**Міністерство освіти і науки України**

**Чернівецький національний університет**

**імені Юрія Федьковича**

\_\_\_\_\_\_Інститут фізико-технічних та комп’ютерних наук\_\_\_\_\_\_

(повна назва інституту/факультету)

\_\_\_\_\_\_\_Кафедра математичних проблем управління і кібернетики\_\_\_\_\_

(повна назва кафедри)

**Тема:**

Фоторедактор

**Дипломна робота**

Виконав: студент (ка) \_4\_курсу, групи\_\_441ск\_\_

напряму підготовки

\_\_\_\_\_\_\_6.050101 – Комп’ютерні науки\_\_\_\_\_\_\_\_

(шифр і назва спеціальності)

\_\_\_\_\_\_\_\_\_\_\_\_Привалов В.Є.\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник\_\_\_\_\_Лазорик В.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (прізвище та ініціали)

**До захисту допущено:**

**Протокол засідання кафедри № \_\_\_\_\_\_**

від „\_\_\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 р.

зав. кафедри\_\_\_\_\_\_\_\_\_ д. ф-м.н. **Дрінь Я.М.**

Чернівці–2018

**Зміст**

[ТЕОРЕТИЧНА ЧАСТИНА - 9 -](#__RefHeading___Toc386013294)

[1. ТЕХНІЧНЕ ЗАВДАННЯ - 10 -](#__RefHeading___Toc386013295)

[1.1. Підстави для розробки - 10 -](#__RefHeading___Toc386013296)

[1.2. Призначення розробки - 10 -](#__RefHeading___Toc386013297)

[1.3. Аналіз вимог до програмного забезпечення - 10 -](#__RefHeading___Toc386013298)

[*1.3.1. Функціональні вимоги* - 10 -](#__RefHeading___Toc386013299)

[*1.3.2. Вимоги до складу та параметрів технічних засобів* - 10 -](#__RefHeading___Toc386013300)

[*1.3.3. Вимоги до інтерфейсу* - 11 -](#__RefHeading___Toc386013301)

[*1.3.4. Вимоги до інформаційної та програмної сумісності* - 11 -](#__RefHeading___Toc386013302)

[*1.3.5. Вимоги до тестування програмного забезпечення* - 11 -](#__RefHeading___Toc386013303)

[1.4. Вимоги до програмної документації - 11 -](#__RefHeading___Toc386013304)

[1.5. Стадії та етапи розробки - 12 -](#__RefHeading___Toc386013307)

[1.6. Порядок контролю і приймання - 12 -](#__RefHeading___Toc386013308)

[ПРАКТИЧНА ЧАСТИНА - 13 -](#__RefHeading___Toc386013294)

[2. АРХІТЕКТУРА, ФУНКЦІОНАЛЬНІ ТА ТЕХНІЧНІ ПОКАЗНИКИ - 14-](#__RefHeading___Toc386013310)

[2.1. Призначення та область застосування - 14 -](#__RefHeading___Toc386013311)

[2.2. Опис та обґрунтування обраної архітектури - 14 -](#__RefHeading___Toc386013312)

[2.3. Функціональна специфікація - 14 -](#__RefHeading___Toc386013313)

[*2.3.1. Опис функціональних можливостей* - 14 -](#__RefHeading___Toc386013314)

[*2.3.2. Опис інтерфейсу користувача* - 16 -](#__RefHeading___Toc386013315)

[2.4. Технічна специфікація - 17 -](#__RefHeading___Toc386013316)

[*2.4.1. Опис модулів* - 17 -](#__RefHeading___Toc386013317)

[*2.4.2. Опис і обґрунтування вхідних і вихідних даних* - 17 -](#__RefHeading___Toc386013318)

[3. КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ - 18 -](#__RefHeading___Toc386013319)

[3.1. Опис і обґрунтування обраних програмних засобів - 18 -](#__RefHeading___Toc386013320)

[3.2. Опис програми - 18 -](#__RefHeading___Toc386013321)

[*3.2.1. Функціональні можливості* - 18 -](#__RefHeading___Toc386013322)

[*3.2.2. Опис логічної структури* - 19 -](#__RefHeading___Toc386013323)

[*3.2.3. Використані технічні засоби* - 22 -](#__RefHeading___Toc386013324)

[4. ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ - 23 -](#__RefHeading___Toc386013325)

[4.1. Об’єкт випробувань - 23 -](#__RefHeading___Toc386013326)

[4.2. Використані технічні засоби - 23 -](#__RefHeading___Toc386013327)

[4.3. Порядок та методика випробувань - 24 -](#__RefHeading___Toc386013328)

[4.4. Результати випробувань - 24 -](#__RefHeading___Toc386013329)

[ВИСНОВКИ - 25 -](#__RefHeading___Toc386013330)

[ВИКОРИСТАНА ЛІТЕРАТУРА - 26 -](#__RefHeading___Toc386013331)

[ДОДАТКИ - 27 -](#__RefHeading___Toc386013332)

[Додаток А. Код програми - 27 –](#__RefHeading___Toc386013333)

[Додаток Б.Порядок та методики випробувань - 37 –](#__RefHeading___Toc386013333)

# ТЕОРЕТИЧНА ЧАСТИНА

# 1. ТЕХНІЧНЕ ЗАВДАННЯ

## 

## 1.1. Підстави для розробки

Розробка програмного забезпечення виконується на підставі рішення засідання кафедри ПЗКС про затвердження тем дипломних та курсових робіт (протокол № від « » 201\_року).

*Повна назва розробки*: «Фоторедактор».

*Коротка назва*: Фоторедактор.

## 1.2. Призначення розробки

*Мета роботи* – редагування фотографій та зображень за допомогою фільтрів.

*Область застосування* – програмне забезпечення для пристроїв з ОС Windows/Linus та браузеру Google Chrome, що генерує задані фільтри за допомогою технології WebGL.

## 1.3. Аналіз вимог до програмного забезпечення

### ***1.3.1. Функціональні вимоги***

До програмного забезпечення висуваються наступні вимоги:

1. Програмне забезпечення повинно містити поле, в якому розміщуватиметься зображення.
2. Виконувач – технологія WebGL, яка задає для зображення кольорову матрицю.
3. Програма повинна надавати можливість завантажувати картинку за допомогою відповідної кнопки та режиму drag&drop.
4. Програма повинна надавати можливість зберігати зображення у форматі .jpg без втрати якості та розширення.
5. Програма повинна надавати можливість задавати значення базових налаштувань зображення: яскравість, контраст, насиченість.
6. Програма повинна надавати можливість застосовувати попередньо підготовлені фільтри до зображення (не менше 10 фільтрів).

### ***1.3.2. Вимоги до складу та параметрів технічних засобів***

Для нормальної роботи розробленої прикладної програми потрібно пристрій із встановленим браузером Google Chrome та наступними мінімальними характеристиками:

|  |  |
| --- | --- |
| Параметр | Значення |
| Процесор | Intel Core 2 Duo або кращий |
| ОЗП | 1024 МБ або більше |
| Об’єм постійної пам’яті | 20 ГБ і більше |
| Роздільна здатність екрану | 1280х1024 і більше |

### ***1.3.3. Вимоги до інтерфейсу***

Інтерфейс програмного продукту повинен бути легким та зрозумілим користувачу.

### ***1.3.4. Вимоги до інформаційної та програмної сумісності***

Необхідною вимогою до програмного забезпечення є встановлена версія браузеру Google Chrome від v.50.

### ***1.3.5. Вимоги до тестування програмного забезпечення***

Для тестування програмного забезпечення необхідно розробити методики тестування, виконання яких забезпечує повне покриття функціональних та нефункціональних вимог:

1. Тестування завантаження зображення за допомогою відповідної кнопки.
2. Тестування завантаження зображення за допомогою режиму drag&drop.
3. Тестування зміни базових налаштувань: яскравість, контраст, насиченість.
4. Тестування кнопки Reset для базових налаштувань.
5. Тестування зміни підготовлених фільтрів.
6. Тестування збереження оброблених зображень.
7. Тестування вимог до складу та параметрів апаратних засобів.
8. Тестування вимог до інформаційної та програмної сумісності.

Після цього програмне забезпечення вважається готовим до роботи.

## 1.4. Вимоги до програмної документації

Програмне забезпечення постачається разом із супроводжувальною документацією в склад якої входить:

1. Технічне завдання.
2. Опис та обґрунтування обраної архітектури.
3. Функціональна специфікація.
4. Технічна специфікація.
5. Опис програми.
6. Програма та методика випробувань.

## 1.5. Стадії та етапи розробки

Таблиця 1.5.1

Стадії та етапи розробки

|  |  |  |
| --- | --- | --- |
| № п/п | Назва етапів кваліфікаційної роботи | Термін виконання етапів роботи |
| 1 | Одержання технічного завдання. | 15.12.15р. |
| 2 | Аналіз літератури. | 20.12.15р. |
| 3 | Розробка інтерфейсу. | 06.01.16р. |
| 4 | Реалізація введення не продуктивного часу. | 13.01.16р. |
| 5 | Реалізація введення задач. | 20.01.16р. |
| 6 | Реалізація аналізу наявних задач. | 15.02.16р. |
| 7 | Реалізація виведення побудованого плану. | 25.03.16р. |
| 8 | Тестування та відлагодження програми. | 10.04.16р. |
| 9 | Оформлення програмної документації. | 10.04.16р. |
| 10 | Представлення готової роботи. | 14.04.16р. |
| 11 | Захист роботи. | згідно розкладу |

## 1.6. Порядок контролю і приймання

Програмне забезпечення повинне бути випробуване та протестоване виробником на наявність помилок.

Порядок контролю і приймання проекту полягає у наступному:

1. Перевірка правильного функціонування програми керівником проекту.
2. Перевірка відповідності стандартам оформлення документації.
3. Перевірка правильного оформлення програмної документації керівником проекту.
4. Проходження нормо-контролю роботи.
5. Попередній захист курсової роботи.

**ПРАКТИЧНА ЧАСТИНА**

# 2. АРХІТЕКТУРА, ФУНКЦІОНАЛЬНІ ТА ТЕХНІЧНІ ПОКАЗНИКИ

## 2.1. Призначення та область застосування

Реалізований двовимірний клітинний автомат Мураха Ленгтона призначений для моделювання поведінки мурахи в певних умовах при певних правилах та спостереження за нею.

*Область застосування* – програмне забезпечення для пристроїв з ОС Windows/Linus та браузеру Google Chrome, що генерує задані фільтри за допомогою технології WebGL.

## 2.2. Опис та обґрунтування обраної архітектури

Для розробки цього проекту було використано модульну архітектуру. Компоненти розробленого ПЗ подано на рис.2.2.1.

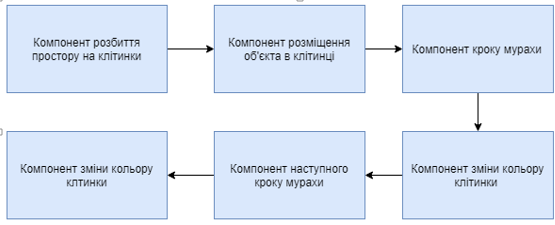


Рисунок 2.2.1 – Компоненти розробленого програмного забезпечення

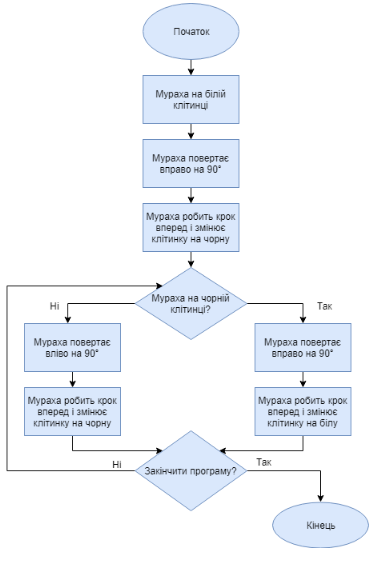


Рисунок 2.2.2 – Загальна блок-схема роботи ПЗ

Як видно з рис.2.2.2, робота з програмою починається з початкової позиції мурахи на білій клітинці, після чого мураха повертає вправо та робить крок в перед змінюючи клітинку на чорну. Опиняючись на білій клітинці мураха повертає вправо та робить крок в перед змінюючи її на чорну , якщо мураха на чорній клітинці вона повертає вліво та робить крок в перед змінюючи її на білу. Програмою передбачено можливість зміни напрямку та можливість додавання інших кольорів .

Натиснувши на кнопку старт починається рух мурахи, за бажанням можна зупинити її рух натиснувши на паузу та повністю почати заново рух натиснувши на кнопку оновити.

У вікні налаштувань є поле вводу секунд для зміни швидкості та поле , яке відображає кількість кроків мурахи.

Також у вікні налаштувань передбачена можливість зміни напрямку та додавання кольору.

## 2.3. Функціональна специфікація

### ***2.3.1. Опис функціональних можливостей***

# З документації що може дана технологія

# Розроблена програма надає можливість користувачу розпізнавати текст на вибраному користувачем зображенні. Вибране зображення можу бути переглянуте в масштабі 1:1 в окремому вікні. Також користувач самостійно обирає мову що використовується на зображенні. Серед функціональних складових програми присутня можливість збереження тексту що був отриманий в ході розпізнавання та можливість вибору користувачем зручного для нього формати збереження тексту.

На рис.2.3.1 наведено блок-схему алгоритму виконання роботи.

### ***2.3.2. Опис інтерфейсу користувача***

Робота з програмою вимагає лише базових знань та навичок роботи з ОС Windows.

Розроблене програмне забезпечення – набір класів, які об’єднані з графічним інтерфейсом користувача.

При запуску програми, мураха знаходиться на початковій позиції. Для початку руху мурахи потрібно натиснути кнопку Play. Перший крок мурахи зображено на рис.3.2

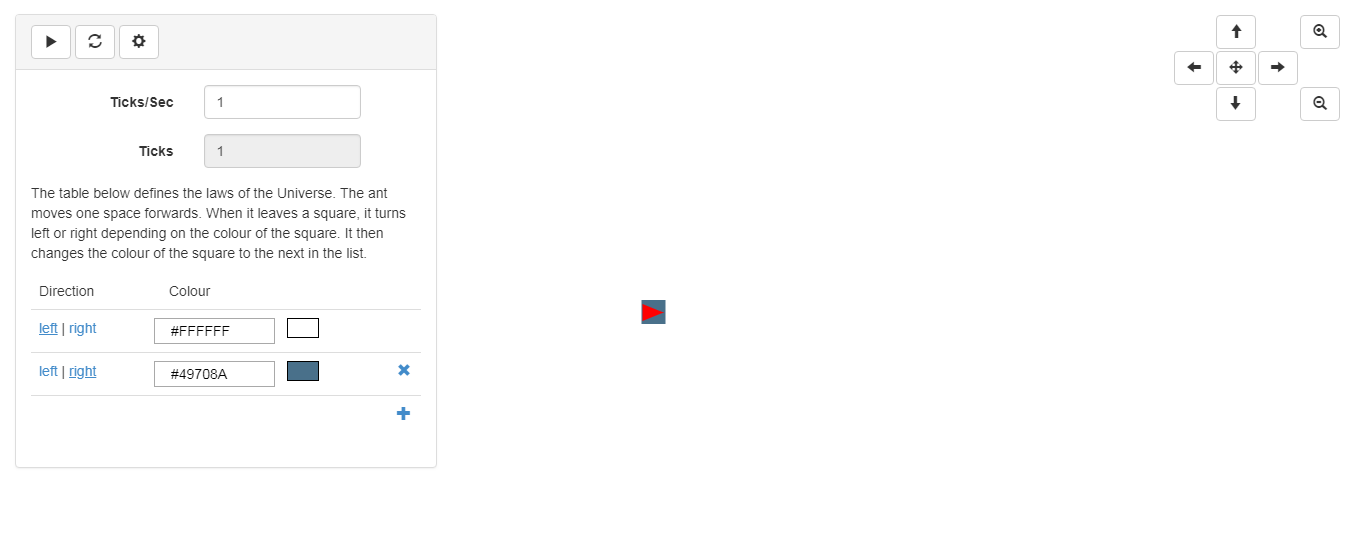


Рисунок 3.2 – Початок руху

Для того щоб зупинити рух потрібно натиснути кнопку Pause. Також передбачена кнопка оновити задля переходу мурахи в початковий стан, тобто видалення всіх попередніх кроків мурахи. По правій стороні екрану є вікно з налаштуванням. В цьому вікні передбачено наступні функції:

* Збільшення та зменшення швидкості руху шляхом вводу секунд.
* Вираховування кількості кроків, які зробила мураха.
* Інструкція по використанню.
* Зміна напрямку руху .
* Додавання кольору .

Після збільшення секунд мураха збільшить швидкістьруху(див. рис.3.3).

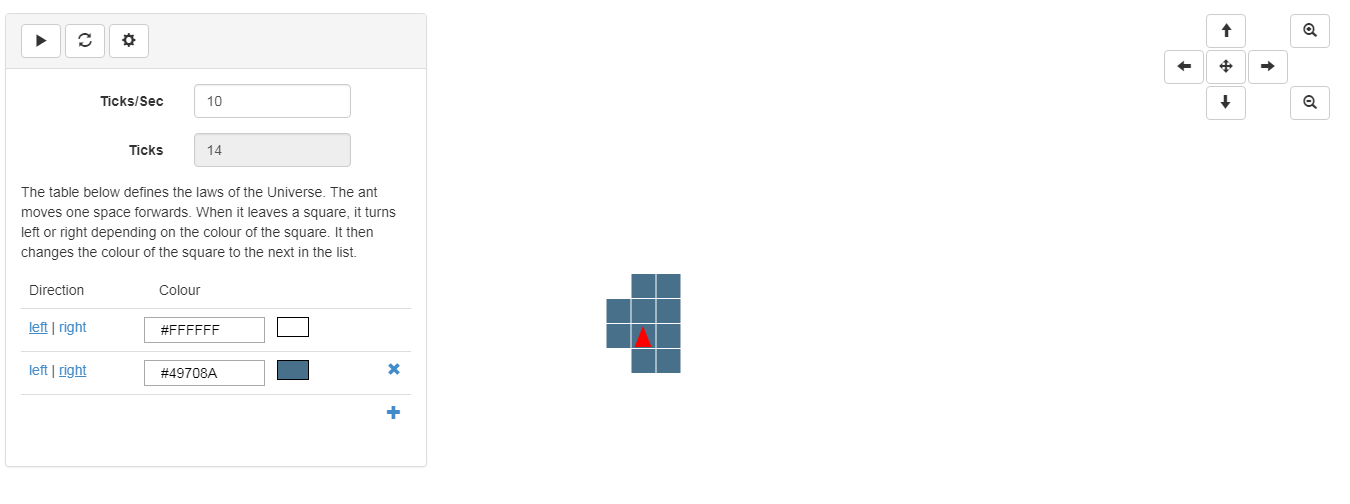


Рисунок 3.3 – Збільшення швидкості руху

Після переключення кнопок left та right мураха змінить напрям руху (див. рис.3.4).

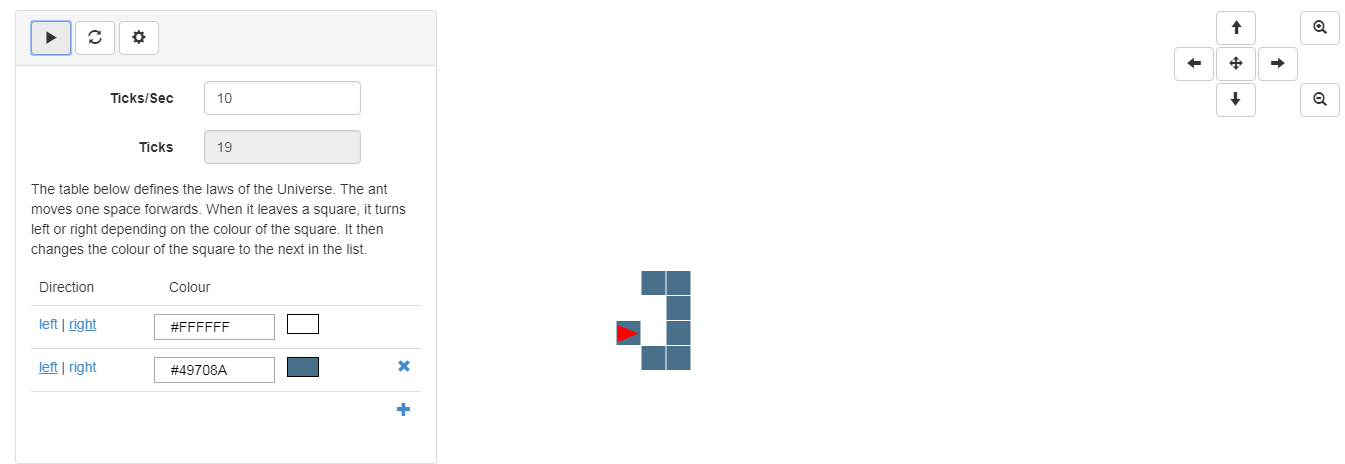


Рисунок 3.4 – Зміна напрямку руху

При натисканні на кнопку + у вікні налаштувань з’явиться додатковий колір, налаштовуємо напрямок та натискаємо кнопку Play (див. рис 3.5).

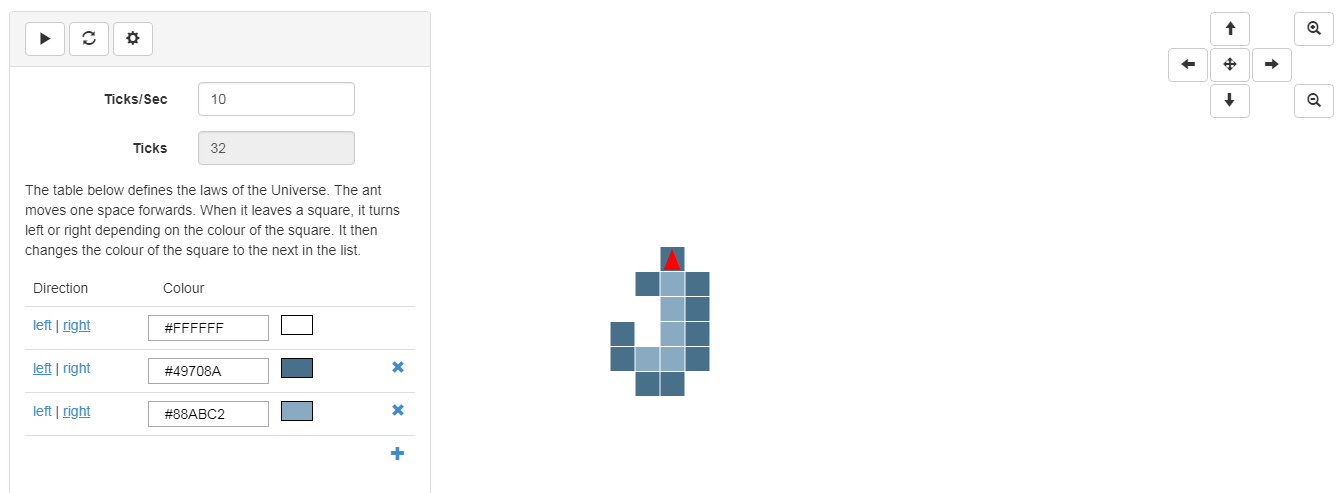


Рисунок 3.5 – Додання кольору

Також передбачено можливість змінити колір за власним бажанням , для цього потрібно вписати код в воле для вводу. (див. рис 3.6).

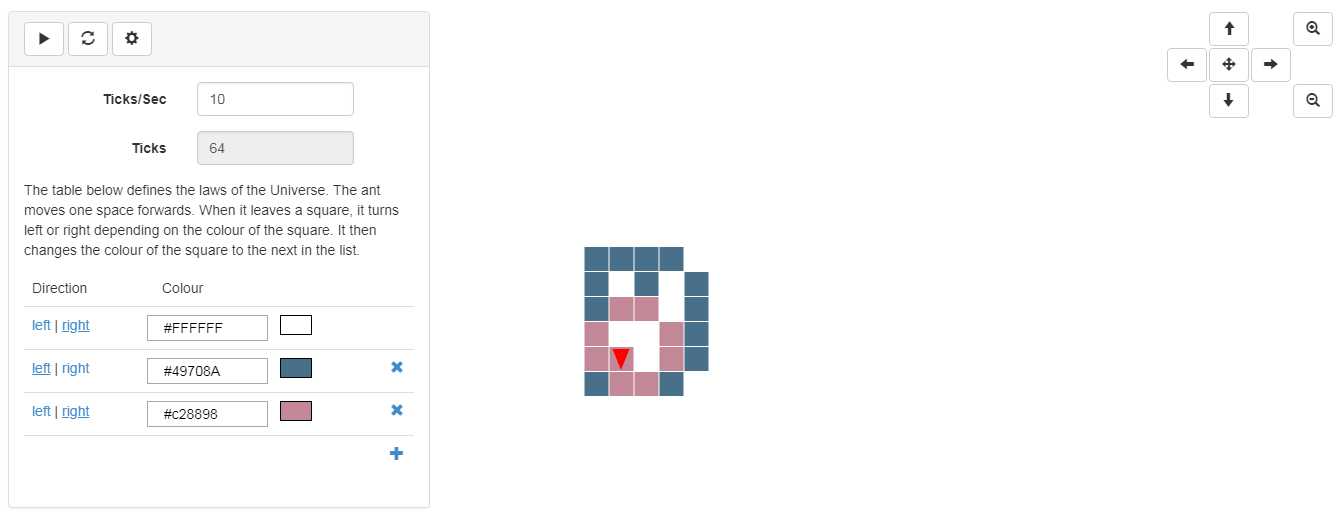


Рисунок 3.6 – Зміна кольору

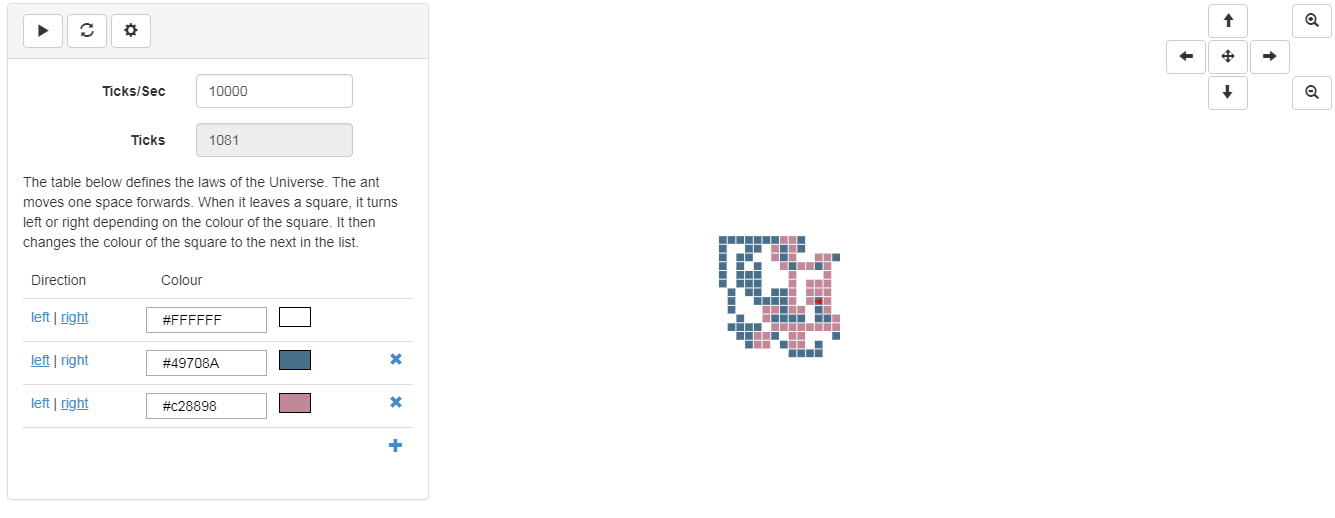
Вправій стороні вікна знаходяться кнопки для переміщення мурахи. Це зроблено для зручності спостерігання за рухом мурахи. Також передбачена можливість змільшити та зменшити масштаб (див. рис. 3.7).

Рисунок 3.7 – Зміщення малюнку вправо

Робота з програмою вимагає лише базових знань та навичок роботи з операційною системою Windows .

Розроблене програмне забезпечення – набір класів, які об’єднані з графічним інтерфейсом користувача.

При запуску програми, користувач має можливість вибрати зображення на якому він збирається розпізнати зображення за допомогою кнопки «Open File» після чого з’являється діалогове вікно з можливістю вибору. Для вибору мови передбачений комбінований список з варіантами мов. Для вибору формату в якому буде збережено розпізнаний текст передбачені радіокнопки «Plain Text», «PDF», «RTF/Word». Для вибору файлу в якому буде зберігатися розпізнаний текст передбачена кнопка «Browse To Save» при натисканні якої з’являється діалогове вікно з можливістю вибору. Для виконання безпосередньо самого розпізнавання та подальшого збереження розпізнаного тексту із заданими вище параметрами призначена кнопка «Process». Також присутнє місце для відображення вибраного зображення та кнопка «Zoom» яка відкриває це зображення в новому вікні в масштабі 1:1 для перегляду. Для закриття зображення в режимі перегляду в новому вікні передбачена кнопка «Close».

## 2.4. Технічна специфікація

### ***2.4.1. Опис модулів***

Робота розробленого програмного забезпечення реалізується наступними файлами класами:

1. **Controls** – фіксована позиція вікна налаштувань.
2. **Btn-default** – клас, в якому задається позиція мурахи.
3. **Glyphicon-pouse** – клас, за допомогою якого зупиняється рух мурахи.
4. **Glyphicon-play** – клас, за допомогою якого розпочинаються та відновлюється рух мурахи.
5. **Glyphicon-refresh** – клас, за допомогою якого очищується поле.
6. **Col-sm-1**– клас, за допомогою якого змінюється рух мурахи.
7. **Col-sm-5**– клас, за допомогою якого додається новий колір.

### ***2.4.2. Опис і обґрунтування вхідних і вихідних даних***

Вхідними даними розробленої програми є:

* швидкість руху;
* зміна кольору;
* зміна руху мурахи.

Вихідними даними розробленої програми є – малюнок руху мурахи.

# 3. КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 3.1. Опис і обґрунтування обраних програмних засобів

Розробка реалізована засобами середовища Visual Studio на мові JavaScript з використанням фреймворку AngularJS . Дане середовище є зручним у використанні для швидкого та якісного створення програмного забезпечення та веб-сторінок.

AngularJS - [JavaScript](https://uk.wikipedia.org/wiki/JavaScript)-[фреймворк](https://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) з [відкритим](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) [програмним кодом](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%80%D1%86%D0%B5%D0%B2%D0%B8%D0%B9_%D0%BA%D0%BE%D0%B4), який розробляє [Google](https://uk.wikipedia.org/wiki/Google" \o "Google). Призначений для розробки [односторінкових додатків](https://uk.wikipedia.org/wiki/Single_Page_Application" \o "Single Page Application), що складаються з одної [HTML](https://uk.wikipedia.org/wiki/HTML" \o "HTML)сторінки з [CSS](https://uk.wikipedia.org/wiki/CSS) і [JavaScript](https://uk.wikipedia.org/wiki/JavaScript" \o "JavaScript).

AngularJS спроектований з переконанням, що [декларативне програмування](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) найкраще пасує для побудови [інтерфейсів користувача](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%87%D0%B0) та опису програмних компонентів, в той час як [імперативне програмування](https://uk.wikipedia.org/wiki/%D0%86%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) пасує для опису [бізнес-логіки](https://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%B7%D0%BD%D0%B5%D1%81-%D0%BB%D0%BE%D0%B3%D1%96%D0%BA%D0%B0). Фреймворк адаптує та розширює традиційний HTML, щоб забезпечити двосторонню прив'язку даних для динамічного контенту, що дозволяє автоматично синхронізувати модель та вид. У результаті AngularJS зменшує роль [DOM](https://uk.wikipedia.org/wiki/DOM)-маніпуляцій з метою підвищення продуктивності і спрощення [тестування](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F).

## 3.2. Опис програми

### ***3.2.1. Функціональні можливості***

В створеному програмному забезпеченні розроблені такі функції:

1. вибирати зображення яке буде скануватись;
2. перегляд зображення в реальному розмірі;
3. вибір мови яка використовується на зображенні;

* зберігати розпізнаний текст;
* вибору файлу для зберігання;

### ***3.2.2. Опис логічної структури***

Структуру даного програмного забезпечення зручно подати у вигляді наступної діаграми варіантів використання (див. рис.3.1):

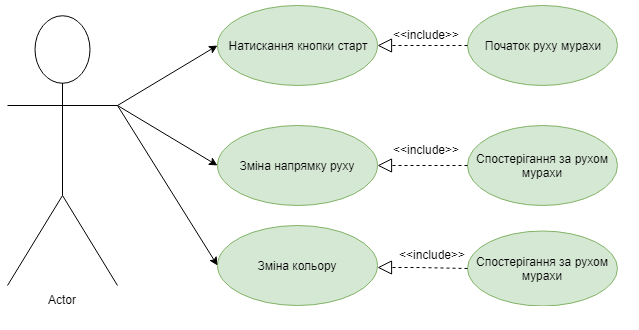


Рисунок 3.1 – Взаємодія користувача з програмою

### ***3.2.3. Використані технічні засоби***

Розробка програмного продукту здійснювалась на персональному комп’ютері наступної конфігурації:

1. Процесор – Intel(R) Core(TM) i7-3612QM CPU @ 2.10 GHz.
2. ОЗП – 8 Гб.
3. Відеоадаптер – ATI AMD Radeon HD 7600M (1024 Мб).
4. Жорсткий диск –Seagate ST1000LM ( 931GB, SATA II).
5. DVD-RW – LG DVD+-RW.

Даний комп’ютер працює під ОС Windows 10 Enterprise.

Використовувалися такі засоби розробки:

1. Visual Studio.

2) AngularJS.

3) JavaScript.

# 4. ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

## 4.1. Об’єкт випробувань

Об’єктом випробування є скрипт, який запускається на пристрію з ОС Windows.

Мета випробувань – задовольнити всі функціональні та нефункціональні вимоги до програмного забезпечення у частині 3.2.4. *Вимоги до тестування програмного забезпечення.*

## 4.2. Використані технічні засоби

Тестування проводилось на пристроях з наступною конфігурацією:

|  |  |  |
| --- | --- | --- |
| Параметри | Пристрій №1 | Пристрій №2 |
| Процесор | Intel Core i3-2350М | Intel Core i3-2100 |
| ОЗП | 4 Гб | 4 Гб |
| Версія ОС | Windows 7 | Windows 10 |
| Пам'ять | 500 ГБ | 500 ГБ |

## 4.3. Порядок та методика випробувань

Порядок проведення випробувань програмного забезпечення:

1. Запуск програми.
2. Перевірка на коректність відображення графічного інтерфейсу.
3. Задання швидкості руху.
4. Натискання на кнопку Play.
5. Спостерігання за рухом.
6. Збільшення швидкості.
7. Спостерігання за рухом.
8. Зміна напрямку руху.
9. Спостерігання за рухом.
10. Додання кольору
11. Спостерігання за рухом.

Під час проведення тестування виконано такі тести:

1. Тестування додання швидкості.
2. Тестування зміни напряму руху.
3. Тестування додавання нових кольорів.

Порядок та методика випробувань кожного з тестів подані у Додатку Б

## 4.4. Результати випробувань

Випробування програмного продукту проводилось на двох різних пристроях на базі ОС Windows. При проведенні випробувань серйозних недоліків не виявлено.

# ВИСНОВКИ

В результаті виконання курсової роботи розроблено програмне забезпечення для реалізації двовимірного клітинного автомату Мураха Ленгтона.

Програма доволі функціональна та зручна у використанні.

Тестування програмного забезпечення показало його працездатність і відповідність технічним вимогам.

Розроблене програмне забезпечення має наступні можливості:

1. Збільшення та зменшення швидкості руху.
2. Зміна напрямку руху.
3. Додавання нового кольору.
4. Переміщення об’єкта.

Проведено усі необхідні роботи по проектуванню архітектури програми, проаналізовані вимоги до неї, приводився опис реалізації, кодування, тестування програми.

У подальших дослідженнях дана розробка може бути розширена через додавання нового функціоналу.

# ВИКОРИСТАНА ЛІТЕРАТУРА

1. David Darling. The Universal Book of Mathematics: From Abracadabra to Zeno's Paradoxes. – John Wiley & Sons, 2004.
2. [Иэн Стюарт](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%8E%D0%B0%D1%80%D1%82,_%D0%98%D1%8D%D0%BD_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA)). Величайшие математические задачи М.: Альпина нон-фикшн, 2016. – 460 с..
3. [Величайшие математические задачи](https://rutlib6.com/book/26401). [Електронний ресурс]. Режим доступу: https://rutlib6.com/book/26401/p/27/

# ДОДАТКИ

**Додаток А. Скролінг (текст) програми**

DOCTYPE html>

<html ng-app="langtonsant">

<head>

<title>Langton's Ant</title>

<link rel="stylesheet" type="text/css" href="bower\_components/bootstrap/dist/css/bootstrap.min.css">

<script src="bower\_components/angular/angular.min.js"></script>

<script src="js/langtonsant.js"></script>

<script src="js/app.js"></script>

<script src="js/controllers.js"></script>

<script src="js/directives.js"></script>

<style>

/\* Styling needed for a fullscreen canvas and no scrollbars. \*/

body, html { width: 100%; height: 100%;margin: 0;

padding: 0;

overflow: hidden;

}

#antcanvas {

display: block;

}

#controls {

position: fixed;

left: 20px;

top: 20px;

}

#collapseSettings {

width: 420px;

}

#locationControls {

position: fixed;

right: 20px;

top: 20px;

}

#info {

position: fixed;

bottom: 20px;

right: 20px;

}

</style>

</head>

<body>

<canvas id="antcanvas"></canvas>

<div id="controls" ng-controller="MainController as main">

<div class="panel panel-default">

<div class="panel-heading">

<div>

<button type="button" ng-click="main.getCurrentState() == 'running' ? main.pause() : main.run()"

class="btn btn-default">

<span ng-show="main.getCurrentState() == 'running'" class="glyphicon glyphicon-pause"></span>

<span ng-show="main.getCurrentState() != 'running'" class="glyphicon glyphicon-play"></span>

</button>

<button type="button" ng-click="main.reset()"

class="btn btn-default"><span class="glyphicon glyphicon-refresh"></span></button>

<button type="button" data-toggle="collapse" data-target="#collapseSettings"

class="btn btn-default"><span class="glyphicon glyphicon-cog"></span></button>

</div>

</div>

<div id="collapseSettings" class="panel-collapse in">

<div class="panel-body">

<div class="row">

<form class="form-horizontal" role="form">

<div class="form-group">

<label for="tps" class="col-sm-5 control-label">Ticks/Sec</label>

<div class="col-sm-5">

<input type="text" class="form-control" id="tps" ng-model="main.tickFrequency">

</div>

</div>

<div class="form-group">

<label for="ticks" class="col-sm-5 control-label">Ticks</label>

<div class="col-sm-5">

<input type="text" class="form-control" id="ticks" readonly ng-model="main.info.currentTicks">

</div>

</div>

</form>

</div>

<p>The table below defines the laws of the Universe. The ant moves one space

forwards. When it leaves a square, it turns left or right depending on the colour

of the square. It then changes the colour of the square to the next in the list.</p>

<table class="table">

<thead>

<tr>

<td>Direction</td>

<td>Colour</td>

<td></td>

</tr>

</thead>

<tr ng-repeat="state in main.states">

<div class="row">

<td class="col-sm-4"><la-leftright value="state.direction"></la-leftright></td>

<td class="col-sm-7"><la-colourpicker colour="state.colour"></la-colourpicker></td>

<td class="col-sm-1"><a href="" ng-hide="$first" ng-click="main.removeState(state)"><span class="glyphicon glyphicon-remove"></span></a></td>

</div>

</tr>

<tr>

<td></td>

<td></td>

<td><a href="" ng-click="main.addState()"><span class="glyphicon glyphicon-plus"></span></a></td>

</tr>

</table>

</div>

</div>

</div>

<div id="locationControls">

<table>

<tr>

<td></td>

<td>

<button class="btn btn-default" title="Move Up" ng-click="main.moveUp()"><span

class="glyphicon glyphicon-arrow-up"></span></button>

</td>

<td></td>

<td>

<button class="btn btn-default" title="Zoom In" ng-click="main.zoomIn()"><span

class="glyphicon glyphicon-zoom-in"></span></button>

</td>

</tr>

<tr>

<td>

<button class="btn btn-default" title="Move Left" ng-click="main.moveLeft()"><span

class="glyphicon glyphicon-arrow-left"></span></button>

</td>

<td>

<button class="btn btn-default" title="Go to Centre" ng-click="main.moveOrigin()"><span

class="glyphicon glyphicon-move"></span></button>

</td>

<td>

<button class="btn btn-default" title="Move Right" ng-click="main.moveRight()"><span

class="glyphicon glyphicon-arrow-right"></span></button>

</td>

<td></td>

</tr>

<tr>

<td></td>

<td>

<button class="btn btn-default" title="Move Down" ng-click="main.moveDown()"><span

class="glyphicon glyphicon-arrow-down"></span></button>

</td>

<td></td>

<td>

<button class="btn btn-default" title="Zoom Out" ng-click="main.zoomOut()"><span

class="glyphicon glyphicon-zoom-out"></span></button>

</td>

</tr>

</table>

</div>

</div>

<script>

//  Setup the canvas.

var canvas = document.getElementById("antcanvas");

function sizeCanvasToWindow() {

canvas.width = window.innerWidth;

canvas.height = window.innerHeight;

}

window.addEventListener('resize', function resize(event) {

sizeCanvasToWindow();

// try and get the service injector. if we can, we can re-render.

var scope = angular.element(document.getElementById('antcanvas')).scope();

if (scope !== null && scope !== undefined) {

scope.render();

}

});

sizeCanvasToWindow();

</script>

<script type="text/javascript">

var \_gaq = \_gaq || [];

\_gaq.push(['\_setAccount', 'UA-41728580-1']);

\_gaq.push(['\_trackPageview']);

(function () {

var ga = document.createElement('script');

ga.type = 'text/javascript';

ga.async = true;

ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';

var s = document.getElementsByTagName('script')[0];

s.parentNode.insertBefore(ga, s);

})();

</script>

<script src="bower\_components/jquery/jquery.min.js"></script>

<script src="bower\_components/bootstrap/dist/js/bootstrap.min.js"></script>

</body>

</html>

## Додаток Б. Порядок та методики випробувань

1. **Тестування програмного продукту.**

*Об’єкт випробувань* – файл програми, який відкривається через Visual Studio та відображається в браузері.

*Мета випробувань* – перевірка функції зміти руху та додоавання кольору.

*Методика випробувань*.

1. Запускається програма.
2. Браузер відображає початкову сторінку (див. рис.Б.1):

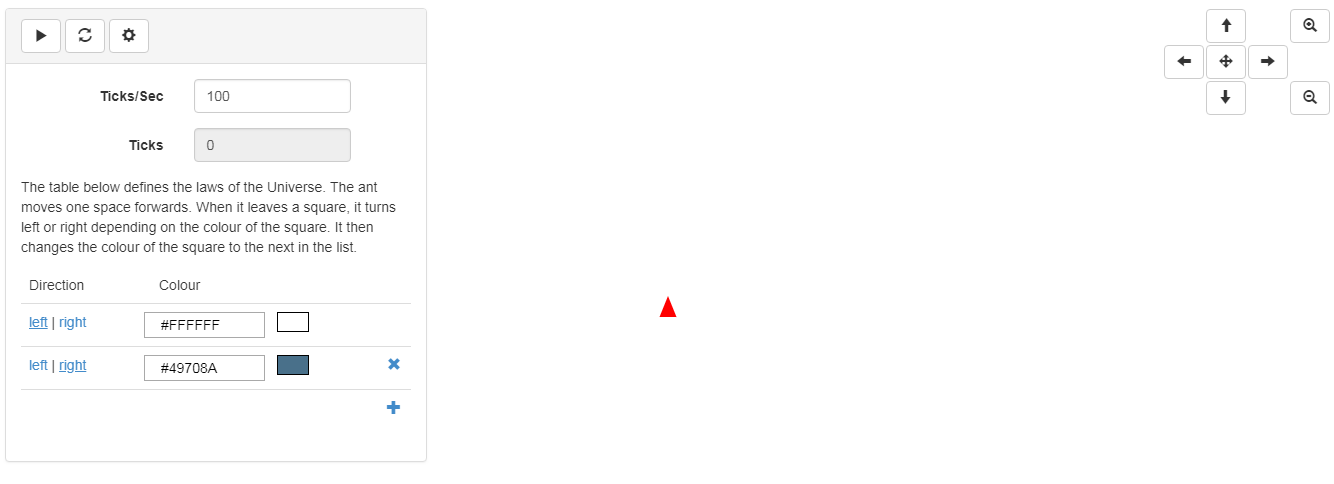


Рисунок Б.1 – Початкова сторінка

1. Для початку руху мурахи потрібно натиснути кнопку Play. Перший крок мурахи зображено на (див. рис.Б.2).

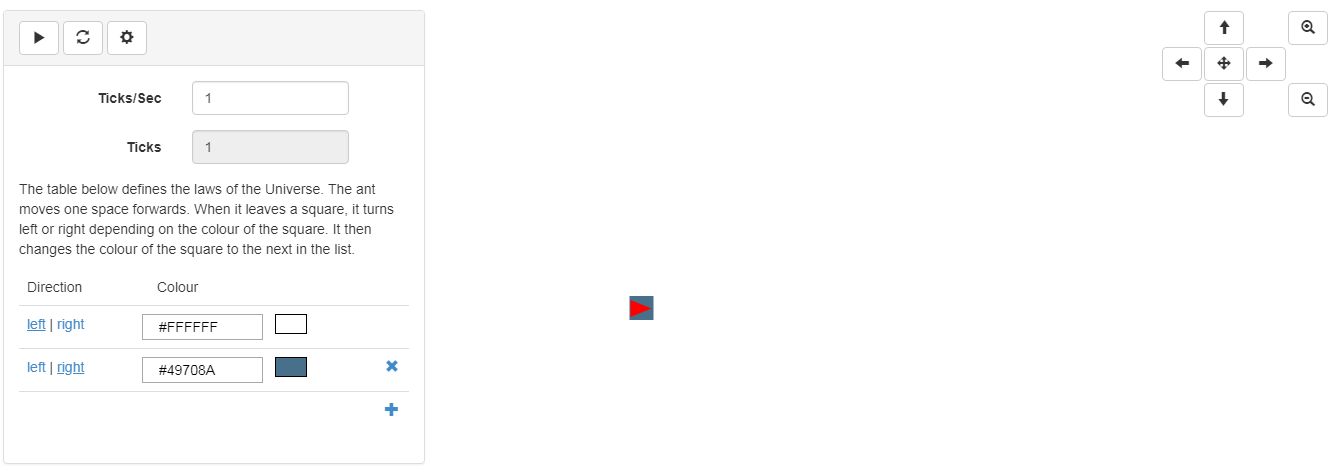


Рисунок Б.2 – Початок руху

Для того щоб зупинити рух потрібно натиснути кнопку Pause. Також передбачена кнопка оновити задля переходу мурахи в початковий стан, тобто видалення всіх попередніх кроків мурахи. По правій стороні екрану є вікно з налаштуванням.

В цьому вікні передбачено наступні функції:

* Збільшення та зменшення швидкості руху шляхом вводу секунд.
* Вираховування кількості кроків, які зробила мураха.
* Інструкція по використанню.
* Зміна напрямку руху.
* Додавання кольору.

Після збільшення секунд мураха збільшить швидкістьруху(див. рис.Б.3).

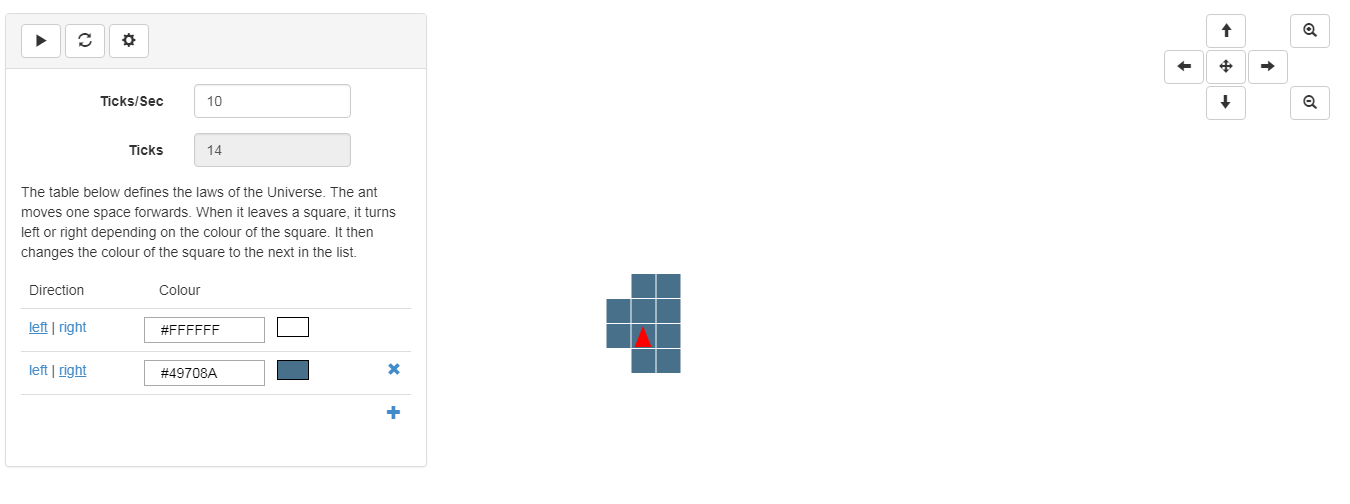


Рисунок Б.3 – Збільшення швидкості руху

Після переключення кнопок left та right мураха змінить напрям руху (див. рис.Б.4).

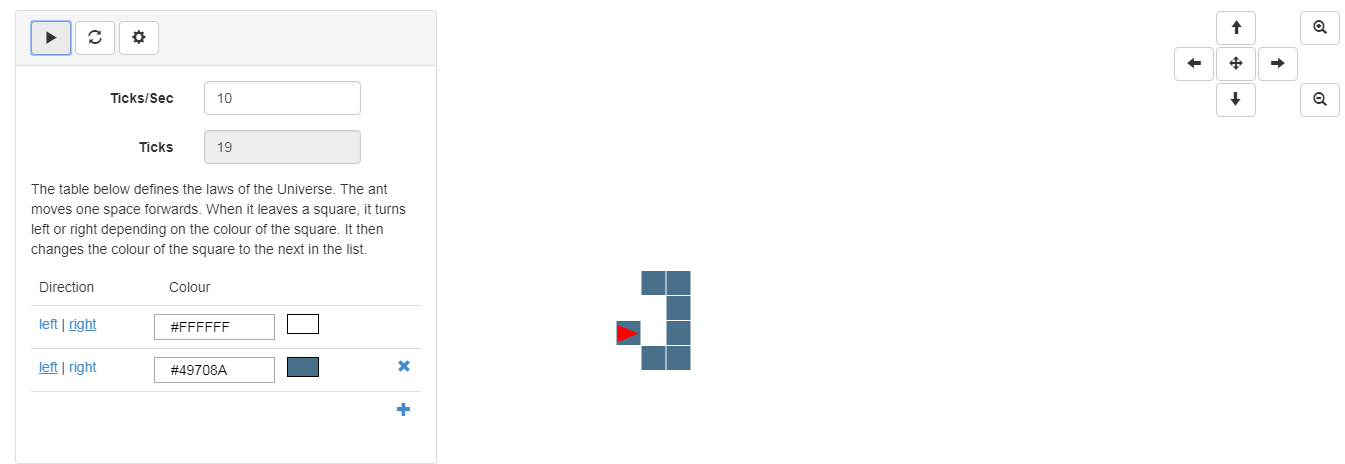


Рисунок Б.4 – Зміна напрямку руху

При натисканні на кнопку + у вікні налаштувань з’явиться додатковий колір, налаштовуємо напрямок та натискаємо кнопку Play (див. рис Б.5).

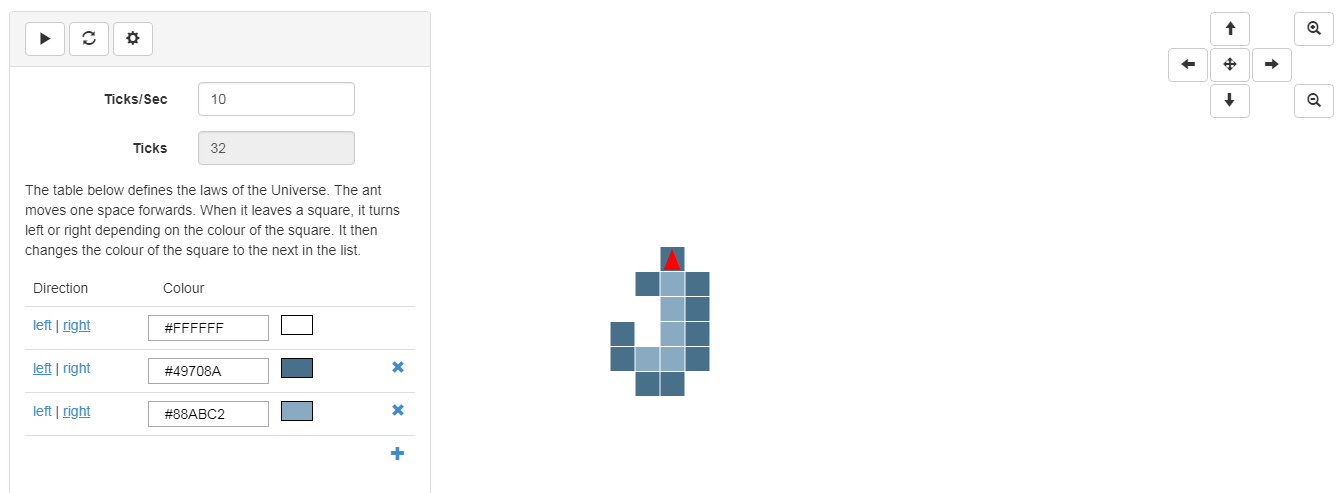


Рисунок Б.5 – Додання кольору

Також передбачено можливість змінити колір за власним бажанням , для цього потрібно вписати код в воле для вводу. (див. рис Б.6).

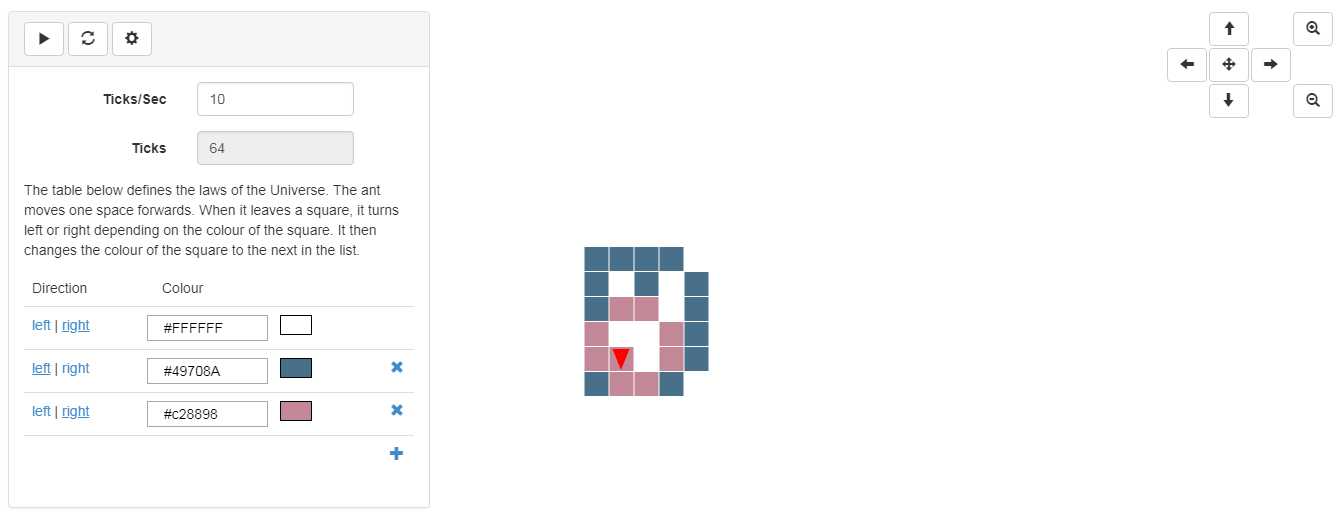


Рисунок Б.6 – Зміна кольору

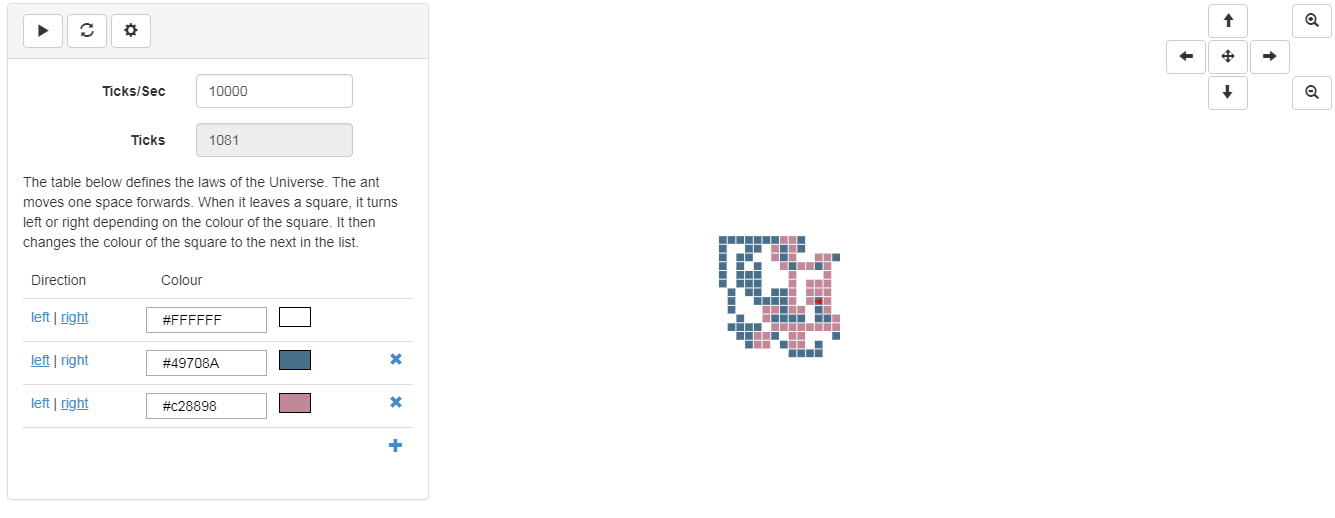
В правій стороні вікна знаходяться кнопки для переміщення мурахи. Це зроблено для зручності спостерігання за рухом мурахи. Також передбачена можливість змільшити та зменшити масштаб (див. рис.Б.7).

Рисунок Б.7 – Зміщення малюнку вправо

*Результати випробувань.*

Тест вважається пройденим, якщо виконуються наступні вимоги:

* Змінюється рух мурахи;
* Додається колір мурахи.

Результатом роботи програми можна вважати магістраль, яку будує мураха після довгих хаотичних кроків. (див. рис.Б.8).

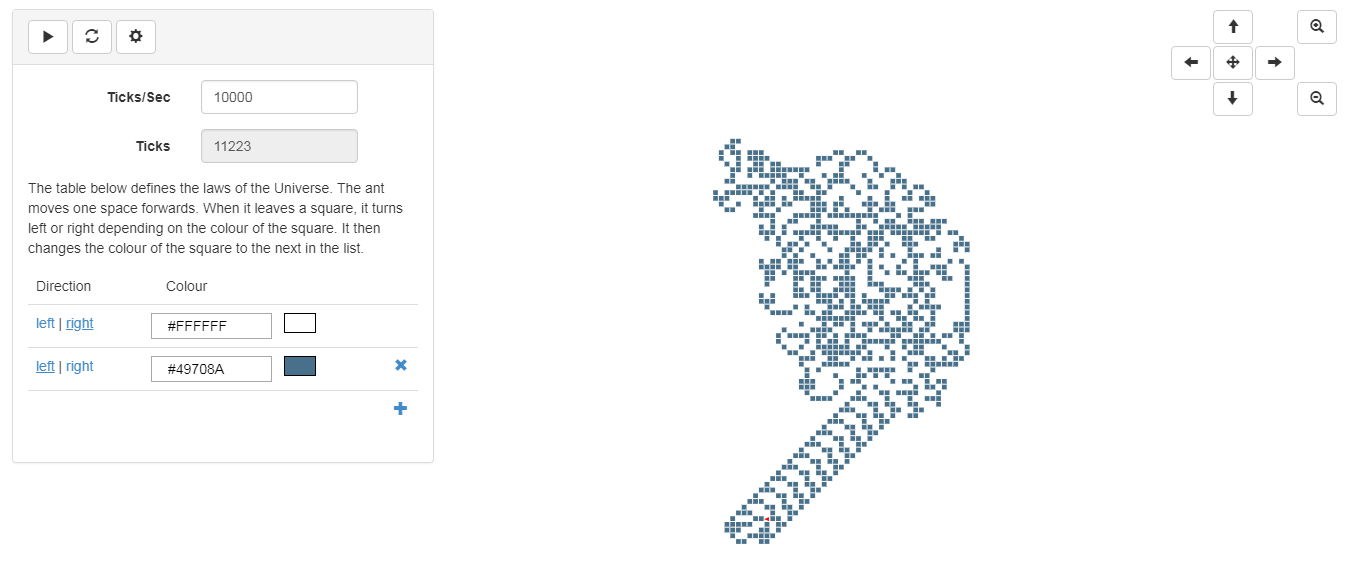


Рисунок Б.8 – Магістраль

Після зміни руху мурахи можна спостерігити те, що через певну кількість кроків мураха почне знову будувати магістраль (див. рис.Б.9).

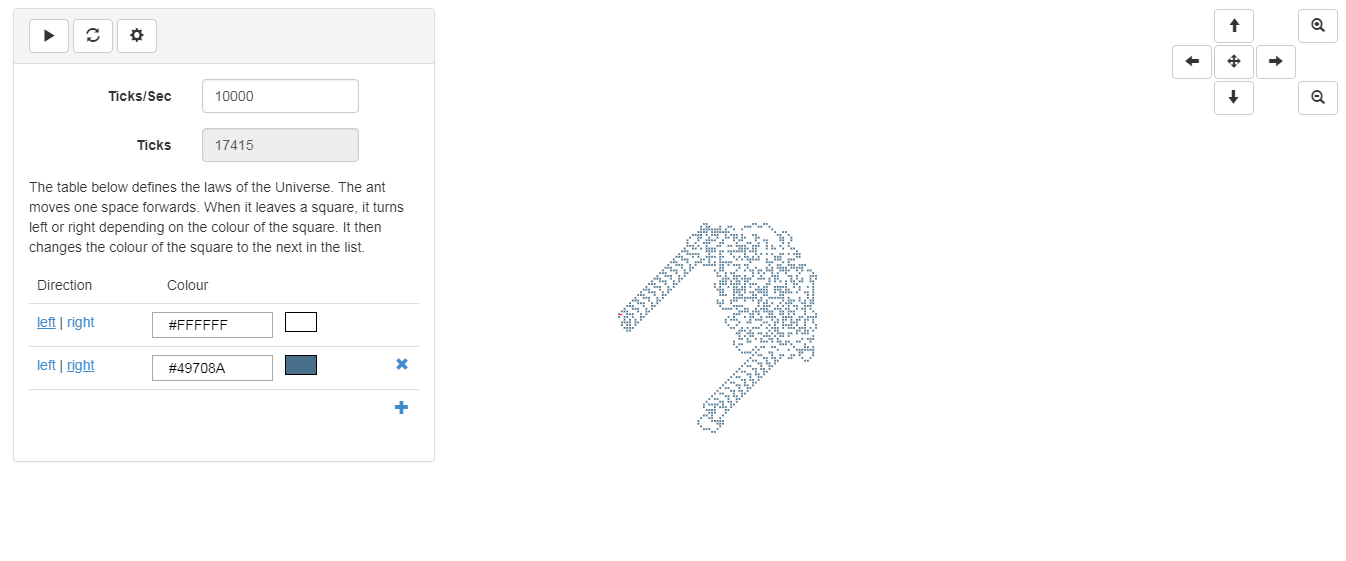


Рисунок Б.9 – Магістраль після зміни руху