

# Ataskaita

## Realizuotos dalys

### Scenarijai

- **Vektoriaus kodavimas:** pilnai realizuota.
- **Teksto kodavimas:** pilnai realizuota.
- **Nuotraukos kodavimas:** pilnai realizuota.

### Techninės dalys

- **Baigtinis kūnas:** pilnai realizuota. (binaryField.js)
- **Veiksmai su vektoriais:** pilnai realizuota. (vector.js)
- **Veiksmai su matricomis:** pilnai realizuota. (matrix.js)
- **Kodavimas kodu:** pilnai realizuota. (golayCode.js)
- **Siuntimas kanalu:** pilnai realizuota. (channel.js)
- **Dekodavimas kodu:** pilnai realizuota (golayCode.js).
- **Teksto/Paveiksluko skaidymas:** pilnai realizuota, tačiau tam nekurtas atskiras modulis.
- Pagalbinės funkcijos su nuotraukomis (antraščių ištraukimas atitinkamam nuotraukos formatui) atskirai įgyvendintos **imgUtils.js** modulyje.
- **constants.js** modulyje inicializuojamos Golėjaus kodo B matrica ir B matrica be paskutinio stulpelio.
- **index.js:** Javascript driver kodas, kuris vykdomas užkrovus puslapį, suteikia vartotojo sąsajos komponentos logikos ir kviečia kitus modulius.
- **index.html:** Pagrindinis html failas, kuris pirmasis pristatomas nuėjus į svetainės adresą.
- **styles.css:** Stiliumo failas.

## Trečiųjų šalių bibliotekos

**Jquery:** [Javascript biblioteka](#). Naudojama tik veiksams su vartotojo sąsajos komponentams. Palengvina įvykių klausymo (mygtukų paspaudimo, teksto įvedimo) įgyvendinimą ir taip pat palengvinami veiksmai modifikuojantys ar skaitantys informaciją komponentuose.

**Bootstrap:** [CSS plėtinys](#). Pagerina svetainės dizainą, palengvina komponentų išdėstymą.

# Užduoties įvykdymo laikas

**Užduoties išsiaiškinimas:** 1h (neskaičiuojant visų teorinių paskaitų, dėl kurių lengviau įsigilinti į užduotį).

**Projektavimas:** 0.5h

**Programavimas:** 14h

**Ataskaitos ruošimas:** 1.5h

## Kaip paleisti programą

### 1 būdas

Programą (svetainę) galima pasiekti <https://karklas.mif.vu.lt/~juru4427/index.html>.

### 2 būdas

Per terminalą reikia nueiti iki programos failų direktorijos (ten kur randasi index.html) ir iš tos vietos paleisti statinį HTTP serverį. Priklausomai nuo turimos programinės įrangos, išvardijamos galimos komandos. (Šaltinis <https://gist.github.com/willurd/5720255>).

**Python 2.x:** `python -m SimpleHTTPServer 8000`

**Python 3.x:** `python -m http.server 8000`  
arba  
`python3 -m http.server 8000`

**Ruby 1.9.2+:** `ruby -run -ehttpd . -p8000`

**Perl:** `cpan HTTP::Server::Brick`  
`perl -MHTTP::Server::Brick -e`  
`'$s=HTTP::Server::Brick->new(port=>8000); $s->mount("/"=>{path=>"."});`  
`$s->start'`

**Node.js:** `npm install -g http-server`  
`http-server -p 8000`

**Php:** `php -S 127.0.0.1:8000`

Paleidus serverį, naršyklėje reikia nueiti adresu [localhost:8000](http://localhost:8000).

## Vartotojo sąsajos aprašymas

### Klaidų tikimybės nustatymas

Per įvedimo lauką arba slankiklį nustatoma klaidos tikimybė (numatyta 0)

## Vektoriaus kodavimas

Pradinis vektorius

101010101010

Užkoduotas vektorius

10101010101001001011110

Siųsti

Persiųstas vektorius

1000010101001000001110

Dekoduoti

Dekoduotas vektorius

100001100000

☐ Koreguoti

Žingsniai kaip užkoduojamas vektorius:

1. Į **Pradinis vektorius** lauką įvedami nuliai ir vienetai. Lauke būtinai turi būti tik nuliai ir vienetai ir simbolių skaičius turi būti būtent 12.
2. Kai įrašomas 12-as simbolis, vektorius užkoduojamas ir pilkame neredaguojamame lauke **Užkoduotas vektorius** parodomas Golėjaus kodo užkoduotas vektorius.
3. Paspaudus mygtuką **Siųsti** užkoduotas vektorius siunčiamas per kanalą ir pagal klaidos tikimybę yra iškraipomas. Iškraipytas vektorius pavaizduojamas **Persiųstas vektorius** lauke, kur iškraipyti bitai yra nuspalvinami raudonai.
4. (papildomas žingsnis) Norint redaguoti persiųsta vektorių, galima paspausti ant **Koreguoti** langelio ir pakeisti persiųsto vektoriaus reikšmę.
5. Paspaudus mygtuką **Dekoduoti**, persiųstas ir galbūt pakoreguotas vektorius yra dekoduojamas, kur gali būti ištaisomos klaidos ir atstatomas pradinis vektorius. Rezultatas įrašomas į **Dekoduotas vektorius** lauką.

## Teksto kodavimas

Įvestis...

Siųsti

Persiųstas tekstas

Ištaisytas tekstas

Norint užkoduoti tekstą, reikia įvesti pradinį tekstą į įvesties lauką ir spausti mygtuką **Siųsti**. Kairėje bus persiųstas kanalu neištaisytas tekstas, o dešinėje persiųstas kanalu ir ištaisytas tekstas.

## Nuotraukos kodavimas

Įkelti failą

Browse



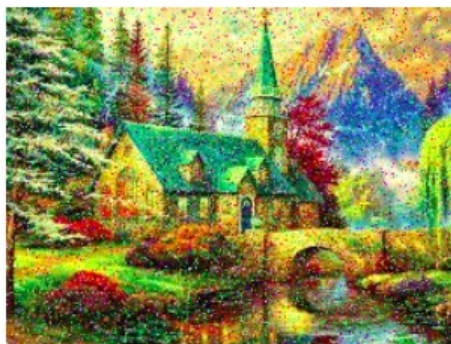
Siųsti

Persiųsta nuotrauka



Iš pradžių iš vartotojo failų sistemos įkeliama nuotrauka paspaudus **Browse**. Palaikomi nuotraukų formatai: BMP, PNG, WEBP, nors rekomenduojama naudoti BMP. Tada paspaudus mygtuką **Siųsti**, nuotrauka siunčiama kanalu ir iškraipoma. Viršuje persiųsta nuotrauka yra neištaisyta, tik persiųsta, o apačioje esanti nuotrauka dekoduoja naudojant Golėjaus kodą ir ištaisyta.

Persiųsta nuotrauka



Ištaisyta nuotrauka



## Programiniai sprendimai

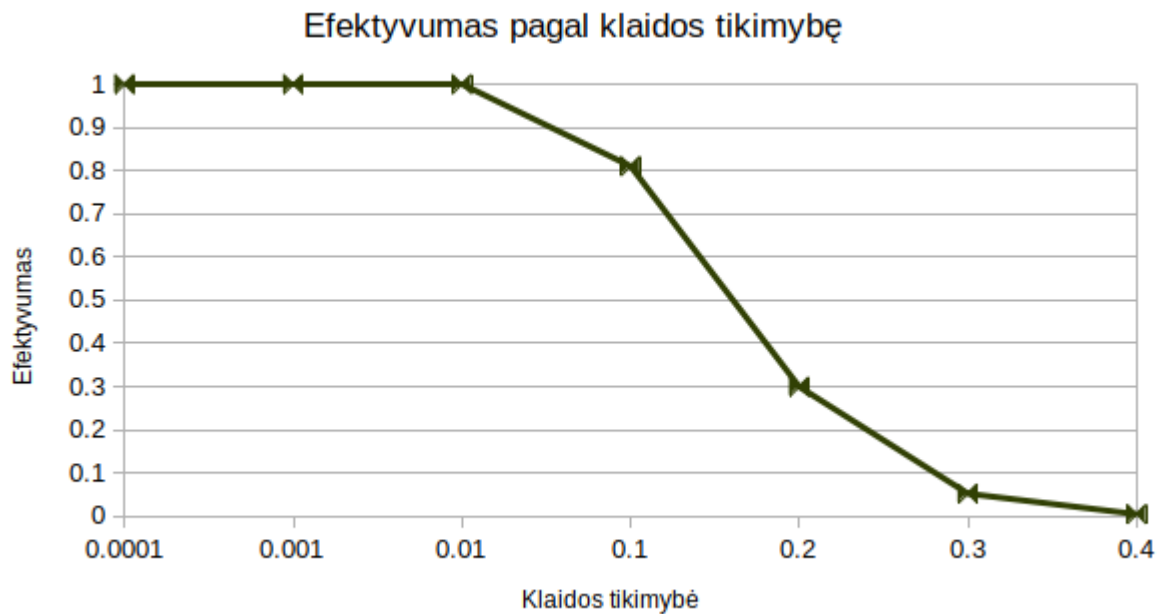
**Jei bitais užkoduota informacija neišsidalina kodo bitais.** Kadangi siunčiamas vektorius dalijamas kas 12 bitų, o mažiausias informacijos vienetas yra 8 bitai, kartais tenka papildyti neišsitekusių paskutinį vektorių nuliukais. Informacija apie kiek bitų papildyta saugoma ir siunčiama atskiru “Patikimu” kanalu. Pavyzdžiui, tekstas “ab” UTF-8 užkoduojamas bitais 01100001 01100010. Tai yra 16 bitų, todėl pirmas vektorius bus 12 bitų ilgio, o antram liko tik 4 bitai, todėl likę 8 bus užpildomi nuliais ir bus prisiminta, kad 8 paskutiniai bitai yra nereikšmingi.

**Teksto pavertimas bitais ir iš bitų vektoriais.** Naudojama TextEncoder klasė, kad tekstą paversti į baitų sąrašą. Tada yra iteruojama per baitus ir pasinaudojant bitų operacijomis iš kiekvieno baito yra paverčiami 8 bitai (Kiekvienas bitas yra JavaScript number tipas, kuris yra arba 0, arba 1). Implementacija yra fromBytes funkcijoje, vector.js modulyje. Pritaikomas atvirkštinis procesas, kad paversti iš bitų į tekstą.

**Nuotraukų pavertimas vektoriais.** Panašus principas kaip ir su tekstu, tik pirmiausiai išsaugoma nuotraukų antraščių informacija ir likusi dalis yra siunčiama kanalu kaip įprasta. Nuotraukos naršyklėje yra užkoduotos Base64 formatu, galima konvertuoti į bitus pasinaudojant atob funkcija (ir atvirkščiai btoa).

## Eksperimentai

Atlikti eksperimentai per testEfficiency funkciją (index.js). 10000 kartų sugeneruojami atsitiktiniai 12-os bitų ilgio vektoriai, kuriems vykdomas pilnas kodavimo-siuntimo-dekodavimo procesas su nurodyta klaidos tikimybe. Tuomet sulyginamas kiekvienas originalus vektorius su dekodotu ir paimamas procentas tų atvejų, kai originalus vektorius yra lygus dekodotajam. Šiuo atveju jei visi vektoriai teisingai dekoduoti, efektyvumas yra 1, jei pusė vektorių teisingai dekoduoti, tuomet efektyvumas 0.5 ir t.t.



## Šaltiniai

1. <https://klevas.mif.vu.lt/~skersys/20r/ktkt/uzduotys/uzduotisA.htm>
2. <https://klevas.mif.vu.lt/~skersys/20r/ktkt/prat.htm>
3. <https://klevas.mif.vu.lt/~skersys/doc/ktkt/literatura12.pdf>
4. <https://stackoverflow.com/>
5. [https://en.wikipedia.org/wiki/Binary\\_Golay\\_code](https://en.wikipedia.org/wiki/Binary_Golay_code)
6. <https://www.mathsisfun.com/algebra/matrix-multiplying.html>
7. <https://matrix.reshish.com/multiplication.php>
8. <https://www.symbolab.com/solver/matrix-multiply-calculator>
9. <https://gist.github.com/willurd/5720255>