

WYDZIAŁ PODSTAWOWYCH PROBLEMÓW TECHNIKI
POLITECHNIKA WROCŁAWSKA

GRAFIKA NIEEUKLIDESOWA NA PRZESTRZENI DWUWYMIAROWEJ

MACIEJ HAJDUK

NR INDEKSU: 236596

Praca inżynierska napisana
pod kierunkiem
Prof. dr hab. Jacka Cichonia



Politechnika
Wrocławska

WROCŁAW 2019

Spis treści

Wstęp	3
Kontekst historyczny	3
Wybrane zagadnienie	4
Analiza problemu	7
Podstawowy podział	7
Geometria Łobaczewskiego-Bólyaia (hiperboliczna)	7
Geometria Riemanna (eliptyczna)	8
Różnice pomiędzy geometriami	8
Najpopularniejsze modele	9
Model Kleina	9
Model półpłaszczyzny Poincaré	10
Model dysku Poincaré	11
Model Hemisfery	12
Uzasadnienie wyboru modelu dysku Poincaré	13
Projekt systemu	15
Cykl pracy silnika	15
Klasy obiektów	16
Klasa Line	16
Klasa Point	16
Klasa Circle	17
Klasa Plane	17
Implementacja systemu	19
Opis technologii	19
Instalacja i wdrożenie	19
Serwer deweloperski	19
Podsumowanie	22
Bibliografia	24
Zawartość płyty CD	26

Wstęp

Kontekst historyczny

We wszystkich szkołach od 2500 lat nauczana jest geometria. Sama nazwa - geometria - kojarzyć się może z nauką przyrodniczą, z rzeczą poznawaną na codzień z doświadczenia. Taka właśnie geometria jest również czymś bardzo instynktownym i dobrze wydawałoby się poznanym. Nasuwa się pytanie, czy może istnieć jakaś konkurencyjna do niej teoria.

Sama nazwa tej geometrii - geometria euklidesowa, bierze się z faktu, że została ona sformułowana aksjomatycznie w dziele *Elementy* około 300 roku przed naszą erą, przez dyrektora Biblioteki Aleksandryjskiej - Euklidesa. Euklides przedstawił 5 postulatów z których wyprowadził całą geometrię, jaką przez kolejne wieki znano:

1. Dowolne dwa punkty można połączyć odcinkiem.
2. Dowolny odcinek można przedłużyć nieograniczenie (uzyskując prostą).
3. Dla danego odcinka można zaznaczyć okrąg o środku w jednym z jego końcowych punktów i promieniu równym jego długości.
4. Wszystkie kąty proste są przystające.
5. Dwie proste, które przecinają trzecią w taki sposób, że suma kątów wewnętrznych po jednej stronie jest mniejsza od dwóch kątów prostych, przetną się z tej właśnie strony.

Piąty aksjomat wywołał wiele wątpliwości. Nawet Euklides unikał używania go w swoim dziele tak długo, jak to było możliwe. Przez kolejne 1500 lat kolejni matematycy próbowali udowodnić, że o wiele bardziej skomplikowany postulat musi wynikać z pozostałych czterech.

Jednym z pierwszych zajmujących się tym problemem uczonych, był żyjący w V wieku naszej ery Proklos. Stwierdził on w swoim komentarzu do dzieł Euklidesa:

Nie jest możliwe, aby uczony tej miary co Euklides godził się na obecność tak długiego postulatu w aksjomatyce – obecność postulatu wzięła się z pośpiesznego kończenia przez niego *Elementów*, tak aby zdążyć przed nadejściem słusznie oczekiwanej rychłej śmierci; my zatem – czcząc jego pamięć – powinniśmy ten postulat usunąć lub co najmniej znacznie uprościć.

Wyzwanie usunięcia piątego aksjomatu podjęło wielu matematyków w kolejnych wiekach, znane są dowody Ibn al-Haythama (Alhazen, XI wiek), Omar Khayyáma (XII wiek), Nasir al-Din al-Tūsī (XIII wiek) i Giovanni Girolamo Saccheriego (XVIII wiek), wszystkie jednak, jak się później okazało, były błędne. Fakt, że mimo prostych błędów funkcjonowały i były nawet nauczane, wskazuje na kłopot, jakim dla uczonych było przyjęcie do wiadomości, że mogą istnieć dwie różne a wręcz wykluczające się, ale jednocześnie poprawne geometrie, a więc dwie teorie opisujące w różny sposób ten sam

obiekt. W szczególności Immanuel Kant w swoim dziele *Krytyka czystego rozumu* stwierdził, że intuicja geometryczna jest wrodzona, więc nie może istnieć wiele równoległych geometrii, a każdy kto chciałby zajmować się alternatywnymi geometriami nie nadaje się do nauki. Spotkało się to ze sprzeciwem. Johann Heinrich Lambert, zajmujący się taką alternatywną geometrią, ogłosił, że jeśli to nie jest nauka, to on chce uprawiać nienaukę. Udano się do największego w tamtym czasie autorytetu - Carla Friedricha Gaussa, ten jednak wycofał się, bojąc się - jak pisał - wrzasku Boetów. Problem jednak był i należało się do niego jakoś odnieść. Odważyło się na to dwóch młodych ludzi, którzy uparli się nie tylko na uprawianie tej geometrii, ale wręcz głosili jej równoprawność. Rosjanin, Nikołaj Łobaczewski oraz Węgier - Janos Bolyai, niezależnie od siebie opublikowali prace w których - chociaż odmiennie - nowa geometria była konsekwentnie wyprowadzona. Obu odkrywców spotkała też za to kara, Łobaczewski został wręcz zmuszony do opuszczenia katedry.

Sprawę nowej geometrii (nazywanej już geometrią Bolyaia-Łobaczewskiego) przejął Felix Klein. Postawił on tezę, że jeżeli za pomocą geometrii euklidesowej jesteśmy w stanie przedstawić tę nieeuklidesową - i odwrotnie, to oba modele są sobie w istocie równoważne. Opublikował też w 1870 roku dzieło, w którym dowiódł równoprawności obu modeli.

Dosadnie do nowego modelu odniósł się fizyk - Hermann Helmholtz, publikując pracę, w której określił matematykę jako skrzynkę z narzędziami dla nauk przyrodniczych, czym odebrał jej walor nakuki przyrodniczej jako takiej.

Wybrane zagadnienie

W niniejszej pracy zaimplementowany zostanie prosty silnik graficzny skupiający się na renderowaniu wizualizacji płaszczyzny dysku w modelu Poincarégo geometrii hiperbolicznej.

Praca swoim zakresem objmie obsługę rysowania linii, okręgów, wielokątów na tejże płaszczyźnie oraz implementacje przykładowych programów obejmujących wizualizacje bardziej skomplikowanych struktur. Na tle innych implementacji, aplikacja wyróżnia się dostarczaniem możliwości i realizacją problemu z pomocą matematycznego opisu pewnego modelu. Przykładowe demonstracje możliwości aplikacji są dostarczone razem z kodem źródłowym, jest to, poza możliwością narysowania dowolnego wielokąta, rysowaniem figur foremnych czy prostych animacji, także interakcja z urządzeniami peryferyjnymi i tesselacja przestrzeni hiperbolicznej. Niewątpliwą zaletą dostarczonej aplikacji jest prostota implementacji własnych rozwiązań, na co składa się silne typowanie języka Typescript wraz z dokładnymi interfejsami dla klas oraz funkcje dostarczone przez silnik, pozwalające na łatwie manipulowanie wyświetlającymi się obiektami, nie wymagające przy tym zrozumienia modelu.

Praca składa się z czterech rozdziałów:

Rozdział pierwszy: W rozdziale omówiono analizę wybranego problemu, przedstawiono motywacje

podjęcia tego tematu oraz uzasadniono wybór modelu płaszczyzny Poincaré. Rozdział zawiera poza tym komentarz do różnych rodzajów geometrii nieeuklidesowych, oraz krótki opis i porównanie innych modeli geometrii hiperbolicznej.

Rozdział drugi: Rozdział zawiera szczegółową charakterystykę systemu wraz z opisem poszczególnych plików oraz przeznaczeniem klas i funkcji składających się na program. Opisane w nim zostały algorytmy przekształcające byty w geometrii Euklidesowej na odpowiadające im elementy geometrii hiperbolicznej, funkcje pomocnicze, reprezentacje punktów i linii w obu modelach.

Rozdział trzeci: W rozdziale wymieniono technologie użyte do implementacji projektu: wybrany język programowania, środowisko składające się na aplikację oraz biblioteki wykorzystane w programie.

Rozdział czwarty: Rozdział zawiera instrukcje instalacji i wdrożenia systemu w środowisku docelowym. Końcowy rozdział stanowi podsumowanie uzyskanych wyników i ewentualne możliwości rozwoju projektu.

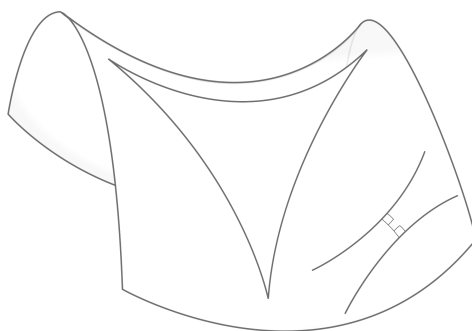
Analiza problemu

W niniejszym rozdziale przedstawiona będzie analiza problemu, opis matematyczny modelu płaszczyzny dysku Poincarégo oraz przegląd kilku wybranych modeli geometrii nieeuklidesowej.

Odkrycie, że piątego aksjomatu nigdy nie można udowodnić na podstawie pozostałych czterech aksjomatów, było dla naukowców niespodzianką. Zrobiono to, demonstrując istnienie geometrii, w której pierwsze cztery aksjomaty utrzymywały się, ale piąty nie. Debata nad piątym postulatem Euklidesa stworzyła problem, jak alternatywna geometria powinna wyglądać. Umiano pokazać zaledwie poszczególne właściwości takich geometrii. Pierwszy model geometrii nieeuklidesowej został stworzony przez Kleina. W sprawę zaangażowało się wielu matematyków, w tym również Bernard Riemann. Stwierdził on, że można opisać nieskończenie wiele struktur matematycznych, które nie będą spełniały postulatów Euklidesa, będąc dalej geometriami.

Podstawowy podział

Geometria nieeuklidesowa to każda geometria, która nie spełnia przynajmniej jednego z postulatów Euklidesa. Geometrie nieeuklidesowe możemy podzielić na dwa rodzaje:



Rysunek 1: Trójkąt oraz dwie proste przedstawione na powierzchni o geometrii hiperbolicznej

Geometria Łobaczewskiego-Bólyaia (hiperboliczna)

Geometria hiperboliczna jest bliżej związana z geometrią euklidesową, niż się wydaje: jedyną różnicą aksjomatyczną jest postulat równoległy. Po usunięciu postulatu równoległego z geometrii euklidesowej geometria wynikowa jest geometrią absolutną. Wszystkie twierdzenia o geometrii absolutnej, w tym pierwsze 28 twierdzeń zaprezentowanych przez Euklidesa, obowiązują w geometrii i euklidesowej i hiperbolicznej.

W modelu hiperbolicznym, w płaszczyźnie dwuwymiarowej, dla dowolnej linii L i punktu X , który nie jest na L , istnieje nieskończenie wiele linii przechodzących przez X , które się nie przecinają L .

Geometria Riemanna (eliptyczna)

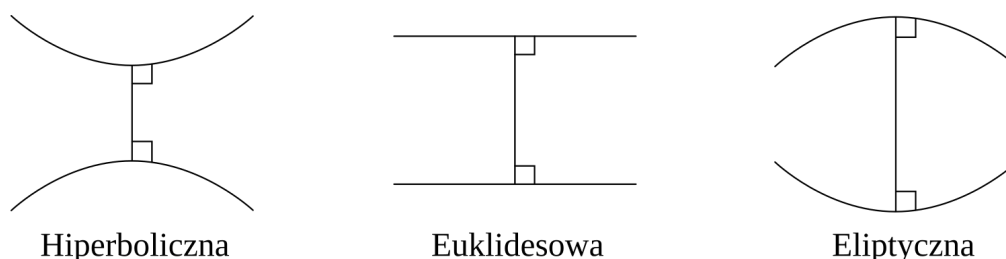
Geometria eliptyczna jest geometrią nieeuklidesową o dodatniej krzywiznie, która zastępuje postulat równoległy stwierdzeniem “przez dowolny punkt na płaszczyźnie, nie ma linii równoległych do danej linii”. Geometria eliptyczna jest czasem nazywana również geometrią Riemannowską. Model można zwizualizować jako powierzchnię kuli, na której linie przyjmowane są jako wielkie koła. W geometrii eliptycznej suma kątów trójkąta wynosi >180 stopni.

W modelu eliptycznym dla dowolnej linii L i punktu X , który nie jest na L , wszystkie linie przechodzące przez X przecinają się L .

Różnice pomiędzy geometriami

Sposobem opisania różnic między tymi geometriami jest rozważenie dwóch linii prostych rozciągniętych w nieskończoność w płaszczyźnie dwuwymiarowej, które są prostopadłe do trzeciej linii:

- W geometrii euklidesowej linie pozostają w stałej odległości od siebie (co oznacza, że linia narysowana prostopadle do jednej linii w dowolnym punkcie przecina drugą linię, a długość odcinka linii łączącego punkty przecięcia pozostaje stała) i są znane jako równoległe.
- W geometrii hiperbolicznej linie *zakrzywiają się od siebie*, zwiększając odległość w miarę przesuwania się dalej od punktów przecięcia ze wspólną prostopadłą; linie te są często nazywane ultrarównoległymi.
- W geometrii eliptycznej linie *zakrzywiają się do siebie* i w końcu przecinają.



Rysunek 2: Zachowanie linii ze wspólną prostopadłą w każdym z trzech rodzajów geometrii

Ta praca skupia się na geometrii hiperbolicznej. Istnieje kilka możliwych sposobów wykorzystania części przestrzeni euklidesowej jako modelu płaszczyzny hiperbolicznej. Wszystkie te modele spełniają

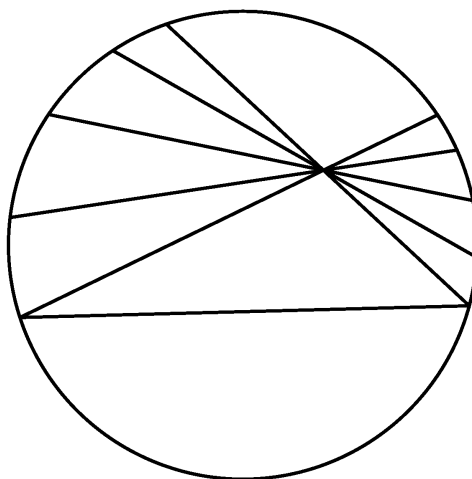
ten sam zestaw aksjomatów i wyrażają tę samą abstrakcyjną płaszczyznę hiperboliczną. Dlatego wybór modelu nie ma znaczenia dla twierdzeń czysto hiperbolicznych, jednak robi to różnicę podczas wizualizacji geometrii hiperbolicznej. Następne podrozdziały są poświęcone krótkiemu omówieniu najpopularniejszych z nich.

Najpopularniejsze modele

Geometria hiperboliczna została opisana za pomocą wielu modeli. Najpopularniejsze przedstawiono poniżej.

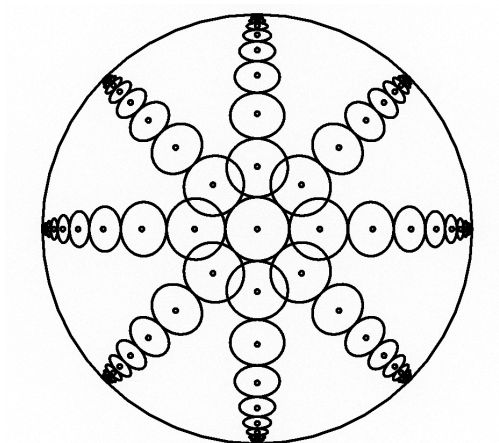
Model Kleina

Model Kleina - a w zasadzie model dysku Beltrami–Kleina jest modelem geometrii hiperbolicznej, w którym punkty są reprezentowane przez punkty we wnętrzu dysku. Przyjmuje on następujące założenia:



Rysunek 3: Model Kleina

- **Płaszczyzną hiperboliczną** jest wnętrze koła bez krawędzi.
- **Prostymi hiperbolicznymi** są cięciwy tego koła (końce prostej).



Rysunek 4: Koła w modelu Kleina

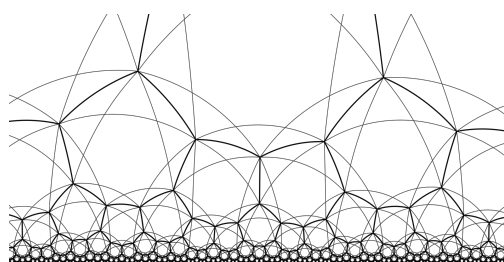
Linie w modelu pozostają proste, a cały model można łatwo osadzić w ramach rzeczywistej geometrii rzutowej. Model ten nie jest jednak zgodny, co oznacza, że kąty są zniekształcone, a okręgi na płaszczyźnie hiperbolicznej na ogół nie są okrągłe w modelu.

Model półpłaszczyzny Poincaré

Model półpłaszczyzny Poincaré to płaszczyzna:

$$\{(x, y) \mid y > 0; x, y \in \mathbb{R}\}$$

Jest to model dwuwymiarowej geometrii hiperbolicznej (geometrii Lobaczewskiego).



Rysunek 5: Tesselacja w modelu półpłaszczyzny Poincaré

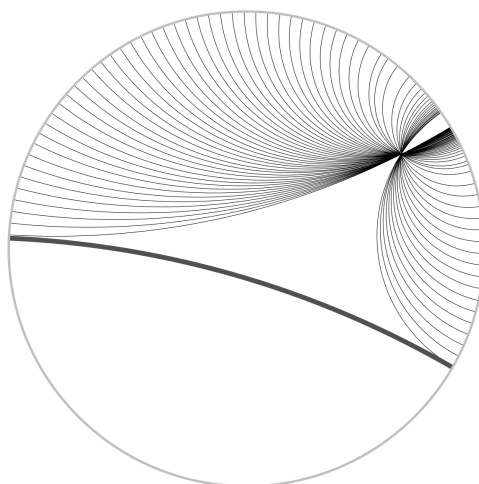
Model nosi imię Henri Poincaré, ale został stworzony przez Eugenio Beltramiego, który użył go wraz z modelem Klein i modelem Poincaré, aby pokazać, że geometria hiperboliczna jest równie spójna, jak spójna jest geometria euklidesowa. Ten model jest zgodny, co oznacza, że kąty zmierzone w punkcie modelu są równe kątom na płaszczyźnie hiperbolicznej.

Model dysku Poincaré

Model dysku Poincaré wykorzystuje wnętrze dysku jako model płaszczyzny hiperbolicznej. Najbardziej oczywistym wyborem dla dysku jest dysk jednostkowy, który będzie również przedmiotem dalszych rozważań.

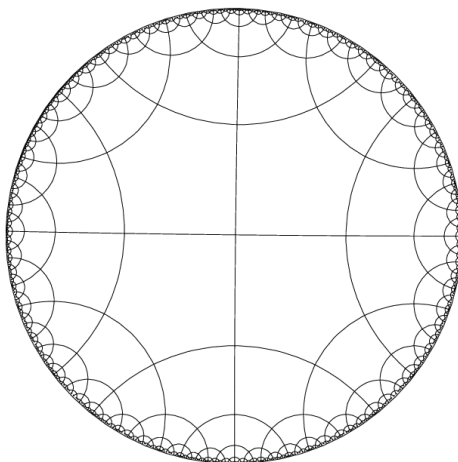
- **Punkty hiperboliczne** to punkty wewnątrz dysku jednostkowego.
- **Linie hiperboliczne** to łuki koła prostopadłe do dysku. Linie hiperboliczne przechodzące przez początek degenerują się do średnic, o których można pomyśleć jako łuki kół o nieskończonym promieniu.
- **Kąty** są mierzone jako kąt euklidesowy między stycznymi w punkcie przecięcia.
- **Odległości** między punktami hiperbolicznymi można mierzyć w oparciu o normę euklidesową:

$$\delta(u, v) = 2 \frac{\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}$$



Rysunek 6: Linie w modelu dysku Poincaré

Ponieważ rozpatrywany jest dysk jednostkowy, formuła nie zawiera w zmiennej dla promienia.



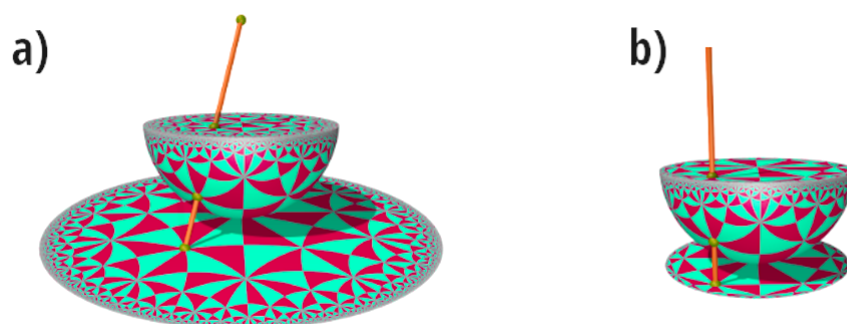
Rysunek 7: Tesselacja w modelu dysku Poincaré

Model jest zgodny, to znaczy, że zachowuje kąty. Oznacza to, że kąty hiperboliczne między krzywymi są równe kątom euklidesowym w punkcie przecięcia. Wadą jest fakt, że ponieważ linia hiperboliczna jest modelowana przez łuk koła euklidesowego, linie proste wydają się zakrzywione.

Model Hemisfery

Hemisfera nie jest często używana jako model płaszczyzny hiperbolicznej jako taka. Jest to jednak bardzo przydatna w łączeniu różnych innych modeli za pomocą różnych rzutów, jak pokazano na poniższym rysunku.

- **Punkty hiperboliczne** to punkty na półkuli południowej.
- **Linie hiperboliczne** to półkola powstałe z przecięcia półkuli południowej z płaszczyznami prostopadłymi do równika.

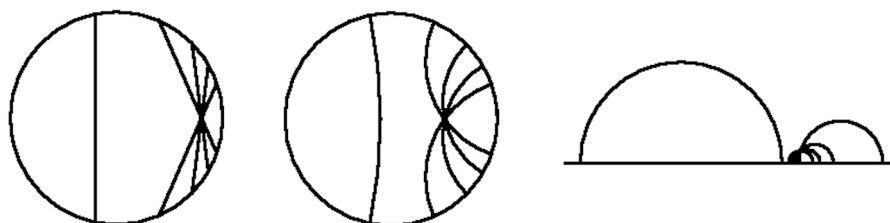


Rysunek 8: Rzut na dysk Poincarégo (a) i projekcja do modelu Klein-Beltrami (b)

Wadą tego rozwiązania, jest dodatkowy wymiar, jaki należy rozpatrywać przy pracy z tym modelem.

Uzasadnienie wyboru modelu dysku Poincaré

Jak stwierdzono na początku tego rozdziału, kolejne rozdziały, a także opisane implementacje będą prawie wyłącznie korzystać z modelu dysku Poincaré. Podczas renderowania geometrii hiperbolicznej wydaje się to być właściwym wyborem, z uwagi na wartości estetyczne i zgodność modelu.



Rysunek 9: Porównanie modeli Kleina, dysku Poincaré i półpłaszczyzny Poincaré

Projekt systemu

W niniejszym rozdziale przedstawiony zostanie szczegółowy projekt systemu, jego matematyczną interpretację, zależności pomiędzy klasami oraz podstawowe algorytmy składające się na logikę funkcjonowania silnika.

Cykl pracy silnika

Głównym plikiem silnika jest `main.ts` znajdujący się w katalogu `/src`. Po załadowaniu programu, tworzy on instancje klasy `Canvas` odpowiedzialnej za rysowanie elementów na ekranie, ładuje konfigurację wyświetlanego programu i tworzy pętlę silnika poprzez wywołanie metody `createLoop()` klasy `Engine`.

Canvas	typeof Canvas
<code>ctx</code>	<code>?any</code>
<code>canvas</code>	<code>?any</code>
<code>__constructor</code>	<code>?Constructor</code>
<code>setupCanvas</code>	<code>?() => void</code>
<code>drawOverlay</code>	<code>?() => void</code>
<code>drawPoint</code>	<code>? (point: Point) => void</code>
<code>drawLine</code>	<code>? (line: Line) => void</code>
<code>drawSection</code>	<code>? (p: Point, q: Point) => void</code>
<code>drawCircle</code>	<code>? (circle: Circle) => void</code>
<code>drawArc</code>	<code>? (line: HypLine) => void</code>
<code>drawHypLine</code>	<code>? (line: HypLine) => void</code>
<code>drawHypPolygon</code>	<code>? (polygon: HypPolygon) => void</code>
<code>setColors</code>	<code>? (color: string) => void</code>
<code>setLineWidth</code>	<code>? (weight: number) => void</code>

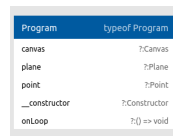
Rysunek 10: Diagram klasy Canvas

Moduł odpowiedzialny za renderowanie obrazu znajduje się w pliku `canvas.ts`. Konstruktor klasy `Canvas` przyjmuje element `canvas` ze strony oraz jego kontekst, oraz inicjuje się poprzez wywołanie funkcji `setupCanvas()`, która ustala szerokość i wysokość elementu. W każdym cyklu silnika, wywoływana jest funkcja `drawOverlay()`, która resetuje element do podstawowego widoku. Kolejne funkcje klasy odpowiadają za rysowanie punktów, linii, łuków i wielokątów. Poza tym klasa udostępnia też funkcje zmiany koloru rysowanych elementów i grubości linii.

Engine	typeof Engine
<code>interval</code>	<code>?number</code>
<code>canvas</code>	<code>?Canvas</code>
<code>__constructor</code>	<code>?Constructor</code>
<code>createLoop</code>	<code>? (program: Program) => void</code>
<code>removeLoop</code>	<code>?() => void</code>

Rysunek 11: Diagram klasy Engine

Klasa `Engine` przyjmuje konfigurację z pliku `/assets/config.json`, która ustala ilość FPS, wywołuje następnie metodę `drawOverlay()` klasy `Canvas` i odpala funkcję `onLoop()` z programu, konfigurację którego dostaje za pomocą *dependency injection* w parametrach konstruktora.

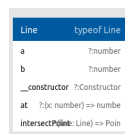
**Rysunek 12:** Diagram klasy Program

Odtwarzany program tworzony jest poprzez wywołanie instancji klasy programu, dziedziczącej po abstrakcyjnej klasie Program, udostępniającej metody takie jak onLoop().

Klasy obiektów

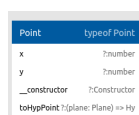
Każdy możliwy do narysowania obiekt jest instancją jednej z klas. W kodzie silnika istnieje wyraźny podział na klasy udostępniające obiekty rysowane w przestrzeni euklidesowej i hiperbolicznej. Wszystkie byty znajdują się w katalogu /src/core/entities. Kolejne rodzaje są poświęcone opisowi i interpretacji poszczególnych klas.

Klasa Line

**Rysunek 13:** Diagram klasy Line

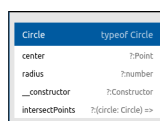
Konstruktor klasy Line przyjmuje dwie zmienne typu number. Programista może skorzystać z metody at(x: number): number, która zwraca wartość w punkcie x oraz intersectPoint(line: Line): Point, która zwraca punkt przecięcia tejże linii z inną linią. Alternatywnymi sposobami na stworzenie instancji klasy Line jest skorzystanie ze statycznych metody fromPoints(p: Point, q: Point), która tworzy linię z dwóch punktów lub fromPointSlope(p: Point, q: number), która do stworzenia linii potrzebuje podania punktu i kąta wyrażonego w radianach.

Klasa Point

**Rysunek 14:** Diagram klasy Point

Konstruktor klasy `Point` przyjmuje dwie zmienne typu `number`, które są reprezentacją bezwzględnych koordynatów punktu na płótnie. Programista może skorzystać z metody `toHypPoint(plane: Plane): HypPoint`, która przyjmuje instancję klasy `Plane` i zwraca dla niej koordynaty punktu w interfejsie `HypPoint`, oraz z metody `inversion(plane: Plane)`, zwracającej punkt odbity względem centralnego punktu obiektu klasy `Plane`.

Klasa Circle



Rysunek 15: Diagram klasy `Circle`

Konstruktor klasy `Circle` przyjmuje punkt centralny będący instancją klasy `Point` i średnicę typu `number`, oraz udostępnia metodę `intersectPoints(circle: Circle): [Point, Point]`, przyjmującą drugi okrąg i zwracającą parę punktów, w których przecinają się oba obiekty. Funkcja `fromPoints(p: Point, q: Point, r: Point)` umożliwia alternatywny sposób stworzenia okręgu z trzech obiektów klasy `Point`.

Klasa Plane

Najważniejszym z pośród omawianych dotychczas bytów jest instancja klasy `Plane`, będąca singletonem i punktem odniesienia do wszystkich obiektów dla geometrii hiperbolicznej. Klasa `Plane` dziedziczy po klasie `Circle`, podobnie jak ona posiada centrum i średnicę, liczone automatycznie na podstawie szerokości i wysokości ekranu przy pobraniu instancji klasy.

Implementacja systemu

W niniejszym rozdziale omówiona zostanie technologia, konfiguracja oraz wdrożenie systemu wraz z krótkim opisem poszczególnych części systemu i kodu źródłowego.

Opis technologii

Do implementacji systemu użyto języka TypeScript w wersji 3.6.3, bundlera (transpilatora nowoczesnych wersji języka JavaScript do wersji zrozumiałych dla przeglądarek) webpack w wersji 2.3.3 oraz CSS3 i HTML5 wraz z elementem `<canvas>` odpowiedzialnym za rysowanie grafiki na ekranie. Pełna lista wszystkich bibliotek wraz z ich wersjami znajduje się w pliku `package.json`, w katalogu głównym projektu.

Instalacja i wdrożenie

Rozdział ten zawiera informacje o sposobie zbudowania aplikacji w celu jej uruchomienia i opcjonalnie - wdrożenia na serwerze WWW.

Do zbudowania aplikacji konieczny będzie menager pakietów npm w wersji przynajmniej 6.5.0 oraz środowisko uruchomieniowe języka JavaScript - node.js w wersji 10.6.0 lub nowszej. Instalacja wymaganych pakietów odbywa się poprzez wpisanie w konsoli polecenia

```
npm install
```

w katalogu głównym projektu. Następnie należy zbudować aplikację poleceniem

```
npm run build
```

Po zbudowaniu aplikacji, w katalogu głównym pojawi się folder `dist` z plikami, które wraz z plikiem `index.html` składają się na gotowy program możliwy do uruchomienia w przeglądarce.

Serwer deweloperski

Aplikacja wspiera tryb deweloperski, w którym bieżące zmiany w kodzie automatycznie są budowane do plików wynikowych. Do uruchomienia trybu deweloperskiego potrzebne są te same pakiety instalowane poleceniem:

```
npm install
```

Wywołanie trybu odbywa się komendą:

`npm run build-watch`

Podsumowanie

Bibliografia

- Caroline Series With assistance from Sara Maloni, Hyperbolic geometry MA448
- Bjørn Jahren, An introduction to hyperbolic geometry, MAT4510/3510
- Martin Freiherr von Gagern, Creation of Hyperbolic Ornaments Algorithmic and Interactive Methods, Technischen Universität München
- Izabela Przezdzińska, Geometria Poincarégo i Kleina. Skrypt do zajęć: Podstawy geometrii i elementy geometrii nieeuklidesowej, Wrocław 2010, Uniwersytet Wrocławski Wydział Matematyki i Informatyki Instytut Matematyczny
- Mateusz Kłęczek, Geometria hiperboliczna, Chrzanów 2016
- Steve Szidlik, Hyperbolic Constructions in Geometer's Sketchpad, December 21, 2001
- Marek Kordos, Geometria Bolyaia–Łobaczewskiego, <http://www.deltami.edu.pl>, Sierpień 2018

Zawartość płyty CD