

# AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Informatyki, Elektroniki i Telekomunikacji



## Projekt i realizacja gry w przestrzeni nieeuklidesowej

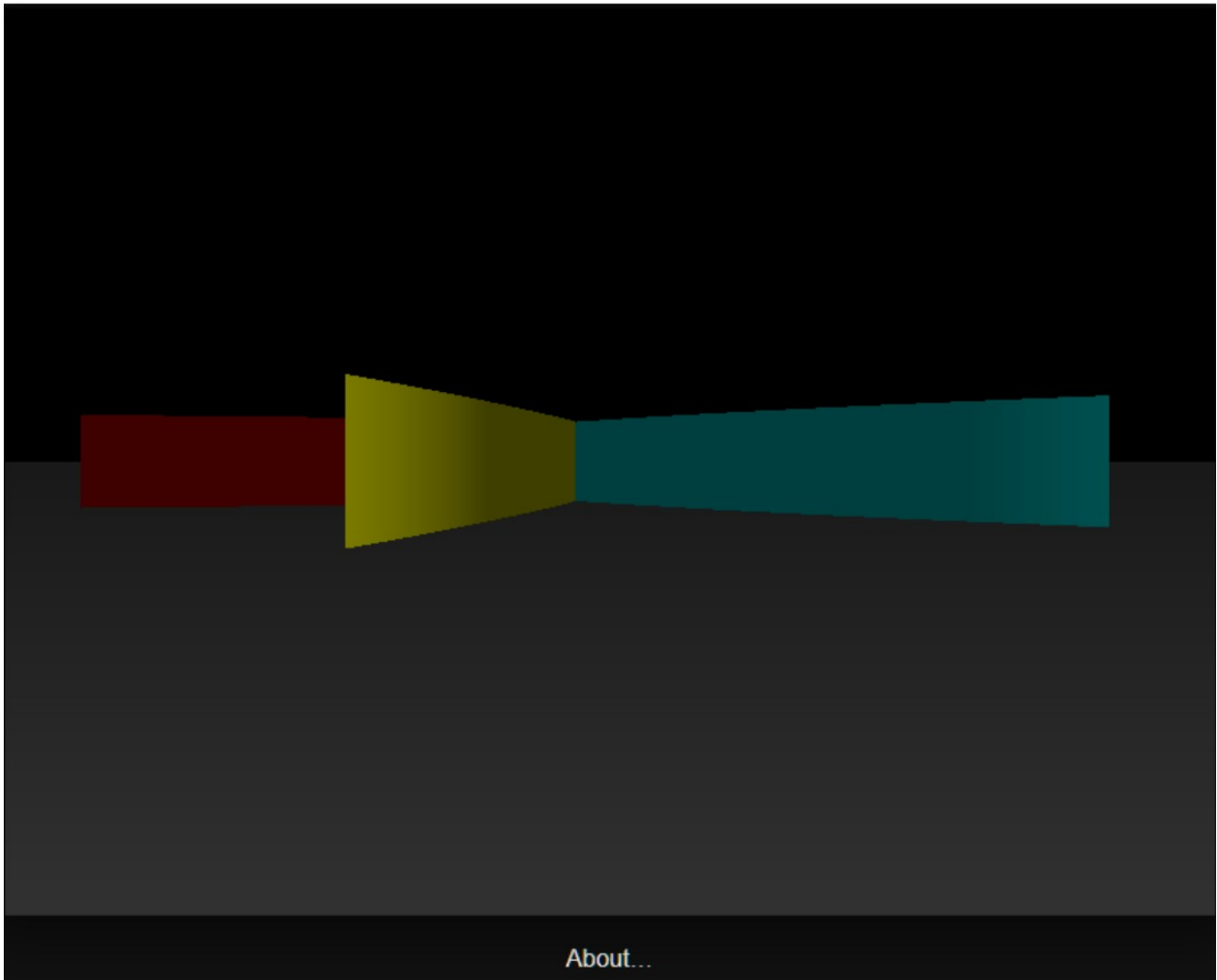
Autor: Michał Flak (294309)

Opiekun: dr. inż. Witold Alda

Dokumentacja deweloperska

# Studium wykonalności

W celu sprawdzenia wykonalności projektu wykonany został prototyp. Oparłem go na prostym raycasterze: <https://github.com/hydrixos/raycaster-rust>



Dostępny on jest do wypróbowania w przeglądarce: <https://elo-siema.github.io/hyperbolic-raycaster-rust/index.html>

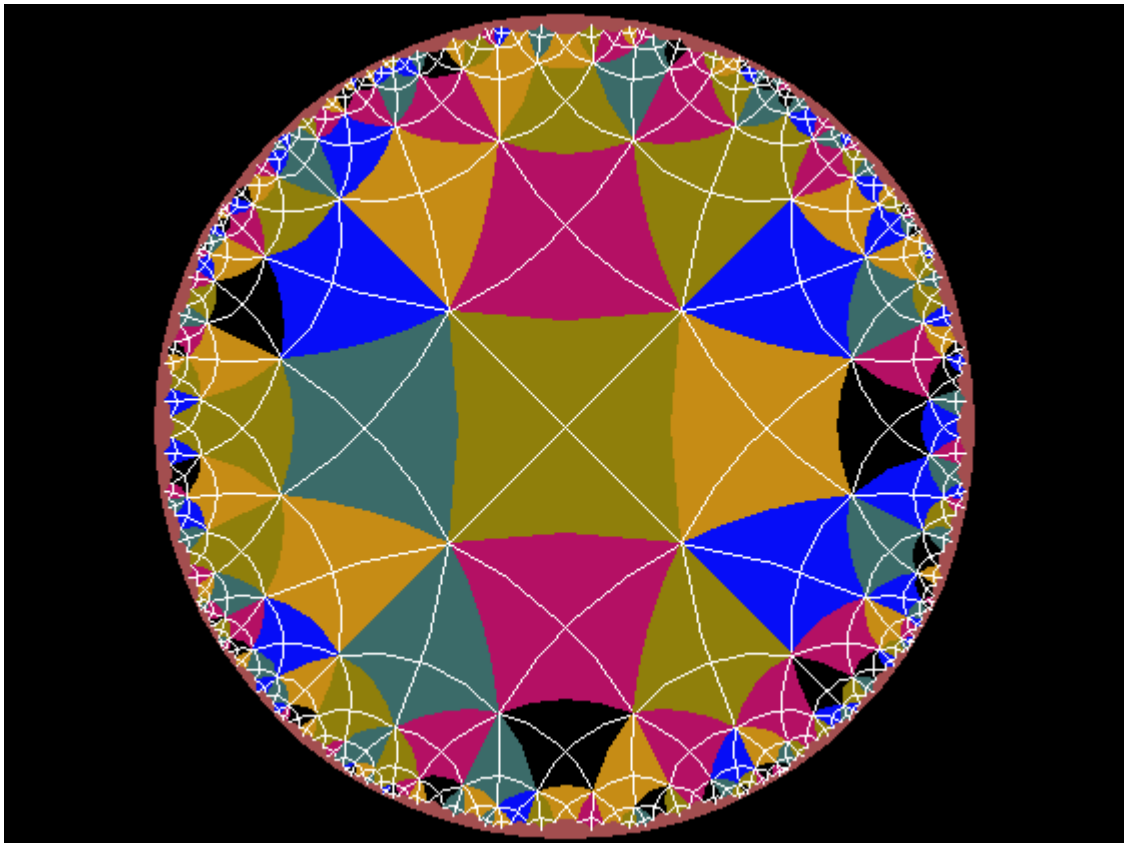
Kod źródłowy: <https://github.com/elo-siema/hyperbolic-raycaster-rust>

Napisany został w języku Rust, z wykorzystaniem biblioteki SDL2. Jest kompilowany do przeglądarki (WebAssembly / Emscripten) oraz do desktopowej aplikacji okienkowej.

Podczas tworzenia prototypu ujawniło się kilka problemów, do których znalazłem rozwiązania:

## 1. Wybór modelu przestrzeni / układu współrzędnych do łatwej edycji mapy

Początkowo rozważałem system oparty na siatce powstałej w wyniku tesselacji dysku Poincaré – numerując poszczególne kwadraty, ale szybko stało się jasne, że to rozwiązanie nie skaluje się oraz jest niezwykle niewygodne do projektowania poziomów:



Następnym pomysłem było stworzenie mapy jako zbioru ścian, z czego każda ma 2 punkty – początek i koniec. Punkty reprezentowane są we współrzędnych dysku Poincaré – pozwala to na łatwe tworzenie poziomu w edytorze graficznym, widząc świat w rzucie z góry. Przykładowy poziom z dwiema ścianami:

```
[  
  {  
    "beginning": [0.0, 0.032],  
    "end": [0.0, 0.31],
```

```

        "color": {
            "red": 255,
            "green": 0,
            "blue": 0
        }
    },
    {
        "beginning": [0.0, 0.032],
        "end": [0.270, 0.129],
        "color": {
            "red": 255,
            "green": 255,
            "blue": 0
        }
    }
]

```

Tworzy to pewien problem – postrzegane odległości zmniejszają się wraz z oddalaniem od środka układu współrzędnych, jest to jednak ciągle najlepsze rozwiązanie jakie widzę.

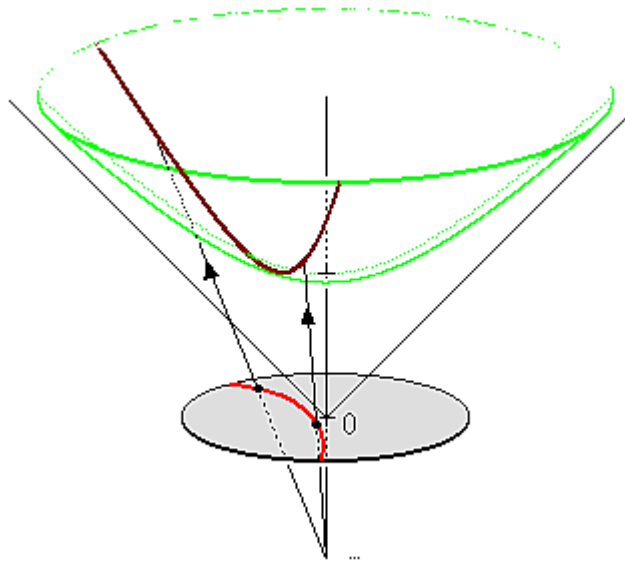
## 2. Wybór modelu przestrzeni do reprezentacji wewnętrznej świata gry

Tym razem priorytetem była łatwość dokonywania transformacji. Początkowo posiłkowałem się materiałami developerskimi z tworzenia gry Hyperbolica, gdzie reprezentacja świata trzymana jest w modelu dysku Poincare, tworzy to jednak spory problem z transformacjami – wymagają użycia żyrowektorów, które nie są jednak wspierane przez biblioteki oraz są zupełnie nieintuicyjne w użyciu.

Zdecydowałem się na reprezentację w postaci modelu hiperboloidu Minkowskiego, ze względu na analogiczne transformacje do przestrzeni euklidesowej – zmieniają się tylko macierze przekształcenia oraz funkcje trygonometryczne na ich hiperboliczne odmiany.

## 3. Rzutowanie świata na ekran

Tutaj wybór był już prosty – ponowne rzutowanie na dysk Poincaré:



Następnie, używając odpowiedniej dla tego modelu metryki, wypuszczamy promienie ze środka układu współrzędnych i mierzymy odległość do punktów przecięcia ze ścianami. Odległości te są odwrotnie proporcjonalne do wysokości ściany rysowanej na ekranie w tej kolumnie – klasyczny raycaster. Dodatkowa transformacja została dodana żeby naprawić efekt “rybiego oka”.

## 4. Wybór technologii

Zdecydowałem się na język Rust oraz bibliotekę graficzną SDL2, ze względu na osobiste doświadczenie, szybkość, doskonały system typów, wsparcie społeczności oraz wsparcie platform na których chciałbym, żeby gra działała.

## 5. Wnioski

Wykonanie prototypu dostarczyło cennych informacji:

- Koncepcja jest wykonalna, problemy zostały rozwiązane od strony matematycznej
- Należy przepisać projekt od podstaw, nie opierając się na istniejącym projekcie, ponieważ zmiany są na tyle daleko idące że pierwotna struktura staje się ograniczająca
- Potrzebny będzie edytor poziomów, oparty w większości na tym samym kodzie, gdyż ręczna edycja jest mało praktyczna