# Анотація

# Зміст

[1 Анотація 1](#_Toc453516939)

[2 Зміст 2](#_Toc453516940)

[3 Вступ 3](#_Toc453516941)

[3.1 Огляд та аналіз сучасного стану 3](#_Toc453516942)

[3.2 Обгрунтування актуальності теми і мети роботи 3](#_Toc453516943)

[4 Аналіз 4](#_Toc453516944)

[5 Вибір засобів реалізації 5](#_Toc453516945)

[6 Реалізація 6](#_Toc453516946)

[7 Висновки 7](#_Toc453516947)

[8 Список використаних джерел 8](#_Toc453516948)

[9 Додатки 9](#_Toc453516949)

# Короткий словник основних термінів

Natural User Interface (NUI) - користувальницький інтерфейс, в якому використовуються природня поведінка користувача, як то жести, розмова, положення тіла, тощо.

# Вступ

## Огляд та аналіз сучасного стану

### Сучасний стан Natural User Interface (NUI)

NUI дозволяють користувачам взаємодіяти з пристроєм за допомогою дій, які є природними, інтуїтивно зрозумілим і загальним для повсякденного людської поведінки. Розрізняють багато різних варіантів таких як: керування жестами, голосом, переміщенням тіла, керування дотиком, тощо.

Поточне рішення використовує керування за допомогою жестів, а також переміщення тіла. Тому розглянемо сучасний стан NUI в цих областях.

Для розпізнавання жестів та переміщення тіла використовуються датчики які базуються

## Обгрунтування актуальності теми і мети роботи

# Аналіз

# Вибір засобів реалізації

## Датчики жестів NUI

В цій секції проводиться аналіз існуючих датчикі жестів в рамках рішення. Головні показники для вибору:

Радіус дії датчика – рішення має працювати на відстані 1-5 метри.

Жести тіла – підтримка датчиком жестів не тільки рук, але і всього тіла.

Кількість об’єктів стеження – потенційно рішенням може користуватися більше ніж одна людина, це означає що датчик має підтримувати більше ніж одну людину

Спеціальне обладнання – датчик може потребувати додаткового обладнання

SDK – наявність бібліотек для розробки під потрібну платформу

Ціна – ціна датчика для інсталяції і розробки

Були розглянуті такі датчики:

Leap Motion – датчик ближньої дії, загальне застосування це керування комп’ютерними програми в звичайному режимі користуванні. Для нашого рішення не підходить, тому що потрібно розпізнавати жести на відстані від 1.5 метрів.

Microsoft Kinect 1.0 – передне покоління ігрових датчиків для Xbox.

Microsoft Kinect 2.0 – поточне покоління ігрових датчиків Xbox, адаптованих для промислового використання. Існують версії для PC.

PlayStation Move – ігровий датчик від Sony. Має кращу точність ніж Microsoft Kinect, але потребує додаткового датчика в руці користувача.

В додатку 11.1 наведена порівняльна характеристика цих датчиків. Базуючись на потрібних характеристиках було обрано датчик Microsoft Kinect 2.0 для PC.

## Бібліотеки виявлення жестів

Kinect 2.0 SDK

OpenNI

OpenKinect

etc

## Платформа для реалізації презентаційного інтерфейсу

Презентаційний інтерфейс це головний інтерфейс який користувачі бачать і використовують. Для вибору платформи реалізації враховувались такі основні особливості.

Швидкість промальовування – інтерфейс активно використовує анімації, складні сцени з ефектами 3D і рендерить їх на великому екрані. Дуже важливо мати підтримку апаратного прискорення.

Наявність бібліотек UI – для проекту важливо мати підтримку широкого спектру бібліотек. В майбутньому планується розширення можливостей.

Комунікація з Kinect - основна маса інтерактивності інтерфейсу базується на реакції на події які генерує Kinect.

Найбільш поширеним способом для розробки додатків з багатим інтерфейсом, які можуть взаємодіяти з Kinect є використання Windows Presentation Foundation (WPF). WPF це дуже потужна бібліотека UI, що дозволяє створювати практично будь-який парадигми призначеного для користувача інтерфейсу. WPF використовує Direct3D, а отже є аппаратно прискоренною платформою.

Другий за поширеністю спосіб це використання HTML5/JavaScript. Сучасні браузери також використовують апаратне прискорення. На відміну від WPF не дозволяє працювати з Kinect на пряму.

Таблиця 6.1 Аналіз платформ для презентаційного інтерфейсу

|  |  |  |  |
| --- | --- | --- | --- |
| **Вимога** | **WPF** | **HTML5/JS** | **Direct3D** |
| Швидкість промальовування | Апаратно прискорене | Апаратно прискорене, але використовуючи DOM | Апаратно прискорене |
| Наявність бібліотек UI | Не відповідає потребам проекту | Дуже широке | Підтримує тільки ручний рендерінг |
| Комунікація з Kinect | Пряма | Через сервер (WebSocets) | N/A |
| Відкритий код | Ні | Частково | Ні |

Аналіз показав, що HTML/JS надає значну перевагу в наявності бібліотек, а також достатню швидкість промальовування. Таким чином, було прийнято рішення не використовуват WPF, і замінити його на HTML/JS, в якості базової бібліотеки інтерфейсу було обрано AngualrJS.

# РеалізаціЯ

## Основні актори та use-case системи

Секція описує основних авторів і пов’язані з ними use-case.

Таблиця 7.1 Основні актори системи

|  |  |
| --- | --- |
| Користувачі | Сторонні люди які користуватимуться системою |
| Адміністратор | Персонал який налаштовує інсталяцію |

Користувач використовує NUI для своїх use-case, натомість адміністратор використовує звичайний веб інтерфейс. В нашій системі адміністратор це сукупна роль, яка робить і адміністрування і налаштування і звітність.

Користувач може бачити загальний план який в довільному порядку показує список проектів доступних для перегляду. Користувач може вибрати один з проектів, продивитись його, і оцінити його.

Адміністратор може подивитись звітність по оцінках і використанню.



Рисунок . Основні use-case системи

## Компоненти системи

Рішення виконане у вигляді поєднання програмного рішення і апаратних датчиків. Система складається з двох автономних пакетів:

Front-end – аплікація для користувачів рішення. Складається з інтерфейсу користувача, Kinect, сервера обробки даних з Kinect та файлової бази даних для реєстрації проектів. Розгортається на комп’ютері з підключеним Kinect.

Back-end – аплікація для адміністраторів рішення, а також для централізованого зберігання статистики. Складається з адміністративного інтерфейсу, серверу статистики і бази даних статистики.

На діаграмі нижче показані зв’язки між компонентами системи, а також актори системи.

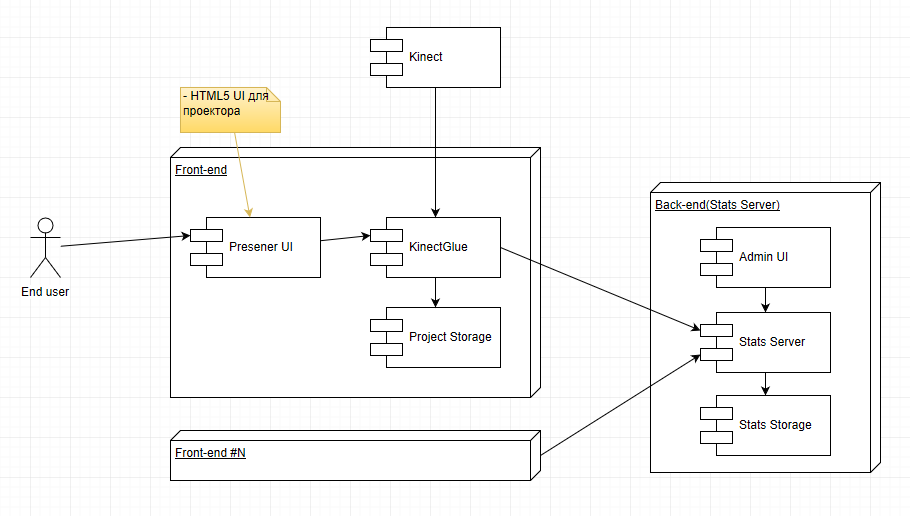


Рисунок . Компонентна діаграма рішення

Компоненти рішення працюють автономно і можуть складати інші топології. Наприклад інсталяцій Front-end може бути більше ніж одна, і вони будуть об’єднані в мережу розподілених інсталяцій, що матимуть спільну статистику і спільне адміністрування.

Модулі системи:

Presenter UI – Інтерфейс користувача, HTML5 інтерфейс який буде виводитись за допомогою проектора на великий екран.

KinectGlue – центральний сервер інсталяції, обробляє потік даних з Kinect: реалізовує алгоритми детектування жестів, оркестрацію користувачів, тощо. Надає API для Presenter UI. Підключається в режимі підписки до сервера статистики і інформує Presenter UI з новою статистикою.

Project Storage – проста файлова БД в якій зберігається контент проекту його метаінформація.

Admin UI – інтерфейс адміністратора, дозволяє дивитись агреговану статистику, а також керувати проектами.

StatsServer – центральний сервер статистики, обробляє виклики з серверів інсталяції, а також підготовлює данні для звітів на Admin UI

StatsStorage – вбудована реляційна БД, зберігає агреговану статистику по проектах

## Технічний стек

Ця секція описує технічний стек фінального рішення. Для прийняття рішення використовувались такі основні фактори:

Відповідність функціональним вимогам - можливіть сторити рішення в рамках заданих вимог

Ціна рішення – перевага була надана компонентам з відкритим кодом

Доступність знань – перевага була надана компонентам з якими був існуючий досвід

### Технічний стек Presenter UI

Інтерфейс користувача побудовано за допомогою веб технологій (TBD мотивацію), базуючись на бібліотеках з відкритим кодом.

Таблиця 7.2 Технічний стек Presenter UI

| Назва | Технологія | Мотивація |
| --- | --- | --- |
| Платформа | HTML5, JavaScript, CSS | * Наявність бібліотек * Можливість розділення логіки, представлення, та стилів |
| Бібліотека інтерфейсу | AngularJS | * Дозволяє писати складні веб сторінки |
| Бібліотека графіки | D3 | * Дозволяє візуалізовувати данні (в нашому випадку точки в просторі) |
| Юніт тести | Jasmine | * JavaScript UnitTest бібліотека |
| Менеджер пакетів | Bower | * Підходить для сайтів без компіляції |

### Технічний стек KinectGlue

Центральний сервер інсталяції побудовано використовуючи платформу .NET яка є ідеальною для роботи з Kinect SDK. Центральний сервер інсталяції використовує чотири типи комунікації. Внутрішню на основі React Extensions, зовнішню для UI – WebSockets і для комунікації з сервером статистики ZeroMQ, а також USB 3.0 для комунікації з Kinect.

Таблиця 7.3 Технічний стек KinectGlue

| Назва | Технологія | Мотивація |
| --- | --- | --- |
| Мова програмування | C# | * Чудова підтримка асинхронного програмування * Мова по замовчуванню для .NET платформи |
| Платформа | .NET | * Чудова підтримка Kinect SDK |
| Внутрішня комунікація | Reactive Extensions | * Реактивна, не блокуюча парадигма * Гнучке фільтрування та абстракція |
| Зовнішнє WebSocket АПІ | Fleck | * Підтримка протоколу WebSockets для передачі даних в реальному режимі часу |
| Юніт тести | NUnit | * .NET UnitTest бібліотека |
| Менеджер пакетів | Nuget | * Менеджер пакетів для .NET |

### Технічний стек Admin UI

Інтерфейс адміністратора побудовано за допомогою веб технологій, базуючись на бібліотеках з відкритим кодом.

Таблиця 7.4 Технічний стек Admin UI

| Назва | Технологія | Мотивація |
| --- | --- | --- |
| Платформа | HTML5, JavaScript, CSS | * Наявність бібліотек * Можливість розділення логіки, представлення, та стилів |
| Бібліотека інтерфейсу | ReactJS | * Дозволяє писати складні веб сторінки |
| Юніт тести | Jasmine | * JavaScript UnitTest бібліотека |
| Менеджер пакетів | npm | * Підходить для сайтів з компіляцією |

### Технічний стек StatsService

Сервер статистики побудовано використовуючи платформу .NET яка підходить для написання такого роду серверів. Сервер статистики використовує три типи комунікації. Внутрішню на основі React Extensions, зовнішню для UI – HTTP JSON і для комунікації з серверами інсталяцій ZeroMQ. В якості бази даних використовується внутрішня БД SQLite.

Таблиця 7.5 Технічний стек StatsServer

| Назва | Технологія | Мотивація |
| --- | --- | --- |
| Мова програмування | C# | * Чудова підтримка асинхронного програмування * Мова по замовчуванню для .NET платформи |
| Платформа | .NET | * Добре підходить для таких рішень |
| Внутрішня комунікація | Reactive Extensions | * Реактивна, не блокуюча парадигма * Гнучке фільтрування та абстракція |
| Зовнішнє HTTP JSON API | NancyFx | * Проста бібліотека для написання HTTP JSON API |
| Юніт тести | NUnit | * .NET UnitTest бібліотека |
| Менеджер пакетів | Nuget | * Менеджер пакетів для .NET |
| База даних | SQLite | * Проста в використанні реляційна БД * Поставляється у вигляді бібліотеки |

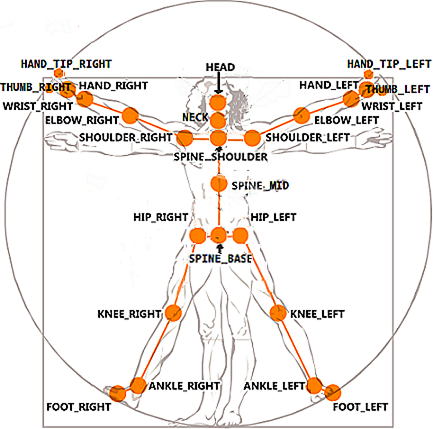
## Алгоритми виявлення жестів

### Короткі загальні відомості

Для виявлення жестів дуже важливо правильно обрати алгоритми, і данні з якими алгоритми будуть працювати. Kinect SDK 2.0 представляє декілька рівнів доступу.

RAW – прямий доступ до даних сенсору. Надаються аудіо, відео високої якості, а також один чи два канали інфрачервоного зображення високої якості. На цьому рівні немає інформації з якою було б легко працювати для виявлення жестів, проте наприклад відео інформацію можна використовувати на час розробки.

Скелетон – на цьому рівні SDK надає можливості отримати вже оброблений скелет людини чи людей за якими слідкує Kinect. Скелет надається у вигляді набору точок в просторі, а також з’єднань між ними. На цьому рівні можна розробляти свої алгоритми, що базуються на цих точках, а також часі.



Інтеракції – наступний рівень на якому SDK надає інформацію про інтерактивні рухи загалом. Інформація про рух складається з поточної позиції керуючої частини тіла (наприклад права рука), стан руки (наприклад: відкрита, захоплена) і стану перетягування.

Виявлення жестів - на цьому рівні SDK дає інструменти для того щоб записати рухи, відмаркувати в спеціальному редакторі і потім згенерувати алгоритм виявлення руху.

## Виявлення активного користувача

Активний користувач - користувач з яким в даний момент працює система. Датчик може стежити за 6 користувачами, користувачі можуть відходити і підходити. Також користувачі можуть «передавати» активність іншому користувачу.

Алгоритм пошуку активного користувача показано нижче на рис 7.2. Алгоритм запускається циклічно, паралельно до основної роботи. Данні про список тіл, та чи слідкує датчик за тілом отримується з датчику. Активний користувач зберігається в пам’яті програми.

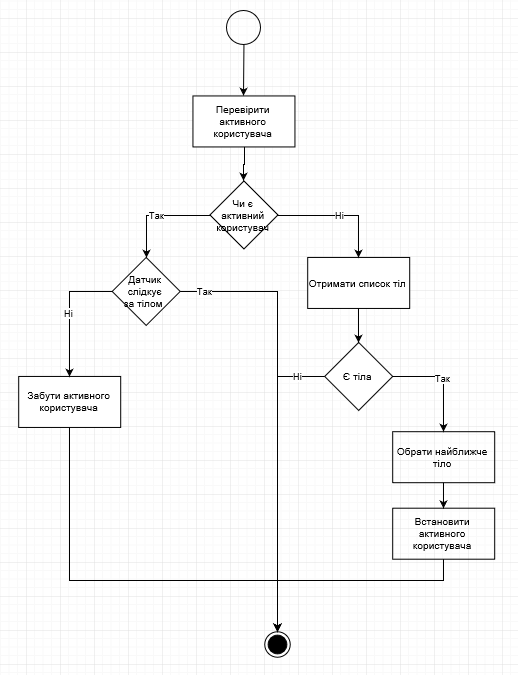


Рисунок . Алгоритм пошуку активного користувача

## Загальне бачення евристичних алгоритмів

В рішенні використовують різні типи алгоритмів. Проміж інших, виділяється група евристичних алгоритмів, які є схожими за поведінкою, а тому мають спільну основу. Усі ці алгоритми працюють з чергою точок у просторі у «вікні» часу. Результатом роботи алгоритму є або пропуск фрейму, або нотифікація виявленого руху. Нижче надано загальну схему алгоритму.

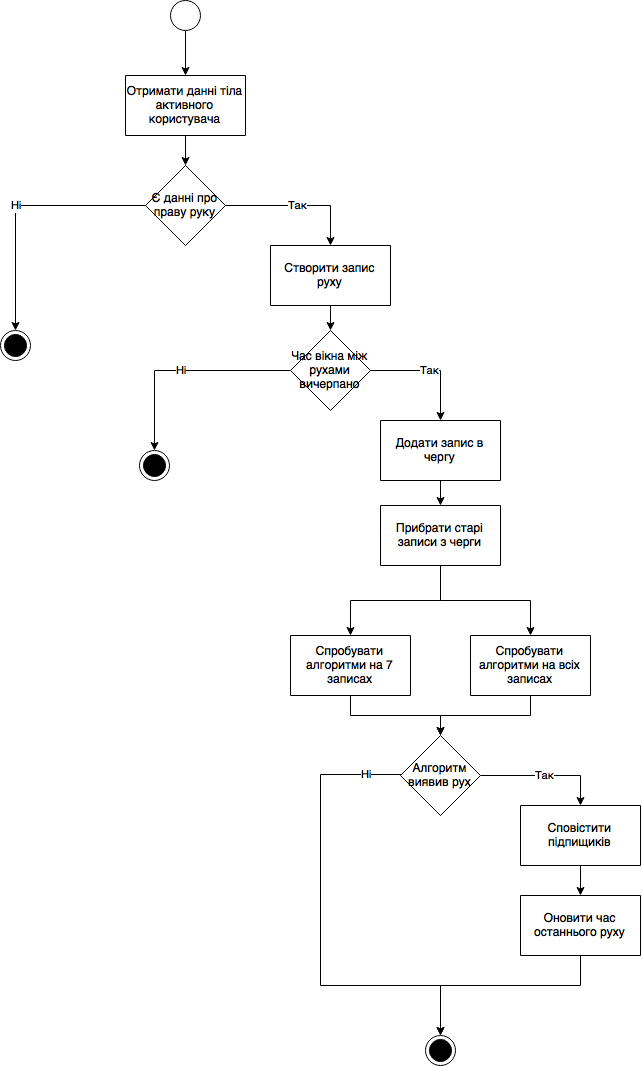


Рисунок . Загальна схема виконання я евристичних алгоритмів

## Виявлення руху перелистування

Для вибору конкретного проекту використовується рух перелистування. Це горизонтальний рух з права на ліво – для пролистування вліво, і з ліва на права – в інший бік.

В рішенні реалізовано два варіанти, перший (SwipeHorizontalGestureDetector) використовує практики евристичного програмування, і використовується по замовчанню. Другий (VgbGesturesEmiter) використовує вбудований в SDK інструментарій для генерації детектора жестів. Для навчання алгоритму було записано рухи 12 людей. За рахунок особливостей руху, перший алгоритм дає кращі результати.

В цій секції описано перший алгоритм з евристичним виявленням руху, загальна схема виконання евристичних алгоритмів надана в секції 7.5.

Алгоритм виконується циклічно, і викликається паралельно з іншими евристичними алгоритмами. Алгоритм використовує експериментально підібрані коефіцієнти. Зазвичай коефіцієнти залежать від середовища в якому буде розгорнута інсталяція, і має підбиратись в кожному конкретному випадку.

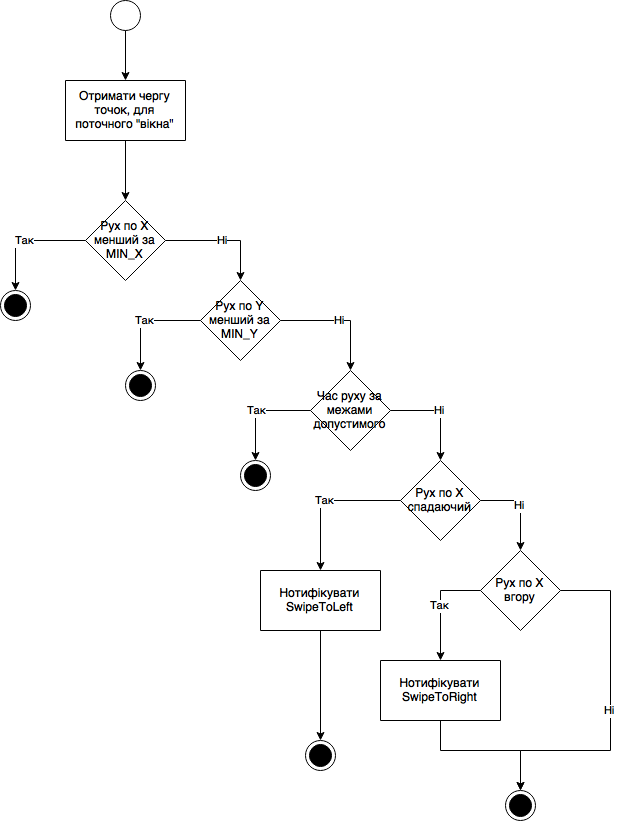


Рисунок . Алгоритм розпізнавання горизонтального руху перелистування

# Висновки

# Список використаних джерел

https://en.wikipedia.org/wiki/Natural\_user\_interface

# Додатки

## Додаток А. Порівняльна характеристика датчиків жестів

Секція надає деталізовану інформацію для прийняття рішення про вибір датчика жестів.

Таблиця 10.1. Порівнянльна таблиця датчиків жестів

| Датчик | Опис | Радіус дії датчика(м) | Жести тіла | Кількість об’єктів стеження | Спеціальне обладнання | SDK | Ціна |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Leap Motion | Датчик жестів на короткій дистанції | 0.3-1 | Ні | 1 | Ні | .NET | 2000 грн |
| Microsoft Kinect 1.0 | Ігровий датчик | 2-6 | Так | 2 | Ні | .NET | 2600грн |
| Microsoft Kinect 2.0 | Ігровий та промисловий датчик | 0.3-6 | Так | 5 | Ні | .NET | 5000грн |
| PlayStation Move | Ігровий датчик | 2-6 | Ні | 1 | Так – датчик в руці | Ні | - |