# Анотація

# Зміст

Анотація 1

Зміст 2

Короткий словник основних термінів 4

Вступ 5

1 Аналіз 8

1.1 Аналіз поведінки користувача 8

1.2 Висновок 14

2 Проектування 15

2.1 Основні актори та use-case системи 15

2.2 Інтерактивний дизайн користувача 16

2.3 Компоненти системи 17

2.4 Аналіз давачів жестів для NUI 19

2.5 Бібліотеки виявлення жестів 20

2.6 Платформа для реалізації презентаційного інтерфейсу 21

2.7 Комунікація реального режиму часу між інтерфейсом користувача і обробником Kinect 22

2.8 Комунікація в режимі підписки між сервером інсталяції і сервером статистики 24

2.9 Алгоритми виявлення жестів 25

2.10 Концепція фізичної інсталяції 31

2.11 Висновок 31

3 Реалізація 33

3.1 Взаємодія між компонентами 33

3.2 Технічний стек 33

3.3 Внутрішня організація системи 37

3.4 Результати роботи системи 43

3.5 Висновок 47

4 Економічна частина 48

4.1 Економічна характеристика проектного рішення (програмного продукту) 48

4.2 Інформаційне забезпечення та формування гіпотези щодо потреби розроблення проектного рішення (програмного продукту) 48

4.3 Оцінювання та аналізування факторів зовнішнього та внутрішнього середовищ 49

4.4 Формування стратегічних альтернатив 51

4.5 Бюджетування 53

4.6 Висновок 59

Висновки 60

Список використаних джерел 62

Додатки 63

# Короткий словник основних термінів

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

Скевоморфізм – в графічних інтерфейсах користувача, імітація реальних предметів для побудови інтуїтивних інтерфейсів.

Kinetic user interface – кінетичний інтерфейс – інтерфейс користувача який реагує на рухи користувача, положення тіла, тощо.

Інтерактивний дизайн – практика проектування інтерактивних цифрових продуктів, систем і сервісів.

Реактивне програмування – це парадигма програмування, побудована на потоках даних і розповсюдженні змін. Це означає, що у мовах програмування має бути можливість легко виразити статичні чи динамічні потоки даних, а реалізована модель виконання буде автоматично розсилати зміни через потік даних.

Евристичний алгоритм - алгоритм, спроможний видати прийнятне рішення проблеми серед багатьох рішень, але неспроможний гарантувати, що це рішення буде найкращим.

MVVM - Model-View-ViewModel – шаблон проектування який дозволяє розділяти модель даних і поведінку. Головна відмінність від класичного MVC – це активне використання автоматичного зв’язування даних і презентації.

Flux – шаблон проектування який організовує обмін даними між логікою і представлення в коло.

# Вступ

NUI дозволяють користувачам взаємодіяти з пристроєм за допомогою дій, які є природними, інтуїтивно зрозумілим і загальним для повсякденного людської поведінки. В загальному випадку користувач має використовувати існуючий досвід, або проекцію існуючого досвіду на нові можливості. Класичним прикладом проекції існуючого досвіду є скевоморфізм – імітація реальних предметів в графічних інтерфейсах користувача – дала можливість розробляти інтуїтивні інтерфейси, такі як інтерфейс медіа програвача у вигляді звичайної стерео системи.

NUI розділяють на такі великі групи:

Керування за допомогою голосу – командні інтерфейси, розмовні інтерфейси, інформативні інтерфейси. Приклади: Apple Siri, Microsoft Cortana

Tangible user interface – керування за допомогою фізичного оточення – поєднання предметів оточення з інтерфейсами програми, наприклад зміна положення предмета на столі, змінює стан роботи системи. Цікавий приклад: Reactable

Кінетичні інтерфейси – керування за допомогою рухів, це включає в себе: жести, зміна положення тіла, рух тіла, переміщення тіла в просторі. Приклади: Microsoft Kinect, Wii Motion, Leap Motion.

Natural language user interfaces – інтерфейси природньої мови що дозволяють використовувати звичайну повсякденну мову для керування. В свою чергу NLUI діляться на командні інтерфейси, розмовні інтерфейси, інформативні інтерфейси. Приклади: Apple Siri, Microsoft Cortana, Wolfram Alpha. Останній час популярність набираються текстові NLUI інтерфейси які інтегруються в загальні комунікаційні платформи, так як: Microsoft Skype, Facebook Messenger, тощо.

В цій роботі досліджується кінетичний інтерфейс - керування за допомогою жестів, а також переміщення тіла. Тому розглянемо сучасний стан NUI в цих областях.

Для розпізнавання жестів та переміщення тіла використовуються давачі сканування положення тіла в просторі. Найбільшу популярність здобули давачі глибини простору, які базуються на скануванні відбиття в інфрачервоному діапазоні мережі генерованої джерелом інфрачервоного світла. Це дає можливість отримати інформацію про відстань до кожної точки в видимому просторі. На основі інформації про відстань, використовуючи алгоритми машинного навчання такі як Random Forest класифікатор виділяє тіло на загальному плані. На наступному етапі методами машинного навчання розпізнаються конкретні частини тіла. Ця інформація стає доступною програмному забезпеченню, яке вже має можливість розпізнавати жести та рухи.

Інформації про жести, рухи, положення тіла не достатньо для побудови інтерфейсу, важливий аспект NUI це конвертація рухів в команди інтерфейсу. На цьому етапі з потоку даних з давача виокремлюються шаблони поведінки, особливості рухів, зміни положення в часі і будуються поведінкові моделі на основі який будується взаємодія з інтерфейсом. Цей аспект вивчається дисципліною інтерактивного дизайну і включає в себе: вивчення поведінки, побудова моделі поведінки, відповідь інтерфейсу на поведінку користувача, моделі навчання користувача, тощо.

NUI знаходиться на етапі ранньої адоптації по класифікації technology adoption life cycle, це означає що розвиток цього напряму буде сильно прискорюватись найближчі роки. Фактично зараз проходить етап пошуку шаблонів, повторюваних рішень і можливих сценаріїв використання. На наступному етапі NUI почне виходити на масового споживача, і відвойовувати сектори які зараз зайняті GUI.

**Метою бакалаврські роботи є** дослідження проблеми візуального керування графічними об’єктами для інтерфейсів загального користування на основі кінетичних інтерфейсів і відкриває можливість вивчення поведінки користувачів, розробки алгоритмів NUI на прикладі керування інтерфейсом загального користування, адоптація і поєднання різних типів керування, вивчення реакції користувачів, розробка цілісного інтерактивного дизайну інсталяції.

**Методологічно робота використовує** широкий спектр загальноприйнятих методів, а саме: методи порівняння, вимірювання, абстрагування, аналізу і синтезу, індукції та дедукції, системний підхід. Для вивчення поведінки користувачів, перевірки гіпотез, а також для перевірки фінального рішення було сформовано групу користувачів, які в межах серії сесій дали великий вклад в розробку проекту і надали великий об‘єм робочих даних для подальшої роботи.

**Науково-технічою новизною** роботи є розробка рішення повного циклу керування графічними об’єктами в інтерфейсах загального використання використовуючи NUI: від заохочення користувача, до аналізу результатів. Це надає можливості до подальшого розширення аналізу поведінки користувачів.

**Практичне значення отриманих результатів** полягає в тому що вони можуть використовуватись для прийняття рішень та розробці інтерактивних дизайнів майбутніх інтерфейсів, а також надає платформу для експериментів в інших областях NUI таких як керування голосом, керування за допомогою фізичних предметів, тощо.

**Структура та зміст роботи.** Бакалаврська робота складається зі вступу, розділів змістовної частини, висновків та списку використаних джерел. Змістовна частина розділена на чотири частини.

Перший розділ надає аналіз поведінки користувача, розробку гіпотез і їхній аналіз.

Другий розділ описує вимоги до рішення, аналізує і обирає технічні рішення, розробляються конкретні алгоритми, тощо.

Третій розділ описує фінальне рішення, його структуру і організацію. Комунікація між компонентами, інтерфейс користувача і адміністратора. Концепція фізичної інсталяції.

Четвертий розділ надає економічне обґрунтування рішення.

В роботі зроблені загальні висновки до бакалаврської роботи.

Повний обсяг дипломної роботи - \_\_\_ сторінок, у тому числі робота містить \_\_\_ таблиць, \_\_\_\_ ілюстрацій, список використаних джерел зі \_\_\_ найменувань.

# Аналіз

Цей розділ описує гіпотези як результат теоретичного аналізу поведінки користувача для рішення проблеми керування інтерфейсами загального користування використовуючи NUI підходи.

## Аналіз поведінки користувача

Для реалізації NUI було проаналізовано досвід користувача і розроблено зв’язану систему рухів що дозволяє інтуїтивно виконати задачі.

Для аналізу рухів були враховані такі фактори

Інтуїтивність – рух має бути інтуїтивним, очікуваним, має мати мінімальне навчання. Якщо можливо повторювати рухи в звичайному оточенні.

Точність спрацьовування – не всі рухи можна виділити на фоні інших рухів.

Відсутність конфліктів – багато рухів повністю чи частково конфліктують між собою. Важливо отримати схему рухів без конфліктів.

Точність виявлення рухів і інтуїтивність було перевірено на сторонніх користувачах включаючи реальну інсталяцію.

### Відповідь інтерфейсу на рух користувача

Для користувача перше знайомство з NUI це один з найважливіших етапів. Користувач не бачить і не чує NUI, на початку для користувача системи не існує взагалі. Для того щоб вирішити цю проблему було вирішено використовувати декілька варіантів заохочення.

Коли потенційний користувач опиняється в зоні детектування давачем, загальний план проектів починає змінювати перспективу базуючись на положенні користувача. Гіпотеза полягає в тому що це зацікавить користувача.

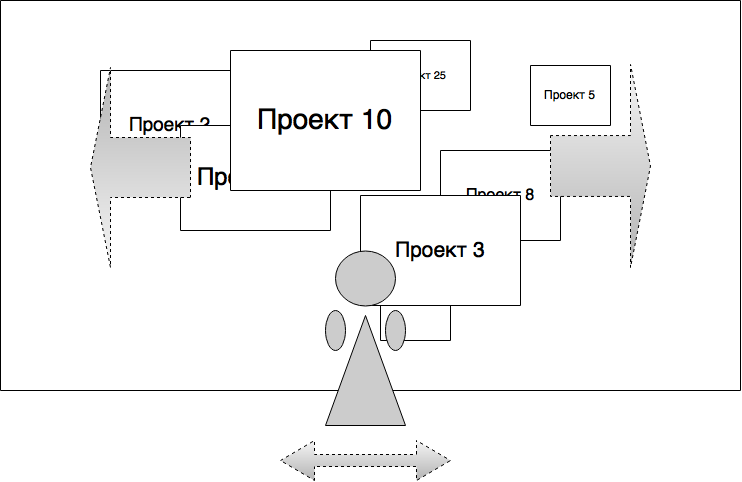


Рисунок 1.1 Відповідь інтферфейсу на рух користувача

Інтуїтивність – висока, від користувача не потребується жодних знань.

Конфлікти – відсутні для користувача, можуть бути конфлікти між користувачами

### Зміна загального плану на режим перегляду проекту

На загальному плані проектів користувач може підійти ближче до екрану. Система переключиться в режим перегляду проекту. Якщо користувач відійде, то переключиться на загальний план.

Інтуїтивність – середня, практика показала що потрібно додаткова мотивація підійти.

Конфлікти – відсутні.

### Жест горизонтального «перегортання» проектів

На екрані перегляду проекту користувач може «перегортати» сторінки з проектами. Різкий рух правої руки з права на ліво, означає перегортання вліво, і навпаки.

Інтуїтивність – висока, деякі люди користувачі пробували використовувати жест «перетягування»

Конфлікти – повернення руки конфліктує з рухом в інший бік, також конфліктує з іншими жестами

### Відповідь на рух користувача в режимі перегляду проекту

Для заохочення користувача спробувати скористатись рухом «перегортання» система реалізовує відповідь на рух в режимі перегляду проекту. Коли активний користувач змінює горизонтальну позицію тіла, череда проектів зміщується, але сторінка не перегортається. Це дає відчуття того що проекти можуть рухатись як карусель.

Інтуїтивність – висока, від користувача не потребується жодних знань.

Конфлікти – відсутні для користувача, можуть бути конфлікти між користувачами

### Жест протягування проектів для зміни проекту

Користувач може «взяти» проект і протягувати його для того щоб перейти на наступний проект.

Інтуїтивність – достатньо висока, але як показала практика рух «перегортання» більш інтуїтивний

Точність – висока, реалізується за допомогою інтеракцій Kinect SDK.

Конфлікти – відсутні.

### Жест вертикального «перегортання» сторінок проекту для перегляду

Різкий рух правої руки вгору перегортає сторінки вниз, різкий рух вниз – відновлює попередній стан.

Інтуїтивність – висока, базується на досвіді користувача з горизонтальним «перегортанням».

Точність – мала, особливість руху потребує дискретних зупинок, для горизонтального перегортання це проекти, для вертикального екрани, але це часто не дозволяє подивитись потрібну частину проекту.

Конфлікти – багато, як з вертикальним «перегортанням», так і з іншими жестами.

### Жест вертикального перетягування проекту для перегляду проекту

Користувач може «взяти» проект і протягувати його вгору чи вниз для того щоб переглянути потрібні деталі проекту.

Інтуїтивність – достатньо висока, іноді потребує навчання за рахунок відмінності від горизонтального руху.

Точність – висока, реалізується за допомогою інтеракцій Kinect SDK.

Конфлікти – відсутні.

### Жест оцінки проекту – великий палець вгору

Жест витягнутої правої руки і верхній палець піднятий догори.

Інтуїтивність – висока, група користувачів знаходила цей жест самостійно.

Точність – низька, роздільна здатність давача не давала можливість точно отримати позицію великого пальця.

Конфлікти – багато, конфліктує з «перегортанням»

### Жест оцінки проекту у вигляді зірки

Для оцінки проекту користувач мав мати позу у вигляді зірки.

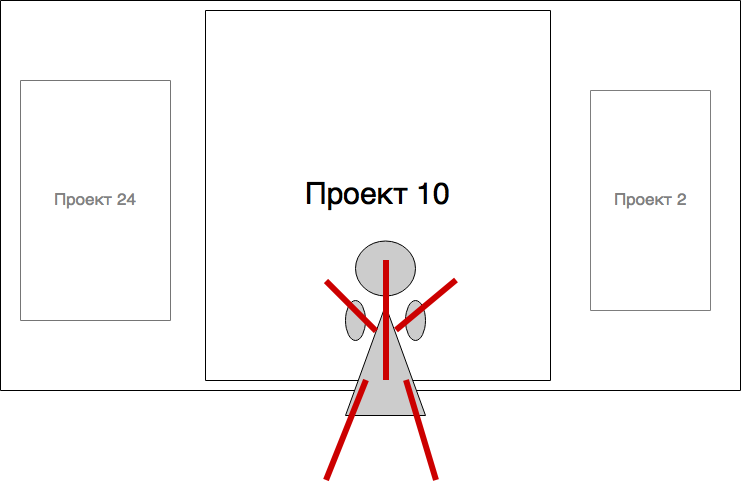


Рисунок 1.2 Жест "зірка"

Інтуїтивність – мала, практика показала що потрібно було допомагати, також складність викликали ноги.

Точність – достатньо висока.

Конфлікти – відстуні.

### Жест оцінки проекту – радість

Варіант оцінки проекту в якому використовуються тільки данні про руки.

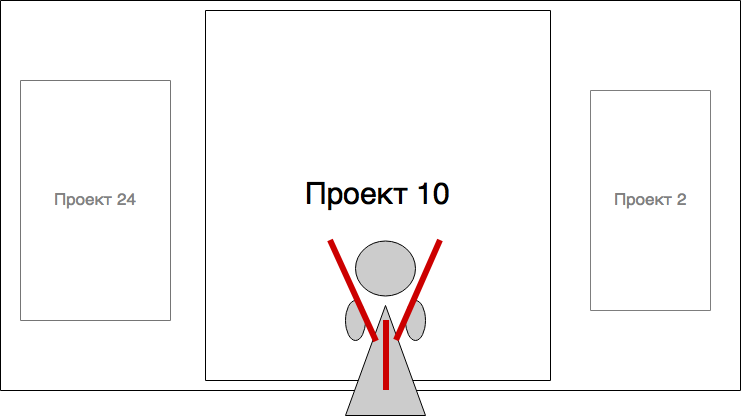


Рисунок 1.3 Жест оцінки проекту радість

Інтуїтивність – висока, багато користувачів самі його знаходили. За рахунок помітності руху, це додатково заохочувало інших користувачів.

Точність – висока.

Конфлікти - відсутні

### Жест зміни активного користувача

Жест схрещених рук сприймався системою як зміна активного користувача.

Інтуїтивність – не висока.

Точність – низка.

Конфлікти – оцінка проекту, «перегортання».

## Висновок

В цьому розділі було проаналізовано поведінку користувача в реальному оточені і розроблено гіпотези про інтуїтивність, точність і наявність конфліктів окремих частин інтерактивного дизайну для рішення. Також надані теоретичні обґрунтування деяких гіпотез.

Наступні розділи проробляють технічне рішення і платформу для перевірки гіпотез описаних в цьому розділі.

# Проектування

Даний розділ описує етап проектування системи, описується проблематика рішення, наводиться аналіз обраних технічних рішень і формується загальний план рішення.

## Основні актори та use-case системи

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

Таблиця 3.1 Основні актори системи

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

Користувач використовує NUI для своїх use-case, натомість адміністратор використовує веб інтерфейс. В 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, переглянути його, і оцінити його.

Адміністратор може подивитись звітність по оцінках і використанню.

На діаграмі нижче показані основні use-case користувача і адміністратора.



Рисунок 2.1 Основні use-case системи

## Інтерактивний дизайн користувача

Для користувача було розроблено інтерактивний дизайн який одним логічним шляхом покриває весь процес огляду і оцінки проектів. Головними критеріями розробки інтерактивного дизайну є:

Відсутність тупіків – шлях користувача має мати логічні входи і виходи. З кожного стану мають бути переходи на інші стани або на початок.

Інтуїтивність – шлях було розроблено базуючись на перевірених гіпотезах про інтуїтивність тих чи інших рухів.

Мінімізація помилок – для станів які можуть мати помилки (наприклад оцінка проекту) було додано додаткові захисні перевірки.

Загальний шлях користувача показано на діаграмі інтеракцій нижче.

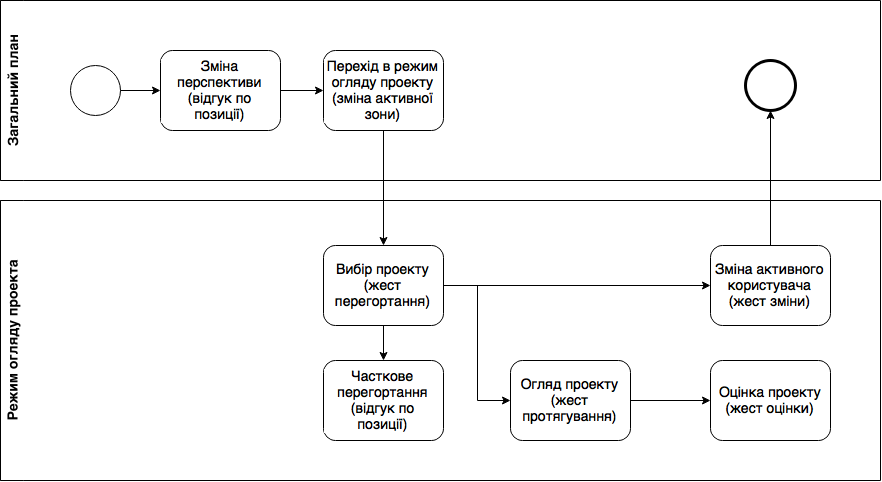


Рисунок 2.2 Схема інтеракцій користувача

Робота користувача системи починається з загального плану, який має додаткові заходи по заохоченню користувача. Після цього загальний план проектів змінюється на режим вибору та оцінки проекта.

## Компоненти системи

Рішення виконане у вигляді поєднання програмного рішення і апаратних давачів. 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.

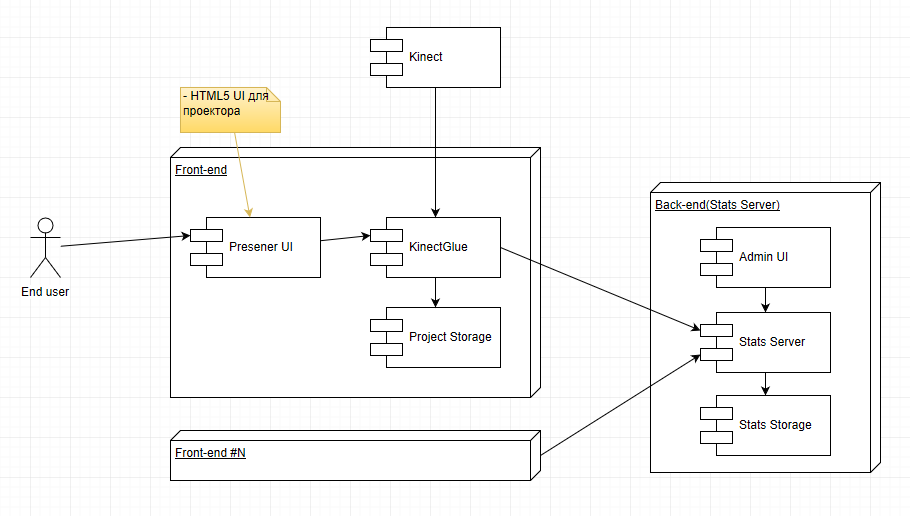


Рисунок 2.3 Компонентна діаграма рішення

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, оркестровці 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

## Аналіз давачів жестів для NUI

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

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

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

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

Спеціальне обладнання – давач може потребувати додаткового обладнання

SDK – наявність бібліотек для розробки під потрібну платформу

Ціна – ціна давача для інсталяції і розробки

Були розглянуті такі давачі:

Leap Motion – давач 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 давачів для Xbox.

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

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

Таблиця 2.1Порівнянльна таблиця давачів жестів

|  | **Leap Motion** | **Microsoft Kinect 1.0** | **Microsoft Kinect 2.0** | **PlayStation Move** |
| --- | --- | --- | --- | --- |
| **Опис** | Давач жестів на короткій дистанції | Ігровий давач | Ігровий та промисловий давач | Ігровий давач |
| **Радіус дії давача(м)** | 0.3-1 | 2-6 | 0.3-6 | 2-6 |
| **Жести тіла** | Ні | Так | Так | Ні |
| **Кількість об’єктів стеження** | 1 | 2 | 5 | 1 |
| **SDK** | .NET | .NET | .NET | Ні |
| **Ціна** | 2000 грн | 2600 грн | 5000 грн | - |

Базуючись на потрібних характеристиках було обрано давач Microsoft Kinect 2.0 для PC.

## Бібліотеки виявлення жестів

Обробка даних від давача потребує розробки алгоритмів оцінки глибини точки, пошуку активних зон, виділення користувачів, зв’язок точок тіла, тощо. Головні фактори які є важливими для вибору бібліотек виявлення жестів такі:

Підтримка Kinect – попередній аналіз дав можливість вибрати Microsoft Kinect 2.0

Швидкість розпізнавання – швидкість виявлення шаблонів, категоризації, тощо

Відкритість коду –важливо розуміти внутрішні деталі реалізації, відкритість коду полегшує цю задачу

Таблиця 2.2 Бібліотеки розпізнавання жестів

| Вимога | OpenNI | OpenKinect | Kinect SDK 2.0 |
| --- | --- | --- | --- |
| Підтримка Kinect | Так | Так | Так |
| Швидкість розпізнавання | Середня | Середня | Швидка |
| Відкритість код | Так | Застарілий | Ні |

Аналіз показав відсутність ідеального рішення. Було обрано використання Kinect SDK 2.0 як рішення рекомендоване Microsoft. Додатковим бонусом є додатковий функціонал як розпізнавання інтеракцій та рішення дл запису і аналізу жестів.

## 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 2.3 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 використовувати WPF, і замінити його на HTML/JS, в якості базової бібліотеки інтерфейсу було обрано AngualrJS.

## Комунікація реального режиму часу між інтерфейсом користувача і обробником Kinect

Один з викликів цього рішення це комунікація реального часу між інтерфейсом користувача і сервером інсталяції. В режимі звичайної роботи, передаються такі типи даних:

Таблиця 2.4 Типи комунікації між інтерфейсом користувача і сервером інсталяції

|  | Напрямок | Об’єм даних | Опис |
| --- | --- | --- | --- |
| Виявлені жести | UI | Малий | Передається ключ виявленого жесту, і невеликий об’єм мета інформації. |
| Інтеракції | UI | Середній | Постійний потік даних з координатами важливих точок в просторі, положенням рук, тощо. |
| Відео | UI | Великий | Потік відео фреймів для тестових сценарієв |
| Команди | Сервер | Малий | Передається команда до серверу, така як оновити оцінки. |

Основні фактори які були прийняті до уваги:

Функціональність – обрана технологія має мати можливість реалізувати функціонал

Дуплексний режим – можливість роботи як від сервера до клієнта, так і навпаки

Швидкість – швидкість передачі даних, затримка відповіді, тощо

Для вибору було розглянуто три варіанти. HTTP – звичайні запити на сервер. HTTP Poll – довгі запити на сервер, які дозволяють передавати данні на клієнт без очікування з’єднання. WebSockets – відносно новий стандарт який привносить в веб двонаправлені сокети для комунікації з сервером.

Таблиця 2.5 Аналіз варіантів комунікації між інтерфейсом користувача і сервером

|  | HTTP | HTTP Poll | WebSockets |
| --- | --- | --- | --- |
| Функціональність | Ні | Так, але потрібно вирішувати проблеми довгих запитів | Так, потребує функціонал пере підключення до сервера. |
| Дуплексний режим | Ні | Емулюється | Так |
| Швидкість | Висока, але не реального часу | Висока, але не реального часу | Дуже висока, за рахунок використання бінарних даних |
| Відео | Так, але потребує стрімінг | Так, але потребує стрімінг | Так |

Аналіз показав, що найкращим рішенням є WebScokets які мають швидкість достатню навіть для пофреймового передавання відео, мають мінімальну затримку за рахунок постійно відкритого з’єднання, а також доступні в усіх сучасних браузерах.

## Комунікація в режимі підписки між сервером інсталяції і сервером статистики

Комунікація між сервером інсталяцій та сервером статитстики складається з обробки факту оцінки, відправки його на сервер статистики і потім розсилки на інші сервевери інсталяцій.

Для вибору технології комунікації було прийнято до уваги такі фактори:

Швидкість розсилки – інтерактивна природа інтеракцій потребує миттєвого відклику на всіх інсталяціях

Можливість Pub/Sub – для спрощення комунікації сервер інсталяції «підписується» на зміни статистики.

Простота розгортання – для мінімізації витрат на розгортання, розгортання має бути відсутнім

Таблиця 2.6 Аналіз технології комунікації між сервером інсталяції і статистики

|  |  |  |  |
| --- | --- | --- | --- |
|  | **RabbitMQ** | **TCP/HTTP** | **ZeroMQ** |
| Швидкість розсилки | Висока | Дуже висока | Дуже висока |
| Pub/Sub | Так | Ні | Так |
| Розгортання | Так | Ні | Ні |
| Підтримка .NET | Так | Так | Так |

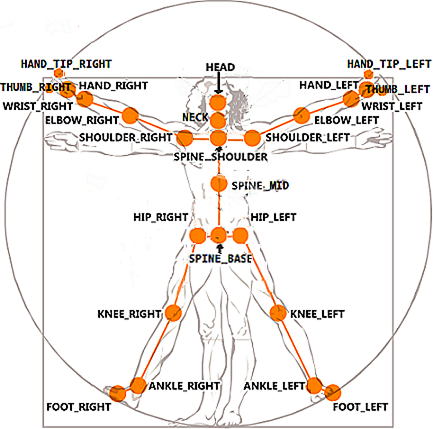
Аналіз показав переваги ZeroMQ, черга і пам’яті з можливість підписки. ZeroMQ не потребує додаткового розгортання і є дуже швидким.

## 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. Давач може стежити за 6 користувачами, користувачі можуть відходити і підходити. Також користувачі можуть «передавати» активність іншому користувачу.

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

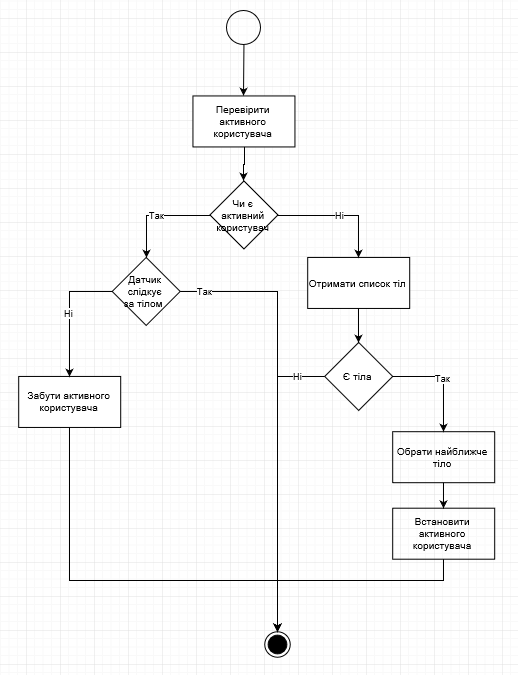


Рисунок 2.4 Алгоритм пошуку активного користувача

### 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.

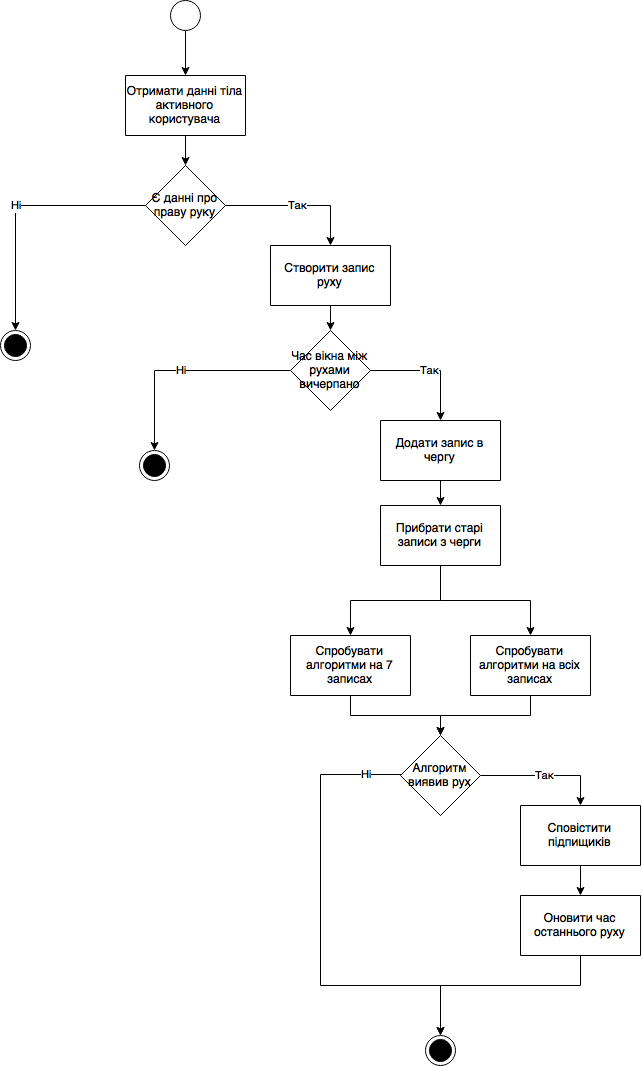


Рисунок 2.5 Загальна схема виконання я евристичних алгоритмів

### Виявлення руху «перегортання»

Для вибору конкретного проекту використовується рух «перегортання». Це горизонтальний рух з права на ліво – для перегортання 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.

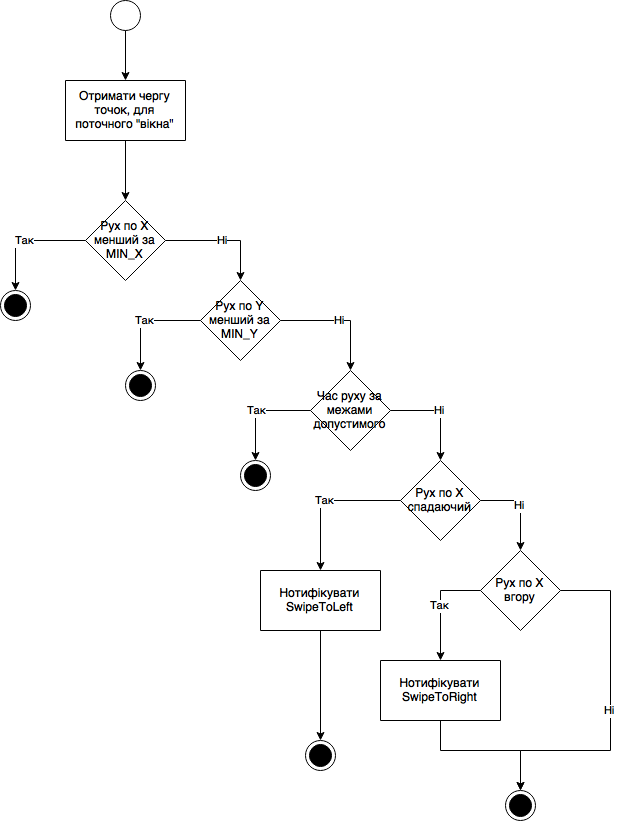


Рисунок 2.6 Алгоритм розпізнавання горизонтального руху «перегортання»

## Концепція фізичної інсталяції

Фінальне рішення складається з сенсора Kinect під’єднаного до сервера інсталяції, на якому йде обробка рухів користувачів, та інтерфейсу що реагує на рухи. За допомогою проектора інтерфейс користувача продукується на стіну або на великий екран. Приблизне розташування елементів інсталяції показано на діаграмі нижче.

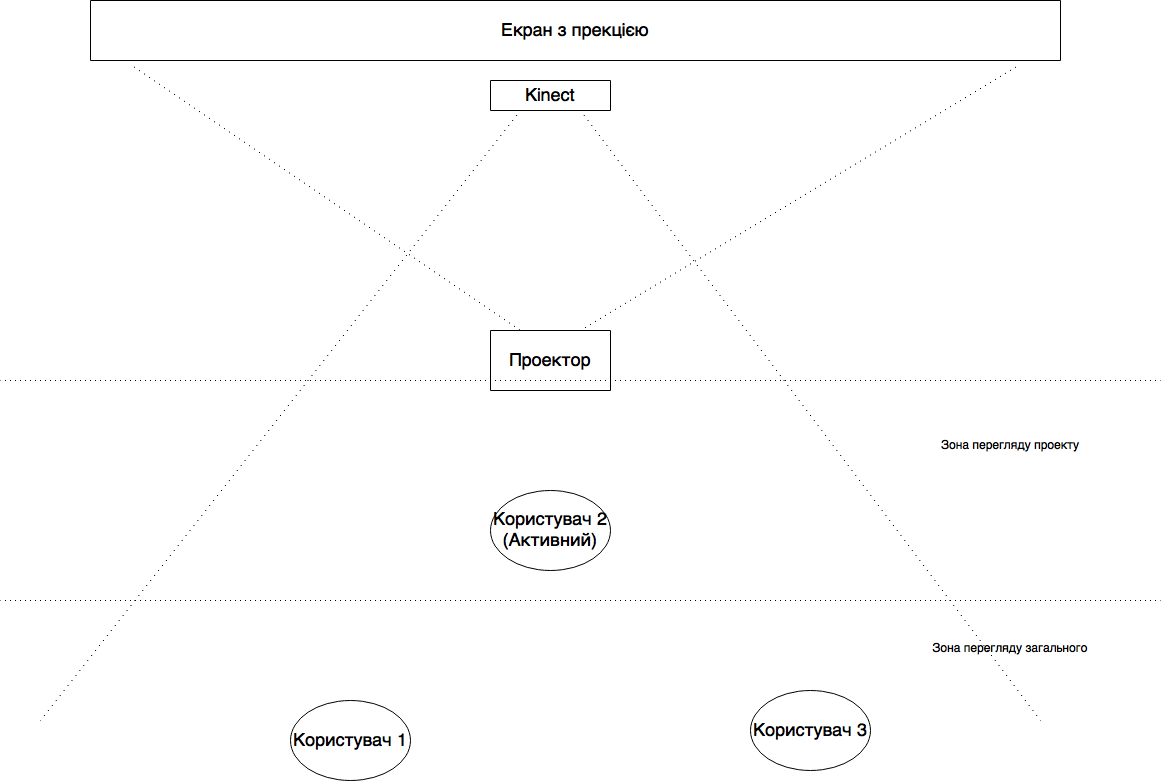


Рисунок 2.7Фізичне розташування елементів інсталяції

Найкращі результати дає робота з великою проекцією, відчуття було протестоване на екрані 3 на 4 метри. Давач руху Kinect має бути розташований на висоті приблизно 30-50см від поверхні підлоги.

## Висновок

В цьому розділі було проаналізовано варіанти технічних опцій для фінального рішення. Було обрано оптимальні опції що вдовольняють вимоги рішення.

У якості давача жестів було обрано Microsoft Kinect 2.0.

Бібліотека виявлення жестів - Kinect SDK 2.0

Платформа для інтерфейсу користувача – JavaScript/HTML5

Комунікація реального режиму часу – HTML5 WebSockets

Комунікація в режимі підписки - ZeroMQ

Було проаналізовано і розроблено алгоритми розпізнавання окремих жестів.

Було розроблено концепцію інсталяції.

# Реалізація

## Взаємодія між компонентами

Ця секція описує комунікаційні протоколи між компонентами системи.

Комунікація між різними компонентами має різні вимоги, тому для них були використані різні технлогії.

Для комунікації між інтерфейсом користувача і сервером інсталяції використовується двонаправлені WebSocket. Для комунікації між сервером інсталяції і сервером статистики використовується схема Pub/Sub реалізована задопомогою in-memory шини ZeroMQ.

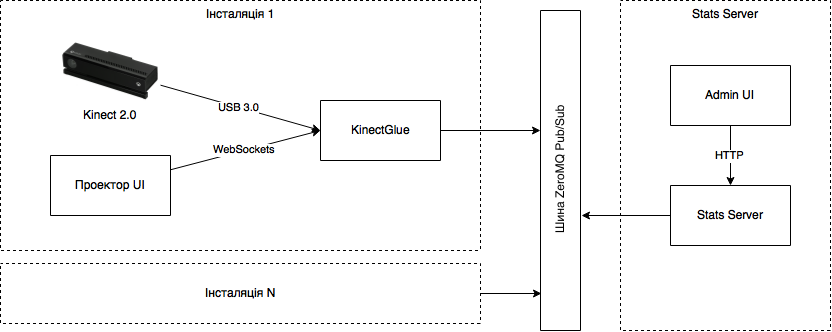


Рисунок 3.1Протоколи і взаємодія між компонентами

## 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, базуючись на бібліотеках з відкритим кодом.

Таблиця 3.2 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 |

Інтерфейс побудовано використовуючи шаблон проектування MVVM який дозволяє розділити логіку інтерфейсу і його представлення використовуючи динамічний зв’язок даних.

### 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 3.3 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 |

Внутрішня комунікація системи побудована на приннципах реактивного програмування, де будь яка зміна в стані описується подією на яку підписані інші частини системи. Це дозволяє будувати не блокуючі багато потокові системи.

### Технічний стек Admin UI

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

Таблиця 3.4 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 |

Для організації інтерфейса адміністратора використовується шаблон проектування Flux – розділення між представленням та даними в одному напрямку по колу.

### 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 3.5 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 |

## Внутрішня організація системи

### Реактивна комунікація між компонентами серверу інсталяції (KinectGlue)

Система використовує шаблони реактивного програмування. Це обумовлено вимогами швидкої обробки даних, мінімальною затримкою (адже кожна затримка буде відбуватися на досвіді користувача). Вся система використовує неблокуючі реалізації.

Для обробки потоку подій в реактивному вигляді використовується .NET Reactive Extensions.

Схема потоку повідомлень серверу інсталяцій показана на діаграмі нижче.

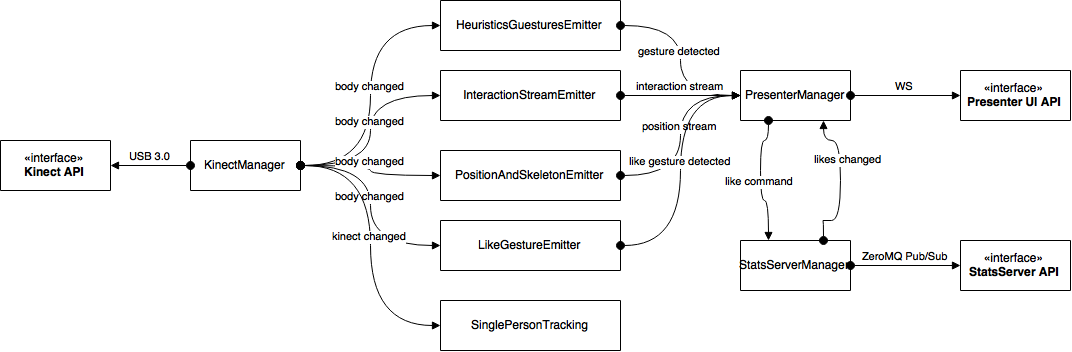


Рисунок 3.2 Схема потоку повідомлень сервера інсталяції

Найбільша частина повідомлень починає свій шлях від Kinect, зазвичай це інформація про положення скелету. Після обробки цієї інформації генеруються події з жестами, інтеракціями, тощо. Ці події спрямовуються на інтерфейс користувача чи сервер статистики. Між сервером статистики та інтерфейсом користувача також є обмін повідомленнями.

### Організація реалізації емітерів, виявлення рухів

Одна з головних частин системи це обробка потоку даних з Kinect, виявлення жестів, рухів. Обробка потоку виконана у вигляді емітерів, об’єктів які на вхід приймають потік фреймів, або поточний фрейм і базуючись на внутрішніх алгоритмах видають чи не видають результат.

Система емітерів показана на діаграмі нижче.

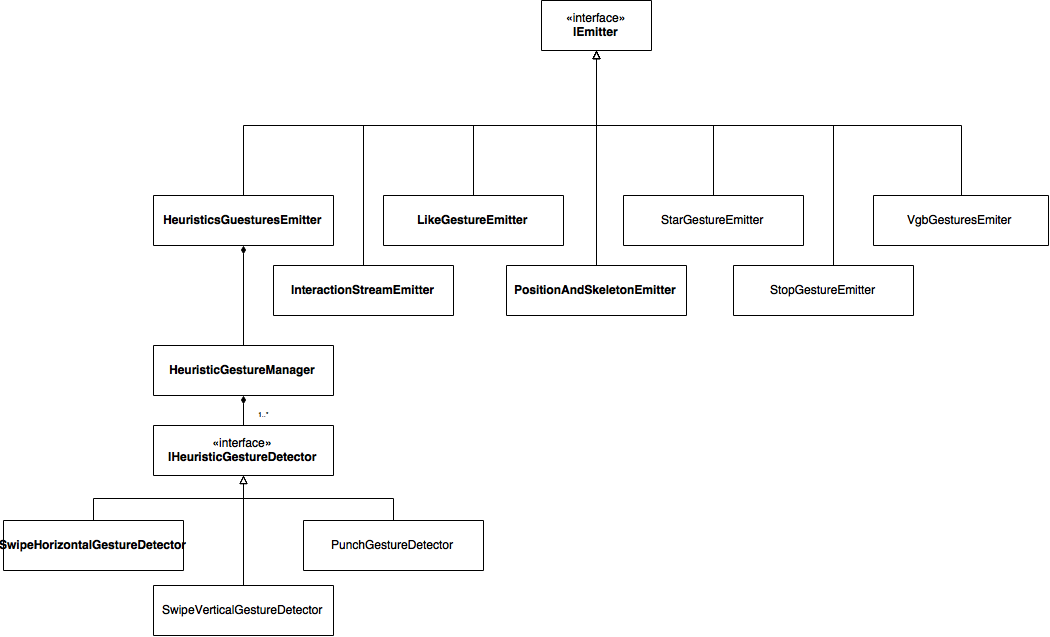


Рисунок 3.3Схема емітерів для обробки потоку даних

Всі емітери реалізовують маркуючий інтерфейс IEmitter. За рахунок відсутності загального контракту IEmitter не нав’язує свій контракт, а отже сторона що використовує конкретний емітер має використовувати конкретну реалізацію емітера.

HeuristicsGuesturesEmiter – емітер для евристичного виявлення жестів. Реалізований у вигляді набору детекорів.

SwipeHorizontalGestureDetector – виявляє горизонтальний жест перегортання сторінки. Використовується в фінальному рішенні.

SwipeVerticalGestureDetector – виявляє вертикальний жест перегортання сторінки. Не використовується в фінальному рішенні, практика показала гіршу інтуїтивність ніж жест захвату і протягування екрана.

PunchGestureDetector – виявляє жест схожий на удар в бік презентації, гіпотеза про те що це може використовуватись для оцінки проекту. Не використовується в фінальному рішенні, практика показала не інтуїтивність цього жесту, а також конфлікти з іншими жестами.

InteractionStreamEmitter – емітер стірму інтеракції. Фактично потік позицій правої руки і її стан. Використовується в фінальному рішенні для реалізація протягування презентації для перегляду.

LikeGestureEmitter – емітер жесту оцінки – обидві руки над головою. Використовується в фінальній реалізації.

PositionAndSkeletonEmitter – емітер позиції та скелетона. Використовується в фінальній реалізації для відстеження положення користувача, а також на тестових екранах.

StarGestureEmitter – другий варіант емітеру для жеста оцінки. В цьому емітері також використовується інформація про положення ніг. Не використовується в фінальному рішенні, практика показала не інтуїтивність.

StopGestureEmitter – жест складених навхрест рук який мав використовуватись для відміни «активності» користувача, і автоматичної передачі іншому. Не використовується в фінальному рішенні, конфліктує з іншими жестами.

VgbGesturesEmiter – емітер жестів що будується на бібліотеці машинного навчання що входить в поставку Kinect SDK. Не використовується в фінальному рішенні, практика показала ненадійність виявлення жестів «перегортання».

### Організація реалізації інтерфейсу користувача

Інтерфейс користувача організовано за допомогою бібліотеки AngularJS з використанням шаблону MVVM. Головні об’єкти апплікації World – загальний план проектів та View – індивідуальний план проекту. Модулі World та View розроблені використовуючи шаблон MVVM де кожен з них має контролер та презентацію у вигляді шаблону.

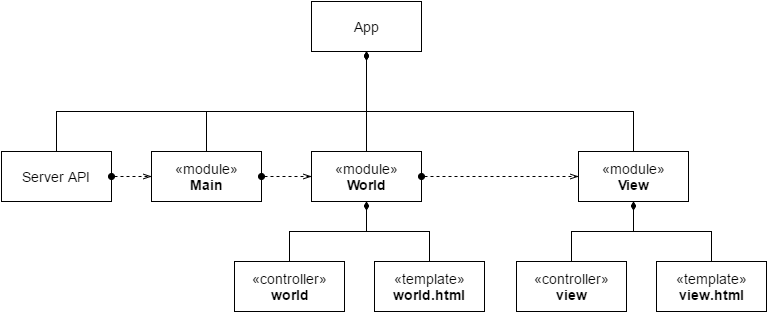


Рисунок 3.4 Організація реалізації інтерфейсу користувача

Вхідна точка додатку це об’єкт App. Додаток це композиція модуля Main який керує даними верхнього рівня як список проектів, синхронізація з сервером та внутрішнім сховищем, утілітних модулів, а також модулів інтерфейсу як World та View.

Комунікація між модулями побудована на основі стану модуля (AngualrJS Scope), це дозволяє декларативно описувати інтерфейс.

Додатково до модулів які використовуються для звичайному режимі, в апплікації є модулі розроблені для потреб тестування та налагодження. Вони доступні зі сторінки dev.html.

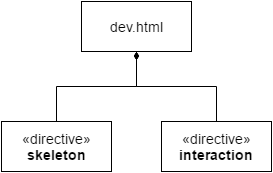


Рисунок 3.5 Організація модулів підтримки розробки

Модуль skeleton – візуалізує активні точки скелета активного користувача і накладає його на загальний відео ряд. Це дозволяє візуально зрозуміти поведінку скелету.

Модуль interaction – візуалізує потік інтеракцій для активного користувача – позиції важливих точок, їхній стан, тощо.

## Результати роботи системи

### Інтерфейс користувача

Головний екран інтерфейсу користувача, показує проекти розміщуючи їх в тривимірному просторі, проекти з найбільшою кількістю оцінок – ближче, з меншою – далі, а також більш розмитої.

Для заохочення користувача в залежності від позиції відносно давача руху, екран зміню точку зору, створюючи ефект паралакса.

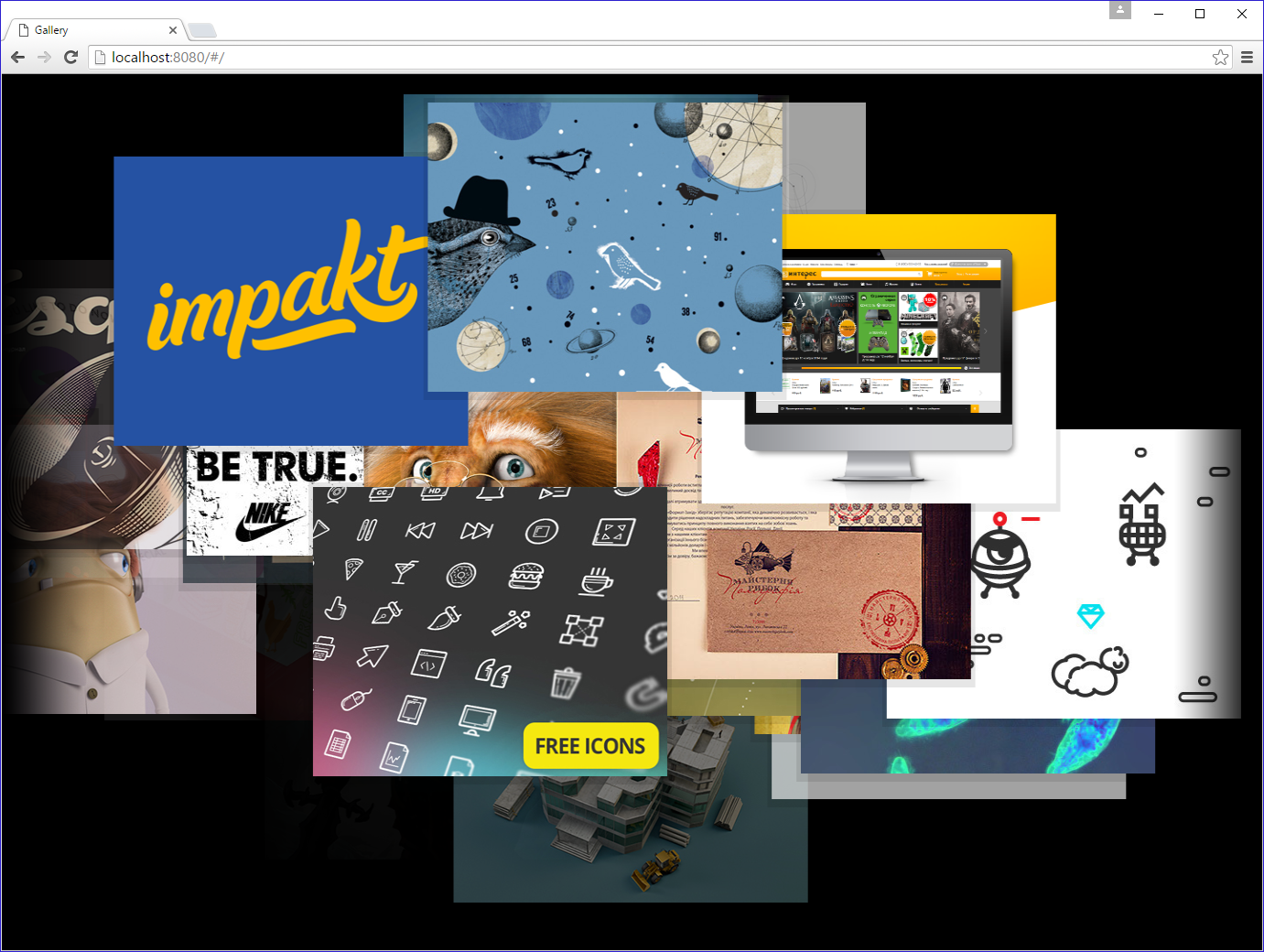


Рисунок 3.6 Екран загального огляду проектів

Після того як користувач підходить ближче до давача (виявлено що користувач зацікавився) автоматично змінюється на показ проектів. В цьому режимі користувач може «прогортати» вправо чи вліво для вибору проекту. Може «потягнути» зображення вверх чи вниз для того шо б переглянути проект.

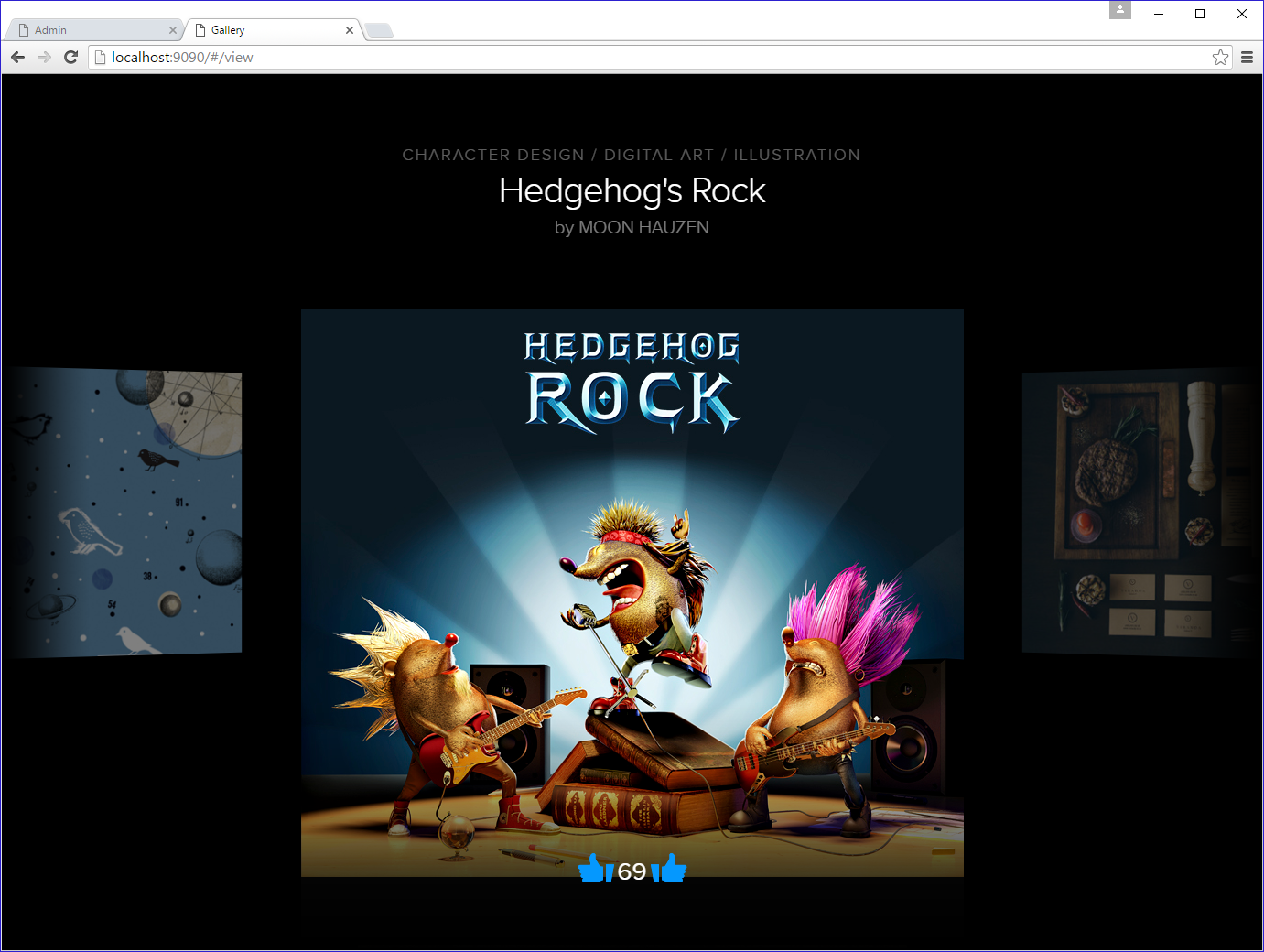


Рисунок 3.7Екран огляду проекта

Користувач має можливість оцінити проект. Для цього потрібно щоб давач виявив жест «зірочка». Оцінка автоматично оновиться на всіх інсталяціях.

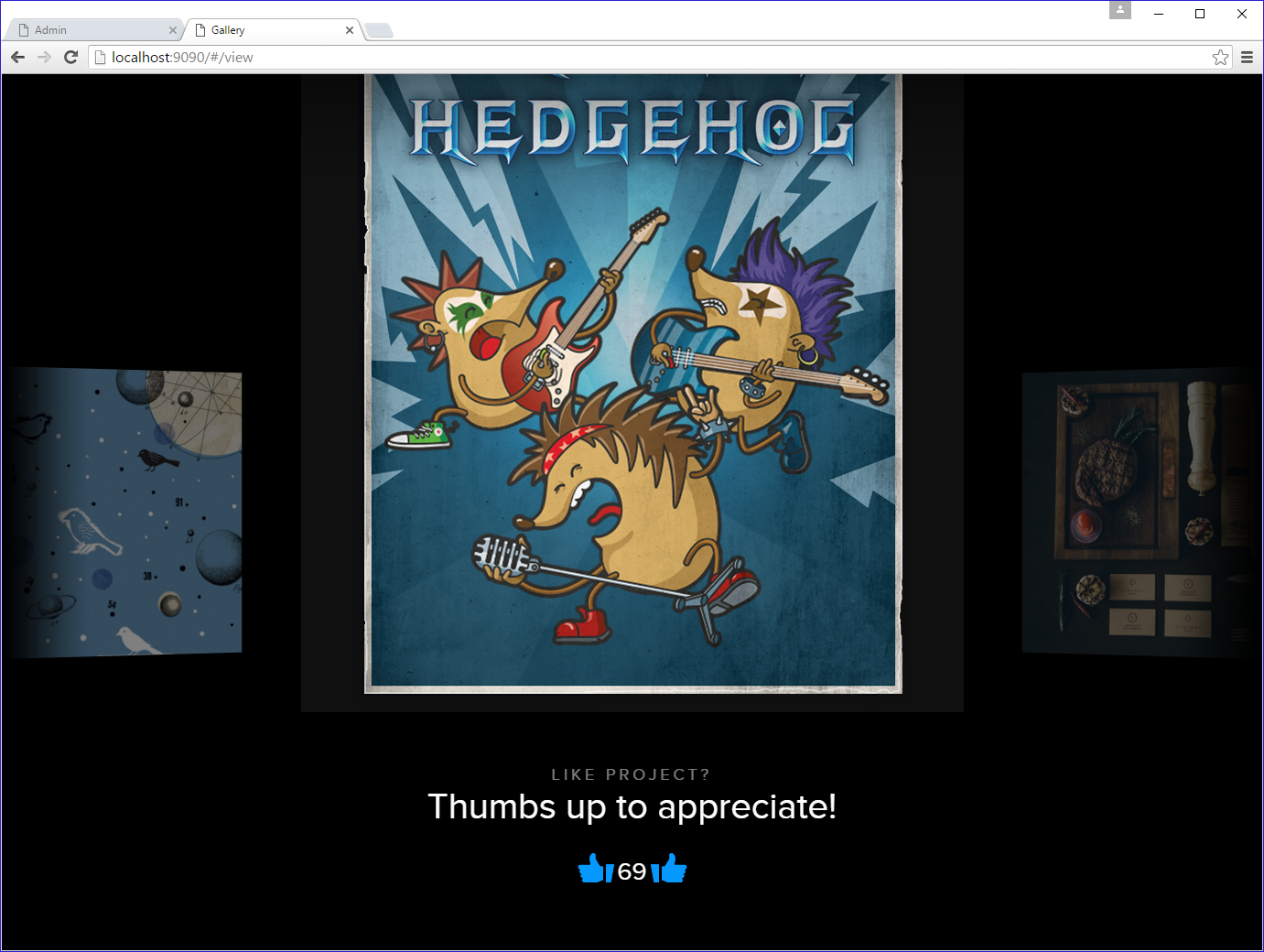


Рисунок 3.8 Екран оцінки проекту

Інтерфейс користувача дуже активно використовує анімацію для заохочення користувачів. В межах документа немає можливості показати анімацію.

### Інтерфейс адміністратора

На відміну від інтерфейсу користувача, адміністративний інтерфейс розрахований тільки для внутрішнього користування . Адміністративний інтерфейс надає можливість звітності.

Нижче наведено екран звітності по відгуках по проектах.

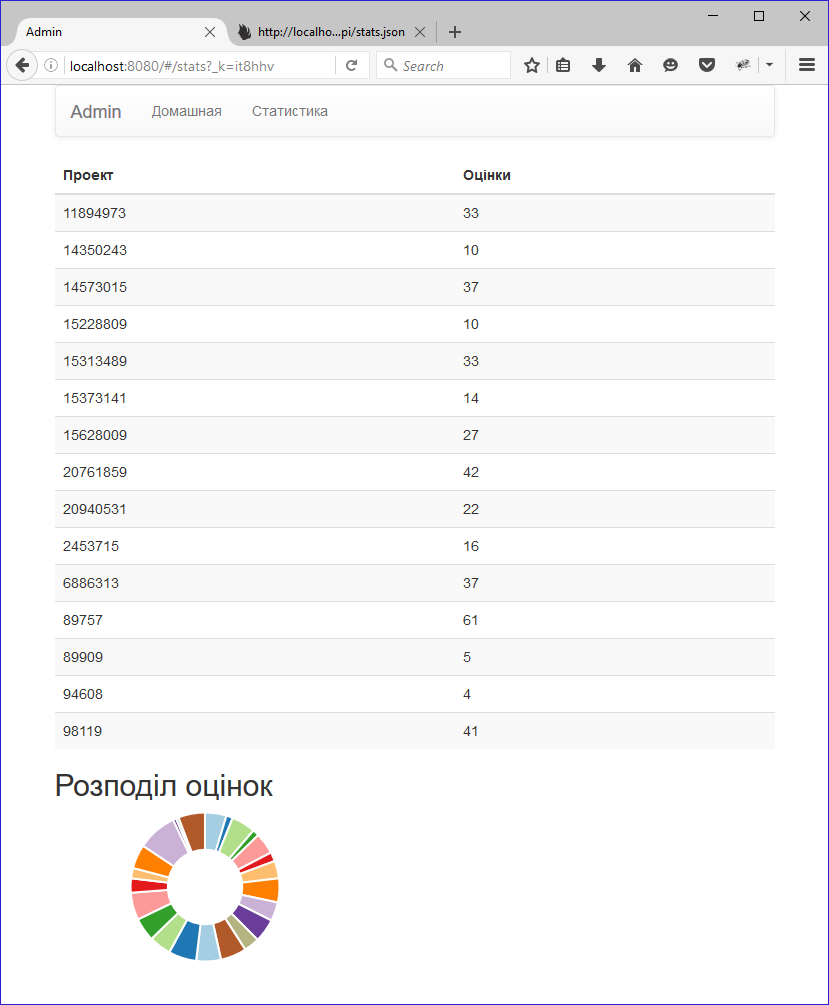


Рисунок 3.9 Екран звіту про оцінки проектів

На екрані можна побачити таблицю з оцінками проектів, так також графік розподілу оцінок між проектами.

## Висновок

Для підтвердження гіпотез розроблених в рамках цієї роботи було розроблено систему яка надала можливість перевірити її на реальних користувачах. Система складається з двох основних компонент, інсталяції і серверу статистики. Інсталяція включає в себе комплекс програмного забезпечення, давача жестів, сервера і проектора.

# Економічна частина

## **Економічна характеристика проектного рішення (програмного продукту)**

Метою виконання бакалаврської кваліфікаційної роботи є розроблення системи для керування відображенням інформації дизайн портфоліо за допомогою природніх жестів. Система розрахована на інсталяцію в приміщеннях проведення дизайнерських подій або аналогічних.

В даній економічній частині наведена економічна оцінка доцільності розроблення програмного продукту з точки зору його окупності.

## **Інформаційне забезпечення та формування гіпотези щодо потреби розроблення проектного рішення (програмного продукту)**

Для організації подій в креативному середовищі, дуже важливо проробляти не тільки контенте наповнення події, а також створювати простір для обміну ідеями, комунікації, розповсюдження інформації, тощо. На сучасних подіях використовується багато варіантів для залучення учасників. Найбільш розповсюдженні, це: перерви на каву, короткі промови (light talks), ігрові\змагальні зони, інформаційні зони, тощо.

Одна з особливостей креативного середовища, це наявність у учасників портфоліо фінальних робіт. Зазвичай для керування портфоліо використовуються сайт компанії, або сторінка в каталогах компанії. В останній час розвиток почали отримувати глобальні сайти для керування портфоліо. Один з найпотужніших таких сайтів є Behance Portfolio Reviews. Behance Portfolio Reviews виступає як єдина платформа, яка дозволяє публікувати портфоліо, описувати деталі креативного процесу, отримувати фідбек, тощо.

Поєднавши два фактори, потребу в додатковому залученні учасників за допомогою покращення комунікації, а також наявності сайтів для обміну портфоліо, було вирішено створити систему залучення учасників у вигляді інсталяції для огляду вибраних проектів з Behance Portfolio Reviews яка використовує природні жести для керування навігацією.

## Оцінювання та аналізування факторів зовнішнього та внутрішнього середовищ

Фактори зовнішні оцінюються за шкалою [-5;5], при цьому межі шкали відображають максимальний негативний та позитивний вплив факторів на організацію, 0 демонструє, що фактор впливає на організацію нейтрально.

Фактори внутрішні оцінюються за шкалою [0;5], при цьому 0 демонструє нерозвинутість, відсутність чи катастрофічний стан фактора внутрішнього середовища, оцінка 5 демонструє високий рівень розвитку даного фактор.

Сума вагомостей усіх факторів становить одиницю, тобто рівень вагомості для кожного фактора визначається за допомогою коефіцієнтів. Зважений рівень впливу факторів розраховується як добуток впливу фактора у балах та рівня вагомості.

Результати експертних оцінок впливу факторів зовнішнього середовища наведено у Таблиця 4.1, внутрішні у Таблиця 4.2

Таблиця 4.1. Результати експертних оцінок впливу факторів зовнішнього середовища

|  |  |  |  |
| --- | --- | --- | --- |
| **Фактори** | **Середня експертна оцінка, бали** | **Середня вагомість факторів** | **Зважений рівень впливу, бали** |
| Споживачі | 5 | 0.11 | 0.55 |
| Постачальники | 0 | 0.1 | 0 |
| Конкуренти | -2 | 0.1 | -0.2 |
| Державні органи влади | 0 | 0.05 | 0 |
| Інфраструктура | 0 | 0.06 | 0 |
| Законодавчі акти | 0 | 0.1 | 0 |
| Профспілки, партії та інші громадські організації | 0 | 0.05 | 0 |
| Система економічних відносин в державі | 0 | 0.06 | 0 |
| Організації-сусіди | 0 | 0.01 | 0 |
| Міжнародні події | 0 | 0.01 | 0 |
| Міжнародне оточення | 0 | 0.03 | 0 |
| Науково-технічний прогрес | 0 | 0.07 | 0 |
| Політичні обставини | 0 | 0.06 | 0 |
| Соціально-культурні обставини | 3 | 0.05 | 0.15 |
| Рівень техніки та технологій | 5 | 0.04 | 0.2 |
| Особливості міжнародних економічних відносин | 0 | 0.02 | 0 |
| Стан економіки | 0 | 0.08 | 0 |
| Загальна сума | 11 | 1 | 0.7 |

Таблиця 4.2. Результати експертних оцінок впливу факторів внутрішнього середовища

|  |  |  |  |
| --- | --- | --- | --- |
| **Фактори** | **Середня експертна оцінка, бали** | **Середня вагомість факторів** | **Зважений рівень впливу, бали** |
| Цілі | 4 | 0.11 | 0.44 |
| Структура | 3 | 0.16 | 0.48 |
| Завдання | 4 | 0.07 | 0.28 |
| Технологія | 5 | 0.2 | 1 |
| Працівники | 5 | 0.21 | 1.05 |
| Ресурси | 5 | 0.25 | 1.25 |
| Загальна сума | 26 | 1 | 4.5 |

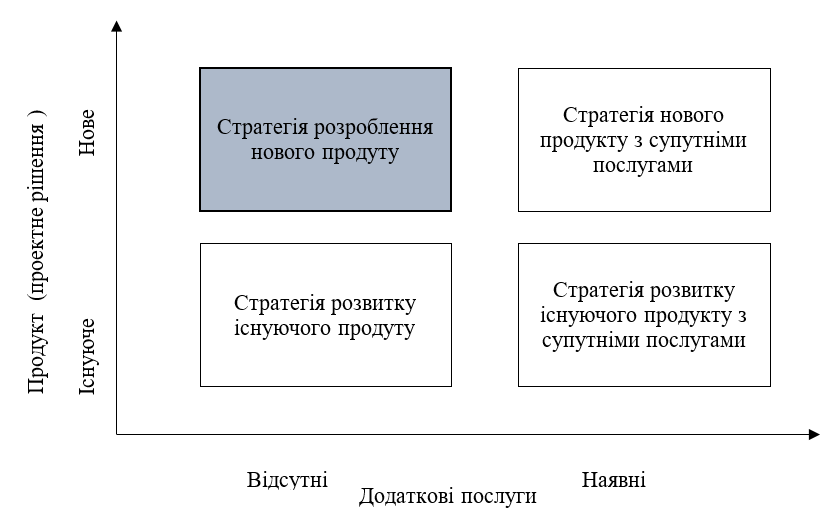
Найбільш сприятливі впливи зі сторони зовнішнього середовища надають споживачі та технології. Система направлена на роботу з користувачами, а технології дозволяють реалізувати систему в існуючих обмеженнях.

Найбільш загрозливий вплив йде зі сторони конкурентів, як технологічних, так і організаційних.

Фактори внутрішнього середовища в нашому випадку є достатньо розвинутими, що, як наслідок, призведе до хорошого результату. Найбільший вплив має рівень працівників та ресурсів.

## Формування стратегічних альтернатив

Дана бакалаврська робота полягає у розробці нового продукту. Стратегічною альтернативою першої групи було обрано: стратегію розроблення нового продукту. Нижче на рисунку буде показано схему розподілу альтернатив, а також вибрано потрібну.

Рисунок 4.1. Вибір стратегічної альтернативи першої групи

Оскільки ми створюємо програмний продукт для існуючого сегменту ринку стратегічною альтернативою другої групи було обрано: стратегію розвитку продукту.

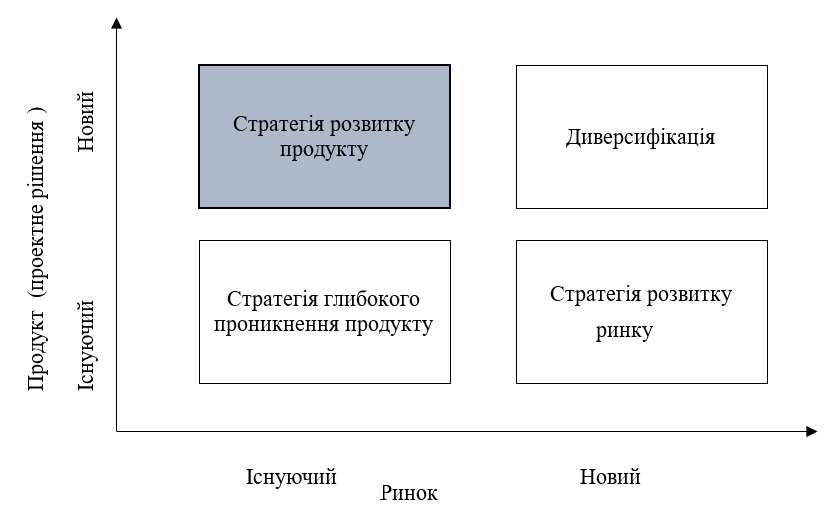


Рисунок 4.2. Вибір стратегічної альтернативи другої групи

Оскільки продукт, що розробляється, є програмним продуктом для існуючого сегменту ринку, тому була вибрана стратегія розроблення нового продукту та стратегія розвитку продукту, які ідеально підходять для розроблення проекту.

## Бюджетування

Бюджетування виступає системою процесів для розрахунку та аналізу витрат, підвищення рентабельності виробництва. Витрати пов’язані з виготовленням та реалізацією розробленого програмного продукту.

Завдання полягає у підрахунку собівартості продукту. Після обчислень потрібно обґрунтувати доцільність однієї із обраних стратегій. Бюджет витрат на матеріали та комплектуючі вироби показано в Таблиця 4.3.

Таблиця 4.3Бюджет витрат на матеріали та комплектуючі вироби

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва матеріалів та комплектуючих | Марка, тип, модель | Фактична кількість, шт. | Ціна за одиницю, грн. | Разом, грн. |
| Сенсор руху | Microsoft Kinect Sensor 2.0 для PC(6L6-00004) | 2 | 4999 | 9998 |
| Мікро сервер | Intel NUC Kit (BOXNUC5PGYH) | 3 | 7271 | 21813 |
| Проектор | Epson EH-TW5300 (V11H707040) | 2 | 27300 | 54600 |
| Кабелі |  | 30 | 40 | 1200 |
| Разом: |  |  |  | 87611 |

Таблиця 4.4 Бюджет витрат на працівників

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Посада, спеціальність** | **Кількість працівників, осіб** | **Час роботи, дні** | **Денна заробітна плата працівників, грн.** | **Сума витрат на оплату праці, грн** |
| Менеджер проекту | 1 | 10 | 2000 | 20000 |
| Дизайнер\Бізнес аналітик | 1 | 10 | 1700 | 17000 |
| Програміст\тех лід | 1 | 10 | 2000 | 20000 |
| Фронт енд програміст | 1 | 10 | 1500 | 15000 |
| Тестер | 1 | 10 | 1000 | 10000 |
| Спеціаліст з інфраструктури | 1 | 5 | 1500 | 7500 |
| Разом: |  |  |  | 89500 |

Таблиця 4.5 Бюджет витрат бюджетні відрахування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Посада, спеціальність** | **Сума основної заробітної плати** | **Сума додаткової заробітної плати** | **Разом витрат на оплату праці** | **Військовий збір (1,5%), грн.** | **Сума податку з доходів фізичних осіб (18%), грн.** |
| Менеджер проекту | 20000 | 0 | 20000 | 300 | 3600 |
| Дизайнер\Бізнес аналітик | 17000 | 0 | 17000 | 255 | 3060 |
| Програміст\тех лід | 20000 | 0 | 20000 | 300 | 3600 |
| Фронт енд програміст | 15000 | 0 | 15000 | 225 | 2700 |
| Тестер | 10000 | 0 | 10000 | 150 | 1800 |
| Спеціаліст з інфраструктури | 7500 | 0 | 7500 | 112.5 | 1350 |
| Разом |  |  |  | 1342.5 | 16110 |

Таблиця 4.6. Бюджет загальновиробничих витрат

|  |  |
| --- | --- |
| **Статті витрат** | **Сума, грн.** |
| Змінні загальновиробничі витрати, у т.ч.: |  |
| - заробітна плата допоміжного персоналу; | 0 |
| - витрати на МШП; | 600 |
| - витрати на електроенергію; | 400 |
| - витрати на ремонт; | 0 |
| - інші змінні витрати; | 0 |
| Разом змінних витрат: | 1000 |
| Постійні загальновиробничі витрати, у т.ч.: |  |
| - заробітна плата допоміжного персоналу; | 0 |
| - комунальні послуги; | 400 |
| - витрати на оренду; | 3000 |
| - витрати на ремонт; | 0 |
| - інші постійні витрати; | 500,00 |
| Разом постійних витрат: | 3900 |
| Разом загальновиробничих витрат: | 4900 |

Таблиця 4.7. Бюджет адміністративних витрат та витрат на збут

|  |  |
| --- | --- |
| **Статті витрат** | **Сума, грн.** |
| *Адміністративні витрати, у т.ч.:* |  |
| - заробітна плата адміністративного персоналу; | 3000 |
| - витрати на МШП; | 1000 |
| - витрати на відрядження; | 500 |
| - витрати на ремонт; | 0 |
| - витрати на паливно-мастильні матеріали; | 0 |
| - витрати на сплату податків і зборів; | 2000 |
| - знос адміністративного обладнання; | 100 |
| - інші адміністративні витрати; | 0 |
| Разом адміністративних витрат: | 6600 |
| *Витрати на збут, у т.ч.:* |  |
| - заробітна плата менеджерів зі збуту; | 0 |
| - витрати на гарантійний ремонт; | 0 |
| - витрати на відрядження; | 0 |
| - витрати на гарантійне обслуговування; | 0 |
| - витрати на налагодження і експлуатацію; | 0 |
| - витрати на паливо-мастильні матеріали; | 0 |
| - витрати на рекламу; | 0 |
| - інші витрати на збут; | 0 |
| Разом витрат на збут: | 0 |

Таблиця 4.8. Зведений кошторис витрат на розробку проектного рішення (продукту)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Статті витрат** | **Одиниці виміру** | **Фактична кількість, шт.** | **Ціна одиниці, грн.** | **Разом, грн.** |
| Сировина і матеріали |  |  |  | 0 |
| Купівельні напівфабрикати та комплектуючі вироби |  |  |  | 87611 |
| Зворотні відходи (вираховуються) |  |  |  | 0 |
| Паливо та електроенергія на технологічні цілі |  |  |  | 0 |
| Основна заробітна плата |  |  |  | 89500 |
| Додаткова заробітна плата |  |  |  | 0 |
| Відрахування на соціальне страхування |  |  |  | 17452.5 |
| Витрати на утримання й експлуатацію устаткування |  |  |  | 0 |
| Загальновиробничі витрати, у т.ч.: |  |  |  | 4900 |
| - змінні; |  |  |  |  |
| - постійні; |  |  |  |  |
| *Разом виробничих витрат:* |  |  |  | 199463.5 |
| Адміністративні витрати |  |  |  | 6600 |
| Витрати на збут |  |  |  | 0 |
| Інші операційні витрати |  |  |  | 0 |
| *Разом виробничих і операційних витрат:* |  |  |  | 206063.5 |

Рішення робиться як інвестиція, тому дохід від реалізації і податки на прибуток та додану вартість рівні 0.

Таблиця 4.9 Підсумкові дані

|  |  |
| --- | --- |
| **Показники** | **Сума, грн.** |
| Дохід від реалізації продукції ( на одиницю товару) | 0 |
| Податок на додану вартість (20%) | 0 |
| Чистий дохід від реалізації продукції | 0 |
| Собівартість реалізованої продукції | 206063.5 |
| Фінансовий результат від операційної діяльності | 0 |
| Податок на прибуток (18%) | 0 |
| Чистий прибуток (збиток) | -206063.5 |

## Висновок

У цьому розділі були отримані і проаналізовані дані щодо ринку, внутрішнього і зовнішнього середовища, стратегічних альтернатив і економічного обґрунтування розробки системи аналізу та регулювання фізичних параметрів спортсмена.

Сьогодні не існує аналогів програмних продуктів. Тому розроблення такої системи, для залучення учасників подій економічно вигідною і є доцільним. Враховуючи також швидкий темп розвитку нових технологій, та сфер їх застосування, даний програмний продукт може бути модифікований і доповнений новими можливостями і застосуваннями.

Стратегічними альтернативами першої та другої груп було обрано стратегію створення нового продукту та стратегію розвитку продукту.

Загальна вартість проекту, на розробку системи, становить 206063.5 грн.

# Висновки

В процесі виконання роботи було досліджено проблему візуального керування графічними об’єктами для інтерфейсів загального користування. Для цього було розроблено набір гіпотез, проведено їхній аналіз, розроблено технічне рішення і проведено аналіз реального використання користувачами.

Виконання роботи дає можливість зробити такі висновки:

Поточний стан NUI потребує кропіткого дослідження, а також переходу до адоптації масовим користувачем. Потенційно це дасть можливість NUI потіснити GUI в області інтерфейсів масового користування.

Точність і інтуїтивність розпізнавання жестів дуже важлива і в сучасному стані є головною проблемою адоптації NUI. Найбільші проблеми це якість давачів і програмне забезпечення виявлення жестів. Також важливим є відсутність ефективних з точки зору ресурсів алгоритмів.

Інтуїтивність поєднання різних типів NUI дуже важлива, і виявляє багато проблем вже на початку проектування. Це включає в себе конфлікти між жестами, конфлікти між користувачами, тощо.

Сучасний стан NUI дозволяє розробляти інтерактивний дизайн який може використовуватись для масового споживача.

Результатом роботи є повне рішення яке підтверджує можливість побудови NUI рішень загального користування. Користувачі з мінімальними вимогами до навчання можуть виконувати закладені дії і досягати результатів керуючи інтерфейсом за допомогою природніх рухів, жестів і дій.

В процесі роботи було підтверджено важливість реалізації додаткових мір по «зацікавленню» користувачів, як то заохочення користувача спробувати зробити жест, або «натяки» на можливі дії.

Рішення дало можливість зібрати важливу статистику використання і інтуїтивності деяких шаблонів поведінки, зробити висновки і використати в фінальній інсталяції.

Розроблене рішення було протестоване на реальній інсталяції з реальними користувачами в межах події Behance Portfolio Review яку організовувала компанія ELEKS у Львові. За час події десятки зовнішніх, незаангажованих користувачів мали можливість пробувати систему.

# Список використаних джерел

Computer and Machine Vision, Fourth Edition: Theory, Algorithms, Practicalities

Practical Python and OpenCV + Case Studies

<https://en.wikipedia.org/wiki/Natural_user_interface>

<https://en.wikipedia.org/wiki/Technology_adoption_life_cycle>

# Додатки

<http://elekslabs.com/2015/03/how-we-made-a-killer-behance-portfolio-review-with-microsoft-kinect.html>