jest podzielenie trójkątów współdzielących wspomniane wyżej wierzchołki na pół .



Rys. 5.2.3. Dwunastościan z podziałem na trójkąty.

Podział ten polega na rozdzieleniu punktów w obrębie linii geometrii na osobne dwa wierzchołki zawierające różne współrzędne UV. Różnice w wielkości 10-6 pomiędzy wierzchołkami nie są zauważalne dla ludzkiego oka, a pozwalają na uzyskanie prawidłowo zmapowanej tekstury na całej powierzchni wielościanu. Poniżej przedstawiono rozdzielenie jednej ze ścian dwunastościanu.



Rys. 5.2.4. Pięciokąt po podziale na dwie części.

A = (- φ, 0, - 0.000001), H = (- φ, 0, 0),  
B = (- φ, 0, - ϕ -1), G = (- φ, 0, ϕ -1),  
C = (- 1, - 1, - 1), F = (-1, -1, 1),  
D = (- ϕ -1, - φ, -0.000001), E = (- ϕ -1, - φ, 0),

Po poprawie mapowania zdjęcie sferyczne jest właściwie nałożone na powierzchnie wielościanu.

Rys. 5.2.5. Dwunastościan przed podziałem wierzchołków Rys. 5.2.6. Dwunastościan po podziale wierzchołków

## Aplikacja webowa

### Strona główna

Strona główna (ang. homepage) jest to część aplikacji webowej która zostaje wyświetlona jako pierwsza po wejściu adres URL aplikacji. Zawiera ważne informacje takie jak przeznaczenie systemu, instrukcje opisującą etapy realizacji jakie użytkownik musi podjąć podczas tworzenia wydruku zdjęcia, jak takie zdjęcie zrobić. Informacje te zostały podzielone na sekcje po których można się poruszać za pomocą nawigacji. Kliknięcie jej konkretnego elementu pozwala na szybkie przemieszczanie się po różnych częściach strony bez konieczności jej przewijania i szukania informacji.



Rys. 5.3.1.Menu nawigacji w wersji desktopowej.

Pierwszą sekcją, która jest widoczna od razu po załadowaniu strony z jest hero image. Jest jednym z najważniejszych elementów portalu, ponieważ ma na celu zwrócenie uwagi użytkownika i przedstawienie głównego przesłania strony. Zawiera obraz służący za tło i napisy opisujące główne zadanie aplikacji. W samym centrum znajduję się obracający dwunastościan z nałożonym na siebie zdjęciem sferycznym. Ma on pokazywać efekt końcowy oferowanej usługi wzbudzając zainteresowanie tematem oraz zachęcić do jej skorzystania.



Rys. 5.3.2. Sekcja hero image strony głównej.

W lewym górnym rogu hero image znajdue się symbol słońca, który pozwala zmienić motyw na stronie. Aplikacja zawiera dwa motywy, jasny oraz ciemny. Pozwala on użytkownikom na zmiane kolorystki strony z jasnej na ciemną lub na odwrót. Polega na zmianie koloru tła, napisów oraz ikon. Dodanie ciemnego trybu jest przydatne dla użytkowników przebywających w ciemniejszym otoczeniu, na przykład przed snem czy w ciemnym pomieszczeniu. Pozwala na większy komfort czytania osobom z wadami wzroku, które mogą mieć problemy z czytaniem na jasnym tle. Zużywa mniej energi co pozwala na zaoszczędzenie baterii na urządzeniach mobilnych. Przycisk „dalej” znajdujący się w prawym dolnym rogu strony przechodzi do kolejnej podstrony z zaimplementowanym systemem wydruku.



Rys. 5.3.3. Sekcja Hero w motywie jasnym.

W kolejnej sekcji zawarty jest krótki opis przedstawiający procedurę drukowania zdjęcia sferycznego jak i opis siatki dwunastościanu na płaszczyźnie. Pod tekstem znajduję się przykładowa zmapowana siatka, która ma uzmysłowić użytkownika o budowie zdjęcia zaraz po jego wydrukowaniu.



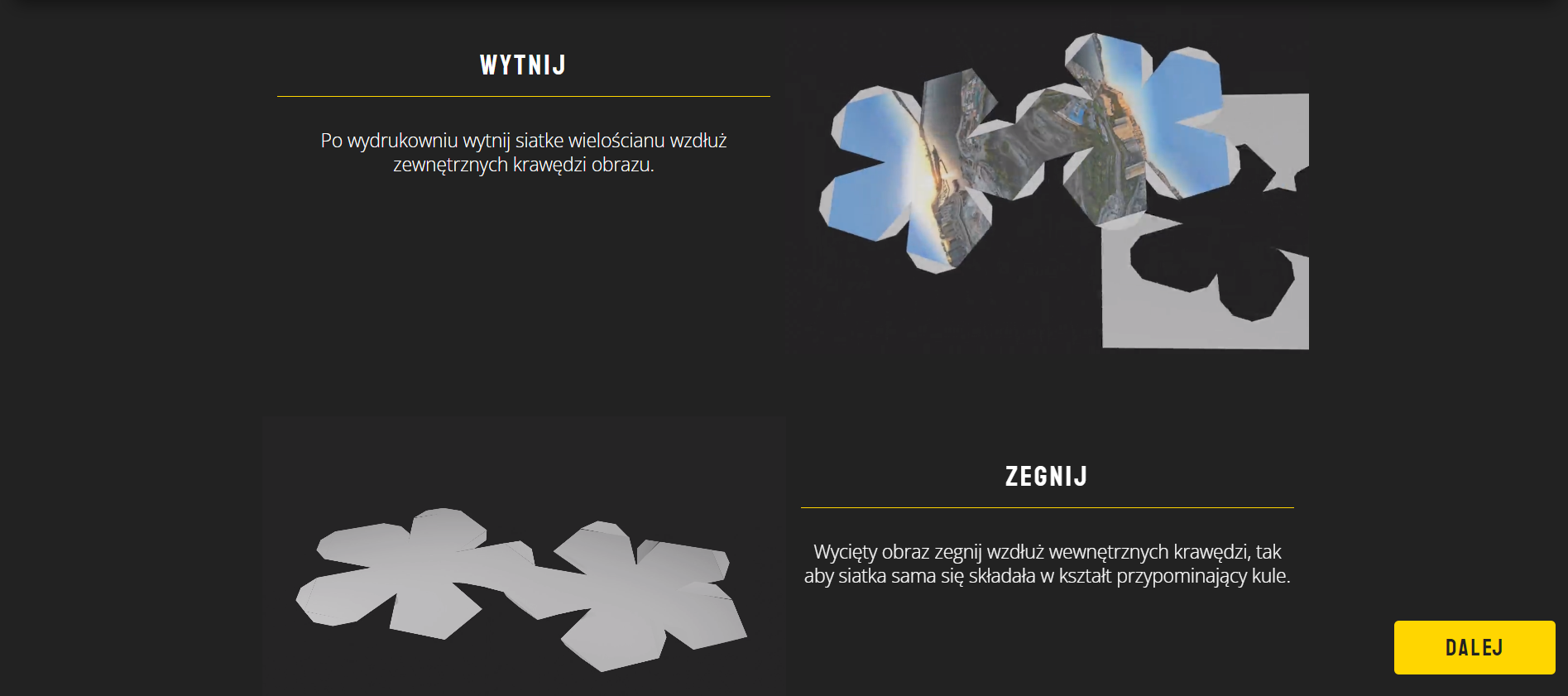
Rys. 5.3.4. Sekcja „About ???” strony głównej

Trzecią już częścią strony głównej jest instrukcja. Podzielona jest na dwie części. Pierwsza informuje użytkownika w jakie przyrządy warto się zaopatrzyć do skorzystania z pełnych możliwości usługi jakie oferuje aplikacja.



Rys. 5.3.5. Fragment instrukcji przedstawiający potrzebne przyrządy.

Druga część przedstawia 4 etapy z jakimi użytkownik musi się zmierzyć podczas tworzenia przestrzennej wersji wydrukowanego zdjęcia sferycznego. Każdy etap jest opisany a jego realizacja przedstawiona za pomocą animacji.



Rys. 5.3.6. Fragment przedstawiający dwa etapy instrukcji strony głównej.

Czwarta sekcją przedstawia informacje o aplikacji umożliwiającej zrobienie własnego zdjęcia sferycznego. Jej celem jest uświadomienie użytkownika o możliwości zrobienia takiego zdjęcia za pomocą telefonu komórkowego z aparatem fotograficznym, co ma zwiększyć szanse na przyszłe skorzystanie z tej aplikacji webowej. Głównym elementem tej sekcji jest ciemny prostokąt zawierający wcześniej wspomniane informacje oraz link do tej aplikacji. Kliknięcie w dowolnym miejscu przenosi użytkownika na stronę stronę internetową.



Rys. 5.3.7. Widok sekcji „zdjęcie sfertyczne” strony głównej.

Na dole homepage’a umieszczony jest duży przycisk przekierowujący do podstrony realizującej drukowanie a pod nim stopka strony w której umieszczone są informacje o prawach autorskich.

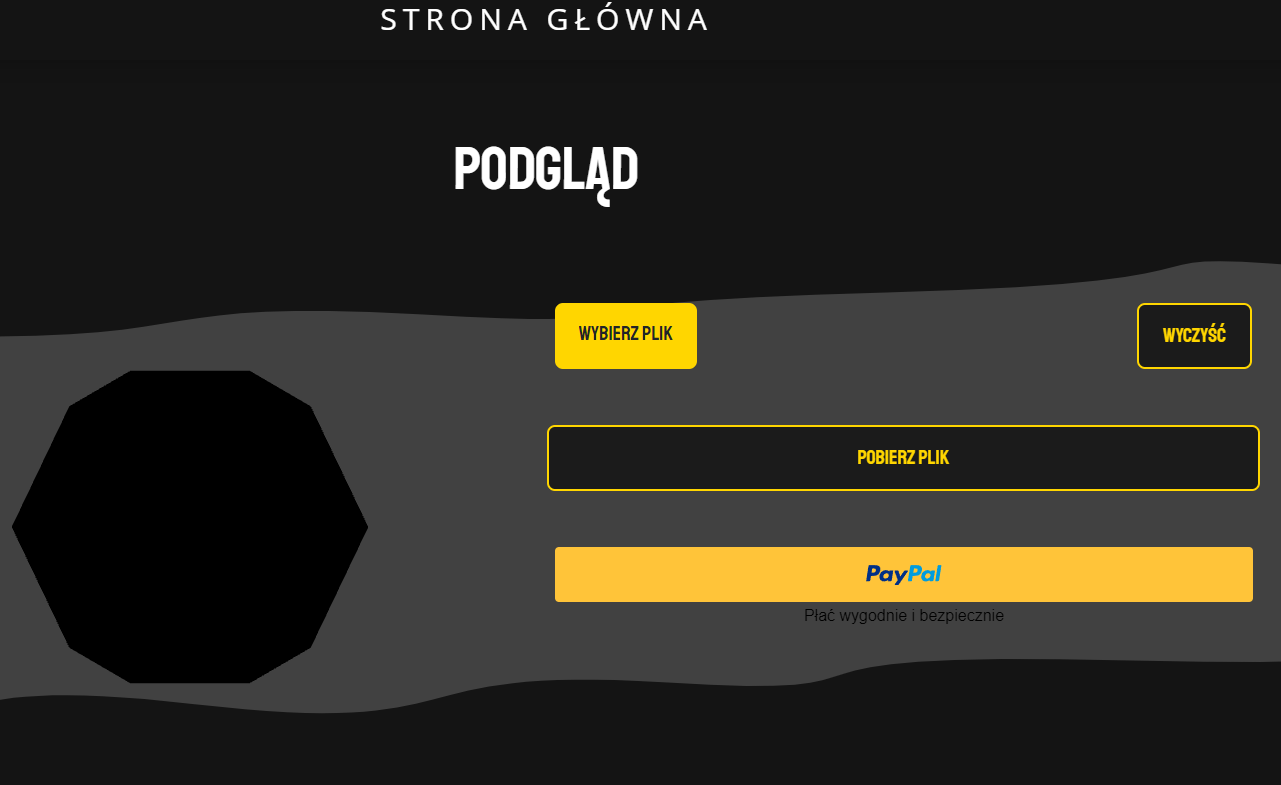
### Strona wydruku

Strona wydruku jest to funkcjonalna część aplikacji webowej, która jest podzielona na dwie podstrony jedna z nich zostaje wyświetlona po zakupie usługi. Zawiera system kupna usługi, mapowania zdjęcia na siatkę 2-D, czyszczenia zmapowanego zdjęcia oraz możliwość pobrania tejże siatki w formacie PDF. Możliwość powrotu na stronę główną został rozwiązany za pomocą nawigacji. Kliknięcie w konkretny element pozwala na powrót na stronę główną aplikacji webowej, same menu nawigacji zostało przyklejone (ang. Sticky) do góry strony co ułatwia użytkownikowi korzystanie ze strony.

Rys. 5.3.8. Menu nawigacji na stronie wydruku.

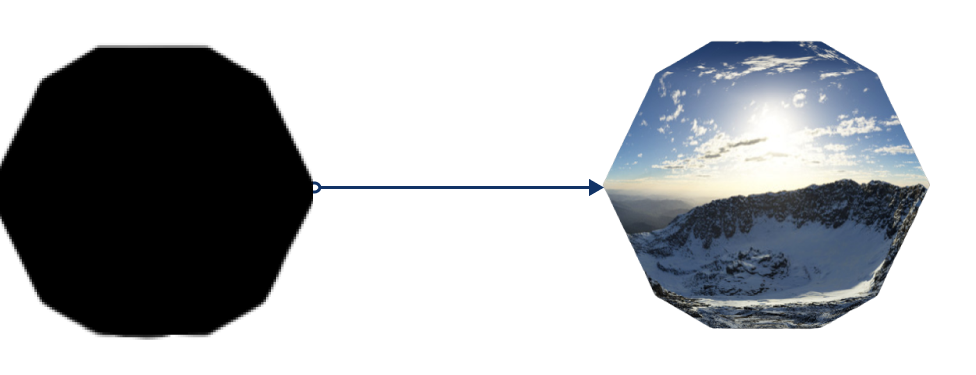
Rys. 5.3.9. Menu nawigacji strony wydruku w wersji desktopowej.

Po załadowaniu strony pojawia się sekcja która mieści całą funkcjonalność danej podstrony. Zawiera dwunastościan na który można nałożyć wybrane zdjęcia dzięki czemu możliwa jest wizualizacja efektu końcowego oferowanej usługi.



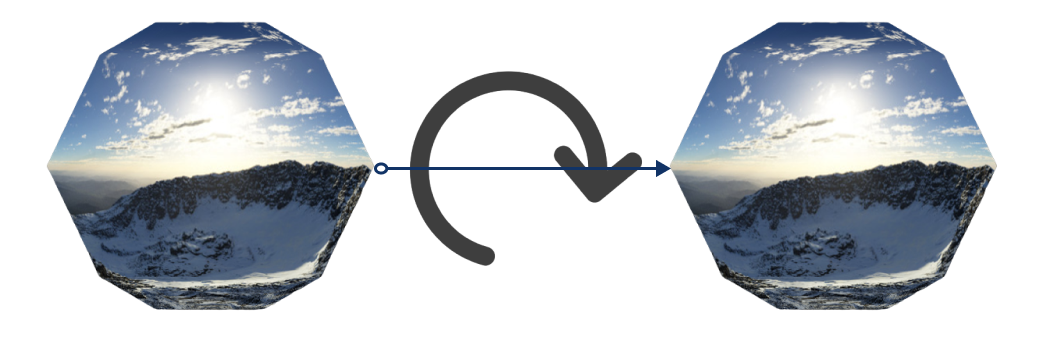
Rys. 5.3.10. Strona wydruku w wersji desktopowej

Mapowanie jest możliwe za pomocą przycisku wybierz plik który znajduje się w centrum strony. Użytkownik ma możliwość wyboru zdjęcia do mapowania, samo zdjęcie musi być plikiem graficznym. Po zmapowaniu dane zdjęcie pojawia się na dwunastościanie.



Rys. 5.3.11. Zmiana dwunastościanu spowodowana mapowaniem zdjęcia..

Zmapowane zdjęcie jest zapisywane automatycznie w asynchronicznym magazynie danych dzięki czemu wrażenia użytkowania aplikacji webowej bez połączenia (ang. Offline) są poprawione. Dzięki użyciu magazynu danych jakim jest localForage przypadkowe odświeżenie strony lub nagły zanik prądu nie przeszkadza w posługiwaniu się strona internetową.



Rys. 5.3.12. Brak zmiany dwunastościanu po przeładowaniu strony.

Gdy zdjęcie zostało nałożone przycisk wybierz plik zmienia swój tryb na tryb wyłączony (ang. disabled) co jest sygnalizowane zmianą koloru przycisku.



Rys. 5.3.13. Zmiana stanu przycisku wybierz plik.

Czyszczenie zmapowanego zdjęcia jest wykonywane poprzez przycisk wyczyść znajduje się w prawej części strony posiada dwie funkcjonalności jedną z nich jest czyszczenie zmapowanego zdjęcia oraz drugą funkcjonalnością jest przywracanie przycisku wybierz plik na tryb włączony. Jednym z główniejszych problemów związanych z czyszczeniem zmapowanego zdjęcia było nie adekwatne wyświetlanie dwunastościanu po wyczyszczeniu. Problem został rozwiązany dodając do przycisku wyczyść kolejną funkcjonalność jaką jest przeładowywanie strony po dokonanym czyszczeniu zmapowanego zdjęcia. Sam przycisk wyczyść również sygnalizuję wyglądem swój tryb jeśli nie mamy zmapowanego zdjęcia to przycisk wyczyść jest w trybie wyłączonym lecz gdy zostaje zmapowane zdjęcie przycisk wyczyść zmienia swój tryb na włączony.



Rys. 5.3.14. Zmiana stanu przycisku wyczyść.

Dzięki przyciskowi PayPal który znajduję się w dolnej części strony użytkownik aplikacji webowej ma możliwość zakupu usługi oraz przekierowania do strony z wydrukiem. Ponad przyciskiem PayPal znajduje się przycisk pobierz plik dzięki któremu możemy w prosty sposób pobrać plik PDF ze zmapowanym zdjęciem sferycznym. Sam przycisk pobierz plik również wskazuję wyglądem swój tryb jeśli nie mamy zmapowanego zdjęcia to przycisk pobierz plik jest w trybie wyłączonym lecz gdy zostaje zmapowane zdjęcie przycisk pobierz plik zmienia swój tryb na włączony.



Rys. 5.3.15. Zmiana stanu przycisku pobierz plik.

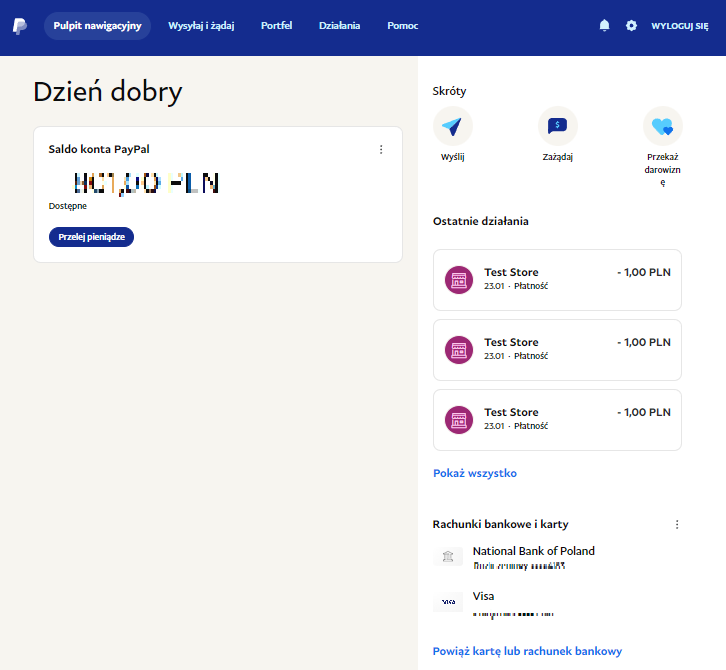
### System płatności

Aby umożliwić użytkownikowi aplikacji webowej zakup usługi mapowania wybranego przez użytkownika zdjęcia, zaimplementowany został system PayPal. Możliwość kupna usługi jest wykonywana poprzez przycisk PayPal.



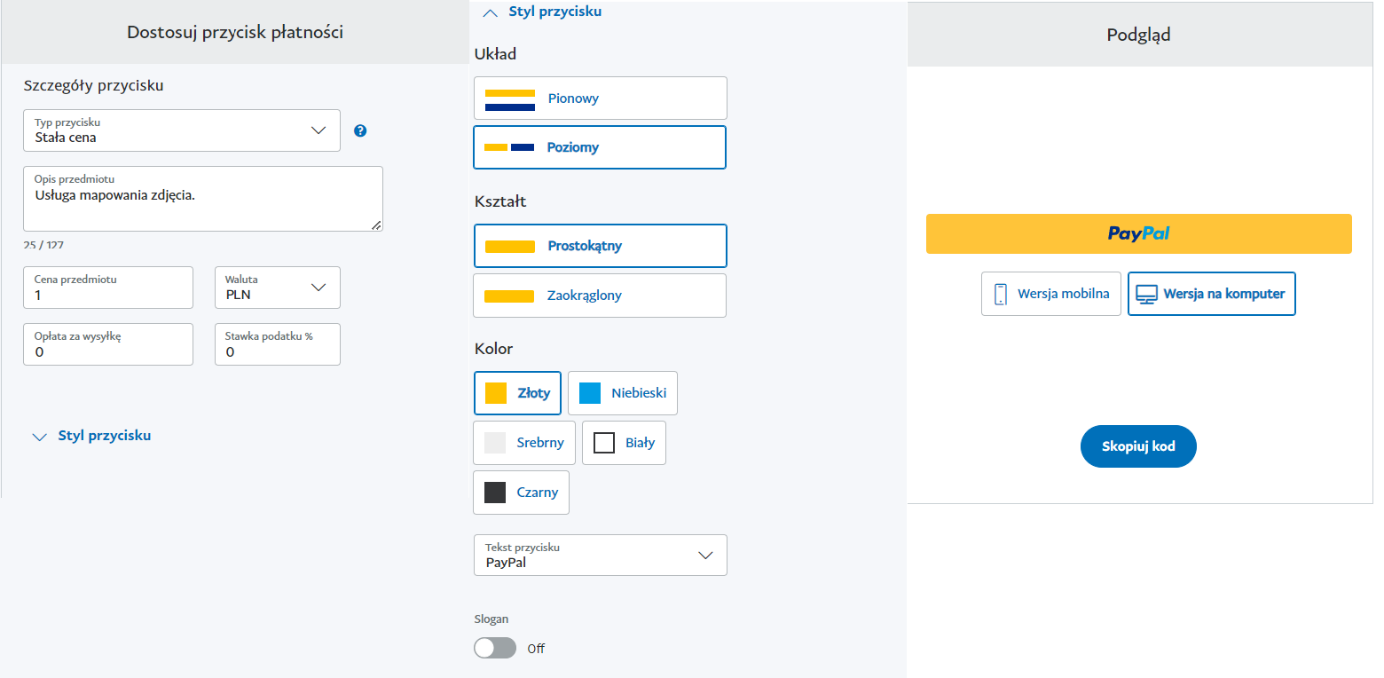
Rys. 5.3.16. Wygląd przycisku PayPal..

System PayPal jest usługą finansową, w której możliwe jest stworzenie wirtualnego portfela. Dzięki takiemu rozwiązaniu transakcje między użytkownikami systemu PayPal są bezpieczne, gdyż wszyscy sprzedawcy, którzy akceptują płatność przez środowisko PayPal muszą zostać przede wszystkim zweryfikowani dzięki czemu użytkownik może być pewny iż sprzedawcy są tymi za kogo się podają. Również jednym z głównych zalet systemu PayPal jest możliwość stworzenia wirtualnego środowiska testowego (ang. Sandbox), w którym możliwe jest symulowanie realnego środowiska PayPal. Dzięki takiemu rozwiązaniu można obserwować jak aplikacja webowa przetwarza transakcje bez tworzenia realnych kont PayPal. Dzięki pulpitowi nawigacyjnemu (ang. Dashboard) do którego użytkownik dostaję dostęp po stworzeniu wirtualnego środowiska testowego możliwe jest między innymi śledzenie statusów transakcji.



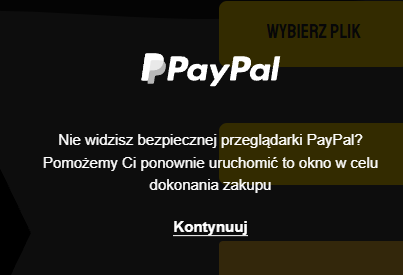
Rys. 5.3.17. Pulpit nawigacyjny środowiska sandbox PayPal.

Na głównej stronie internetowej PayPal-a możliwe jest konfigurowanie przycisku kupna usługi można wybrać typ przycisku, między stałą cena a zmienna ceną, dodać opis sprzedawanej usługi, wyznaczyć cenę usługi, walutę w jakiej będzie przebiegać transakcja, dodać opłatę za wysyłkę oraz stawkę podatku w procentach również można wybrać układ przycisków pionowy albo poziomy oraz sam kształt przycisku prostokątny albo zaokrąglony jaki i również można wybrać kolor przycisku.



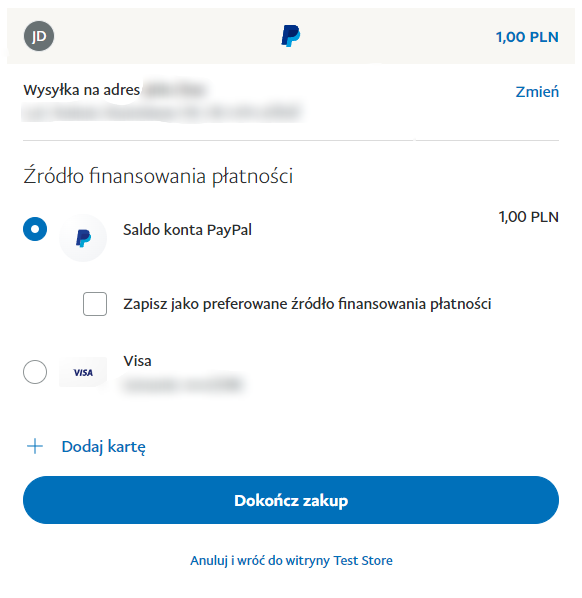
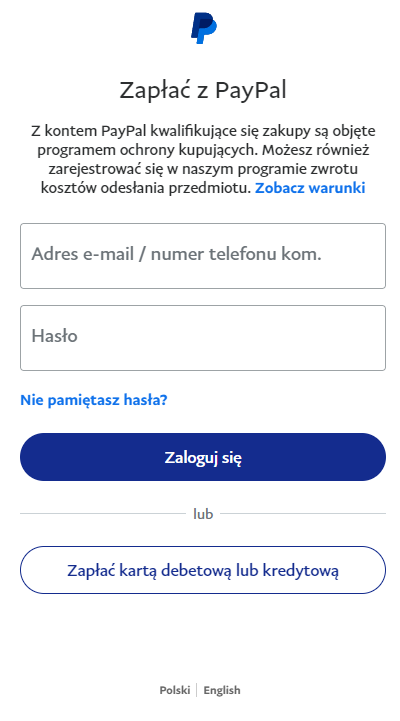
Rys. 5.3.18. Konfiguracja przycisku PayPal

Usługa kupna jest automatycznie wykonywana na oddzielnej stronie PayPal co jest sygnalizowane tekstem wyświetlanym po naciśnięciu na przycisk PayPal.



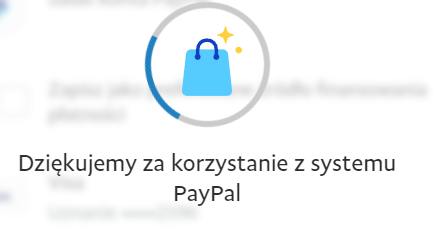
Rys. 5.3.19. Tekst sygnalizujący PayPal

Po przekierowaniu na bezpieczną przeglądarkę PayPal, użytkownik proszony jest o zalogowanie się do systemu PayPal w celu uregulowania należności za zakup usługi mapowania. Możliwe jest uregulowanie należności poprzez saldo konta PayPal lub kartę płatniczą.

. 

Rys. 5.3.20. Widok logowania / zakupu poprzez PayPal.

Po wprowadzeniu wszystkich potrzebnych informacji użytkownik klikając przycisk dokończ zakup jest informowany o przetwarzaniu transakcji.



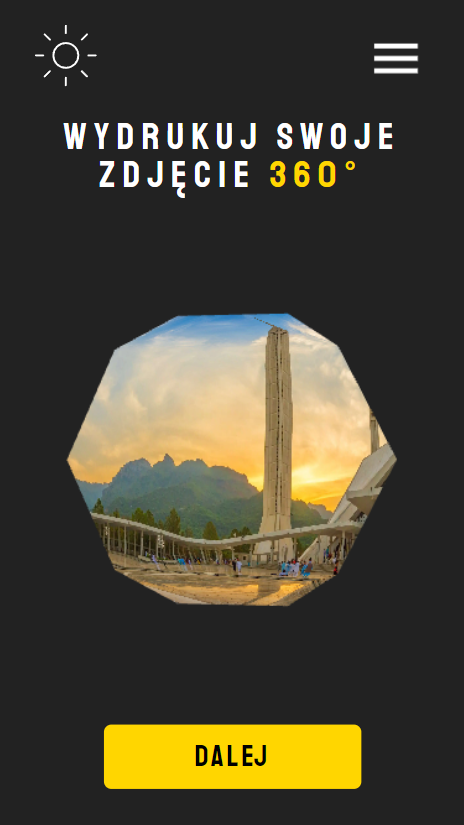
Rys. 5.3.21. Przetwarzanie transakcji.

Jeśli transakcja powiodła się użytkownik jest przekierowywany automatycznie na stronę z możliwością wydruku zmapowanego zdjęcia lecz gdy transakcja nie powiodła się użytkownik powraca na stronę z możliwością zakupu usługi.

### Wersja mobilna

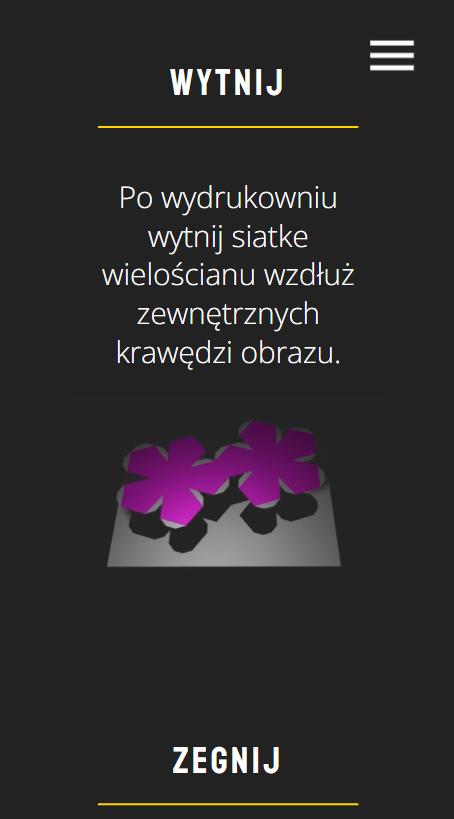
Aplikacja webowa jest przystosowana do użytku na wielu urządzeniach. Jest elastyczna i dostosowuje się do różnych rozmiarów ekranów, dzięki czemu jest ona czytelna i łatwa do nawigacji na różnych urządzeniach, takich jak smartfony, tablety, komputery oraz innych urządzeń z wyświetlaczem. Zmiana rozdzielczości skutkuje zmianą wielkości elementów strony oraz ich pozycjonowania. Poniżej wymieniono elementy dla których zaszły najistotniejsze zmiany.

* Hero image na urządzeniach o mniejszej rozdzielczości o pionowych ekranach zmieniło tło z napisu na jednolity kolor. Przycisk zmienił swoją pozycje z prawej strony na centrum.
* Nawigacja na urządzeniach mobilnych zmieniła swój kształt z poziomego paska szerokiego na cały ekran w znajdującą się w prawym górnym rogu drobną ikonę tzw. hamburgera (ang. hamburger menu) zbudowaną z trzech poziomych linii jedna nad drugą. Kliknięcie tej ikony powoduję rozwinięcie się menu nawigacyjnego, które zawiera odwołania do trzech sekcji strony. Składane menu pozwala zaoszczędzić dużo miejsca co jest szczególnie przydatne na urządzeniach mobilnych.

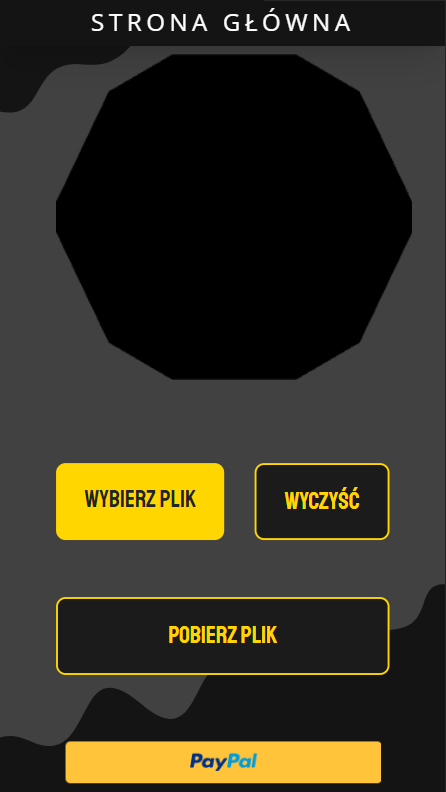
Rys. 5.3.22. Widok sekcji hero image w wersji mobilnej. Rys. 5.3.23. Rozwinięte menu nawigacji w wersji mobilnej

* Elementy instrukcji pozycjonowanie z poziomego na pionowe. Pod opisem etapu z instrukcji znajduję się pomniejszona animacja. Pozwala to na uzyskanie większego rozmiaru animacji co za tym idzie lepszej jakości co przenosi się na czytelniejsze przeglądanie strony.

Rys. 5.3.24. Fragment instrukcji przedstawiający Rys. 5.3.25. Fragment przedstawiający etapy instrukcji. potrzebne przyrządy w wersji mobilnej

* Na stronie na, której znajduję się system mapowania zdjęcia oraz pobrania zmapowanego zdjęcia jako plik PDF pozycjonowanie danych elementów takich ja przyciski oraz obiekt pozwalający na dynamiczne, skryptowe renderowanie figur i widoków bitmapowych (ang. Canvas) zmienia się z pozycjonowania poziomego na pozycjonowanie pionowe.



Rys. 5.3.26. Strona wydruku wersji mobilnej

# Implementacja systemu

## Środowisko i narzędzia programistyczne

Środowisko programistyczne, które zostało użyte przy tworzeniu aplikacji webowej to Visual Studio Code. Jest to zaawansowane narzędzie programistyczne pozwalające na tworzenie miedzy innymi aplikacji webowych. Narzędzie to zostało stworzone przez firmę Microsoft oraz dalej jest przez nią rozwijane. Visual Studio Code oferuję ogromną liczbę narzędzi, wyszczególniamy zwłaszcza edytor kodu źródłowego który posiada wbudowaną obsługę miedzy innymi JavaScript lub Node.js, bogaty wybór dodatków (ang. Plugin) które rozszerzają ekosystem języków programowania (takich jak Python, Java, C++, C#) oraz uzupełnianie kodu IntelliSense jest to funkcja która przewiduję kolejną część kodu na podstawie bieżącego kodu i przedstawia te przewidywanie jako szary tekst. Języki HTML, CSS oraz JS zostały wybrane jako główne narzędzia, w którym pisany był kod aplikacji webowej. Figma czyli aplikacja do prototypowania i projektowania UI/UX (ang. user interface / user experience) bardzo dobrze uzupełnia lukę która powstaję przy tworzeniu stron internetowych tylko za pomocą narzędzi programistycznych takich jak na przykład Visual Studio Code. Luką tą jest brak możliwości wizualizacji efektu końcowego strony internetowej bez używania jakiegokolwiek języka programowania. Sama Figma została stworzona przez dwóch przyjaciół, Dylana Fielda oraz Evana Wallace’a w 2012 roku. Figma kładzie nacisk na pracę zespołową, która może być wykonywana w czasie rzeczywistym. Udostępnianie plików, komentowanie, integracja z czatem grupowym Slack oraz historia wersji projektu pozwalających na powrót do poprzednich faz projektu umożliwia efektywną współpracę z innymi projektantami oraz z klientami.

Dodatkowym oprogramowaniem które zostało użyte w tworzeniu aplikacji webowej jest Blender. Blender został stworzony przez firmę NeoGeo lecz od 2002 jest rozwijany przez firmę Blender Foundation. Aplikacja ta jest wykorzystywana do tworzenia zaawansowanej grafiki 2D i 3D oraz tworzenia między innymi animacji. Blender daje ogromne możliwości lecz krzywa uczenia (ang. learning curve) się tego oprogramowania jest stroma gdyż Blender posiada bardzo rozbudowany interfejs który dla początkujących użytkowników możę sprawiać problemy w nauce. W aplikacji webowej zostało wykorzystane localStorage oraz localForage jako forma przechowywania danych dzięki którym doświadczenia użytkowania aplikacji webowej są polepszone oraz również system kontroli wersji Git dzięki któremu możliwa jest kontrola postępów w kreowaniu kodu źródłowego oraz zapisywanie wcześniejszych wersji kodu.

## Modele figury

### Reprezentacja modelu

Model dwunastościanu został zaimplementowany za pośrednictwem interfejsu programowania aplikacji WebGL. Dane figury są przechowywane w VBO (Vertex Buffer Object) za pośrednictwem trzech buforów: dwóch buforów wierzchołków (ARRAY-BUFFER) oraz buforu indeksów (ELEMENT\_ARRAY\_BUFFER).

* Pierwszy z buforów wierzchołków przechowuje informacje o wierzchołkach. Są to punkty w przestrzeni 3D, które reprezentują pozycje i położenie wszystkich elementów modelu. Składają się z trzech wartości x, y, z i służą do tworzenia trójkątów z których zbudowany jest dwunastościan.
* Drugi bufor wierzchołków zawiera współrzędne UV dwunastościanu. Są to dwuwymiarowe punkty określające położenie punktu na obrazie. Są przypisywane do każdego wierzchołka po czym interpolowane między nimi, określając położenie tekstury na całym trójkącie.
* Bufor indeksów jest specjalnym rodzajem VBO przechowującym indeksy wierzchołków. W dwunastościanie każdy wierzchołek należy do trzech trójkątów. Wprowadzenie indeksowania umożliwiają przesyłanie tylko unikalnych wierzchołków zamiast każdego z osobna. Poprawia to wydajność systemu poprzez zmniejszenie ilości danych przesłanych do karty graficznej.

Wszystkie trzy bufory są powiązane z programem cieniowania (ang. shader program), który odpowiada za przetwarzanie danych oraz rastrowanie pikseli na ekranie. Składa się z dwóch rodzajów shaderów: Vertex Shader oraz Fragment Shader.

* Vertex shader odpowiada za przetwarzanie danych wierzchołków.
* Fragment Shader odpowiada za przetwarzanie danych tekstury oraz określenie kolorów każdego piksela na ekranie.

(dopisać o glsl???)

    gl.attachShader(shaderProgram, vertexShader)

    gl.attachShader(shaderProgram, fragmentShader)

### Model przestrzenny

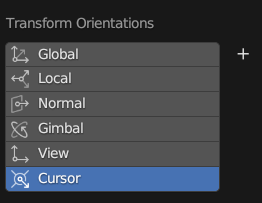
Przestrzenna wersja modelu dwunastościanu jest w całości zbudowana w WebGL’u. Współrzędne wierzchołków oraz indeksy zostały ręcznie przypisane na podstawie obliczeń matematycznych. Przechowywane są w specjalnych strukturach danych nazwanych tablicami. Obie tablice zawierają po 114 elementów.

Każda liczba w pierwszej tablicy reprezentuje jedną ze współrzędnych x, y lub z. Są one ułożone w takiej samej kolejności a trzy z nich reprezentują jeden punkt. Całą tablica zawiera 38 wierzchołków, z których dodatkowych 18 zostało stworzonych do poprawnego mapowania tekstury (Rozdział 5.2.2.).

Druga tablica zawiera ciąg indeksów numerowanych od 0 do 37. Pojedynczy indeks odnosi się do trzech koordynatów z poprzedniego zboru danych, więc pojedynczy indeks reprezentuje jeden wierzchołek dwunastościanu, a trzy z nich cały trójkąt.

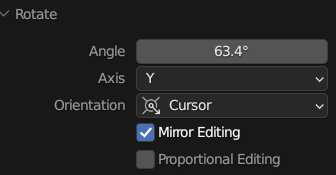
### Model płaski

Model płaski jest zbudowany w oparciu o jego przestrzenną wersje. Powstał poprzez rozłożenie ścian bryły na dwuwymiarową powierzchnie w programie Blender. Każda ściana została obrócona za pomocą narzędzia rotacji. Do transformacji kąta konieczne jest ustawienie orientacji wzdłuż kursora oraz przypisanie go do krawędzi zawierającej kąt miedzy ścienny w obracanej powierzchni.



Rys. 6.2.1. Ustawienie orientacji dla kursora.

Kąt rotacji jest równy 63.4°. Do utworzenia wstępnego modelu siatki każda ze ścian musi przejść przez taki proces do czasu aż współrzędna Z każdego punktu będzie równa lub bardzo zbliżona do liczby 0.



Rys. 6.2.2. Konfiguracja narzędzia rotacji.

Ostanim krokiem przy implementacji siatki w programie graficznym jest stworzenie elementów przeznaczonych do klejenia. Każdy z nich jest identyczny i są zbudowany na kształt trapeza. Dłuższa krawędź trapezu jest połączona z zewnętrzną krawędzią pięciokąta przy użyciu narzędzia łączenia (ang. merge).

Stworzona siatka dwunastościanu jest eksportowana do pliku OBJ i importowana w kodzie programu. Wartości współrzędnych wierzchołków, indeksów oraz koordynatów UV są wpisane w tablice, które są powiązane z buforami danych.

## System mapowania

Implementacja mapowania jest określona za pomocą jednej funkcji. Stworzone są w niej dwie zmienne lokalne:

- position: zawiera przypisane wartości współrzędnych wierzchołków dwunastościanu

- vertexCoords: tablica przechowująca współrzędne tekstury.

Uzupełnianie elementów tablicy współrzędnych uv odbywa się w pętli for. Pętla powtarza się 38 razy, tyle ile jest wierzchołków w dwunastościanie.

for(let i = 0; i < position.length; i += 3)

Do nowo utworzonych zmiennych X, Y, Z przypisane są wartości kolejnych elementów tablicy będących współrzędnymi jednego punktu. Po każdym powtórzeniu się pętli wartości zmiennych są nadpisywane kolejnymi współrzędnymi wierzchołków.

const X = position[i+0];

const Y = position[i+1];

const Z = position[i+2];

Na ich podstawie za pomocą funkcji arctan2(-Z, X) oraz arcsin(Y/R) wyliczany jest kolejno kąt azymutalny oraz kąt elewacji.

const fi = Math.atan2(-Z, X); //atan(-Z/X)

const phi = Math.asin(Y / Math.sqrt(X\*X+Y\*Y+Z\*Z)); //asin(Y/|R|)

Wyniki tych działań są skalowane do przedziału (0, 1), wpisane w zmienne u,v i dodawane do tablicy, która zawiera wszystkie koordynaty tekstury.

const u = fi / (2 \* Math.PI) + 0.5;

const v = phi / (Math.PI) + 0.5;

vertexCoords.push(...[u,v]);

Wynikiem funkcji mapowania jest zwrócenie całej tablicy vertexCoords, która jest przechowywana w buforze wierzchołków (VBO).

## Aplikacja webowa

### Wczytywanie modelu (objloader)

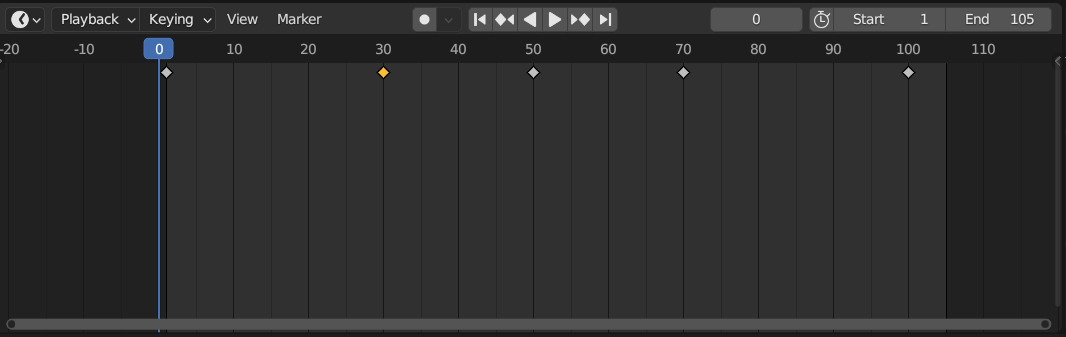
### Zapisywanie Modelu (pdf)

### Animacje

Implementacja animacji procesem składającym się z dwóch etapów:

* Stworzenie w Blenderze,
* Zaimplementowanie w kodzie.

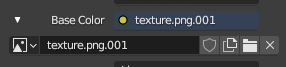
W pierwszym etapie głównym narzędziem jest Timeline, które pozwalana przypisanie położenia obiektu oraz kamery w danej klatce. Każda ze czterech animacji zawiera od 105 do 250 obrazów w zależności od jej złożoności. Najprostsza przedstawiająca wycinanie siatki z powierzchni kartki jest złożona ze 105, a reszta z 250 klatek.



Rys. 6.4.1. Oś czasu w narzędziu timeline

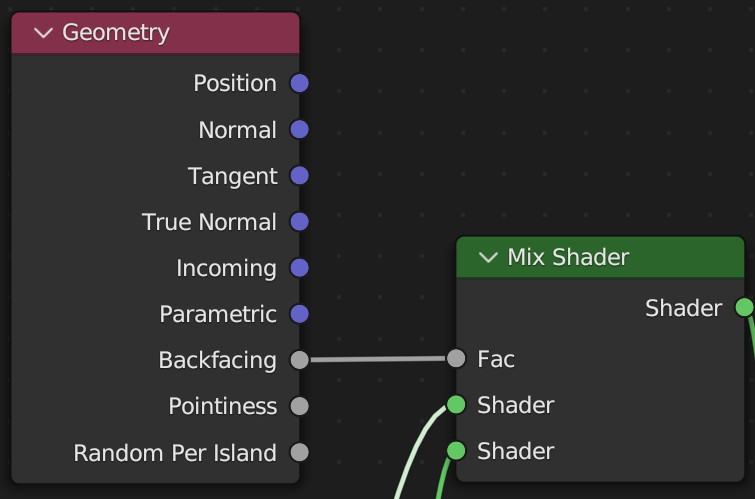
Kolejnym narzędziem jest compositing, które pozwala na dokonanie zmian w materiale obiektu. Dwunastościan przed rozpoczęciem animacji musi zostać należycie skonfigurowany. Do zmian jakich doszło to:

- przypisanie tekstury, które zaimplementowano poprzez ustawienie zdjęcia jako koloru głównego w obiekcie.

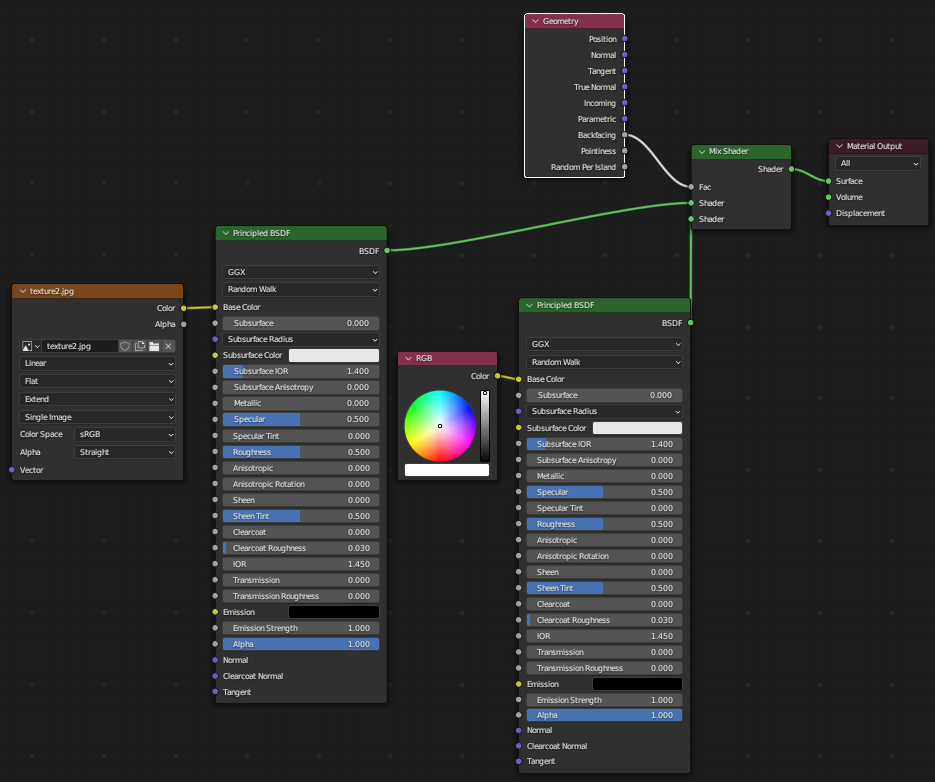


Rys. 6.4.2. Przypisanie tekstury jako koloru bazowego dla dwunastościanu.

- zmiana geometrii, która oprócz modyfikacji współrzędnych w jakich znajdują się wierzchołki w danej klatce dostosowuje mapowanie tekstury. Domyślnie w blenderze obraz jest nakładany po obu stronach ścian obiektu. Do poprawnego odzwierciedlenia dwunastościanu zdjęcie musi zostać nałożone jednostronnie, aby druga strona reprezentowała pustą część kartki. Dokonano to poprzez dodanie funkcji backfacing. (rys.)

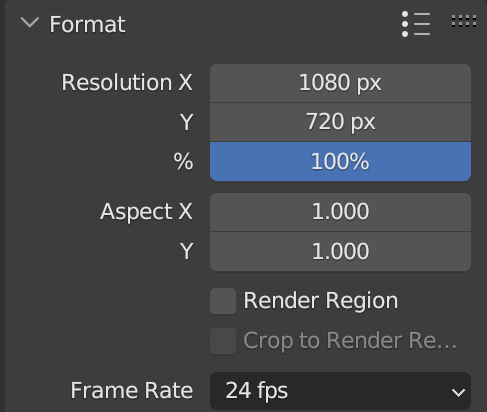


Rys. 6.4.3. Dodanie funkcji backfacing.



Rys. 6.4.4. Właściwości materiału dwunastościanu w narzędziu „compositing

Ostatnim etapem jest wyrenderowanie serii obrazów do postaci wideo. Zostało to zrobione w rozdzielczości 1080x720 pikseli z częstotliwością 24 klatek na sekundę.



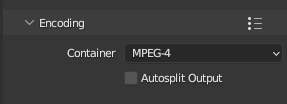
Rys. 6.4.5. Ustawienia rozdzielczości oraz częstotliwości animacji.

Zastosowany format to FFmpeg video będący otwartoźródłowym narzędziem do kodowania i dekodowania wideo. Pozwala na eksportowanie animacji w formacie mp4.



Rys. 6.4.6. Ustawienia formatu animacji.

Do kompresji danych użyto standardu MPEG-4 z kodekiem H.264 co pozwoliło na zmniejszenie rozmiaru przy zachowaniu odpowiedniej jakości obrazu. Wyjsciowa jakość wideo została zmniejszona do niskiej w celu uzyskania jak najmniejszego rozmiaru pliku. Poprzez niewielkie rozmiary oraz złożoność animacja nie utraciła wiele na jakości.

Rys. 6.4.8. Ustawienia kodeka dla animacji.

### System płatność i cache

Aby zaimplementować możliwość kupna usługi poprzez system PayPal najpierw trzeba posiadać identyfikator klienta aplikacji PayPal (ang. Client ID) oraz tajny klucz (ang. Secret Key) które znajdują się w pulpicie nawigacyjnym systemu PayPal. Za pomocą tych kluczy użytkownik ma możliwość tworzenia tokenu dostępu (ang. Access Token), dzięki któremu upoważniany jest dostęp do korzystania z PayPal REST API. Przykładowa odpowiedź serwera na żądanie zdobycia tokenu dostępu obrazuję budowę odpowiedzi. W "scope" otrzymywana jest informacja o liście uprawnień związanych z danym tokenem dostępu, następnie "access\_token" zawiera informację o tokenie, który został użyty do wywołania punktu końcowego informacji o użytkowniku, w "token\_type" dostarczana jest informacja o typie tokena, a w " expires\_in " zostaje określona liczba sekund do wygaśnięcia tokena dostępu oraz w " nonce " otrzymywany jest losowy ciąg który jest używany do zapobiegania atakom.

|  |
| --- |
|  |

Następnie używając uzyskany token dostępu możliwe jest tworzenie zamówienia oraz przechwycenie płatności związanej z danym zamówieniem.

# Testy subiektywne

# Podsumowanie

# Bibliografia

S. Krug, Don't make me think, revisited: A common sense approach to web usability., 2014.

Dizajn na co dzien