A może **DOOM** ~~pod choinkę~~ na choince?

Zdecydowanie zbyt skomplikowany kontroler lampek choinkowych

Krzysztof Mazur, Mariusz Różycki

5 grudnia 2023 r.

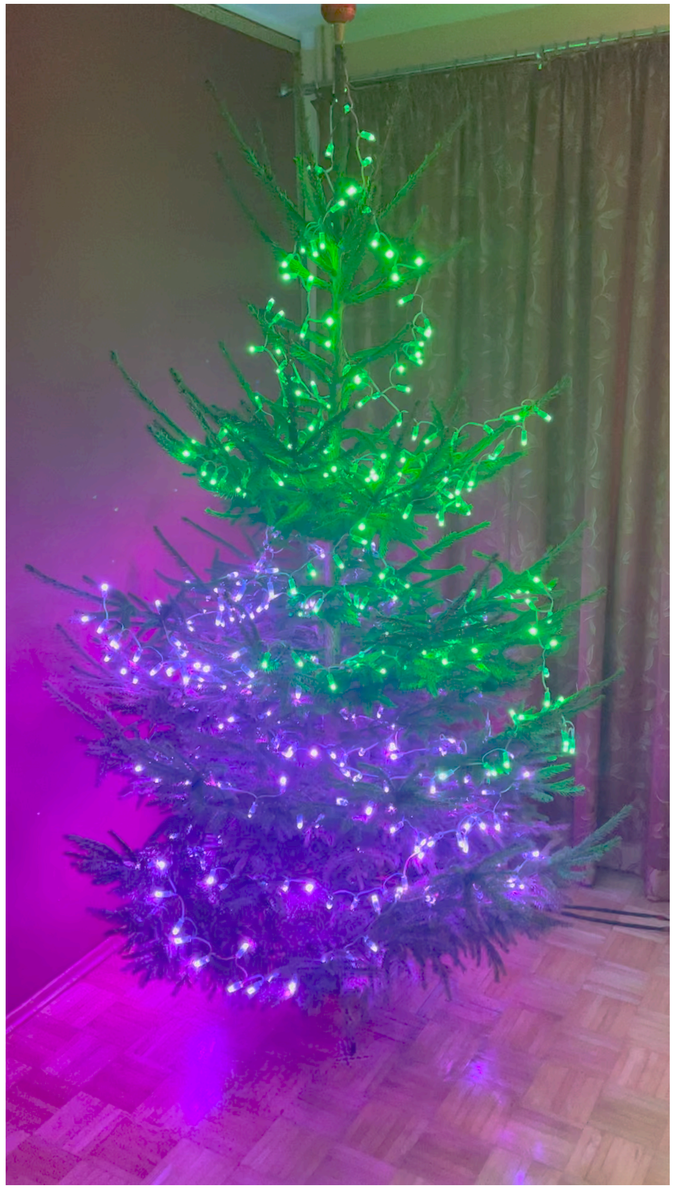
Inspiracja: Matt Parker

I wired my tree with 500 LED lights and calculated their 3D coordinates

Rustmas

Santa Crab is coming to town

- Programowalne lampki na WS281x
- Podłączone do Raspberry Pi Pico
- Bezprzewodowo podłączone do Raspberry Pi
- Na którym uruchomiony jest Rustmas
- Na podstawie wcześniej zebranych koordynatów wyświetla animacje 3D



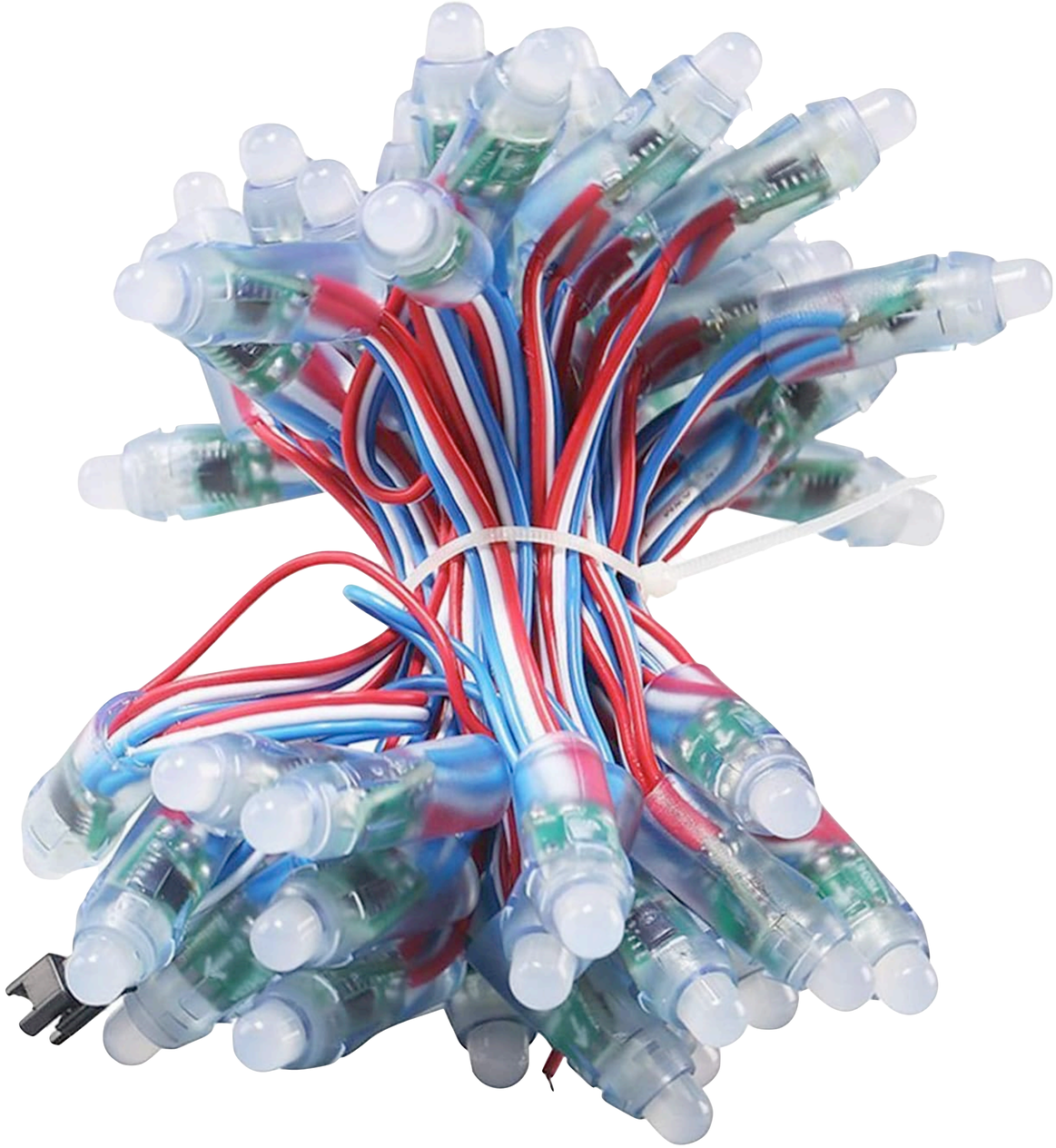
Demo

Hardware

Lampki, kabelki, ograniczenia

Użyte lampki

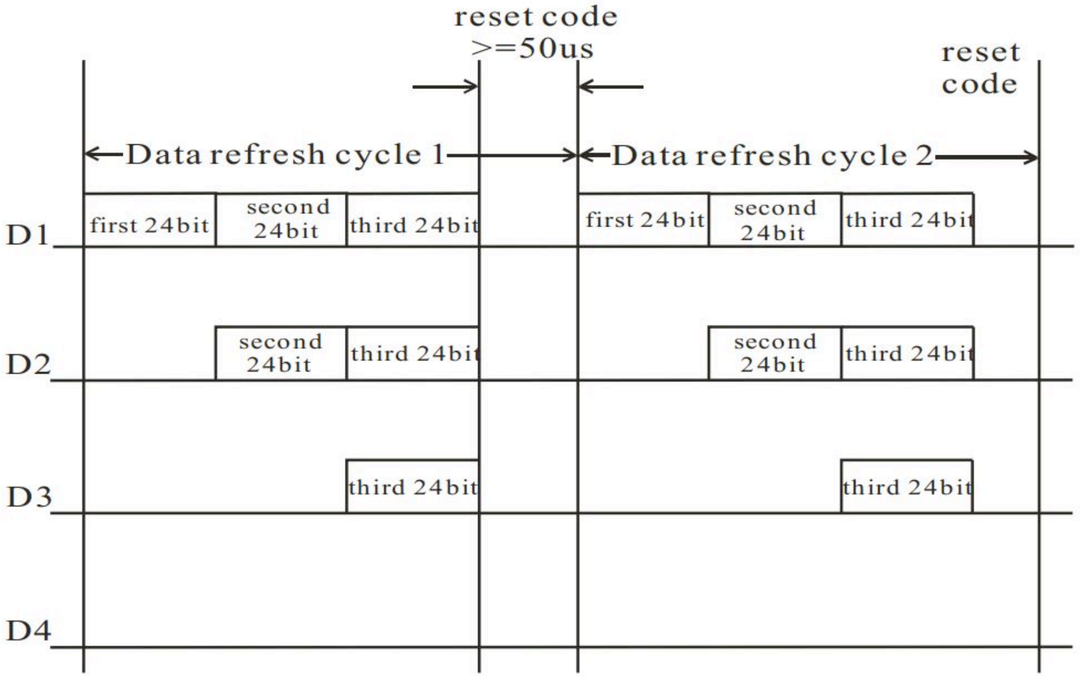
- WS2811 + 12mm RGB LED
- Każda lampka zatopiona w żywicy epoksydowej, co pozwala na wykorzystanie ich na zewnątrz
- Zasilacz 250W do 500 lampek



WS281x "Neopixel"

- Każda lampka bierze ze strumienia danych pierwszą trójkę RGB i przekazuje resztę dalej

- Aby zmienić kolor n-tej lampki, musimy wysłać co najmniej n kolorów



Ile klatek na sekundę*?

- Komunikacja przy 800 kHz daje 1 bit co 1,25µs
- 24 bity na lampkę = 30µs
- 50µs po ostatniej wartości aby „zapisać” kolory
- 500 · 30µs + 50µs = 15,05ms → 66 fps
- 1000 · 30µs + 50µs = 30,05ms → 33 fps

* w teorii

Potencjalne problemy

- Napięcie linii przesyłu danych
  - RaspberryPi Pico pracuje na 3.3V
  - Może być to wystarczające, ale nie jest zgodne ze specyfikacją
- Częstotliwość sygnału
  - RaspberryPi 4 ma procesor o zmiennej częstotliwości taktowania
  - Może sprawiać problemy przy przesyle danych po GPIO
- RGB vs GRB — różne lampki mogą oczekiwać innej kolejności bajtów

Potencjalne problemy

- Spadek napięcia wraz z długością kabla
  - Lampki robią się wyraźnie ciemniejsze
  - Zasilanie możemy dostarczać w wielu miejscach
  - Przy 12V lampkach spadki napięcia są mniej istotne niż przy 5V
  - Mimo wszystko dostarczamy zasilanie na obu końcach przy 500 lampkach

Potencjalne problemy

- Korekcja gamma
  - Mariusz, oddychaj
  - Lampki działają liniowo: 128 oznacza, że lampki świecą połowę czasu
  - Ludzkie oko nie postrzega natężenia światła liniowo
  - Połowa jasności to 1/4 czasu świecenia, czyli wartość 64

Programujemy

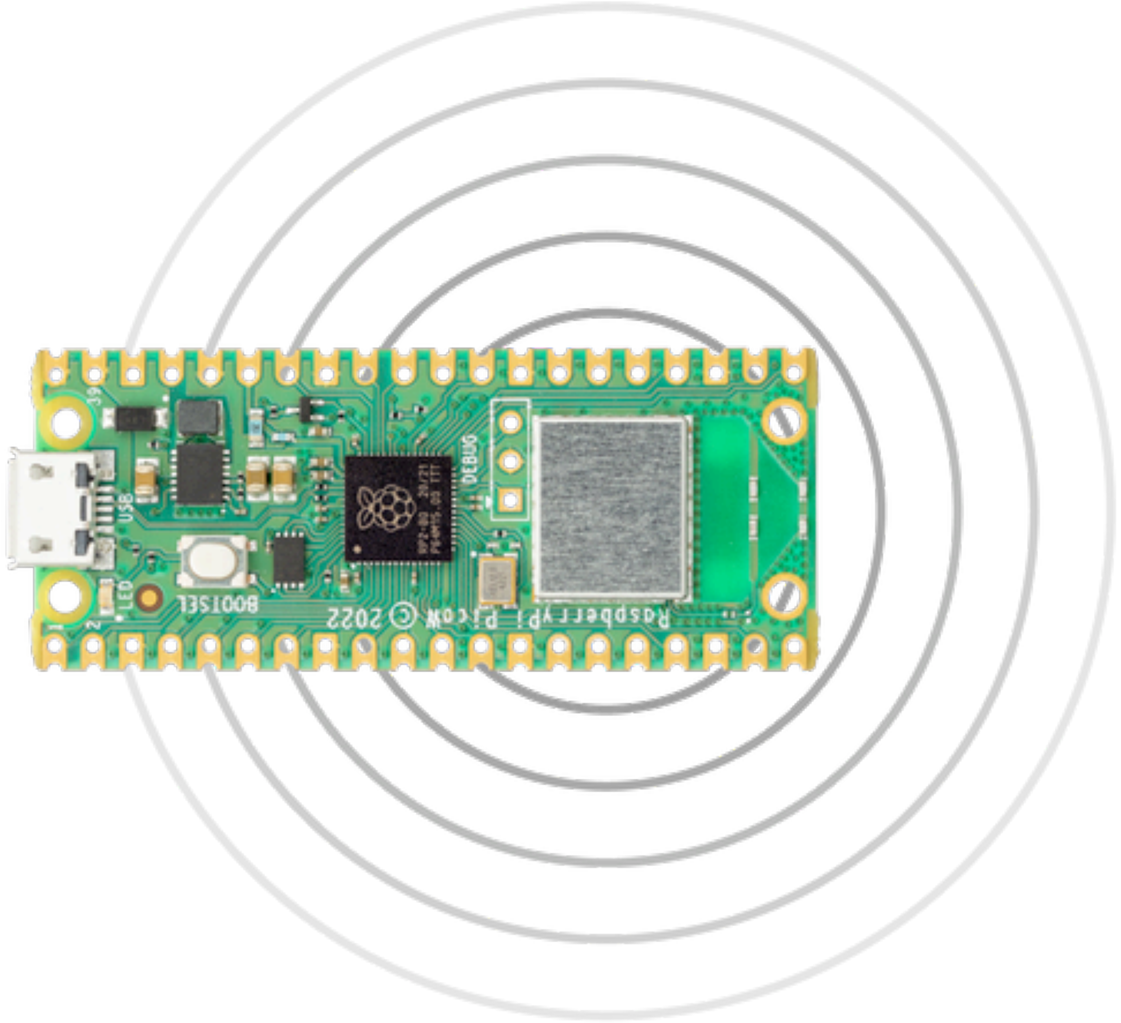
Pierwsza warstwa abstrakcji nad kabelkami

Sterowanie neopixelami

- RaspberryPi z lampkami podpiętymi do GPIO
  - Duże, drogie, ryzyko zniszczenia przez zalanie/ubrudzenie żywicą
  - Problemy przez zmienną częstotliwość taktowania procesora
- WLED na chipie z ESP32
  - Nastawione głównie na wysyłanie tylko parametrów animacji
  - Nie wiadomo czy udźwignie tyle lampek
  - Pico było pod ręką ¯_(ツ)_/¯

RaspberryPi Pico

- Tanie (nie szkoda jeżeli się zepsuje)
- Małe (można ukryć w choince)
- Wystarczająco mocne, żeby bezpośrednio sterować lampkami
- Wersja W wspiera komunikację bezprzewodową
- Prosty klient na Pico, bardziej zaawansowane obliczenia na innej maszynie (np. zwykłe RPi)



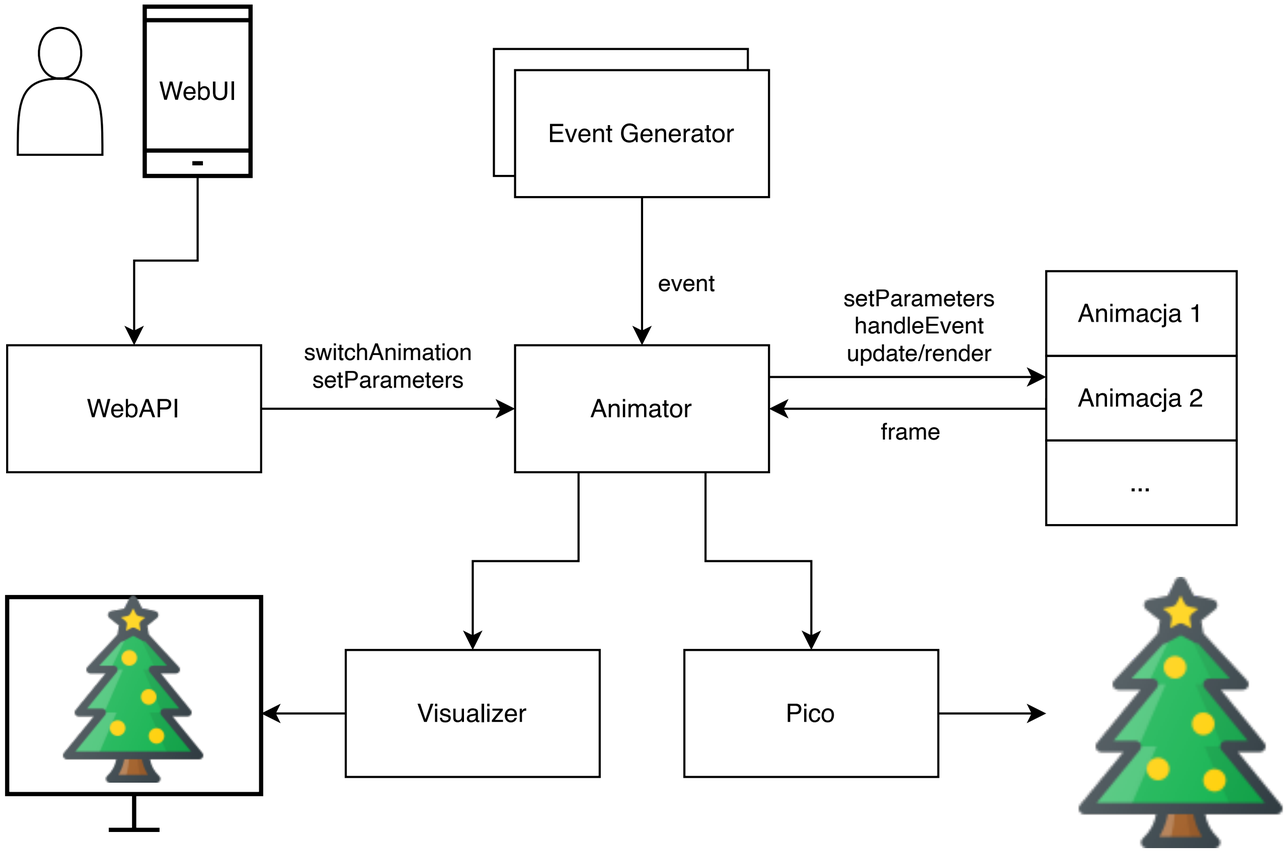
Komunikacja z Pico

- Prosty serwer HTTP na bazie lwip (C++)
- UDP (Rust)
- TCP (Rust)
- Przewodowo przez serial port po USB (Rust)

Rustmas

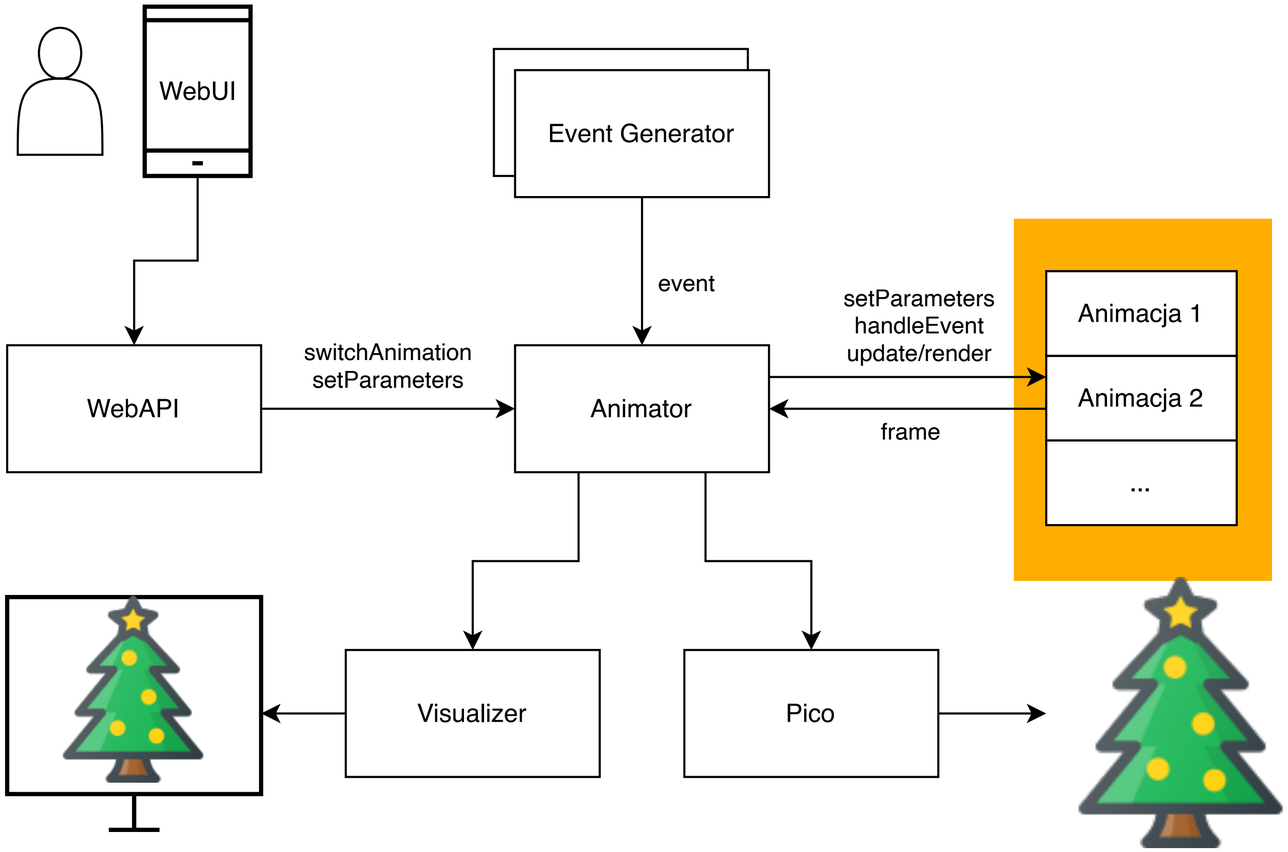
Wchodzi cały na kolorowo

Rustmas z lotu ptaka



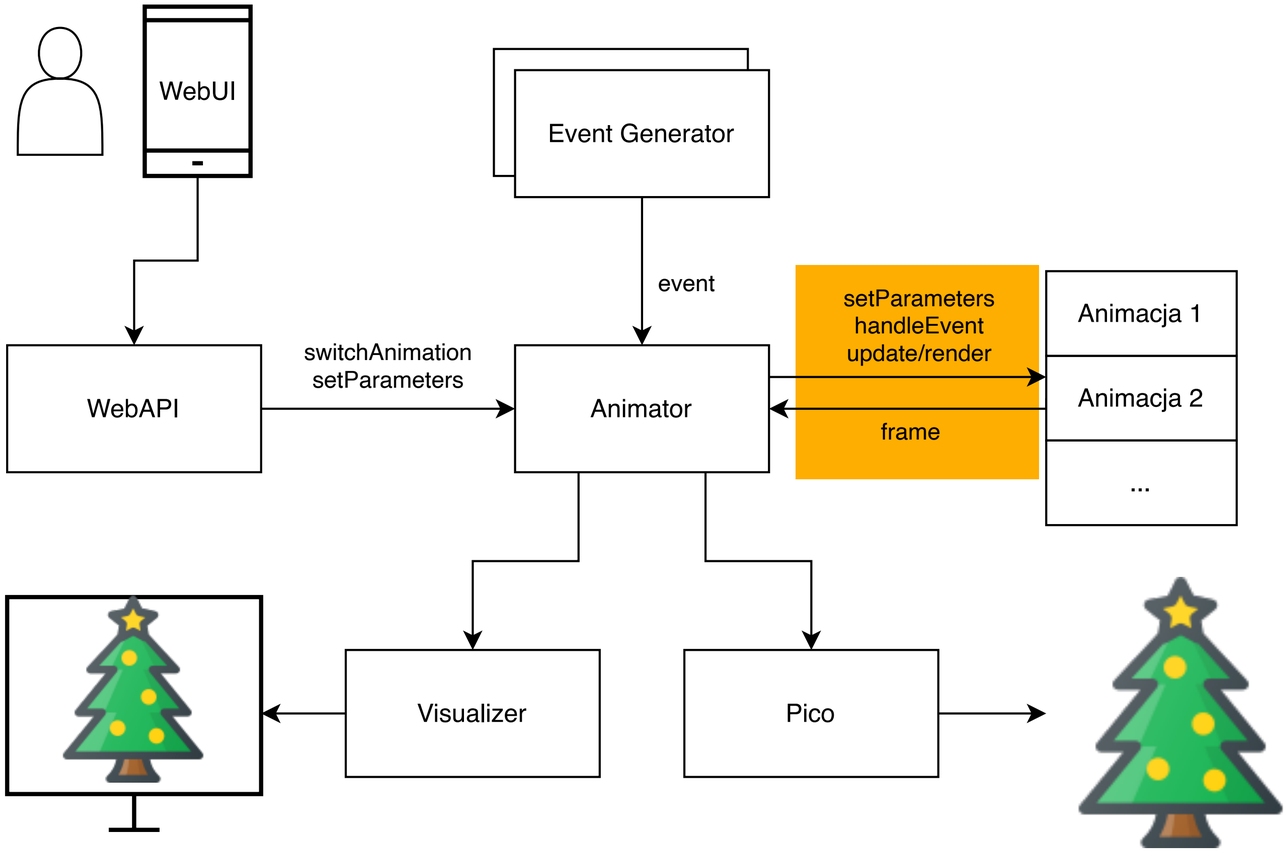
Rustmas z lotu ptaka

- Każda animacja to osobny program



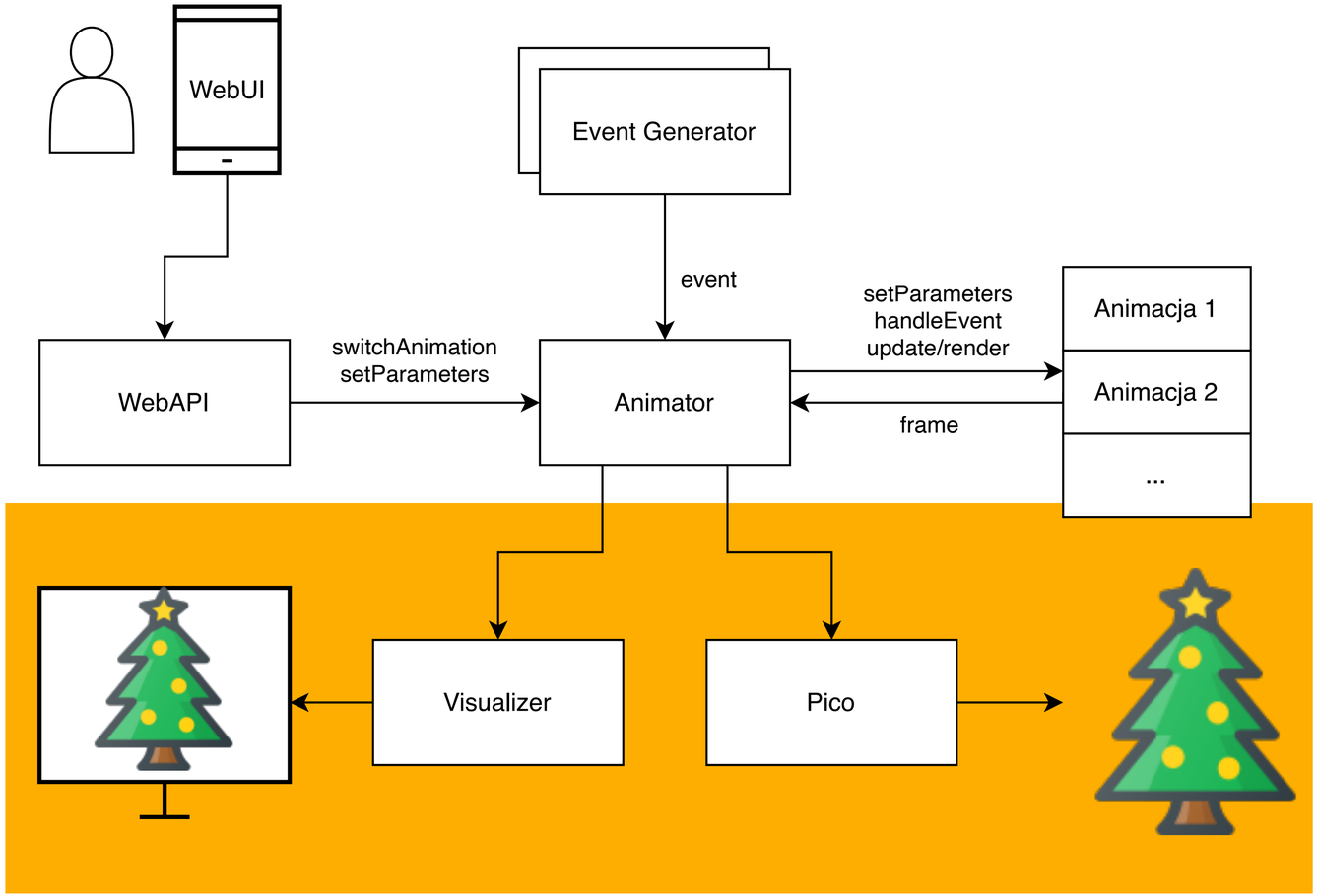
Rustmas z lotu ptaka

- Każda animacja to osobny program
- Animator periodycznie odpytuje animację o nowe ramki



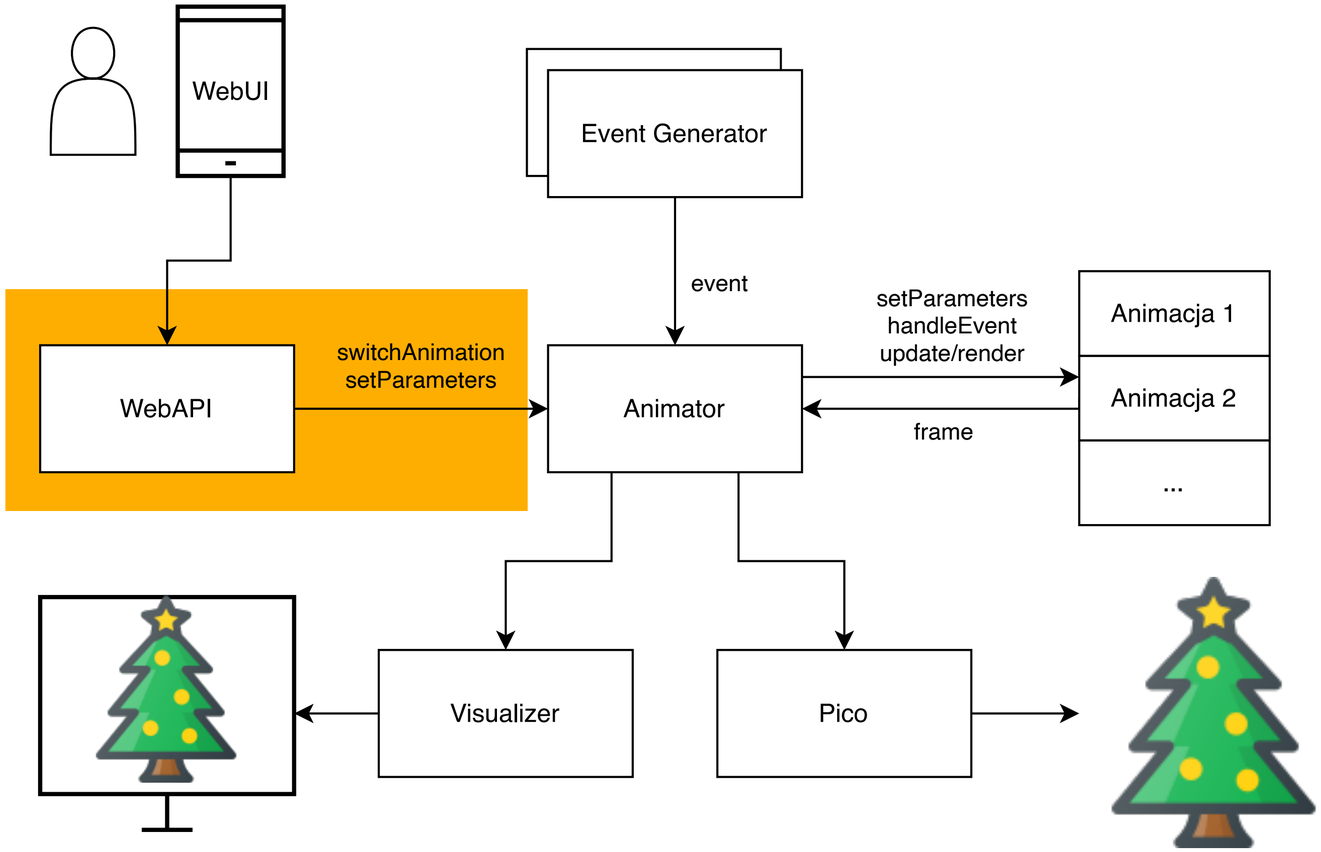
Rustmas z lotu ptaka

- Każda animacja to osobny program
- Animator periodycznie odpytuje animację o nowe ramki i przekazuje je do lampek



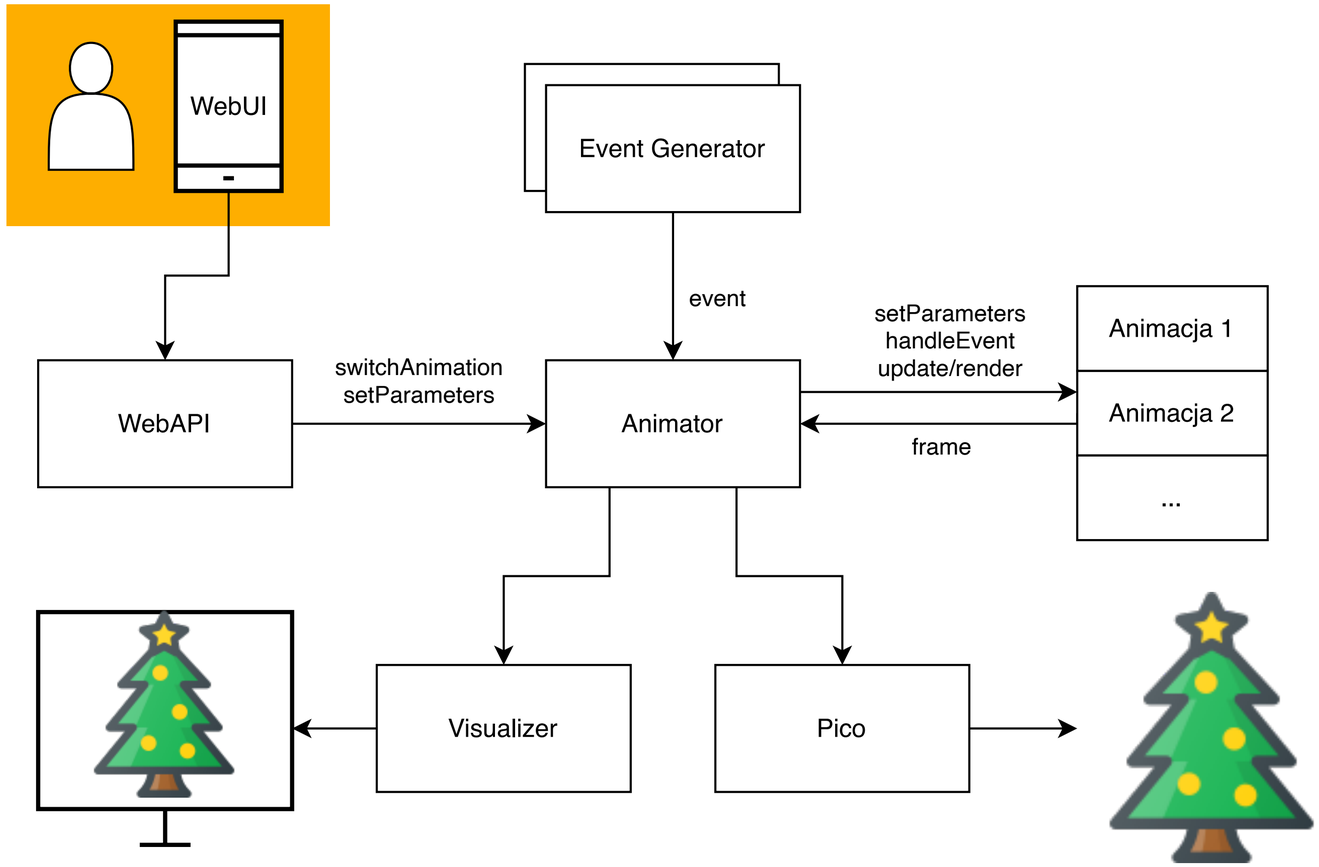
Rustmas z lotu ptaka

- Każda animacja to osobny program
- Animator periodycznie odpytuje animację o nowe ramki i przekazuje je do lampek
- WebAPI obsługuje zapytania z zewnątrz i przekazuje je do animatora



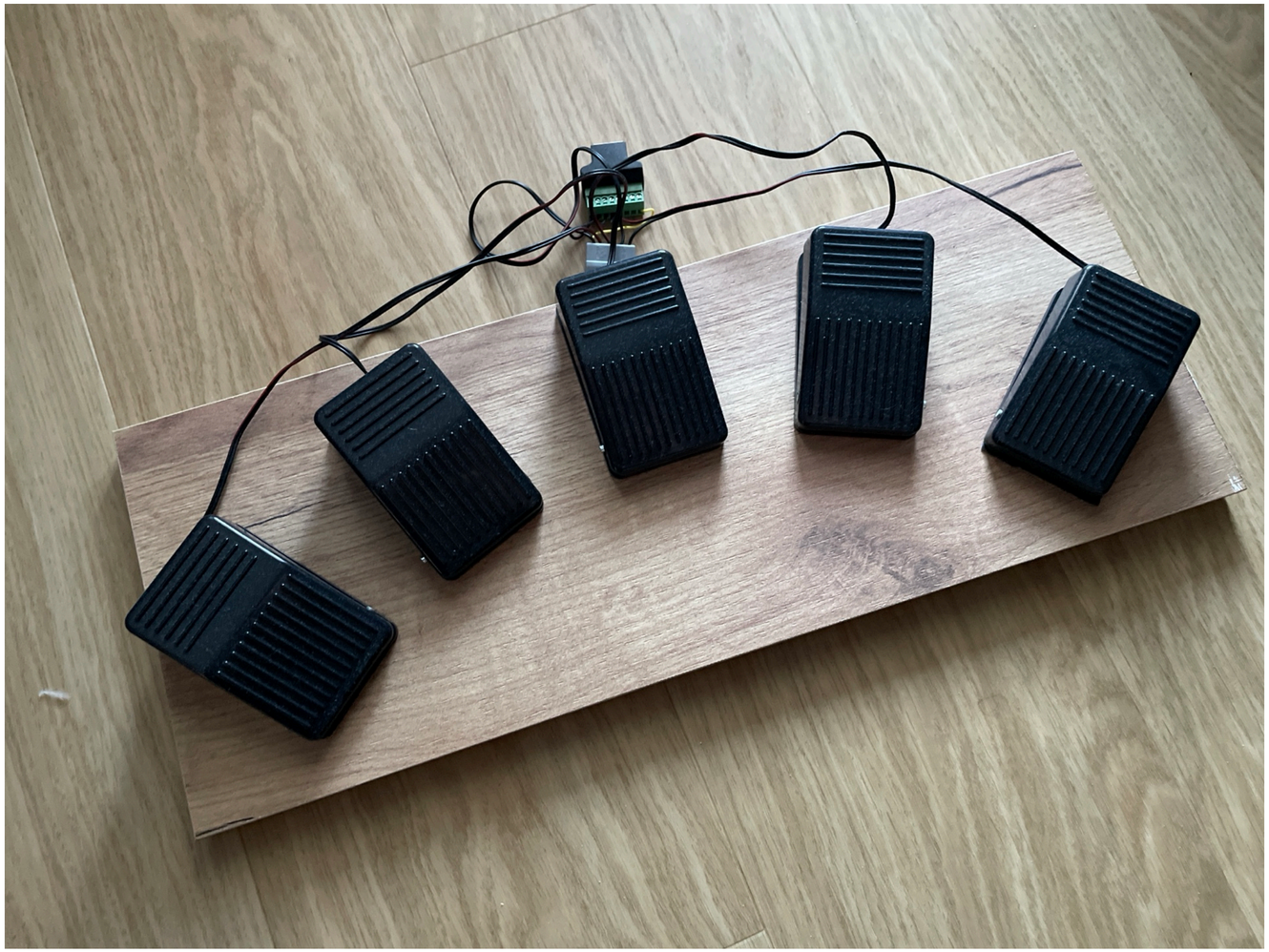
Rustmas z lotu ptaka

- Każda animacja to osobny program
- Animator periodycznie odpytuje animację o nowe ramki i przekazuje je do lampek
- WebAPI obsługuje zapytania z zewnątrz i przekazuje je do animatora
- WebUI służy za wygodny interfejs dla użytkownika



Po co zastępować WebUI?

Po co zastępować WebUI?



Własne animacje

Narzędzia deweloperskie, API animacji

Przykładowa animacja

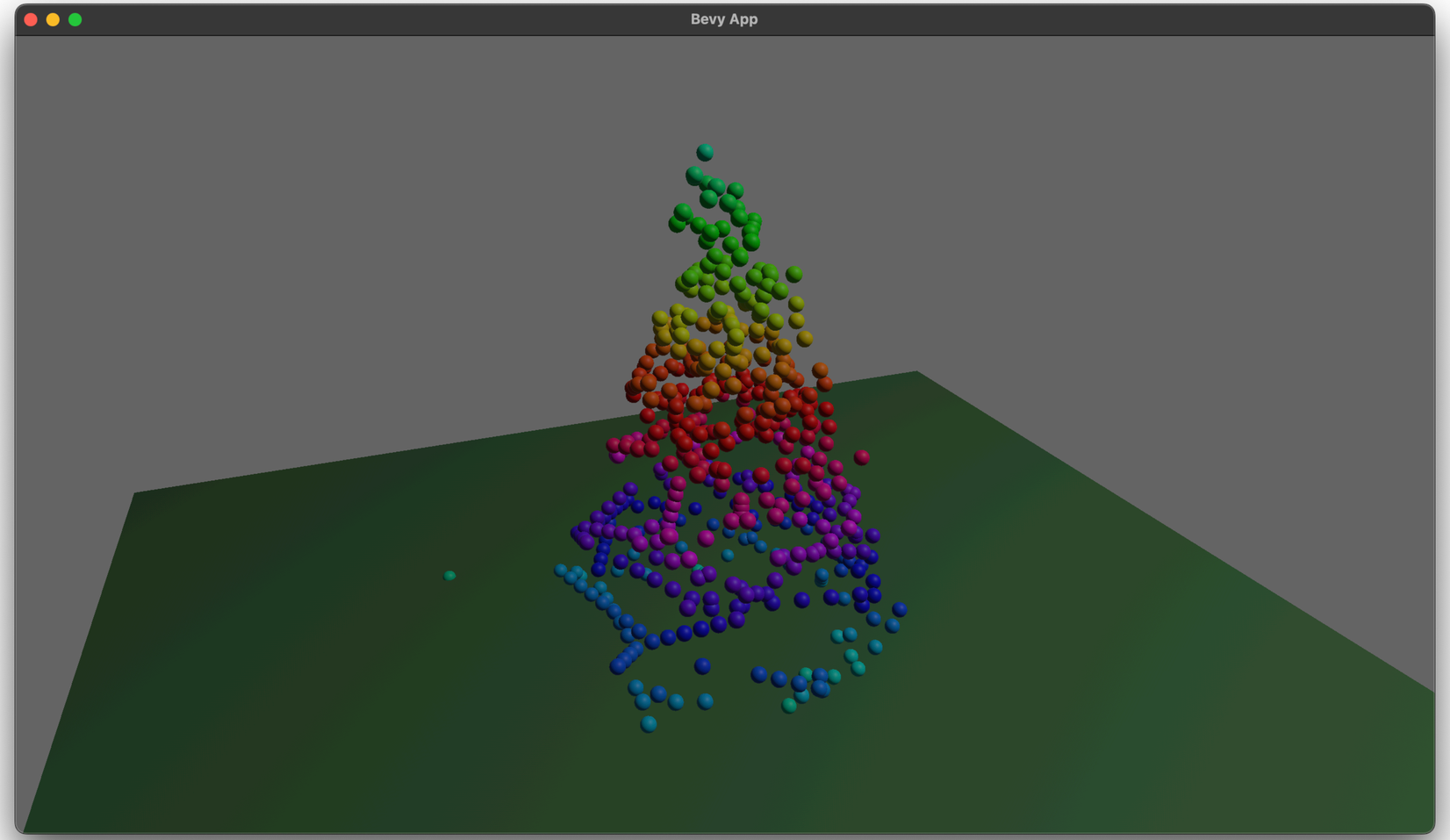
```rust
impl Animation for RainbowCable {
    fn update(&mut self, delta: f64) {
        self.time += delta;
    }

    fn render(&self) -> lightfx::Frame {
        (0..self.points_count)
            .map(|i| {
                lightfx::Color::hsv(
                    i as f64 / self.points_count as f64 + self.time,
                    1.0,
                    1.0,
                )
            })
            .into()
    }
}
```

API animacji

- Każda animacja to osobny program
- Komunikacja z animacjami odbywa się przez JsonRPC, więc w teorii mogą być napisane w dowolnym języku programowania
- W praktyce dostarczamy API i makra rustowe, które ułatwiają pisanie animacji w tym języku

```rust
#[derive(Serialize, Deserialize)]
#[serde(tag = "method", content = "params")]
pub enum JsonRpcMethod {
    Initialize { points: Vec<(f64, f64, f64)> },
    AnimationName,
    ParameterSchema,
    SetParameters { params: serde_json::Value },
    GetParameters,
    GetFps,
    Update { time_delta: f64 },
    OnEvent { event: crate::event::Event },
    Render,
}
```

Wizualizator



Animacje w przestrzeni

Dzięki Algorytmom Sztucznej Inteligencji

[redacted] • 3.+

Generative AI Founder | Advisor ...

2 t •

+ Obserwuj

This awesome AI-powered plate holds a ball in balance on a plate. (+ Code)💥

A webcam films the system and a python program analyzes the images to find the position of the ball.

The Python program calculates the tilting of the tray to prevent the ball from falling.

A proportional-integral-derivative (PID) regulator is used to compensate for the movements of the ball.

The position and speed of the ball are measured by the camera and these measurements are used by the PID regulator in the Python program.

The base is moved using a Microchip ATmega 32u4 microcontroller while the computer vision portion is on a laptop with a connected USB camera

Click "Follow" for more AI Innovations and Insights

[redacted]

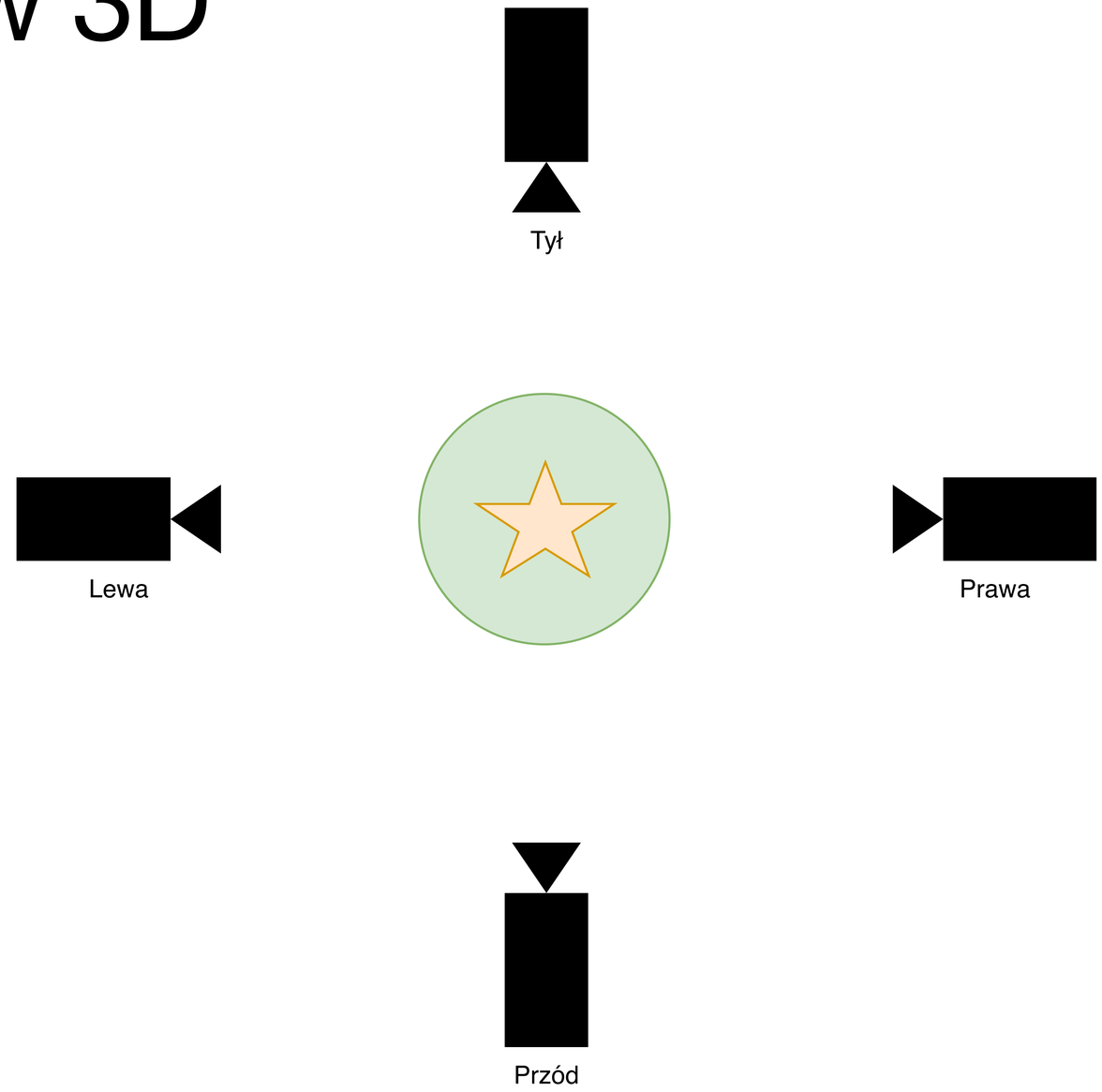
#innovation #machinelearning #artificialintelligence

Znajdowanie lampek w 2D

- Gasimy lampki i robimy zdjęcie 1
- Zapalamy jedną lampkę i robimy zdjęcie 2
- Znajdujemy różnicę zdjęć 1 i 2
- Znajdujemy najjaśniejszy punkt
- Powtarzamy dla każdej lampki na łańcuchu

Znajdowanie lampek w 3D

- Ten sam proces powtarzamy kolejno z przodu, prawej, tyłu i lewej
- Łączymy perspektywy w jeden zestaw koordynatów 3D
- Wykrywamy błędnie znalezione lampki i usuwamy ich koordynaty
- Luki uzupełniamy interpolacją i ekstrapolacją



Co może pójść nie tak?

- Proces trwa ~5 minut na stronę + ustawienie kamery
- Zmiana położenia lampek po konfiguracji
- Błędnie wykryte lampki — ruch w kadrze, odbicia
- Okluzje — powinny poradzić sobie z nimi inter- i ekstrapolacja
- Zerwane połączenie z lampkami — zalecamy używanie kabla
- Brak dostępu do choinki z 4 stron — można próbować ją obracać
- Konfigurator działa w wierszu poleceń, nie każdy sobie z nim poradzi

Animacja w 3D

```rust
impl RainbowWaterfall {
    pub fn create(points: Vec<(f64, f64, f64)>) -> impl Animation {
        Self {
            time: 0.0,
            points_height: points
                .into_iter()
                .map(|(_, h, _)| (h + 1.0) / 2.0)
                .collect(),
        }
    }
}

impl Animation for RainbowWaterfall {
    fn update(&mut self, delta: f64) {
        self.time += delta;
    }

    fn render(&self) -> lightfx::Frame {
        self.points_height
            .iter()
            .map(|h| lightfx::Color::hsv(h + self.time, 1.0, 1.0))
            .into()
    }
}
```

DOOM na choince?

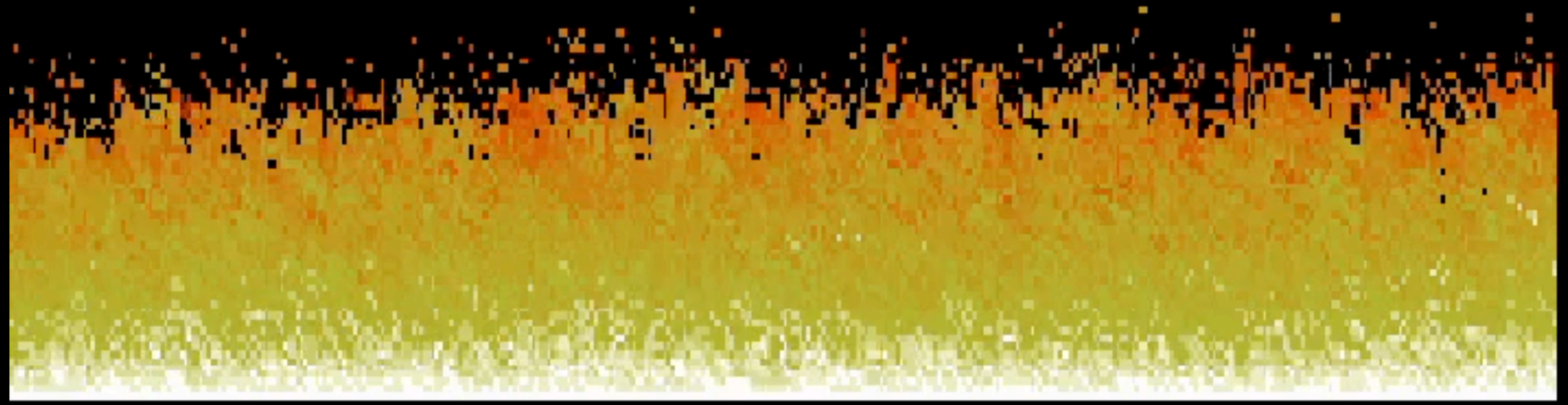
Prawo nagłówków Betteridge’a w praktyce

Prawo nagłówków Betteridge’a

Reguła, zgodnie z którą na każdy nagłówek prasowy kończący się znakiem zapytania można odpowiedzieć „nie”.

https://pl.wikipedia.org/wiki/Prawo_nagłówków_Betteridge'a

Ale…



Co dalej?

Nad czym pracujemy teraz

- Animacje wykorzystujące wejście audio i MIDI
- Ustawienia dla Event Generatorów
- Wizualizator w WebUI, przekazywanie dotyku do animacji
  - Cel: rysowanie po choince
- Mniejsze i większe poprawki w konfiguratorze
- Łatwiejsza instalacja pluginów

Co chcemy osiągnąć w przyszłości

- Konfigurator w WebUI — prostszy w obsłudze
- Obsługa większej liczby lampek — poprzez wiele Pico?
- Event Generatory jako pluginy
- Migracja animacji do WASM?

Co wy możecie zrobić

- Cały kod (wraz z instrukcjami uruchomienia) dostępny jest na GitHubie
- Cały system można zestawić lokalnie bez lampek i testować
- Zawsze chętnie przyjmiemy ciekawe animacje
- Zawsze chętnie zobaczymy rustmasowe lampki na innych choinkach

mrozycki/rustmas

mrozycki

krzmaz

Dziękujemy za uwagę!

Pytania?