

Одеський національний політехнічний університет  
Інститут комп'ютерних систем  
Кафедра інформаційних систем

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи  
бакалавра

на тему: «Розробка віртуального тренажера на базі контролеру Kinect»

Виконав: студент 4 курсу, групи AI – 153

Напрямок підготовки 6.050101 – «Комп'ютерні науки»

\_\_\_\_\_  
Єржов Є. О.

(прізвище та ініціали)

Керівник Блажко О. А.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Одеса – 2019 року

Одеський національний політехнічний університет

Інститут комп'ютерних систем  
Кафедра інформаційних систем  
Освітньо–кваліфікаційний рівень – бакалавр  
Напрямок підготовки 6.050101 – «Комп'ютерні науки»

ЗАТВЕРДЖУЮ  
Завідувач кафедри ІС

\_\_\_\_\_  
(підпис)  
д.т.н., проф. Арсірій О. О.  
“    “    \_\_\_\_\_ 2019 р.

**З А В Д А Н Н Я**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

\_\_\_\_\_ Єржову Євгенію Олександровичу

1. Тема роботи: «Розробка віртуального тренажера на базі контролеру Kinect»

керівник роботи Блажко Олександр Анатолійович, канд. техн. наук,  
доцент \_\_\_\_\_,

( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 14.05.2019 року № 231-в

2. Строк подання студентом роботи 5.07.2019 р.

3. Вихідні дані до роботи Технічна документація з використання технологій  
розробки та збірки мобільного застосовання.

4. Зміст розрахунково–пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Огляд системі віртуальних тренажерів та їх технологій.  
Проектування віртуального тренажера на базі контролера Kinect. Реалізація  
віртуального тренажера на базі контролера Kinect. Забезпечення якості ІС. Охорона  
праці. Висновок. Перелік посилань. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Комп'ютерна презентація

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Козерацький Г.В., ст. викл.		

7. Дата видачі завдання 29.01.19

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної Роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	15.02.2019	Виконано
2	Порівняння існуючих систем–аналогів віртуального тренажеру	28.02.2019	Виконано
3	Аналіз технологій для розробки віртуального тренажеру	10.03.2019	Виконано
4	Проектування системи віртуального тренажеру	25.04.2019	Виконано
5	Програмна реалізація системи віртуального тренажеру	30.04.2019	Виконано
6	Тестування системи віртуального тренажеру	19.05.2019	Виконано
7	Розгляд питань охорони праці	29.05.2019	Виконано
8	Оформлення пояснювальної записки	10.06.2019	Виконано
9	Нормоконтроль		
10	Попередній захист		
11	Захист		

Студент

\_\_\_\_\_  
( підпис )

Єржов Є. О.

(прізвище та ініціали)

Керівник проекту

\_\_\_\_\_  
( підпис )

Блажко О. А.

(прізвище та ініціали)

## ЗАВДАННЯ

на розробку розділу «Охорона праці» у кваліфікаційній роботі бакалавра:

студента Єржова Є. О.  
(прізвище та ініціали)  
групи AI-153 інституту комп'ютерних систем.

Дата отримання завдання 01.03.2019 р.

Консультації “12” червня 2019 р.

Дата закінчення розробки “26” червня 2019 р.

Тема дипломної роботи: «Розробка віртуального тренажера на базі контролера Kinect»

Розділ розробляється на основі «Методичних вказівок по виконанню розділу «Охорона праці» у дипломних проектах (роботах) для студентів всіх напрямів» із використанням рекомендованих літературних джерел.

Розділ «Охорона праці», залежно від профілю дипломного проекту може включати такі підрозділи:

1. Аналіз можливих небезпечних і шкідливих виробничих факторів, створюваних об'єктом проектування.
2. Розробку заходів, спрямованих на усунення або зниження шкідливого впливу виявлених факторів та забезпечення пожежної і вибухопожежної безпеки, які базуються на нормативних документах (ГОСТ, ССБТ, ДНАОП, НПАОП, ДСТУ, ДСН, ДСанПІН, СНиП, ДБН, НАПБ, РД та інші, в першу чергу ГОСТ 12.4.011–75).

Рекомендована література: Охрана труда в машиностроении /Под ред. Е.Я. Юдина и С.В. Белова. 2-е изд. – М.: Машиностроение, 1983. – 432 с.

Розділ «Охорона праці» оформлюється окремим розділом із зазначенням назви підрозділів відповідно до п. 1, 2, 3.

Керівник роботи

Блажко О.А.  
(підпис) (прізвище та ініціали)

«    »    2019 р.

Консультант з охорони праці

Козерацький Г. В.  
(підпис) (прізвище та ініціали)

«    »    2019 р.

## ЗМІСТ

Вступ.....	7
1 Огляд систем віртуальних тренажерів та їх технологій.....	11
1.1 Порівняння аналогів віртуальних тренажерів.....	11
1.1.1 Застосунок everyone piano .....	11
1.1.2 Застосунок pianoworld .....	12
1.1.3 Застосунок virtual piano .....	14
1.1.4 Результат порівнянь .....	15
1.2 Вибір технологій для розробки системи.....	16
1.3 Висновок до першого розділу .....	21
2 Проектування віртуального тренажера на базі контролера kinect.....	23
2.1 Мета та задачі іс .....	23
2.2 Типи користувачів.....	24
2.3 Функціональні вимоги .....	25
2.4 Нефункціональні вимоги .....	26
2.5 Ідентифікація архетипу іс.....	27
2.6 Користувацький інтерфейс (ui view).....	27
2.7 Логічне уявлення про іс (logical view) .....	29
2.8 Уявлення розгортання іс (deployment view) .....	30
2.9 Уявлення слоїв іс (design view).....	31
2.10 Уявлення процесів іс (process view) .....	32
2.11 Уявлення безпеки іс (security view).....	38
2.12 Опис стеку технологій .....	38
2.13 Висновок до другого розділу .....	39
3. Розробка віртуального тренажера на базі контролера kinect.....	40

					ІС КРБ 6.050101 019 ПЗ						
Змін.	Арк.	№ докум.	Підпис	Дата	Віртуальний тренажер на базі контролера Kinect			Літ.	Лист	Листів	
Розробник	Єржов Є. О.										
Керівник	Блажко О. А.								5	74	
								ОНПУ, ІКС, каф. ІС			
Нормконтр.	Ядрова М.В.										
Затв.	Арсірій О.О.										

3.1 Уявлення про структуру проекту іс .....	40
3.2 Уявлення про класи іс.....	40
3.3 Інфраструктурне уявлення іс (infrastructure view) .....	42
3.4 Стратегія доставки продукту (delivery strategy view).....	42
3.5 Документація іс .....	42
3.6 Керування програмним кодом іс .....	42
3.7 Розрахунок метрик програмного коду іс .....	43
3.8 Контрольний список з якості реалізації іс.....	43
3.9 Протокол проведення функціонального тестування .....	44
3.10 Проведення тестування точності.....	45
3.11 Висновки до третього розділу.....	46
4.1 Аналіз небезпечних і шкідливих факторів .....	47
4.1.1. Організація робочого місця інженера–програміста.....	47
4.1.2 Мікроклімат робочої зони інженера–програміста.....	49
4.1.3 Освітлення робочого місця .....	49
4.1.4 Вплив шуму на інженера–програміста .....	51
4.1.5 Виробничі випромінювання.....	52
4.2 Розробка заходів з охорони праці.....	54
4.2.1 Ергономіка та організація робочого місця .....	54
4.2.2 Нормалізація повітря робочої зони .....	55
4.2.4 Захист від виробничого шуму.....	58
4.2.5. Захист від електромагнітних полів.....	59
4.2.6 Електробезпека .....	59
4.3. Пожежна безпека.....	60
4.3.1 Причини виникнення пожежі .....	60
4.3.2 Профілактика пожежі .....	61
4.4 Висновки до четвертого розділу.....	62
Висновки .....	64
Перелік посилань.....	66
Додаток А Лістинг програмного коду.....	67

Додаток Б Інструкція користувача .....	71
Додаток В Відомість до кваліфікаційної роботи бакалавра .....	76

					ІС КРБ 6.050101 019 ПЗ	
Змін	Лист	№ докум.	Підпис	Дата		7

## ВСТУП

Головною метою дипломної роботи є створення віртуального тренажеру на базі контролера Kinect. Серед віртуальних тренажерів першими приходять на думку тренажери з гри на музичних інструментах. Для розробки було обрано віртуальне піаніно.

Актуальність тренажерів у спортивних залах ні в кого не викликає сумнівів, в цьому ж проекті будуть поєднані переваги фізичних і розумових навантажень. Оскільки для того, щоб займатися на фізичних тренажерах потрібно кудись йти, віртуальний стане у нагоді будь-кому, у кого є пристрій Kinect.

Піаніно (італ. *Pianino*, дослівно: маленьке фортепіано) – струнно-клавішний музичний інструмент з ударним (молоточковим) способом звуковидобування, створений спеціально для кімнатного музикування в невеликих приміщеннях. Піаніно є зменшеним за розміром різновидом фортепіано, в якій струни, дека і механічна частина розташовані вертикально, а не горизонтально, внаслідок чого піаніно займає набагато менше місця, ніж рояль.

Сервіс надає користувачам можливість як програвати довільну мелодію, рухаючись по віртуальним клавішам, що проєціюються проектором, так і грати послідовні пісні за допомогою створеної бази пісень.

Грі на піаніно навчитися за тиждень, місяць або рік неможливо. Тим більше це завдання стає нездійсненним, якщо немає можливості займатися музикою, наприклад, через відсутність в будинку відповідного музичного інструменту. Але з нинішніми гаджетами і новинками неможливе стає цілком реальним. Віртуальні піаніно – ось відмінне рішення для тих, хто давно хотів спробувати грати улюблені мелодії, але не робив це з різних вагомих причин.

Найважливіше з того, що потрібно знати – цифрове піаніно створює звук способом, аналогічним свого дерев'яного предку – механічно (в дорогих моделях використовується навіть вже відома нам молоточкова структура). Так що, фактично, цифрове піаніно неправильно було б називати електронним

					ІС КРБ 6.050101 019 ПЗ	8
Змін	Лист	№ докум.	Підпис	Дата		



інструментом: це – електронно–механічний інструмент, на відміну, наприклад, від того ж синтезатора.

При грі на електронному піаніно ви не помітите жодної різниці з грою на звичайному – клавіші є повновагими і, як наслідок куди більш підходять для навчання, на відміну від синтезаторів, які мають менший розмір клавіатури, менше октав, і натискання відбувається практично без зусиль. Коли ви вирішите показати майстер–клас, маючи тільки досвід гри на синтезаторі, то ви переоціните поняття «фортепіано» і «клавішні інструменти».

Застосування підходить як для розваги, так і для навчання, тренувань в домашніх умовах. Інтерфейс інструменту максимально простий і зрозумілий, тому труднощів з ознайомленням його функціональності виникати не повинно. До того ж це відмінний варіант, якщо грати хочеться зараз, а можливостей для реалізації свого бажання на даний момент немає. Гра на піаніно онлайн на клавіатурі максимально схожа на гру на оригінальному інструменті. А це в свою чергу дозволить отримати максимальне задоволення від музикування.

В роботі розглядається та розробляється віртуальне піаніно на базі контролера Kinect.

Для реалізації цього застосування необхідно виконати наступні задачі:

- аналіз інших віртуальних піаніно;
- визначення функціональних вимог до системи
- вибір інструментарію для її проектування та реалізації
- проектування системи віртуального піаніно на базі контролера Kinect;
- реалізація системи віртуального піаніно на базі контролера Kinect;
- тестування системи віртуального піаніно на базі контролера Kinect

Подібна інформаційна система відкриває можливості для самореалізації студентів та поглиблення їх знань у галузях програмування, розробки застосовань, командної роботи, комунікацій та інших.

# 1 ОГЛЯД СИСТЕМ ВІРТУАЛЬНИХ ТРЕНАЖЕРІВ ТА ЇХ ТЕХНОЛОГІЙ

## 1.1 Порівняння аналогів віртуальних тренажерів

Перед початком робочого процесу треба провести дослідження предметної області та знайти певні аналоги майбутньої системи. В час коли технології стали доступні будь-кому знайдуться аналоги для будь-якого програмного продукту.

В результаті пошуку аналогів було знайдено три десктоп-застосування:

- десктоп-застосування «Everyone Piano» [1];
- десктоп-застосування «PianoWorld» [2];
- десктоп-застосування «Virtual Piano» [3];

Розглянемо ці три системи з точки зору повсякденного користування та можливостей, які вони надають.

### 1.1.1 Застосунок Everyone Piano

Everyone Piano – це витончений інструмент, який перетворює ПК в класичне піаніно, яке звучить практично так само, як справжнє. З Everyone Piano ви зможете практикуватися в грі прямо на комп'ютері, покращуючи свою майстерність, або почати навчання гри на піаніно з нуля.

Даний симулятор призначає кожній клавіші на клавіатурі певну ноту, і навіть клавіші для кожної з педалей. Ця програма також дозволяє записувати уривки своїх імпровізацій або гри, щоб пізніше можна було прослухати і виправити помилки. Додаток також включає величезну музичну бібліотеку з демо-композиціями

Клавіатура Everyone Piano повністю настроюється – ви зможете налаштовувати навіть півтонни і тональність, в якій будете грати. Інструмент має кілька розширених налаштувань, що дозволяють імпортувати звуки з інших джерел і підключати MIDI-клавіатуру або електронний орган.

					ІС КРБ 6.050101 019 ПЗ	10
Змін	Лист	№ докум.	Підпис	Дата		

Основні переваги цього застосунку:

- досить великий спектр можливостей для налаштування;
- можливість персоніфікації зовнішнього вигляду програми;
- наявність регулятора гучності та регулятора тонів;

Основні недоліки цього застосунку;

- досить складний інтерфейс для початківця;
- перевантаженість програми функціями;



Рисунок 1.1 – Інтерфейс застосунку Everyone Piano

Отже, підсумовуючи наш аналіз: застосунок більш підходить для людини, яка вже мала неабиякий досвід як у роботі із комп'ютером, так і у грі із реальним чи віртуальним піаніно. Можливостей для налаштування чи персоніфікації програми настільки багато, що це подовжує час на знайомство з програмою та віртуальним піаніно.

### 1.1.2 Застосунок PianoWorld

Реальне Піаніно – це не тільки віртуальний симулятор піаніно, а й захоплюючі ігри, мелодії і пісні, за допомогою яких можна навчитися грати на вашому улюбленому музичному інструменті.

Основні функції та переваги застосунку:

- можливість записувати свою мелодію;
- повний діапазон фортепіано – 7 ½ октав (88 клавіш)
- 2 ігрових режими;
- міні-ігри;
- 9 різних клавішних і струнних інструментів: піаніно, фортепіано, орган, клавесин, акордеон, електрична гітара, арфа, піццикато, вінтажне піаніно;
- 1 або 2 ряди клавіш зі зручним скролом;
- безкоштовні популярні пісні для гри на інструменті;
- набір класичних мелодій;



Рисунок 1.2 – Інтерфейс застосунку PianoWorld

Недоліки цього застосунку полягають у наступному:

- він мультифункціональний, тому користувач, що хоче грати саме на піаніно, а не на іншому інструменті, може не знайти достатньо функцій саме для

					ІС КРБ 6.050101 019 ПЗ	12
Змін	Лист	№ докум.	Підпис	Дата		

оптимізації гри на піаніно;

– застосунок орієнтований скоріше на проведення вільного часу, аніж на навчання гри на піаніно.

### 1.1.3 Застосунок Virtual Piano

Virtual Piano – програма буде корисна музикантові в самих різних ситуаціях (наприклад, налаштувати на слух музичний інструмент, з'ясувати тональність почутої музики, озвучити уроки музичної грамоти, підібрати на слух акорди і т.д.). Ця програма в сукупності з відеопроєктором дозволить ефектно пояснити що–небудь для великої аудиторії учнів музичних шкіл, коледжів і консерваторій.

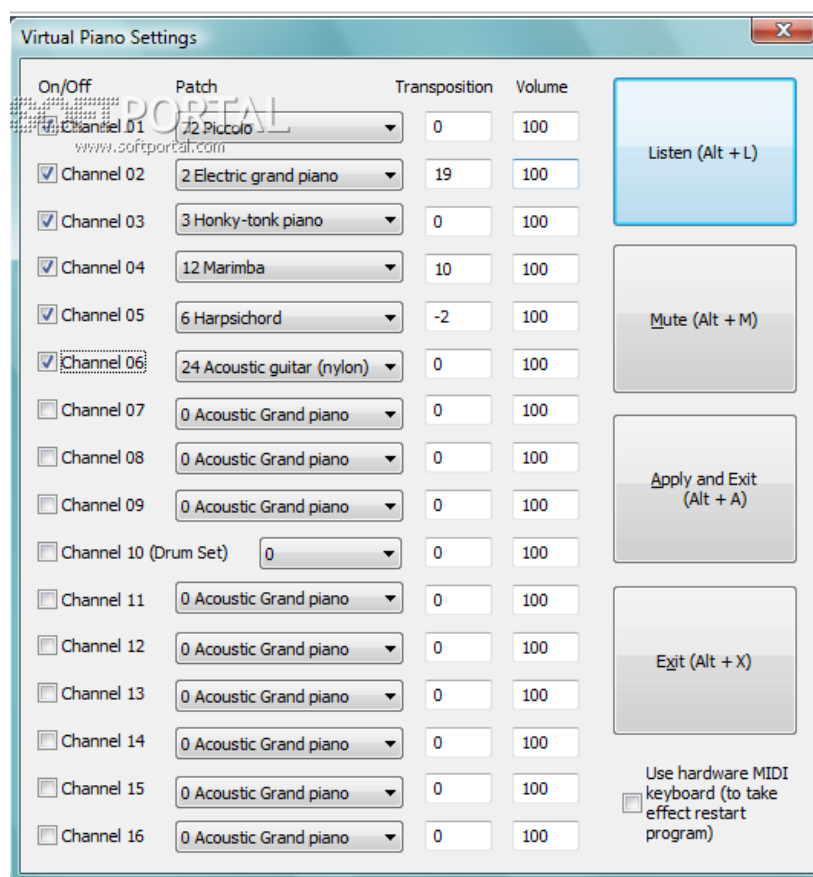


Рисунок 1.3. – Меню налаштувань застосування «Virtual Piano»

Virtual Piano дозволяє створювати і зберігати у вигляді файлів колекцію улюблених тембрів, заснованих на можливостях міді–синтезатора звукової карти.

Кожен тембр може представляти із себе сукупність з одного або декількох різних по гучності і транспонірованію тембрів (всього можуть бути задіяні від 1 до 16 міді-каналів). Можливість грати в режимі сустейна і негайно приглушати всі звуки. (рис. 1.4)

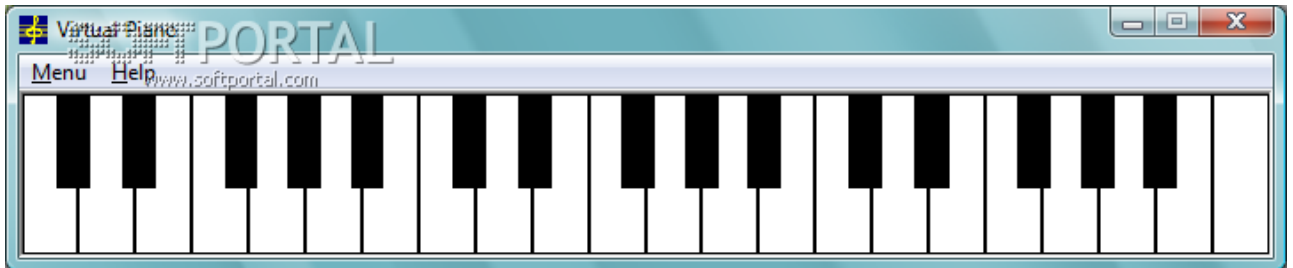


Рисунок 1.4. – Інтерфейс застосунку «Virtual Piano».

Основними перевагами застосунку є:

- простота інтерфейсу;
- можливість налаштування каналів програвання;
- можливість налаштування спектру звуків для програвання.

Основними недоліками є:

- малі можливості для персоніфікації застосунку;
- мінімум необхідних функцій для гри.

#### 1.1.4 Результат порівнянь

В результаті дослідження декількох аналогів розробляємої системи можна зробити висновок, що всі представлені варіанти дуже відрізняються один від одного.

Перші два застосування схожі між собою великою кількістю функціоналу, проте перше орієнтоване на гравця з досвідом, а друге на дітей, що прагнуть якось провести вільний час.

Система, що розробляється для дипломної роботи, буде працювати завдяки середовищу Scratch та системі розпізнавання Kinect. Проектор транслюватиме на

					ІС КРБ 6.050101 019 ПЗ	14
Змін	Лист	№ докум.	Підпис	Дата		

підлогу віртуальне піаніно, а система Kinect буде фіксувати фігуру гравця і розпізнавати його рухи.

Для більш наглядного порівняння створимо таблицю в якій відзначимо основні схожості та розбіжності систем.

Таблиця 1.1 – Результати порівнянь аналогів

Ознаки	Назва системи			
	EveryOne Piano	PianoWorld	Virtual Piano	Віртуальний тренажер
Зручний інтерфейс	–	+	+	+
Оптимальне співвідношення стандартного і розширеного функціоналу	–	+	–	+
Можливість налаштовувати дизайн	+	–	–	–
Зручність для новачків	–	+	+	+
Максимальне охоплення аудиторії незалежно від навичок	–	+	–	+
Можливість грати заготовлені пісні	+	+	–	+
Підтримка системи розпізнавання Kinect	–	–	–	+
Можливість зберігати пісні	–	–	+	+

Таким чином, система, що розроблюється, поєднає в собі більшість переваг інших подібних застосувань та пропонуватиме значно ширший функціонал.

## 1.2 Вибір технологій для розробки системи

Є лише два достатньо відомих середовища, що взаємодіють із Kinect – Unity та Scratch.

Unity – більше, ніж рушій, це середовище для розробки комп'ютерних ігор, в якій об'єднані різні програмні засоби, що використовуються при створенні ПЗ – текстовий редактор, компілятор, відладчик і так далі. При цьому, завдяки зручності використання, Unity робить створення ігор максимально простим і комфортним, а

мультиплатформеність движка дозволяє ігоробів охопити якомога більшу кількість ігрових платформ і операційних систем. Міжплатформовість є досить вагомим аргументом на користь цього середовища, проте він перш за все комерційний. Також недоліком цього середовища є те, що воно оптимізовано під великі проекти професійних компаній. Проаналізуємо переваги Unity.

В першу чергу, рушій Unity3D дає можливість розробляти гри, не вимагаючи для цього якихось особливих знань. Тут використовується компонентно-орієнтований підхід, в рамках якого розробник створює об'єкти (наприклад, головного героя) і до них додає різні компоненти (наприклад, візуальне відображення персонажа і способи управління ним). Завдяки зручному Drag & Drop інтерфейсу і функціональному графічному редактору рушій дозволяє малювати карти і розставляти об'єкти в реальному часі і відразу ж тестувати вийшов результат.

Друга перевага рушія – наявність величезної бібліотеки ассетів і плагінів, за допомогою яких можна значно прискорити процес розробки гри. Їх можна імпортувати і експортувати, додавати в гру цілі заготовки – рівні, ворогів, патерни поведінки штучного інтелекту і так далі. Ніякої метушні з програмуванням. Багато ассетів надаються безкоштовно, інші пропонуються за невелику суму, і при бажанні можна створювати власний контент, публікувати його в Unity Asset Store і отримувати від цього прибуток.

Третя сильна сторона Unity 3D – підтримка величезної кількості платформ, технологій, API. Створені на рушії ігри можна легко перенести між ОС Windows, Linux, OS X, Android, iOS, на консолі сімейств PlayStation, Xbox, Nintendo, на VR- і AR-пристрої. Unity підтримує DirectX і OpenGL, працює з усіма сучасними ефектами рендеринга, включаючи новітню технологію трасування променів в реальному часі.

Нарешті, Unity доступний безкоштовно, що відкриває перед незалежними розробниками двері в ігрову індустрію. Звичайно, існують обмеження: безкоштовна версія рушія демонструє лого Unity перед запуском гри, а проект, створений з її допомогою, не повинен приносити розробнику більше \$ 100 тисяч в

					ІС КРБ 6.050101 019 ПЗ	16
Змін	Лист	№ докум.	Підпис	Дата		



рік. Втім, тарифи на підписку не спустошать гаманці навіть починаючої команди: про-версія коштує \$ 125 на місяць, що не так вже й багато в порівнянні з іншими двигунами, причому базова версія містить рівно той же функціонал, що і професійна.

При всіх своїх перевагах, рушій має і свої недоліки. Так, якщо команда захоче розробити що-небудь складніше простого клікера або платформера, то їй доведеться шукати хорошого програміста на C #, який напише скрипти і компоненти, запровадить їх у гру і змусить працювати.

З цього випливає інша проблема рушія Unity – повільність. Створення масштабних, складних сцен з великою кількістю компонентів може негативно вплинути на продуктивність гри, в результаті чого розробникам доведеться витратити додатковий час і ресурси на оптимізацію, а можливо – і видалення деяких елементів з проекту. Крім того, додатки, створені на Unity, досить «великовагових»: навіть найпростіша піксельна гра може займати кілька сотень мегабайт на ПК. Так, для жорсткого диска комп'ютерів це невеликий обсяг, але, якщо проект розробляється і для мобільних платформ, слід задуматися про оптимізацію його розміру.

Для розробки нашого віртуального тренажеру ми будемо використовувати середовище Scratch, оскільки воно більш оптимізоване під некомерційні маленькі проекти. Проекти у ньому пишуться також на Scratch – освітній мові програмування з чисто графічним інтерфейсом.

Головне вікно середовища розробки Scratch розділене на кілька частин, згрупованих в три колонки.

Ліва колонка містить палітру блоків. Блок – це мінімальний фрагмент програми в Scratch: змінна, оператор, функція або керуюча структура. Блоки згруповані в наступні 8 категорій:

- Motion (управління рухом спрайтів)
- Looks (графічні ефекти і функції виведення think і say)
- Sound (звукові ефекти)
- Pen (спрайтовий еквівалент черепащачої графіки)

- Control (керуючі структури)
- Sensing (обробка сигналів від миші, клавіатури і сенсора)
- Operators (математичні, логічні та рядкові функції і оператори)
- Variables (скалярні і векторні змінні і операції над ними).

– Центральна колонка містить поточного спрайта і його скрипти. Спрайт – це об'єкт в стилі Scratch, асоційований з зображенням і набором змінних і скриптів, які визначають його поведінку. Скрипти створюються методом з'єднання окремих блоків – або послідовно, або розміщуючи блок в певному місці іншого блоку (керуючої структури, функції і т.д.). Один спрайт може володіти декількома скриптами, які запускаються незалежно – дією користувача (натисканням клавіші або кнопки миші), таймером або отриманням повідомлення від іншого спрайту.

Права колонка містить сцену і список спрайтів. Сцена – ділянка, у якій діють спрайти при виконанні програми. Вона містить зображення видимих спрайтів, видимі змінні і будь-які повідомлення та графічні ефекти, створені при виконанні програми. Кумедна і проста мова Scratch ілюструє кілька важливих парадигм:

- структурна (в низкоурівневому розумінні): всі програми конструюються з обмеженого набору елементів (блоків);
- об'єктно-орієнтована: кожен спрайт насправді є об'єктом зі своїми властивостями (змінними) і поведінкою (скриптами), і різні об'єкти можуть взаємодіяти;
- багатопоточна: об'єкти взаємодіють за допомогою обміну повідомленнями через блоки broadcast і when I receive;

Також ключовим пристроєм, що буде використаний у цій роботі, є Microsoft Kinect.

Kinect – це інтерактивний ігровий безконтактний контролер, спочатку представлений для консолі Xbox 360 (1 червня 2009). 1 лютого 2012 вийшов новий контролер Kinect For Windows (разом з однойменною бібліотекою), який відрізняється від свого "ігрового" колеги тим, що здатний обчислювати глибину сцени починаючи від 40см.

Серцем сенсора Kinect, є 3D-технологія від компанії PrimeSense, яка

використовує процесор для вимірювання відстані від камери до сцени.

Ці виміри проводяться по всьому полю зору сенсора. В результаті, виходить хмара точок, що складається з 307200 вимірювань відстаней між сенсором і сценою. Точки проектується на заданих кутах від лазера. Лазерні точки відбиваються від об'єктів сцени і фіксуються ІК-камерою. Об'єкти сцени, розташовані близько до камери відбиваються у точки зображення, які знаходяться близько один до одного, а об'єкти, розташовані далеко від камери – відбиваються у точки зображення, які знаходяться далі один від одного.

Пристрої Kinect самі по собі передають користувачеві дані з RGB камери і камери глибини. Працювати з чистими потоками даних було б дуже важко і, звичайно ж, існують бібліотеки, які надають більш високорівневий інтерфейс для роботи з Kinect.

Одним з основоположних рішень був вибір бібліотеки взаємодії з Kinect. На даний момент відомі 3 найбільш популярні бібліотеки:

- Libreenect [4];
- OpenNI [5];
- KinectForScratch [6];

Перша бібліотека libfreenect надає самий низькорівневий доступ до Kinect: зокрема, тільки кольорове і глибинне зображення. Після короткого порівняльного аналізу стає ясно, що libfreenect не підходить за відсутністю підтримки розпізнавання скелета. При порівнянні OpenNI і KinectForWindows підсумковий вибір припав на офіційну бібліотеку від Microsoft з наступних причин:

- Найкраще розпізнавання кінцівок;
- Підтримка запису звуку з Kinect – отже, є можливість управління голосом за допомогою технології Microsoft Speech [7];
- Підтримка декількох сенсорів Kinect [8];
- Пряма підтримка контролерів Kinect For Windows, Kinect [9]

Створимо таблицю для порівняння, щоб систематизувати переваги і недоліки бібліотек. (табл. 1.2)

					ІС КРБ 6.050101 019 ПЗ	19
Змін	Лист	№ докум.	Підпис	Дата		

Таблиця 1.2 – Порівняння бібліотек

Критерії	Libfreenect	OpenNI	Kinectforscratch
Перший реліз	11.2010	11.2010	06.2011
Open Source	+	+	–
Вільне розповсюдження	+	+	+
Низькорівневий доступ до камер	+	+	+
Доступ до мікрофонного масиву	–	–	+
Розпізнавання скелета	–	+	+
Кроссплатформовість	+	+	–
Керування телеметрією	+	–	+
Підтримка	Невідомо	+-	+
Розпізнавання пальців	–	–	–
Вбудоване калібрування глибини і кольору	–	+	+

### 1.3 Висновок до першого розділу

В результаті проведення порівняння аналогів були розглянуті три системи, що мають в собі подібний функціонал. Ці системи працюють з активним підключенням до мережі інтернет, реалізують інструментарій щодо поширення ідей і проектів та дають можливість формування проектних команд.

Для побудови майбутньої системи, будуть використовуватись основні переваги інших систем та додані свої унікальні ознаки, які б відзначили та відрізняли застосування з ряду інших та привнесла дещо нове.

Також були розглянуті основні технології, які будуть потрібні на етапі реалізації системи, а деякі вже на етапі проектування. Всі технології є безкоштовними та розповсюджуються за відкритими, вільними ліцензіями, а тому їх використання не накладає ніяких обмежень.

Було вирішено не використовувати середовище та рушій Unity через те, що він спрямований на більш комерційні та великі проекти. Також недоліком є те, що для відкриття додаткового функціоналу, що передбачений розробниками, треба додатково використовувати гроші з бюджету проекту.

Гравець буде розпізнаватися за допомогою пристрою Kinect, наше застосування буде ініціалізуватися у середовищі Scratch, а за взаємодію між Scratch і Kinect буде відповідати Scratch2Kinect. Scratch2Kinect було обрано через те, що в нього найбільш придатний функціонал для розробки нашого застосування.

На основі аналізу аналогів наступні їх недоліки були визначені як пріоритетні переваги у віртуальному тренажері Kinect:

- оптимальне співвідношення стандартного і розширеного функціоналу
- максимальне охоплення аудиторії незалежно від навичок
- зручний інтерфейс
- можливість грати заготовлені пісні

					ІС КРБ 6.050101 019 ПЗ	
Змін	Лист	№ докум.	Підпис	Дата		21

## 2 ПРОЕКТУВАННЯ ВІРТУАЛЬНОГО ПІАНІНО НА БАЗІ КОНТРОЛЕРА KINECT

### 2.1 Мета та задачі ІС

Розробляема інформаційна система – це RCA застосовання, основною функцією якого є надання користувачу можливості грати на піаніно за допомогою Kinect [9].

Мета ІС: інформаційна система повинна допомогти освоїти гру на піаніно навіть тим, хто не вміє на ньому грати.

Цільова аудиторія: офісні працівники, яким бракує рухливого образу життя; молодь, що прагне вчитися грі на піаніно.

Головними функціями застосовання є:

- фіксація рухів користувача у реальному часі;
- програвання клавіш, із якими взаємодіє користувач;
- надання інструкції взаємодії із додатком;

Інформаційні потоки ІС зображені на рисунку 2.1.



Рисунок 2.1 – Інформаційні потоки розробляємої ІС.

Можна побачити, що на виході користувач отримує два типи інформаційних

одиниць, тобто результати розпізнавання у вигляді охоплення фігури користувача (виділяється червоним) та реакція програми на рухи.

Охоплення фігури спільно з реакцією програми на рухи утворюють сутність «Мелодія».

Знаходження повинно мати наступні атрибути:

- приналежність мелодії до пісні;
- кількість зіграних нот;
- послідовність зіграних нот;
- кількість зіграних акордів;

## 2.2 Типи користувачів

В інформаційній системі, що розроблюється, можливий лише один тип користувачів – це користувач.

Інформаційна система має певні функції, які на мові UML вказуються на діаграмі варіантів використання. Найголовніші варіанти використання наведені в таблиці 2.1.

Таблиця 2.1 – Визначені варіанти використання

Код	Основний актор	Найменування
A1	Виконавець	Запустити середовище виконання
A2	Виконавець	Відкрити список наявних пісень
A3	Виконавець	Додати пісню
A4	Виконавець	Видалити пісню
A5	Виконавець	Запустити систему розпізнавання Kinect
A6	Виконавець	Провести калібрування
A7	Виконавець	Грати за нотами пісень
A8	Виконавець	Грати довільну мелодію
A9	Виконавець	Обрати пісню

## 2.3 Функціональні вимоги

Вимога 1 – розпізнавання фігури та рухів користувача

F1.1 Функція налаштування системи розпізнавання

FR1.1.1 Гравець має стати на таку відстань від пристрою Kinect, щоб пристрій його повністю охопив і виділив червоним кольором.

FR1.1.2 Обмеження: якщо гравець стоїть занадто далеко, або близько, система його не розпізнає.

FR1.1.3 Обмеження: якщо у межах пристроя стоїть більше однієї людини, він може не розпізнати, або навести фокус не на гравця.

FR1.1.4 Обмеження: якщо пристрій Kinect не був увімкнений, або потрібне програмне обладнання не було запущено, користувача розпізнано не буде.

Вимога 2 – взаємодія програми із списками пісень

F2.1 Функція додавання пісень

FR2.1.1 Для додавання пісні потрібно «прив'язати» текстовий файл у форматі \*.txt, в якому буде послідовність нот потрібної мелодії. Для зручності є папка «ноти», в якій в подальшому будуть зберігатися всі пісні.

FR2.1.2 Обмеження: якщо зберегти файл у форматі, відмінному від txt, програма може його не розпізнати. Ноти пісні потрібно писати по одній у стовпчик.

FR2.2 Функція видалення пісень

FR2.2.1 Для видалення пісні достатньо видалити іконку для нової пісні з панелі керування, а також текстовий файл з її нотами.

FR2.2.2 Обмеження: якщо видалити лише текстовий файл, іконка продовжить посилатись на пусту пісню.

FR2.3 Функція програвання обраної пісні

FR2.3.1 Для програвання нот лише обраної пісні, треба перейти до меню вибора пісень «Choose melody» та обрати пісню.

FR2.3.2 Обмеження: якщо гравець буде грати інші ноти (або в іншій послідовності), аніж ті, що підсвічені на піаніно, програма буде подавати сигнали

					ІС КРБ 6.050101 019 ПЗ	24
Змін	Лист	№ докум.	Підпис	Дата		



та лише після правильних рухів продовжить грати мелодію.

Вимога 3 – синхронізація рухів користувача із віртуальним піаніно

F3.1 Функція калібрування рухів користувача

FR3.1.1 Після запуску віртуального піаніно, будет проводитися калібрування рухів користувача для його подальшої злагодженої взаємодії із віртуальним піаніно.

FR3.1.2 Про послідовність калібрування сповістить программа. Для початку гравець потрібен стати однією ногою на потрібний маркер, а потім іншою. Після цього він повинен повторити те саме, але зафіксувати рух при піднятті ноги.

## 2.4 Нефункціональні вимоги

NF1. Віртуальне піаніно повинно з точністю як мінімум у 80% відтворювати аккорди реального аналогу.

NF2. Робота із програмою має бути інтуїтивно зрозумілою та не викликати труднощів навіть у тих, хто не мав раніше справу з Kinect

NF3. Незалежно від статури гравця, пристрій Kinect повинен його розпізнавати.

NF3. Реакція програми на рух гравця повинна бути не більше 2–3 секунд.

Вимоги визначають можливості застосування, отже на підставі списку вимог можна побудувати діаграму варіантів використання (рис 2.2) ІС.

Для детальнішого розгляду процесу є можливість побудови додактові, більш детальніші діаграми використання. Наприклад, побудуємо діаграму прецедентів для блоку отримання інформації з пристрою (рис. 2.3).

Ця діаграма надасть нам більше розуміння про можливі варіанти поведінки користувача у обраному сценарії. Аналогічні діаграми можуть бути створені для будь-якого процесу.

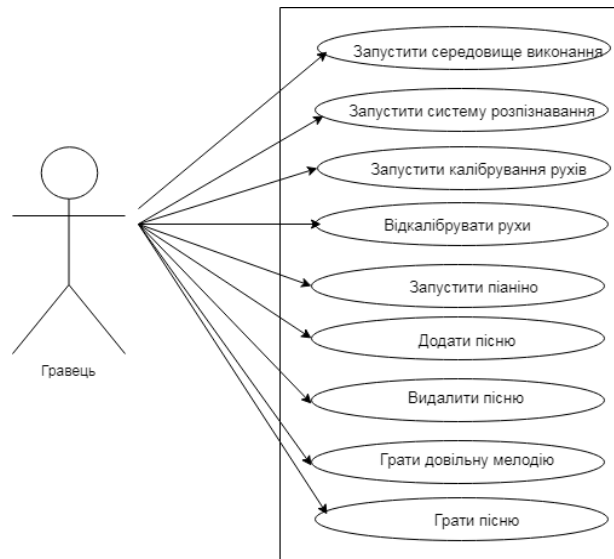


Рисунок 2.2 – Діаграма варіантів використання ІС

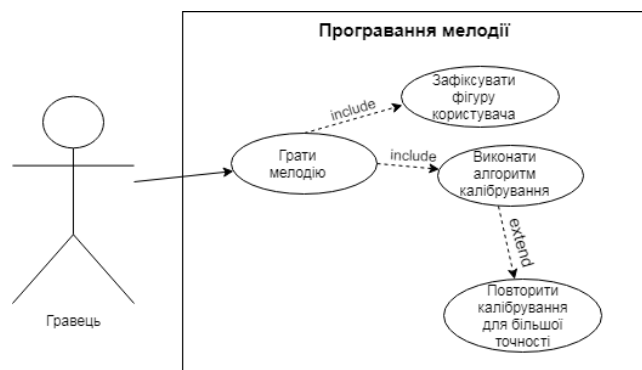


Рисунок 2.3 – Діаграма прецедентів для блоку «програвання довільної мелодії»

## 2.5 Ідентифікація архетипу ІС

Інформаційна система буде RCA–застосуванням. Для роботи з віртуальним тренажером, треба інсталиувати середовище Scratch, яке буде взаємодіяти з Kinect через клієнт Scratch2Kinect.

## 2.6 Користувацький інтерфейс (UI View)

Для наглядної візуалізації майбутньої системи розробимо набір макетів.

Перше, що треба буде налаштувати виконавцю перед грою – це розкладка піаніно. (рис. 2.4).

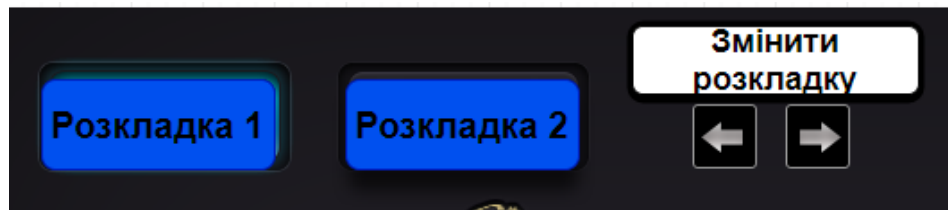


Рисунок 2.4 – Частина панелі, що відповідає за взаємодію із розкладками

Перша розкладка більш компактна та мінімалістична, проте вона містить все необхідне для гри. (рис. 2.5).



Рисунок 2.5 – Перший варіант розкладки

Друга розкладка повністю відтворює реальне піаніно за кількістю клавіш (рис. 2.6)



Рисунок 2.6 – Другий варіант розкладки

Інша частина верхньої панелі (рис. 2.7) відповідатиме за:

- гучність
- моніторинг ноти, що програється

- графічну візуалізацію програваної мелодії

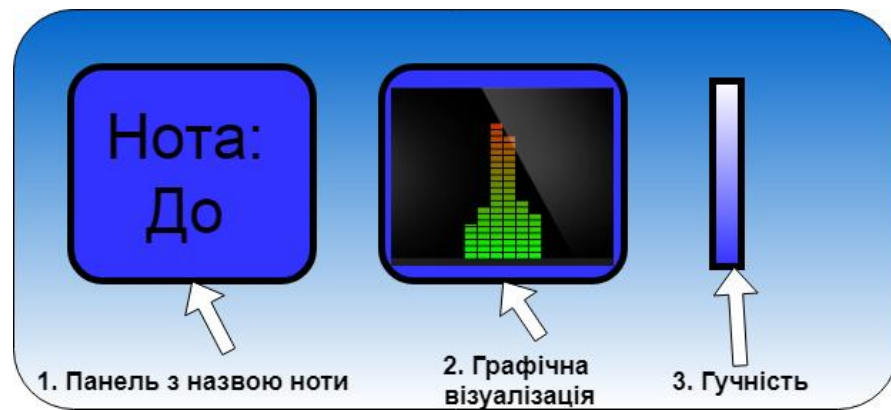


Рисунок 2.7. – Друга частина панелі керування

## 2.7 Логічне уявлення про ІС (Logical View)

Для уявлення майбутньої системи, побудуємо логічну діаграму зв'язку компонентів системи (рис. 2.7). Модель складається з гравця, що першим чином ініціалізує середовище Scratch, а потім ініціалізує проект піаніно. Scratch починає калібрування рухів і після виконання гравцем інструкцій він отримує підтвердження вдалої операції. Потім гравець може грати на піаніно.

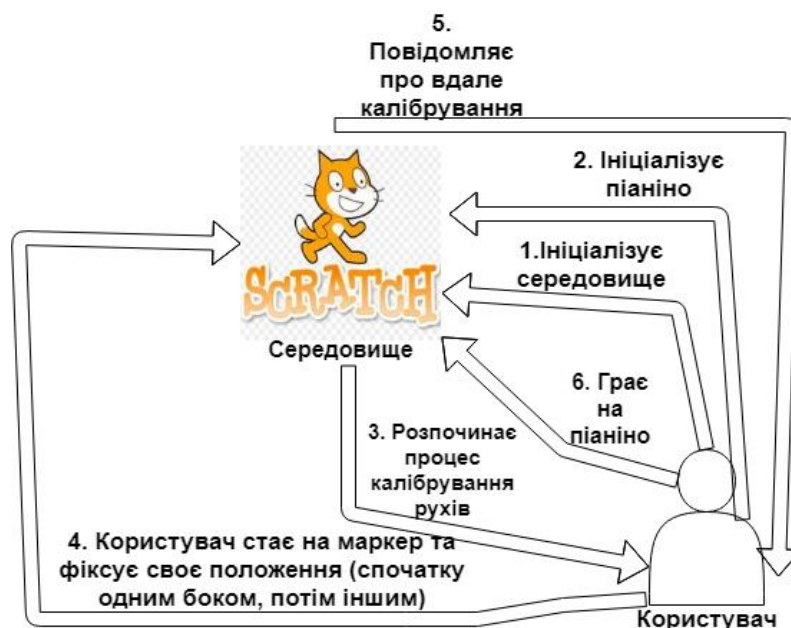


Рисунок 2.8 – Схема логіки роботи системи та зв'язку компонентів

## 2.8 Уявлення розгортання IC (Deployment View)

Робота застосунку насамперед базується на системі розпізнавання Kinect.

Kinect (раніше Project Natal) – безконтактний сенсорний ігровий контролер, спочатку представлений для консолі Xbox 360, і значно пізніше для Xbox One і персональних комп'ютерів під управлінням ОС Windows.

Спочатку ми ініціалізуємо систему розпізнавання Kinect. Вона спочатку повинна охопити фігуру. Потім, щоб запустити віртуальне піаніно, починаємо калібрування рухів користувача у Scratch. Середовище Scratch взаємодіє із системою розпізнавання Kinect через додаток Kinect2Scratch. Коли калібрування рухів завершилося, ми можемо запустити віртуальне піаніно. (рис. 2.8)

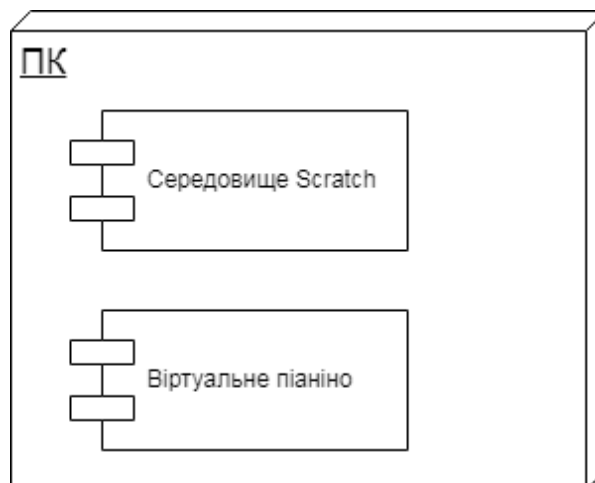


Рисунок 2.9 – Діаграма розгортання (Deployment View)

## 2.9 Уявлення слоїв IC (Design View)

Система складається з наступних рівнів (рис. 2.10):

– Presentation Layer (містить компоненти, які відповідають за функціонування різних сервісів: камери, мікрофона, системи розпізнавання, інфрачервоного датчику тощо);

- Application Layer (містить сервіси, які складають основну бізнес логіку застосування; також виконують додакові функції, такі як збереження даних в пам'яті, запис звуку через мікрофон, тощо).
- Domain Layer (сервіси, ціллю яких є збереження стану системи та бути єдиним джерелом збережених даних для усіх частин застосування).
- Communication Layer (сервіси для взаємодії застосування з зовнішніми компонентами системи, в нашому випадку на цьому рівні є клієнт Kinect2Scratch).
- Data access Layer (сервіси для збереження даних, в нашому випадку це DataStore і SongRepository).

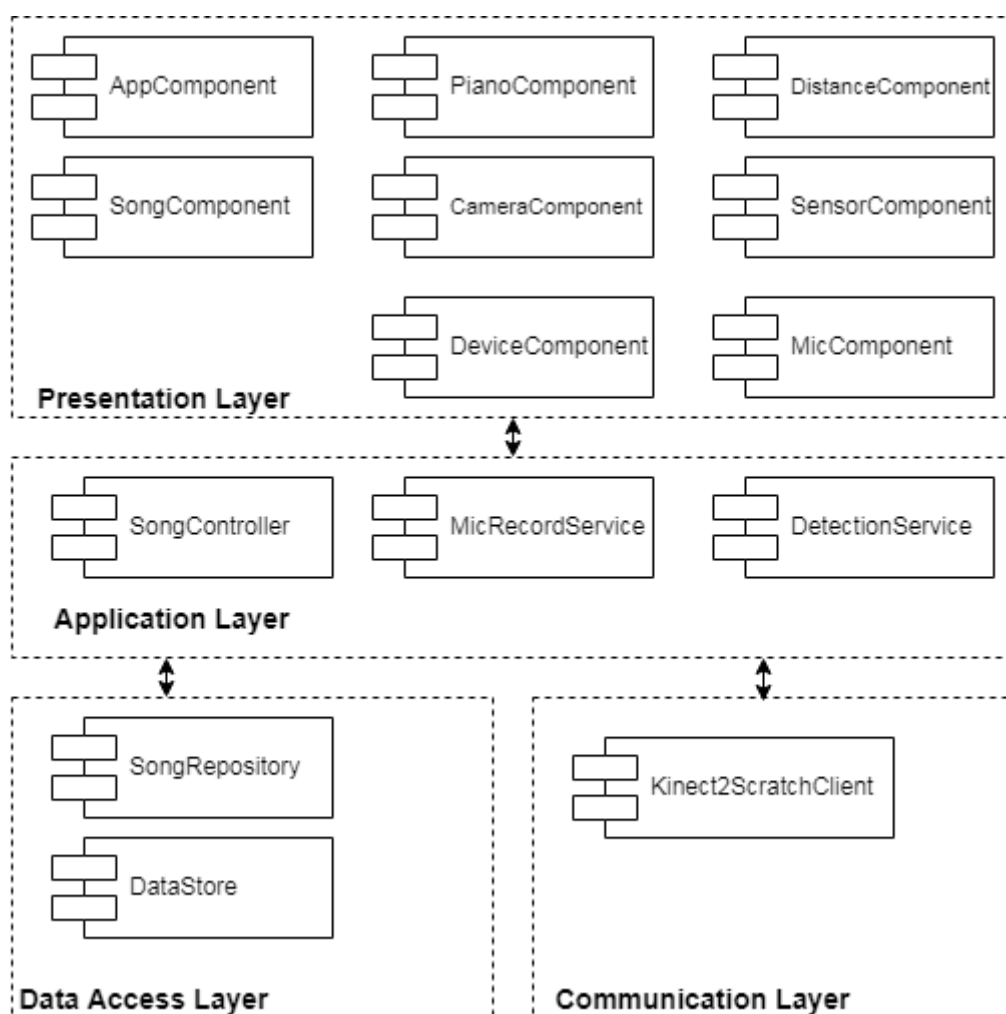


Рисунок 2.10 – Діаграма слоїв IC (Design View)

Проаналізуємо, за що відповідають основні компоненти:

- Appcomponent – компонент для взаємодії із середовищем Scratch.
- MicComponent – компонент для взаємодії із мікрофоном
- DistanceComponent – компонент, що розпізнає стіни і відстань до об'єкта.
- CameraComponent – компонент для взаємодії із камерою.
- SongComponent – компонент для відображення пісень
- PianoComponent – компонент для взаємодії із піаніно.

## 2.10 Уявлення процесів IC (Process View)

Оскільки основною ціллю системи є проведення розпізнавання образів, слід виділити два процеси, один з яких є автономним (без взаємодії з користувачем) та один з яких є інтерактивним (ініціюється користувачем).

Сценарій 1. Проведення розпізнавання користувача.

Серцем сенсора Kinect, є 3D–технологія від компанії PrimeSense, яка використовує структурований світ, інфрачервоні камери і спеціалізований процесор для вимірювання відстані від камери до сцени.

Ці виміри проводяться по всьому полю зору сенсора. В результаті, виходить хмара точок, що складається з 307200 вимірювань відстаней між сенсором і сценою. Точки проєктуються на заданих кутах від лазера. Лазерні точки відбиваються від об'єктів сцени і фіксуються ІК–камерою. Об'єкти сцени, розташовані близько до камери відображаються в точки зображення, які знаходяться близько один до одного, а об'єкти, розташовані далеко від камери – відображаються в точки зображення, які знаходяться далі один від одного. Задля детальної візуалізації процесу побудуємо діаграми активності (рис 2.12) та послідовності (рис. 2.13) IC для сценарію розпізнавання.

					IC КРБ 6.050101 019 ПЗ	31
Змін	Лист	№ докум.	Підпис	Дата		

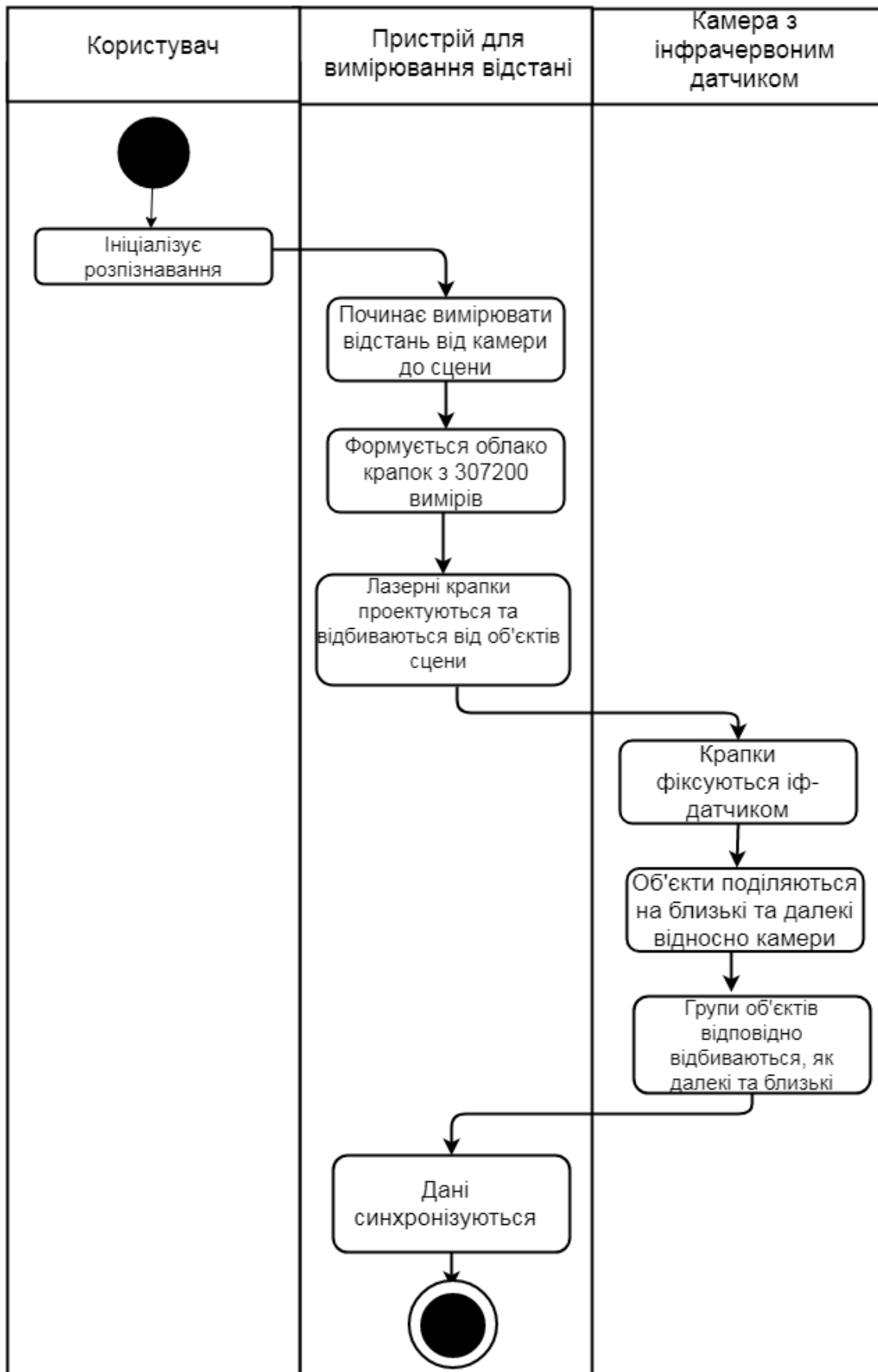


Рисунок 2.11 – Діаграма активності ІС для сценарію розпізнавання



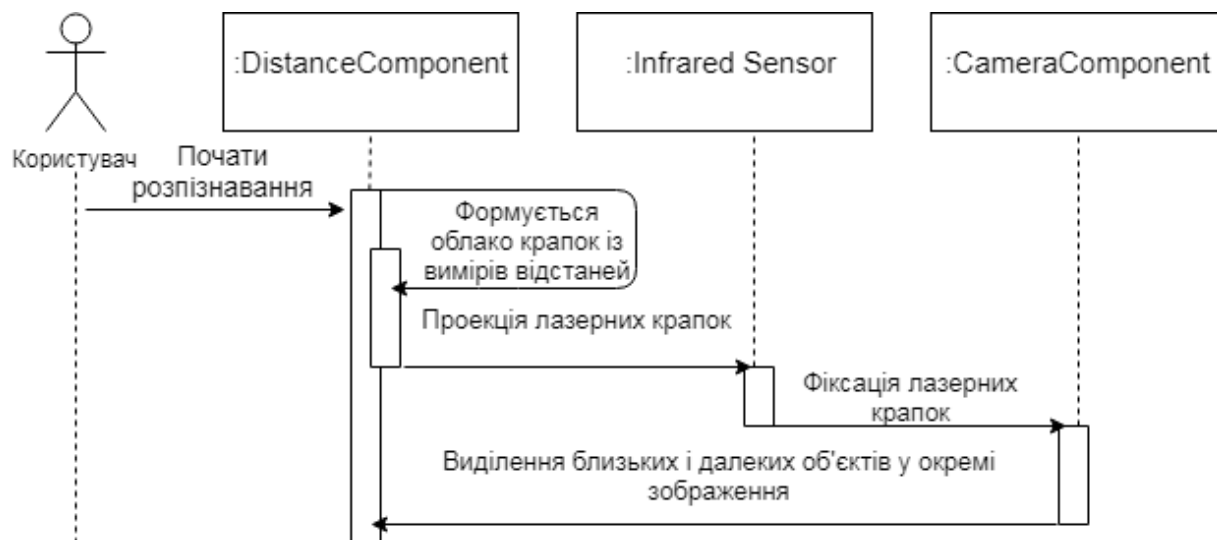


Рисунок 2.12 – Діаграма послідовності ІС для сценарію розпізнавання у відеопотоці

Сценарій 2. Робочий процес взаємодії користувача із віртуальним піаніно.

Спочатку користувач ініціалізує середовище виконання Scratch. Середовище у свою чергу завантажує файли проекту віртуального піаніно. Потім користувач ініціалізує піаніно і воно розпочинає калібрування рухів. Користувач виконує послідовно інструкції з калібрування, фіксуючи своє тіло у певному положенні та наостанок калібрує маркери. Потім користувач натискає кнопку «обрати мелодію» та застосування надає йому список наявних мелодій. Користувач обирає мелодію та застосування підсвічує у певній послідовності наявні у обраній мелодії ноти. Якщо користувач натискає правильну клавішу, застосування підсвічує наступну. Якщо ні, мелодія далі не грає, доки користувач не зіграє правильну клавішу.

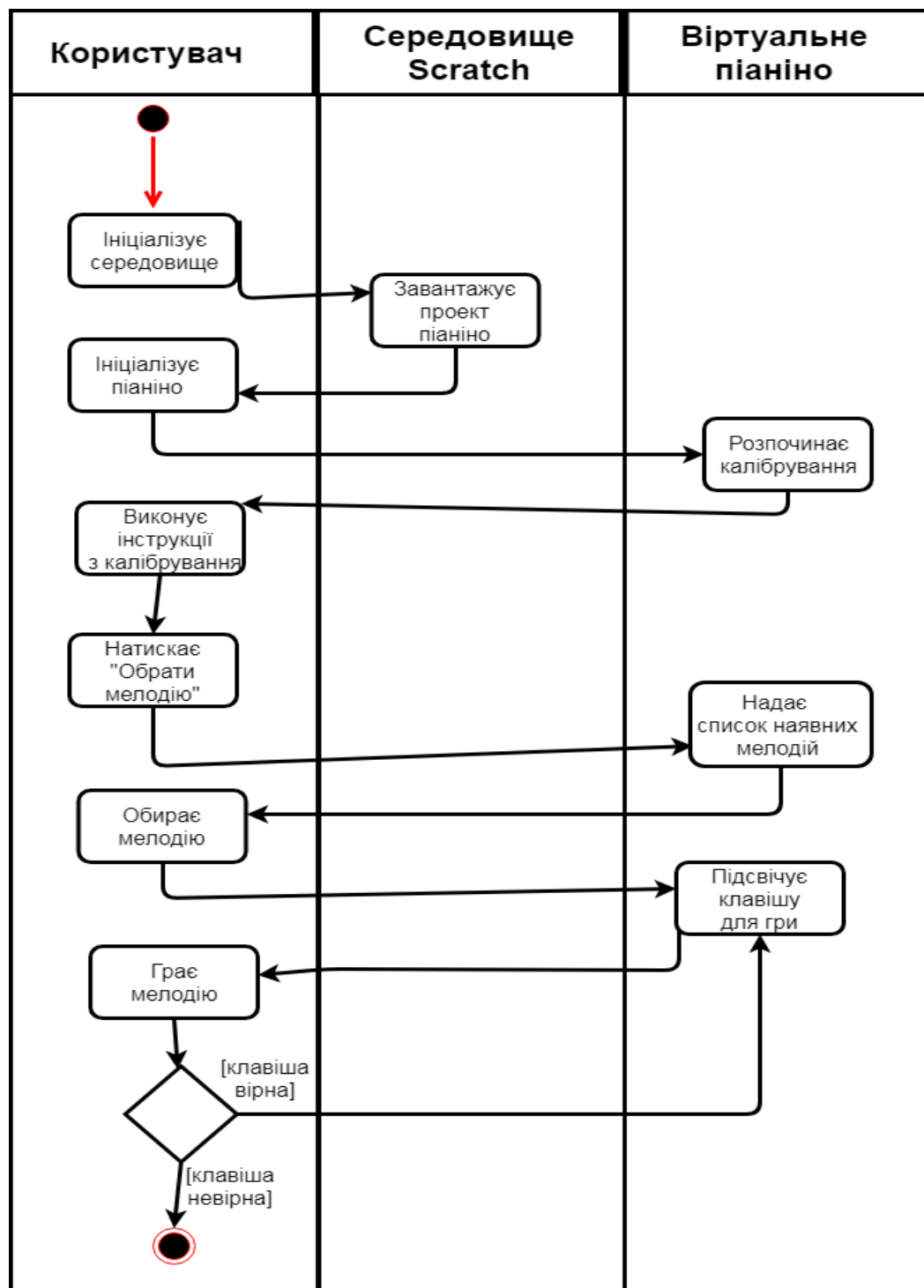


Рисунок 2.13 – Діаграма активності ІС для сценарію взаємодії користувача із віртуальним піаніно

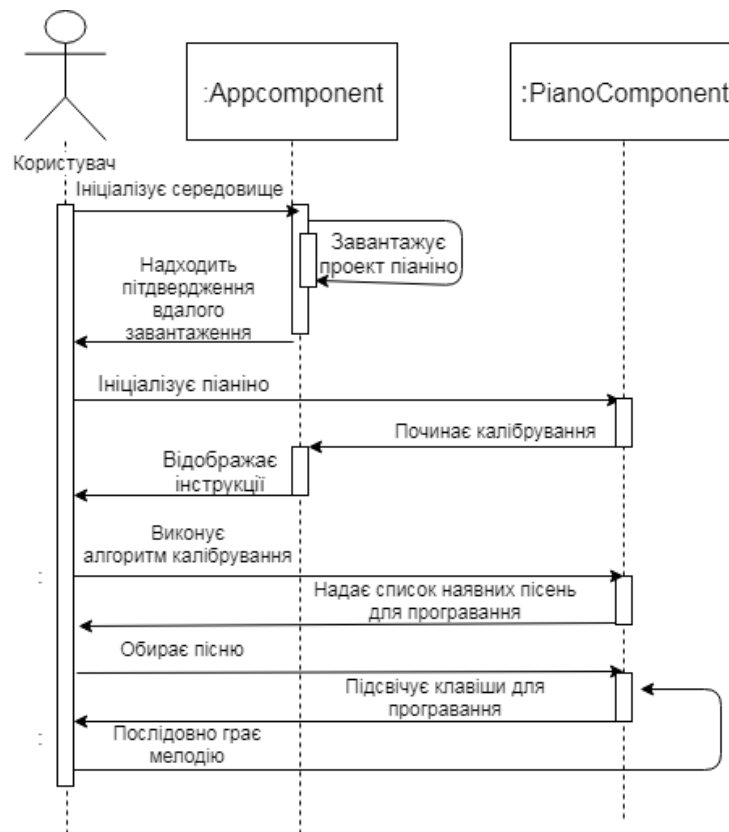


Рисунок 2.14 – Діаграма послідовності ІС для сценарію взаємодії

## 2.11 Уявлення комунікацій ІС (Communication View)

Для розуміння взаємодії компонентів ІС, було зроблено діаграму комунікацій. (рис. 2.15)

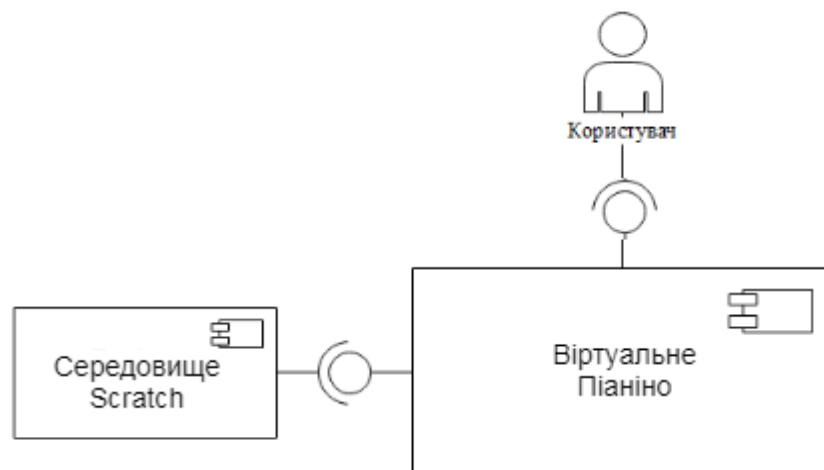


Рисунок 2.15– Уявлення комунікацій

## 2.12 Уявлення безпеки IC (Security View)

Scratch працює через HTTPS, що гарантує, що сторінки дійсно надходять з серверів Scratch і що сторінки дійсно призначені для свого браузера.

Крім того, Scratch використовує техніку "дезактивації запитів до бази даних", яка запобігає доступу до довільних даних зловмисникам. Крім того, більша частина коду Scratch є відкритим кодом, що дозволяє учасникам спільноти вказувати на недоліки, перш ніж вони можуть бути виявлені зловмисниками.

## 2.13 Опис стеку технологій

Нижче наведені найменування технологій, що були використані в цьому проекті та їх опис.

Таблиця 2.3 – стек технологій клієнтської частини

Тип	Назва	Веб-сайт	Призначення
IDE	Scratch 2	<a href="https://scratch.mit.edu/download/scratch2">https://scratch.mit.edu/download/scratch2</a>	Скратч – візуальне середовище програмування, яке було розроблено для некомерційного розповсюдження.
Мова	ScratchJr	<a href="https://scratch.mit.edu">https://scratch.mit.edu</a>	Scratch – візуальна мова програмування, метою якої є створення програми для ЕОМ шляхом маніпулювання графічними об'єктами замість написання її тексту.

### Продовження таблиці 2.3

Тип	Назва	Веб-сайт	Призначення
Пристрій	Kinect	<a href="https://support.xbox.com/en-US/xbox-on-windows/accessories/kinect-for-windows-v2-setup">https://support.xbox.com/en-US/xbox-on-windows/accessories/kinect-for-windows-v2-setup</a>	Kinect - це інтерактивний ігровий безконтактний контролер.
Клієнт	Kinect2Scratchch	<a href="https://stephenhowell.github.io/kinect2scratch/">https://stephenhowell.github.io/kinect2scratch/</a>	Застосування, що виконує функції взаємодії Scratch із коолером Kinect.

#### 2.14 Висновок до другого розділу

У даному розділі було проведено проектування програмного забезпечення віртуального тренажера, де були сформовані функціональні та нефункціональні вимоги, побудована діаграма прецедентів, яка ілюструє ці вимоги та спроектований інтерфейс користувача. Також був описаний стек технологій, що використовуються при розробці віртуального тренажера.

Були побудовані діаграми активності для відображення процесу розпізнавання фігури користувача та для процесу взаємодії користувача із грою. Діаграми послідовності також відтворили ці процеси через компоненти, що взаємодіють із собою. Розпізнавання відбувається за допомогою системи Kinect.

## 3 РОЗРОБКА ВІРТУАЛЬНОГО ТРЕНАЖЕРА НА ОСНОВІ КОНТРОЛЕРА KINECT

### 3.1 Уявлення про структуру проекту ІС

Нижче приведена структура директорії проекту. (рис. 3.1)

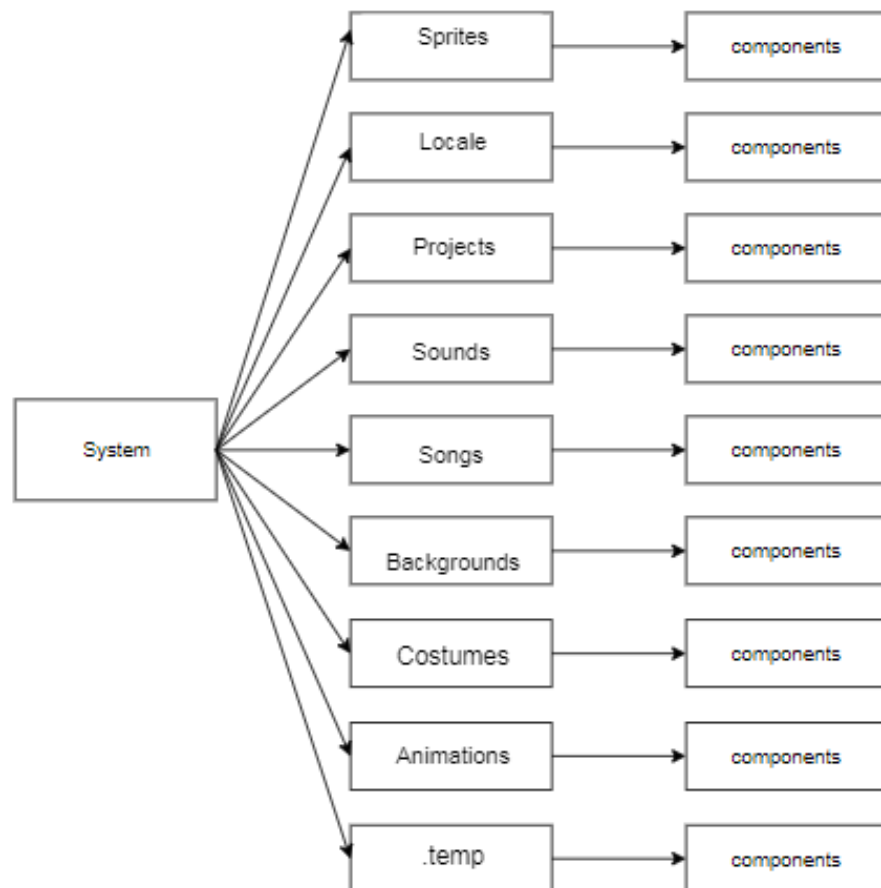


Рисунок 3.1 – Структура проекту

### 3.2 Уявлення про класи ІС

Для кращого розуміння зв'язку між компонентами необхідно представити загальну діаграму класів(рис 3.3).

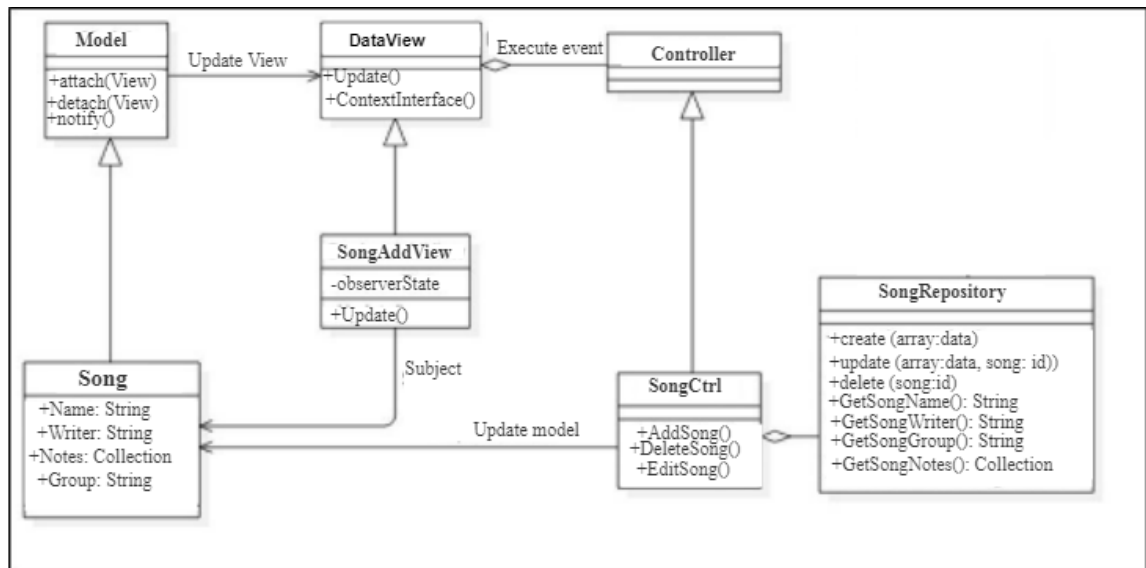


Рисунок 3.2 – Діаграма класів

У класі Song, що відповідає за пісні, містяться наступні атрибути:

- Name – назва пісні
- Writer – той, хто написав пісню.
- Notes – ноти у пісні.
- Group – приналежність пісні до групи за акордами.

У класі SongCtrl, що відповідає за операції над піснями, містяться наступні методи:

- Addsong() – додати пісню.
- DeleteSong() – видалити пісню.
- EditSong() – змінити ноти у пісні.

У класі SongRepository, що відповідає за зберігання пісень, містяться наступні методи:

- GetSongName() – отримати назву пісні.
- GetSongWriter() – отримати ім'я автора пісні.
- GetSongGroup() – отримати дані про приналежність пісні до групи.
- GetSongNotes() – отримати ноти пісні.

### 3.3 Інфраструктурне уявлення IC (Infrastructure View)

Оскільки наразі застосування «Kinect2Scratch» працює лише для Windows через брак драйверів до інших операційних систем, він підтримує наступні ОС:

- Windows 8;
- Windows 8.1;
- Windows 10;

### 3.4 Стратегія доставки продукту (Delivery Strategy View)

Кінцевий продукт буде поширюватися через хмарне сховище «google drive», середовище Scratch для ініціалізації проекту знаходиться у відкритому доступі на офіційному сайті. Інструкцію з користування проектом буде наведено у додатку.

### 3.5 Документація IC

Для застосування було створено керівництво користувача, яке допоможе новим користувачам розібратись у структурі програми

### 3.6 Керування програмним кодом IC

Для керування програмним кодом при розробці програми використовувались системи контролю версій Git. Для детального огляду репозиторію розрахуємо певні метрики, які дадуть бачення роботи(табл. 3.1).

Таблиця 3.1 – Метрики системи контролю версій Git

Метрика	Значення
Кількість комітів у master branch	8
Загальна кількість бранчів	5
Кількість закритих PR	7



### 3.7 Розрахунок метрик програмного коду ІС

Для розрахунку метрик коду представимо їх у відповідній таблиці (табл. 3.2).

Таблиця 3.2 – Метрика програмного коду проекту

Метрика	Значення
Загальні кількість строчок коду в проекті	1054
Середня кількість строчок коду в класі	73,7
Максимальна кількість строчок коду в одному класі	96
Середня кількість строчок коду в одному методі	20
Максимальна кількість строчок коду в одному методі	56
Максимальна глибина дерева наслідування	3
Середня цикломатична складність методів	1,4
Максимальна цикломатична складність методів	4
Відсоток коментарів у коді	20%

### 3.8 Контрольний список з якості реалізації ІС

Було створено контрольний список та відображено у таблиці (табл. 3.3).

Таблиця 3.3 – Контрольний список з якості реалізації ІС

Твердження	Відповідь
Використання Dependency Injection	Так
Використання логування (logging)	Ні
Використання модульного тестування	Ні
Захист від SQL ін'єкцій	Так
Захист від Javascript/XSS ін'єкцій	Ні
Використання валідації даних у всіх полях вводу для користувачьких інтерфейсів	Так
Використання інсталяторів/інтернет магазинів	Ні
Використання засобів синхронізації даних у випадку багатопоточного застосування	Ні
Підтримка глобалізації СППР (інтернаціоналізація, локалізація)	Так
Відсутність зашитих в програмний код конфігураційних параметрів застосування	Ні
Використання UI патернів при розробці UI	Ні
Врахування coding style guide lines для обраного мови програмування	Ні
Врахування рекомендованих guide lines при розробці користувачького інтерфейсу для тієї чи іншої ОС	Ні
Підтримка локалізації та глобалізації СППР	Так

Для проведення функціонального тестування треба розробити спеціальні тест-кейси. Для цього треба створити список з роз'ясненням кожного тест-кейсу.

Список має таку структуру: ідентифікатор – номер функціональної вимоги – опис тесту – передумови – що треба зробити – що повинно відбутись.

Тест кейс №1. FR1.1.1 – перевірка позиції для розпізнавання – зробити декілька кроків назад від пристрою Kinect – перевірити статус розпізнавання – фігура гравця повинна виділитися червоним кольором.

Тест кейс №2. FR1.1.2 – перевірка синхронізації рухів – виконати алгоритм калібрування – зробити декілька кроків – маркери на екрані, що відповідають за ноги гравця, повинні рухатися разом з гравцем.

Тест кейс №3. FR2.1.1 – додавання пісні – зробити текстовий файл у форматі txt у директорії пісень – додати файл у середовищі Scratch – при відкритті списку наявних пісень нова пісня повинна з'явитися.

Тест кейс №4. FR2.1.2 – видалення пісні – видалити текстовий файл із директорії пісень – видалити пісню у меню пісень середовища Scratch – при відкритті списку наявних пісень видаленої вже не буде.

Тест кейс №5. FR2.1.3 – обирання пісні для програвання користувачем – запустити віртуальне піаніно та виконати калібрування рухів – натиснути на кнопку «обрати мелодію» – для програвання підсвітиться перша клавіша.

Тест кейс №6. FR 3.1.1 – програвання довільної мелодії – запустити віртуальне піаніно та провести калібрування рухів – грати на довільних клавішах – має грати музика незалежно від обраних клавіш.

Тест кейс №7. FR 3.1.2 – фіксація гучності – перемістити повзунок із гучністю у середину – грати мелодію до та після фіксації – гучність повинна зменшитися.

### 3.9 Протокол проведення функціонального тестування

Скориставшись планом проведення функціонального тестування у попередньому пункті, виконаємо тестування розробленої системи та проведемо

					ІС КРБ 6.050101 019 ПЗ	42
Змін	Лист	№ докум.	Підпис	Дата		

відповідність між очікуваним результатом та отриманим. Дані, що отримаємо занесемо до таблиці 3.4

Таблиця 3.4 – Функціональне тестування

Номер тест-кейсу	Очікуваний результат	Отриманий результат	Коментарі
Тест кейс № 1	Фігура гравця повинна виділитися червоним кольором, що підтвердить вдалість розпізнавання.	Фігура гравця виділилася червоним кольором.	Виконано
Тест кейс № 2	Маркери, що відповідають за ноги гравця, повинні рухатися синхронно з ним.	Маркери, що відповідають за ноги гравця, рухалися синхронно з ним.	Виконано
Тест кейс № 3	При відкритті списку наявних пісень додана пісня повинна з'явитися.	При відкритті списку наявних пісень додана пісня з'явилася.	Виконано
Тест кейс № 4	При видаленні пісні, вона має зникнути у списку наявних пісень.	При видаленні пісні, вона зникла у списку наявних пісень.	Виконано
Тест кейс № 5	Після натискання кнопки «обрати мелодію» та вибору пісні, підсвітіться перша клавіша для програвання.	Після натискання кнопки «обрати мелодію» та вибору пісні, підсвітилася перша клавіша для програвання.	Виконано
Тест кейс № 6	Має програватися довільна мелодія, незалежно від послідовності натиснутих клавіш.	Програється довільна мелодія, незалежно від послідовності натиснутих клавіш.	Виконано
Тест кейс № 7	Має мінятися гучність, в залежності від позиції повзунка.	Гучність міняється, в залежності від позиції повзунка.	Виконано

### 3.10 Проведення тестування точності

Для проведення тестування точності реакції віртуального тренажера на рухи, була відібрана група з 10 людей. Кожен повинен був раз спробувати зіграти ноту після вдалого калібрування. Результати наведено у таблиці. (табл. 3.5)

Як ми бачимо, точність натискання складає у середньому не менше 80,3%.

Таблиця 3.5 – тестування точності

#	До	Ре	Мі	Фа	Соль	Ля	Сі	#До	#Ре	Точність	Середня точність
1	+	–	+	+	+	+	+	+	–	77%	80,3%
2	+	+	+	+	–	+	+	+	–	77%	
3	+	–	+	+	+	+	+	+	+	88%	
4	+	+	+	+	–	+	–	–	+	66%	
5	+	+	–	+	+	+	+	+	+	88%	
6	–	–	+	+	+	+	+	+	+	77%	
7	+	+	+	–	+	+	+	+	+	88%	
8	+	+	+	+	+	+	+	+	+	100%	
9	+	–	+	–	+	+	+	+	–	66%	
10	+	+	+	–	–	–	+	+	+	66%	

### 3.11 Висновки до третього розділу

У даному розділі були проведені заходи з забезпечення якості та надійності системи. Також були розглянуті питання управління програмним кодом з використанням системи контролю версій Git та її веб-оболонки GitHub. Розраховані метрики програмного коду та оформлений контрольний список по якості реалізації інформаційної системи. Були розроблені тест-кейси, що перевіряли справність усіх вказаних функціональних вимог у віртуальному тренажері Kinect. Далі була зібрана фокус-група з 10 людей, які перевіряли точність розпізнавання рухів. Для цього кожен спробував натиснути всі сім клавіш у довільній послідовності. У функціональних вимогах було заявлено, що точність розпізнавання повинна бути не меншою за 80%, тестування точності підтвердило ці цифри.

## 4 ОХОРОНА ПРАЦІ

Питання охорони праці людини необхідно вирішувати на всіх стадіях трудового процесу незалежно від виду професійної діяльності.

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних, шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

У даному розділі вирішується питання охорони праці системного програміста на стадії розробки ним програмного комплексу класифікації систем.

4.1 Аналіз небезпечних і шкідливих факторів, що впливають на інженера–програміста при розробці даної системи

### 4.1.1. Організація робочого місця інженера–програміста

Приміщення, в якому працює інженер–програміст, має загальну площу 46,8 м.кв., висоту стелі 4 м. У приміщенні знаходиться 8 робочих місць з ПК. Кожне робоче місце обладнане робочим столом площею 1,2 м<sup>2</sup>, стільцем та персональним комп'ютером, що складається з монітора AOC Q3279VWF, системного блоку, клавіатури та миші. Слід відзначити, що площа одного робочого місця оператора ПК не повинна бути меншою за 6м. кв, а об'єм не менший за 20м. куб., тобто площі та об'єму даного приміщення вистачає для розташування 8 робочих місць інженерів–програмістів.

У приміщенні лабораторії знаходиться два вікна розміром 2,3×1,8 м., на які встановлено склопакети фірми KBE Classica, а джерелом штучного освітлення є 6 світильників з дзеркальним растром Light Case 3 x TC–L (36Вт). Лабораторія обладнана кондиціонером Airwell HKD 012–N11/YKD 012–H11.

					ІС КРБ 6.050101 019 ПЗ	45
Змін	Лист	№ докум.	Підпис	Дата		

Аналіз умов праці показує, що у приміщенні лабораторії на інженера–програміста можуть негативно впливати наступні шкідливі та небезпечні чинники: фізичні:

- підвищений рівень «м'якого» рентгенівського випромінювання;
- підвищений рівень електромагнітного випромінювання радіочастотного діапазону;
- підвищений рівень інфрачервоного та ультрафіолетового випромінювання;
- підвищений рівень шуму та вібрації на робочих місцях;
- підвищене значення напруги в електричній мережі, замикання якої може відбутися через тіло людини; підвищений рівень статичної електрики;
- підвищені або понижені температура, відносна вологість та швидкість руху повітря робочої зони; підвищена запиленість і загазованість повітря робочої зони; підвищений або понижений вміст позитивних і негативних аероіонів у повітрі робочої зони;
- недостатня освітленість робочої зони; підвищена яскравість світла; понижена контрастність між об'єктом і фоном; прямий та віддзеркалений відблиск; підвищена пульсація світлового потоку;

хімічні:

- підвищений вміст у повітрі робочої зони озону й аміаку;

психофізіологічні:

- надмірні статичні та динамічні навантаження;
- розумове перенавантаження; перенавантаження аналізаторів;
- монотонність праці; надмірні емоційні навантаження;
- нерациональна організація робочого місця;

біологічні:

- підвищений вміст мікроорганізмів у повітрі робочої зони.

Кожний із наведених чинників може зумовити небажані наслідки впливу ВДТ, ЕОМ, ПЕОМ на здоров'я користувачів, а їх сумісна дія підсилити цей вплив.

					ІС КРБ 6.050101 019 ПЗ	46
Змін	Лист	№ докум.	Підпис	Дата		

#### 4.1.2 Мікроклімат робочої зони інженера–програміста

Робота інженера–програміста за енерговитратами відноситься до категорії легких робіт Іа, Іб, тому повинні дотримуватися наступні вимоги згідно ДСН 3.3.6.042–99: оптимальна температура повітря – 220С (допустима – 20–240С), оптимальна відносна вологість – 40–60% (допустима – не більш 75%), швидкість руху повітря не більш 0,1 м/с.

Виміряні за допомогою приладів (психрометр Августа) температура та вологість у лабораторії відповідають вказаним у таблиці для теплого періоду року.

Розташовані у приміщенні 8 ПК являються джерелами тепловиділень, крім того для підтримання у приміщенні в холодний період року оптимальних параметрів мікроклімату використовуються нагріті поверхні опалювальної системи. Нормованим показником ІЧВ являється гранично допустима густина потоку енергії  $I_{г.д}$ , Вт/м<sup>2</sup>, яка встановлюється в залежності від площі опромінюваної поверхні тіла людини ( $S_{опр}$ ). Нормовані рівні складають:

- $I_{г.д} = 35$  Вт/м<sup>2</sup> при  $S_{опр} > 50\%$ ;
- $I_{г.д} = 70$  Вт/м<sup>2</sup> при  $S_{опр} \sim 25\text{--}50\%$ ;
- $I_{г.д} = 100$  Вт/м<sup>2</sup> при  $S_{опр} < 25\%$

#### 4.1.3 Освітлення робочого місця

Нормованим параметром природного освітлення згідно ДБН В.2.5–28 – 2006 являється коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт. Для штучного освітлення нормованим параметром виступає  $E_{мін}$  – мінімальний рівень освітленості, та  $K_p$  – коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж 20%. Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300–500 лк.

Причинами виникнення небезпечних і шкідливих факторів є:

- відсутність чи недостатність природного освітлення;

					ІС КРБ 6.050101 019 ПЗ	47
Змін	Лист	№ докум.	Підпис	Дата		

- недостатня освітленість робочої зони;
- прямий чи відбитий блиск.

Причини виникнення вищевказаних факторів:

- відсутність чи недостатні розміри вікон;
- відсутність чи недостатня кількість світильників;
- недоліки освітлення і/або неправильне розташування дисплея.

Найменший об'єкт розрізнення залежить від класу монітора. У нашому випадку використовується монітор AOC Q3279VWF з найменшим розміром об'єкта розрізнення 0,273 мм. [11]

Технічні характеристики монітора AOC Q3279VWF: тип – ЖК–монітор, широкоформатний; діагональ – 31.5"; дозвіл – 2560x1440 (16:9); тип ЖК–матриці – TFT MVA; яскравість – 250 кд/м.кв.; максимальна кількість кольорів – більше 1 млрд.; кольори – 8 біт + FRC = 10 біт; підсвічування – WLED; підсвічування без мерехтіння (Flicker–Free); час відгуку – 5 мс. GtG; область огляду – по горизонталі –178°; по вертикалі – 178°; контрастність (динамічна контрастність) 3000: 1 (20M: 1); крок точки по горизонталі – 0.273 мм.; мін. частота рядків – 30 кГц.; максимальна частота оновлення кадрів: 75 Гц.; покриття sRGB – 98%; енергоспоживання – клас С. Відповідає стандартам: DDC 2B; Plug&Play; TCO'03 (стандарт безпеки); MPR–II (стандарт безпеки).

Найменшим об'єктом розрізнення на екрані монітора є крапка розміром 0,27×0,27 мм. Фон, з яким працює користувач, має чорний або білий, або синій колір, а колір символів коливається від чисто білого до чорного, включаючи всі можливі кольори, однак вибір залежить від працюючого на ЕОМ. Контраст і чіткість зображення можна регулювати, не втручаючись в програму, для чого необхідно скористатися регулюванням на моніторі.

Нормування природного освітлення (табл. 4.1) проводиться згідно з ДБН В.2.5–28–2006 «Природне та штучне освітлення».

					ІС КРБ 6.050101 019 ПЗ	48
Змін	Лист	№ докум.	Підпис	Дата		



Таблиця 4.1. – Нормування природного освітлення

Найменший розмір об'єкта роздільної здатності	Розряд зорової роботи	Коефіцієнт природного висвітлення при бічному освітленні
0,26×0,25	5	0,6

Примітка: природне освітлення нормується в залежності від контрасту об'єкта з фоном і характеристики фону.

Нормування штучного освітлення (табл.4.2.) здійснюється згідно ДБН В.2.5–28–2006 «Природне та штучне освітлення».

Таблиця 4.2. – Нормування штучного освітлення

Найменший розмір об'єкта роздільної здатності	Розряд зорової роботи	Підрозряд	Контраст із фоном	Фон	Штучне висвітлення при комбінованому висвітленні
0,26 – 0,25	5	а	малий	темний	300

#### 4.1.4 Вплив шуму на інженера–програміста

Як було вказано вище, в лабораторії знаходиться вісім робочих місць з ПК, кожне з яких устатковане монітором, вінчестером в системному блоці, трьома вентиляторами системи охолодження ПК та клавіатурою. Крім того працює периферійна техніка. Таким чином у приміщенні мають місце шуми механічного і аеродинамічного походження, широкосмугові із аперіодичним підсиленням при роботі принтерів. Допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА.

Відповідно до документації на використовувані периферійні пристрої, рівень шуму не перевищує 45 дБ, що відповідає нормам, тому спеціальних додаткових заходів по захисту від шуму застосовувати не потрібно.

Нормування визначається згідно ГОСТ 12.1.003–83 «Шум. Общие требования безопасности».

#### 4.1.5 Виробничі випромінювання

Допустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітору комп'ютера представлені в таблиці 4.3. Нормованим параметром невикористаного рентгенівського випромінювання виступає потужність експозиційної дози. Максимальний рівень рентгенівського випромінювання на робочому місці програміста зазвичай не перевищує 20 мкР/год.

Таблиця 4.3 Допустимі значення параметрів неіонізуючих електромагнітних випромінювань

Найменування параметра	Допустимі значення
Напруженість електричної складової електромагнітного поля на відстані 50 см від поверхні відеомонітора	10 В/м
Напруженість магнітної складової електромагнітного поля на відстані 50 см від поверхні відеомонітора	0,3 А/м
Напруженість електростатичного поля не повинна перевищувати: для дорослих користувачів для дітей дошкільних установ і що вчаться середніх спеціальних і вищих учбових закладів	20кВ/м 15кВ/м

На відстані 5–10 см від екрана і корпусу монітора рівні напруженості можуть досягати 140 В/м по електричній складовій, що значно перевищує допустимі значення.

#### 4.1.6 Електробезпека. Статична електрика.

Приміщення лабораторії за небезпекою ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, без пилу, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

Основні причини ураження людини електричним струмом на робочому місці:

- дотик до металевих неструмоведучих частин (корпусу, периферії комп'ютера), що можуть виявитися під напругою в результаті ушкодження ізоляції;
- нерегламентоване використання електричних приладів;
- відсутність інструктажу співробітників з правил електробезпеки.

На протязі роботи на корпусі комп'ютера накопичується статична електрика. На відстані 5–10 см від екрана напруженість електростатичного поля складає 60–280 кВ/м, тобто в 10 разів перевищує норму 20 кВ/м. [12]

#### 4.1.7 Важкість та напруженість праці

Оцінка напруженості праці здійснювалась на підставі обліку всіх наявних значущих показників, які можуть перевищувати нормативні рівні .

Розподіл функцій за ступенем складності завдання – належить до класу 2 (обробка, виконання завдання та його перевірка). Характер виконуваної роботи – належить до класу 2 (робота за встановленим графіком з можливим його коректуванням у ході діяльності). Навантаження на зоровий аналізатор (при відстані від очей працюючого до об'єкта розрізнення не більше 0,5 м), при тривалості зосередженого спостереження (% часу зміни) – належить до класу 2 (5,0–1,1 мм більше 50 % часу; 1,0–0,3 мм до 50 % часу; менше 0,3 мм до 25 %).

Спостереження за екранами відеотерміналів (годин на зміну) – належить до класу 3.2 (більше 4 годин). Монотонність праці. Кількість елементів (приймів, необхідних для реалізації простого завдання або в операціях, які повторюються багаторазово) – належить до класу 3.1 (5–3 прийоми). Режим праці (фактична тривалість робочого дня (год.) – належить до класу 1 (6–7 годин). Наявність регламентованих перерв та їх тривалість – належить до класу 2 (перерви регламентовані, недостатньої тривалості: від 3 % до 7 % часу зміни). Отже робоче місце за показниками напруженості трудового процесу відноситься до класу 3.1 – Шкідливий (напружена праця).

					ІС КРБ 6.050101 019 ПЗ	51
Змін	Лист	№ докум.	Підпис	Дата		

Важкість праці. Оцінка важкості праці здійснюється на підставі обліку всіх наявних значущих показників. При цьому спочатку встановлюється клас кожного із вимірюваних показників, а кінцева оцінка важкості праці встановлюється за показником, який має найвищий ступінь важкості.

Стереотипні робочі рухи (кількість за зміну): при локальному навантаженні (за участю м'язів кистей та пальців рук) – належить до класу 1 (до 20000). При загальному навантаженні (при роботі з переважною участю м'язів рук та плечового поясу) – належить до класу 1 (до 10000). Робоча поза – належить до класу 2 (періодичне перебування в незручній позі (робота з поворотом) тулуба, незручним розташуванням кінцівок та/або фіксованій позі (неможливість зміни взаєморозташування різних частин тіла відносно одна одної) до 25 % часу зміни.) Нахили корпуса (вимушені, більше 30), кількість за зміну – належить до класу 1 (до 50). Переміщення у просторі (переходи, обумовлені технологічним процесом протягом зміни), км: по горизонталі – належить до класу 1 (до 4). По вертикалі – належить до класу 1 (до 2). Отже робоче місце за показниками важкості трудового процесу відноситься до класу 2 – Допустимий, середнє фізичне навантаження.

## 4.2 Розробка заходів з охорони праці

### 4.2.1 Ергономіка та організація робочого місця

Після проведення аналізу робочого місця інженера–програміста в лабораторії було з'ясовано, що воно відповідає встановленим вимогам. Але, в результаті аналізу були виявлені порушення в організації безпосередньо самого робочого місця програміста. У зв'язку з цим пропонується організувати робоче місце програміста наступним способом:

– висота над рівнем підлоги робочої поверхні, на якій працює програміст, повинна складати 720 мм. Бажано, щоб робочий стіл при необхідності можна було регулювати по висоті в межах 680–780 мм;

					ІС КРБ 6.050101 019 ПЗ	52
Змін	Лист	№ докум.	Підпис	Дата		

– оптимальний розмір поверхні столу 1600 x 1000 мм<sup>2</sup>. Під столом повинен бути простір для ніг з розмірами по глибині 650 мм. Робочий стіл оператора повинен також мати підставку для ніг, розташовану під кутом 15 град. до поверхні столу. Довжина підставки – 400 мм, ширина – 350 мм. Відстань клавіатури від краю столу повинна бути не більш 300 мм, що забезпечить програмісту зручну опору для передпліч. Відстань між очима й екраном монітору повинне складати 40–80 см;

– робочий стілець інженера–програміста повинен бути оснащений підйомно–поворотним механізмом. Висота сидіння повинна регулюватися в межах 400–500 мм. Глибина сидіння повинна складати не менш 380 мм, а ширина – не менш 400 мм. Висота опорної поверхні спинки не менш 300 мм, ширина – не менш 380 мм. Кут нахилу спинки стільця до площини сидіння повинен змінюватися в межах 90 – 1100.

Виходячи з результатів аналізу важкості та напруженості праці пропоную скоротити час роботи за комп'ютером, робити перерви сумарний час яких повинен складати 50 хвилин при 8–ми годинній зміні.

#### 4.2.2 Нормалізація повітря робочої зони

Для створення й автоматичної підтримки в лабораторії незалежно від зовнішніх умов оптимальних значень температури, вологості, чистоти і швидкості руху повітря, у холодний час року використовується водяне опалення, у теплий час року застосовується кондиціонування повітря.

Приміщення обладнано кондиціонером (спліт система) Airwell HKD 012–N11/YKD 012–N11, що має наступні основні технічні характеристики:

- режим роботи – охолодження/обігрів;
- рекомендована площа приміщення – 35м<sup>2</sup>;
- витрата повітря – 350/500 м.куб. / год.
- рівень шуму внутрішнього блоку, Дб. – min / max 32/37;
- потужність, що споживається – 1 кВт.
- фотокаталітичний і антибактеріальний фільтри;

					ІС КРБ 6.050101 019 ПЗ	53
Змін	Лист	№ докум.	Підпис	Дата		

- функція «I Feel» з точним контролем температури приміщення.

Приміщення обладнане склопакетами KBE Classica, які приймають значну участь в регуляції мікрокліматичних показників. Характеристики склопакетів вказано нижче:

- Кількість камер – 6 камер;
- Товщина зовнішньої стінки – 2,8 мм.;
- Монтажна глибина – 70 мм.;
- Коефіцієнт опору теплопередачі – 0,87 м.кв. °C / Вт.;
- Можливість установки склопакетів – до 38 мм.;
- стійкість до морозів – до -60°C;

#### 4.2.3 Виробниче освітлення

Під час аналізу освітлення на робочому місті інженера–програміста було встановлено, що воно відповідає встановленим нормам. Для підтримки запроектованого освітлення у чистому виді необхідно скласти графік, де передбачити очищення віконних блоків і світильників не менше 2 разів на рік.

Інженерні (технічні) рішення з охорони праці

Необхідно обрати систему природного освітлення та розрахувати її параметри, в приміщенні лабораторії, яке має розміри: довжина 8,5м., ширина 5,5м., висота 4м.

Вікна лабораторії виходять на західну сторону, використані дерев'яні спарені рами для вікон житлових, громадських і допоміжних будівель з віконним листовим одинарним склом. Вікна обладнані регулюючі жалюзіями, що регулюються та складаються, а також шторами (міжскляні, внутрішні, зовнішні) які виготовлені з подвійними дерев'яними рамами, в яких уставлене віконне скло. [13]

Розрахунок природного освітлення виконують за методикою ДБН В.2.5–28–2006 «Природне і штучне освітлення». В задачу розрахунку входять визначення площі світлових отворів, які повинні забезпечувати належну штучну освітленість. Площа отворів для забезпечення бічної природної освітленості знаходять за

					ІС КРБ 6.050101 019 ПЗ	54
Змін	Лист	№ докум.	Підпис	Дата		

формулою:

$$S_0 = \frac{S_n \cdot E_n \cdot k_3 \cdot \eta_0}{100 \cdot t_0 \cdot r_1} \cdot k_{3\partial} \quad (4.1)$$

де  $S_n$  – площа підлоги приміщення ;

$E_n$  – нормоване значення коефіцієнта природної освітленості;

$k_3$  – коефіцієнт запасу, який приймаємо за 1,5;

$k_{3\partial}$  – коефіцієнт, який враховує затінення сусідніми спорудам;

$\eta_0$  – світлові характеристики бічних отворів;

$t_0$  – загальний коефіцієнт світло пропускання;

$r_1$  – коефіцієнт, який враховує відображення світла при бічній і верхній освітленості, приймаємо за 1,8.

Визначимо спочатку необхідні для розрахунку значення.

Нормоване значення КПО знайдемо, скориставшись формулою:

$$E_n = e_n \cdot m_n \quad (4.2)$$

де  $e_n$  – значення КПО, значення якого приймаємо за 1,5;

$m_n$  – коефіцієнт світлового клімату, значення якого приймаємо за 0,8;

$n$  – номер групи забезпеченості природним світлом V.

$$E_n = 1,5 \cdot 0,8 = 1,2$$

Загальний коефіцієнт світло пропускання знайдемо за формулою:

$$t_0 = t_1 \cdot t_2 \cdot t_3 \cdot t_4 \cdot t_5 \quad (4.3)$$

де  $t_1$  – коефіцієнт світлопропускання матеріалу;

$t_2$  – коефіцієнт, який враховує втрати світла в переплетіннях віконних отворів;

$t_3$  – коефіцієнт, який враховує втрати світла в несучих конструкціях;

$t_4$  – коефіцієнт, який враховує втрати світла в сонцезахисних пристроях;

$t_5$  – коефіцієнт, який враховує втрати світла в захисній сітці.

За заданими параметрами вікон у лабораторії знайдемо значення коефіцієнтів  $t_1, t_2, t_3, t_4, t_5$  в таблицях, та визначимо загальний коефіцієнт світлопропускання вікон:

					ІС КРБ 6.050101 019 ПЗ	55
Змін	Лист	№ докум.	Підпис	Дата		

$$t_1 = 0,9;$$

$$t_2 = 0,75;$$

$$t_3 = 1;$$

$$t_4 = 1;$$

$$t_5 = 0,9.$$

Отже отримаємо:

$$t_0 = 0,9 \cdot 0,75 \cdot 1 \cdot 1 \cdot 0,9 = 0,6$$

Знаходимо площу підлоги

$$S_n = 8,5 \cdot 5,5 = 46,75 \text{ м}^2$$

Отримуємо необхідну площу світлових отворів:

$$S_0 = \frac{46,75 \cdot 1,2 \cdot 1,5 \cdot 10,5}{100 \cdot 0,6 \cdot 1,8} \cdot 1 = 8,2 \text{ м}^2$$

Кількість вікон визначаємо за формулою :

$$n = \frac{S_o}{S} \quad (4.4)$$

де  $S$  – стандартний розмір вікна.

За стандартний розмір вікна взято

$$S = 2,3 \cdot 1,8 = 4,1 \text{ м}^2$$

Тому кількість вікон

$$m = \frac{8,2}{4,1} = 2$$

#### 4.2.4 Захист від виробничого шуму

Як міри по зниженню шуму можна запропонувати:

- облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6–8 дБ);
- екранування робочого місця (постановкою перегородок, діафрагм);



- установка в комп'ютерних приміщеннях устаткування, що робить
- мінімальний шум;
- раціональне планування приміщення.

Для зменшення шуму в аналізованій лабораторії пропоную використовувати замість матричного принтера, що створює багато шуму, більш тихий – лазерний принтер.

#### 4.2.5. Захист від електромагнітних полів

Для попередження впровадження небезпечної техніки всі дисплеї повинні бути сертифіковані.

#### 4.2.6 Електробезпека

Електробезпечність у приміщенні лабораторії пропоную забезпечити наступними технічними способами і засобами захисту:

- для зменшення накопичення статичної електрики застосовувати зволожувачі і нейтралізатори, антистатичне покриття підлоги;
  - забезпечити приєднання металевих корпусів устаткування до жили, що заземлює.
  - заземлення корпусу ПК забезпечити підведенням жили, що заземлює, до розеток. Опір заземлення 4 Ом, згідно (ПУЕ) для електроустановок з напругою до 1000 В.;
  - для захисту приладів від перенавантажень та коротких замикань використовується запобіжник;
- а також організаційними заходами:
- своєчасне проведення інструктажів з охорони праці;
  - заборонення використання непередбачених у лабораторії електричних приладів, таких як електричні чайники, обігрівачі.

					ІС КРБ 6.050101 019 ПЗ	57
Змін	Лист	№ докум.	Підпис	Дата		

#### 4.3. Пожежна безпека

Ступінь вогнестійкості будинків приймається в залежності від їхнього призначення, категорії по вибухопожежній і пожежній небезпеці, по поверховості, площі поверху в межах пожежного відсіку згідно ДСТУ Б В.1.1–36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою».

Будинок, у якому знаходиться лабораторія по пожежній небезпеці будівельних конструкцій відноситься до категорії Д (зниженопожежонебезпечні), оскільки тут присутні займисті (книги, документи, меблі, оргтехніка і т.д.) і легкогорючі речовини (сейфи, різне устаткування і т.д.), що при взаємодії з вогнем можуть горіти без вибуху.

По конструктивних характеристиках будинок можна віднести до будинків з несучими і огорожуючими конструкціями із природних або штучних кам'яних матеріалів, бетону або залізобетону, де для перекриттів допускається використання дерев'яних конструкцій, захищених штукатуркою або важкогорючими листовими, а також плитними матеріалами.

Отже, ступінь вогнестійкості будинку можна визначити як третю (III).

Приміщення лабораторії по функціональній пожежній небезпеці відноситься до класу Ф4.2 – будівлі освітніх установ вищої професійної освіти і додаткової професійної освіти (підвищення кваліфікації) фахівців.

##### 4.3.1 Причини виникнення пожежі

Пожежа в лабораторії, може привести до дуже несприятливих наслідків (втрати кошовної інформації, псування майна, загибель людей і т.д.), тому необхідно: виявити й усунути всі причини виникнення пожежі; розробити план заходів для ліквідації пожежі в будинку; план евакуації людей з будинку.

Причинами виникнення пожежі можуть бути:

					ІС КРБ 6.050101 019 ПЗ	58
Змін	Лист	№ докум.	Підпис	Дата		

- несправності електропроводки, розеток і вимикачів які можуть привести до короткого замикання або пробою ізоляції;
- використання ушкоджених (несправних) електроприладів;
- використання в приміщенні електронагрівальних приладів з відкритими нагрівальними елементами;
- виникнення пожежі внаслідок влучення блискавки в будинок;
- загоряння будинку внаслідок зовнішніх впливів;
- неакуратне поводження з вогнем і недотримання мір пожежної безпеки.

#### 4.3.2 Профілактика пожежі

Для профілактики пожежі надзвичайно важлива правильна оцінка пожежонебезпеки будинку, визначення небезпечних факторів і обґрунтування способів і засобів пожежопередження і захисту.

Одне з умов забезпечення пожежобезпеки – ліквідація можливих джерел запалення.

У лабораторії джерелами запалення можуть бути:

- несправне електроустаткування, несправності в електропроводці, електричних розетках і вимикачах. Для виключення виникнення пожежі з цих причин необхідно вчасно виявляти й усувати несправності, проводити плановий огляд і вчасно усувати всі несправності;
- несправні електроприлади. Необхідні міри для виключення пожежі містять у собі своєчасний ремонт електроприладів, якісне виправлення поломок, не використання несправних електроприладів;
- обігрівання приміщення електронагрівальними приладами з відкритими нагрівальними елементами. Відкриті нагрівальні поверхні можуть спричинити пожежу, тому що в приміщенні знаходяться паперові документи і довідкова література у виді книг, посібників, а папір – легкозаймистий предмет;

– влучення в будинок блискавки. У літній період під час грози можливе влучення блискавки внаслідок чого можливий пожежа. Щоб уникнути цього я рекомендую установити на даху будинку блискавковідвід;

– недотримання мір пожежної безпеки і паління в приміщенні також може спричинити пожежу.

З метою запобігання пожежі пропоную проводити з інженерами–програмістами, що працюють у лабораторії, протипожежний інструктаж, на якому ознайомити працівників із правилами протипожежної безпеки, а також навчити використанню первинних засобів пожежогасіння.

У випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефоні пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації, приведеному на малюнку 1 і приступити до ліквідації пожежі вогнегасниками.



Рисунок 4.1 – План евакуації з приміщення лабораторії

#### 4.4 Висновки до четвертого розділу

В результаті роботи були досліджені основні питання з охорони праці на підприємстві. Був проаналізований стан робочого місця програміста та всього кабінету в цілому за параметрами освітлення, мікроклімату пожежної та

електробезпеки. Після аналізу біли запропоновані рішення по оптимізації робочої зони для безпечної повсякденної роботи.

Також було вирішено індивідуальне завдання з охорони праці. Були проаналізовані потенційно шкідливі та небезпечні виробничі чинники, що можуть мати місце в процесі роботи інженера–програміста. Також були визначені вимоги санітарних норм до гігієнічних умов праці, наведено розрахунок системи природної вентиляції приміщення, де працює інженер–програміст.

У розділі обґрунтовані та визначені заходи по створенню безпечних умов праці під час роботи програміста, а також розглянуті правила пожежної безпеки випромінювання ЕОМ, дотримання яких є життєво важливим для користувача ЕОМ і правила поведінки при аварійних режимах роботи ЕОМ.

					ІС КРБ 6.050101 019 ПЗ	
Змін	Лист	№ докум.	Підпис	Дата		61

## ВИСНОВКИ

В процесі виконання дипломної роботи був розроблений віртуальний тренажер на основі контролера Kinect. А саме віртуальне піаніно. Грі на піаніно навчитися за тиждень, місяць або рік неможливо. Тим більше це завдання стає нездійсненним, якщо немає можливості займатися музикою, наприклад, через відсутність в будинку відповідного музичного інструменту. Але з нинішніми гаджетами і новинками неможливе стає цілком реальним. Віртуальні піаніно – ось відмінне рішення для тих, хто давно хотів спробувати грати улюблені мелодії, але не робив це з різних вагомих причин.

Проаналізувавши існуючі аналоги, було вирішено спроектувати та реалізувати систему у вигляді десктоп-застосування. Розглянувши технології для розробки даного типу системи, було обрано: середовище Scratch для розробки тренажера, бібліотеку Kinect2Scratch, що виконує функцію взаємодії середовища із тренажером Kinect.

Було проведено проектування програмного забезпечення віртуального тренажера, де були сформовані функціональні та нефункціональні вимоги, побудована діаграма прецедентів, яка ілюструє ці вимоги та спроектований інтерфейс користувача.

Були побудовані діаграми активності для відображення процесу розпізнавання фігури користувача та для процесу взаємодії користувача із грою. Діаграми послідовності також відтворили ці процеси за допомогою компонентів. Розпізнавання відбувається за допомогою системи Kinect.

Також були розглянуті питання операційного представлення та описаний стек технологій, що задіяний у розробці даної інформаційної системи.

Були проведені заходи з забезпечення якості та надійності системи. Також були розглянуті питання управління програмним кодом з використанням системи контролю версій Git та її веб-оболонки GitHub. Розраховані метрики програмного коду та оформлений контрольний список по якості реалізації інформаційної системи. Були розроблені тест-кейси, що перевіряли справність усіх вказаних

					ІС КРБ 6.050101 019 ПЗ	62
Змін	Лист	№ докум.	Підпис	Дата		

функціональних вимог у віртуальному тренажері Kinect. Далі була зібрана фокус-група з 10 людей, які перевіряли точність розпізнавання рухів. Для цього кожен спробував натиснути всі сім клавіш у довільній послідовності. У функціональних вимогах було заявлено, що точність розпізнавання повинна бути не меншою за 80%, тестування точності підтвердило ці цифри.

Також були розглянуті питання охорони праці програміста на стадії розробки ним програмно-апаратного комплексу. Були вирішені питання, що пов'язані з розробкою заходів та обиранням засобів, які забезпечують повну безпеку при роботі з технікою, обладнанням, які також забезпечують здорову атмосферу робочого приміщення, зокрема питання оптимізації параметрів мікроклімату у приміщенні з ПК та вибір кондиціонера для забезпечення необхідних кліматичних умов.

					ІС КРБ 6.050101 019 ПЗ	
Змін	Лист	№ докум.	Підпис	Дата		63

## ПЕРЕЛІК ПОСИЛАНЬ

1. Сайт застосування EveryonePiano [Електронний ресурс] / Monsters Inc. – Режим доступу: <http://www.everyonepiano.com>.
2. Сайт застосування Virtual Piano [Електронний ресурс] / Jokers. – Режим доступу: <https://virtualpiano.net/>.
3. Сайт застосування Piano world [Електронний ресурс] / Piano World Software. – Режим доступу <https://coolpiano.ru>.
4. Libfreenect [Електронний ресурс] / Ros – стаття. – Режим доступу: <http://wiki.ros.org/libfreenect>
5. OpenNI [Електронний ресурс] / Structure – стаття. Режим доступу: <https://structure.io/openni>.
6. Scratch2Kinect [Електронний ресурс] / Stephen Howell. Режим доступу: <http://stephenhowell.github.io/>.
7. Microsoft Speech [Електронний ресурс] / Wikipedia – стаття. Режим доступу: [https://ru.wikipedia.org/wiki/Microsoft\\_Speech\\_API](https://ru.wikipedia.org/wiki/Microsoft_Speech_API).
8. Kinect [Електронний ресурс] / Wikipedia – стаття. Режим доступу: <https://ru.wikipedia.org/wiki/Kinect>.
9. KinectforWindows [Електронний ресурс] / Kinect – стаття. Режим доступу: <https://habr.com/ru/post/150955/>.
10. Karma [Електронний ресурс] / Karma. Режим доступу: <https://karma-runner.github.io/2.0/index.html>.
11. Git [Електронний ресурс] / Wikipedia – стаття. Режим доступу: <https://en.wikipedia.org/wiki/Git>.
12. Довідник з охорони праці на промислових підприємствах. К.М. Ткачук, Д. Ф. Іванчук, Р. В. Сабарно та ін–К .: Техніка, 1991.–286 с.
13. ГОСТ 12.1.007–76 ССБТ. Шкідливі речовини. Класифікація і загальні вимоги безпеки.

					ІС КРБ 6.050101 019 ПЗ	64
Змін	Лист	№ докум.	Підпис	Дата		



## ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО КОДУ

### Код компоненту Scripts.js:

```
import Block from '../blocks/Block';
import BlockSpecs from '../blocks/BlockSpecs';
import ScriptsPane from './ScriptsPane';
import Events from '../../utils/Events';
import ScratchAudio from '../../utils/ScratchAudio';
import {gn, newHTML, setCanvasSize, setProps,
  localx, locally, scaleMultiplier, hit3DRect, isTablet} from
  '../../utils/lib';

export default class Scripts {
  constructor (spr) {
    this.flowCaret = null;
    this.spr = spr;
    this.dragList = [];
    var dc = gn('scriptscontainer');
    this.sc = newHTML('div', 'look', dc);
    setCanvasSize(this.sc, dc.offsetWidth, dc.offsetHeight);
    this.sc.setAttribute('id', spr.id + '_scripts');
    this.sc.setAttribute('class', 'look');
    this.sc.owner = this;
    this.sc.top = 0;
    this.sc.left = 0;
  }

  activate () {
    setProps(this.sc.style, {
      visibility: 'visible'
    });
  }

  deactivate () {
    setProps(this.sc.style, {
      visibility: 'hidden'
    });
  }

  ////////////////////////////////////////////
  //  Events MouseDown
  ////////////////////////////////////////////
```

```

scriptsMouseDown (e) {
    if (isTablet && e.touches && (e.touches.length > 1)) {
        return;
    }
    if (ScratchJr.onHold) {
        return;
    }
    if (window.event) {
        t = window.event.srcElement;
    } else {
        t = e.target;
    }
    if ((t.nodeName == 'H3') && (t.owner ==
ScratchJr.activeFocus)) {
        return;
    } // editing the current field
    ScratchJr.clearSelection();
    if (t.nodeName == 'H3') {
        ScratchJr.blur();
        ScratchJr.editArg(e, t);
        return;
    }

    if (t.firstChild && t.firstChild.nodeName == 'H3') {
        ScratchJr.blur();
        ScratchJr.editArg(e, t.firstChild);
        return;
    }

    ScratchJr.unfocus(e);
    var sc = ScratchJr.getActiveScript();
    var spt = Events.getTargetPoint(e);
    var pt = {
        x: localx(sc, spt.x),
        y: localy(sc, spt.y)
    };
    for (var i = sc.childElementCount - 1; i > -1; i--) {
        var ths = sc.childNodes[i];
        if (!ths.owner) {
            continue;
        }
        if (ths.owner.isCaret) {
            continue;
        }
        if (!hit3DRect(ths, pt)) {
            continue;
        }
        var t = new
WebKitCSSMatrix(window.getComputedStyle(ths).webkitTransform);

```

```

        // This line was causing repeat blocks to only drag when
        touched in the front and top
        // It seems to have been checking if the drag was on the
        invisible shadow of the repeat block
        // It's not clear to me why we would want this, and seems
        functional without it. -- TM
        //if ((ths.owner.blocktype == "repeat") &&
!hitTest(ths.childNodes[1], pixel)) continue;
        Events.startDrag(e, ths, ScriptsPane.prepareToDrag,
        ScriptsPane.dropBlock, ScriptsPane.draggingBlock,
ScriptsPane.runBlock);
        return;
    }
    ScriptsPane.dragBackground(e);
}

```

```

////////////////////////////////////
//  Events MouseUP
////////////////////////////////////

```

```

addBlockToScripts (b, dx, dy) {
    if ((this.flowCaret != null) &&
(this.flowCaret.div.parentNode == this.sc)) {
        this.sc.removeChild(this.flowCaret.div);
    }
    this.flowCaret = null;
    Events.dragDiv.removeChild(b);
    this.sc.appendChild(b);
    //  b.owner.drop();
    b.owner.moveBlock(dx, dy);
    for (var i = 1; i < this.dragList.length; i++) {
        var piece = this.dragList[i].div;
        piece.parentNode.removeChild(piece);
        this.sc.appendChild(piece);
        //  piece.owner.drop();
    }
    this.layout(b.owner);
    this.snapToPlace(this.dragList);
    if (b.owner.cShape) {
        this.sendToBack(b.owner);
    }
    this.dragList = [];
}

```

```

sendToBack (b) {
    if (!b.inside) {

```

```

        return;
    }
    var you = b.inside;
    while (you != null) {
        var p = you.div.parentNode;
        p.appendChild(you.div);
        if (you.cShape) {
            this.sendToBack(you);
        }
        you = you.next;
    }
    this.layout(b);
}

snapToPlace (drag) {
    if ((drag.length < 2) && drag[0].cShape) {
        this.snapCshape(drag);
    } else {
        this.snapBlock(drag);
    }
}

snapBlock (drag) {
    var me = drag[0];
    var last = me.findLast();
    var res = this.findClosest(this.available(0, me, drag), me);
    if (this.isValid(me, res, 0)) {
        this.snapToDock(res, me, 0, drag);
        return;
    }
    res = this.findClosest(this.available(last.cShape ? 2 : 1,
last, drag), last);
    if (!this.isValid(last, res, last.cShape ? 2 : 1)) {
        return;
    }
    this.snapToDock(res, last, last.cShape ? 2 : 1, drag);
}

snapCshape (drag) {
    var me = drag[0];
    var last = me.findLast();
    var res = this.findClosest(this.available(0, me, drag), me);
    if (this.isValid(me, res, 0)) {
        this.snapToDock(res, me, 0, drag);
        return;
    }
}

```

## ДОДАТОК Б ІНСТРУКЦІЯ КОРИСТУВАЧА

До калібрування встановіть такі мінімальні і максимальні значення для «важелів» в лівій частині екрана (за допомогою ПКМ – «Встановити мінімальне і максимальне значення»): CLX, CRX –  $[-100; 100]$ , CLZ, CRZ –  $[-500; 500]$ , Scale\_X –  $[0; 100]$ , Scale\_Z –  $[0; 500]$ . Перейдіть в режим презентації, і, навівши курсор миші на центр поля (синя точка посередині екрану) збільште масштаб за допомогою екранної лупи Windows (для цього використовуємо поєднання клавіш Windows + "+")

### Інструкція з калібрування

#### 1. Натиснути на сірий коло лівою кнопкою миші (ЛКМ)

Повинні з'явитися 2 червоні точки, які рухаються разом з рухом ніг гравця (по ходу процесу калібрування вони вирівнятися під ноги)

2. Гравцеві необхідно стати лівою ногою на великий червоний круг (рекомендуємо стати правим боком до кінект), коли положення зафіксоване, натиснути на сірий коло (ЛФМ).

3. Аналогічно робимо в протилежному кутку ігрового поля, стаючи лівим боком.

4. Далі гравець стає на центр лівою ногою, і робимо те ж саме.

5. Після підводиться ногу на 15–20 см від підлоги (на тому ж місці) і, зафіксувавши це положення ноги, натискаємо на сірий коло (ЛФМ).

6. Тепер залишилося підсунути маркери під ноги гравця. Для цього гравцеві краще встати на ігрове піаніно – туди, де буде відбуватися процес гри, і за допомогою «важелів» в лівій частині екрана налаштувати маркери так, щоб вони рухалися рівно за ногами. CLX і CRX – зміни координат маркерів по осі X лівої і правої ноги відповідно. CLZ і CRZ – зміна координат маркерів по осі Z лівої і правої ноги відповідно. Scale\_X і Scale\_Z – зміна коефіцієнта переміщення маркерів зі зміною координат ноги від центру ігрового поля по осях X і Z відповідно (Наприклад, якщо при пересуванні від центру поля до країв, маркери

тікають вперед від ніг – Scale зменшуємо, а якщо маркери відстають від ніг – збільшуємо коефіцієнт).

7. В кінці калібрування потрібно натиснути на помаранчевий коло (ЛФМ).

Процес калібрування завершено, тепер можете приступити до гри, для цього:

1. Натисніть на кнопку вибору мелодії «Choose melody». Це робиться ЛФМ або натисканням ногою на кнопці на ігровому полі.

2. Виберіть мелодію, яку хочете зіграти. Так само ЛФМ або ногою.

3. Далі почнуть підсвічуватися клавіші світло-сірим (або синім) кольором, які потрібно зіграти (якщо клавіша змінилася, значить потрібно її грати знову).

4. Після натискання гравцем на останню клавішу мелодії гра припиниться, кольорові клавіші з'являтися не будуть, можна буде вибрати нову мелодію.

Білий хрест в червоному квадраті, при натисканні ЛКМ або ногою, зупиняє програму.

#### Додавання нової мелодії

1. Дублюємо вже наявний спрайт мелодію (горизонтально розташований прямокутник з назвою) для зручності і швидкості. // дублювати 'МК'

2. Змінюємо назву спрайту на зрозуміле і короткий. // Поміняли назву на 'М1'

3. Натискаємо на кнопку 'Choose melody', щоб всі мелодії з'явилися на екрані.

4. Тепер можна змінити розмір, положення всіх пісень для подальшої зручності.

5. Міняємо костюм (зовнішній вигляд) кнопки на потрібний нам. // Біла кнопка з написом 'Melody 1'

6. Створюємо (Змінюємо) блоки коду в скрипті:



Рисунок Б.1 – Скрипт для додавання мелодії

1) «Коли клацнуть по М1» – «передати М1» (створюємо новий сигнал, який буде позначати що потрібно грати саме цю мелодію, наприклад для «Коника» – «МК», для «Melody 1» – «М1».)

2) Для кожної мелодії створюємо окремий список. Для цього переходимо у вкладку «Змінні», натискаємо «Створити список». Вводимо ім'я, наприклад «Melody1» – натискаємо «ОК».

3) До цього списку потрібно «прив'язати» текстовий файл, в якому буде послідовність нот потрібної мелодії. Для зручності є папка «ноти», в якій створимо файл Мелодія 1.txt. Стандартного блокнота вистачить для заповнення документа, нам важливий тільки порядок нот, які буде натискати гравець. Для цього пишемо їх в стовпчик цифрами, наприклад:

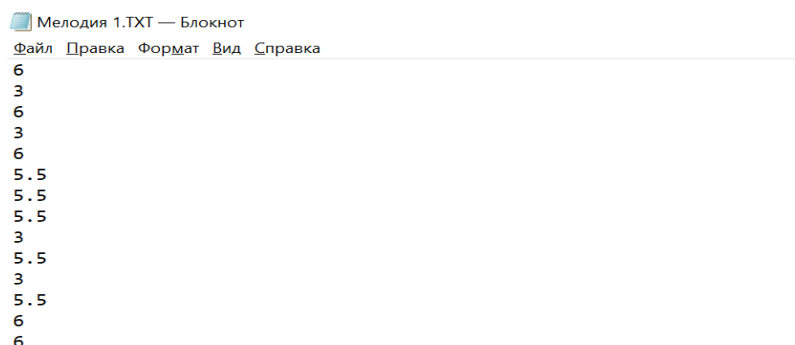


Рисунок Б.2 – Формат запису нот у мелодії

5) Коли документ готовий, переходимо в Scratch, якщо перелік не з'явився на екрані, у вкладці «Змінні» потрібно поставити галочку навпроти його назви.

6) Натискаємо ПКМ з цього списку, який з'явився на екрані – «Імпорт» і вибираємо наш файл.

7) Далі в скрипті спрайту потрібно записати всі можливі ноти, які є у нас в списку. У подблок «Завжди» для кожної ноти створюємо (змінюємо наявний) блок коду.

«Якщо елемент [під номером (тут залишаємо як є)] з [нашого списку (вибираємо наш список)] дорівнює [нота (яку потрібно буде зіграти)]» – «Поставити [MelodyNote (не змінюємо)] в [ту ж ноту ( яку потрібно буде зіграти)] ». І так для кожної ноти. Уважно стежимо за крапками і комами (всюди ставимо крапки!) На зображенні можна подивитися нумерацію нот (Нота 1 – «До» [C (60)]) (рис. Б.3)

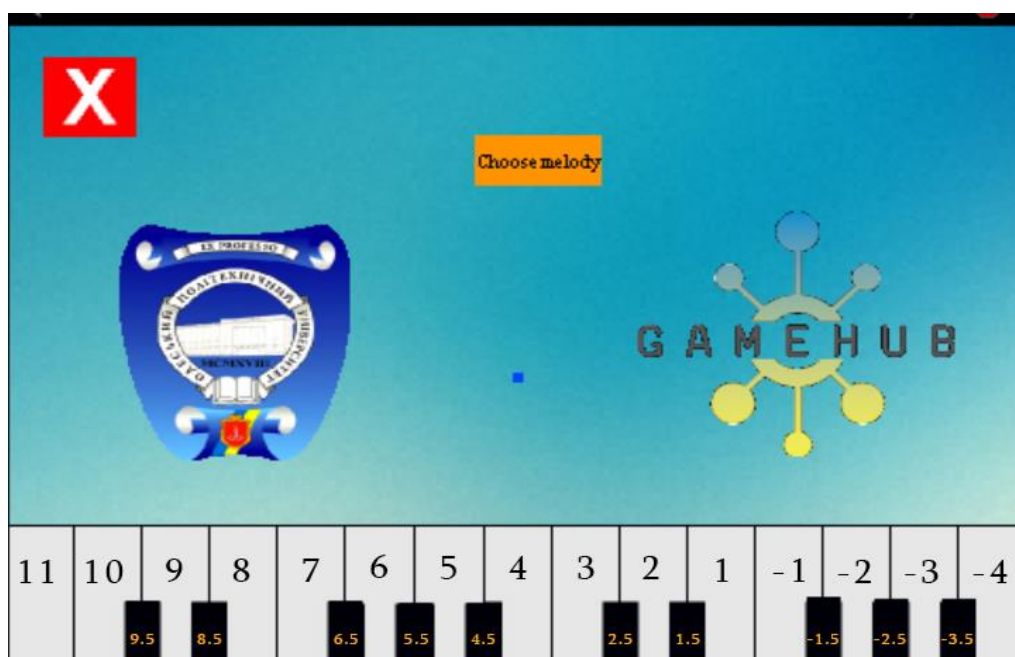


Рисунок Б.3 – Нумерація нот на піаніно

Готово! Скрипт нової мелодії (рис. Б.4):



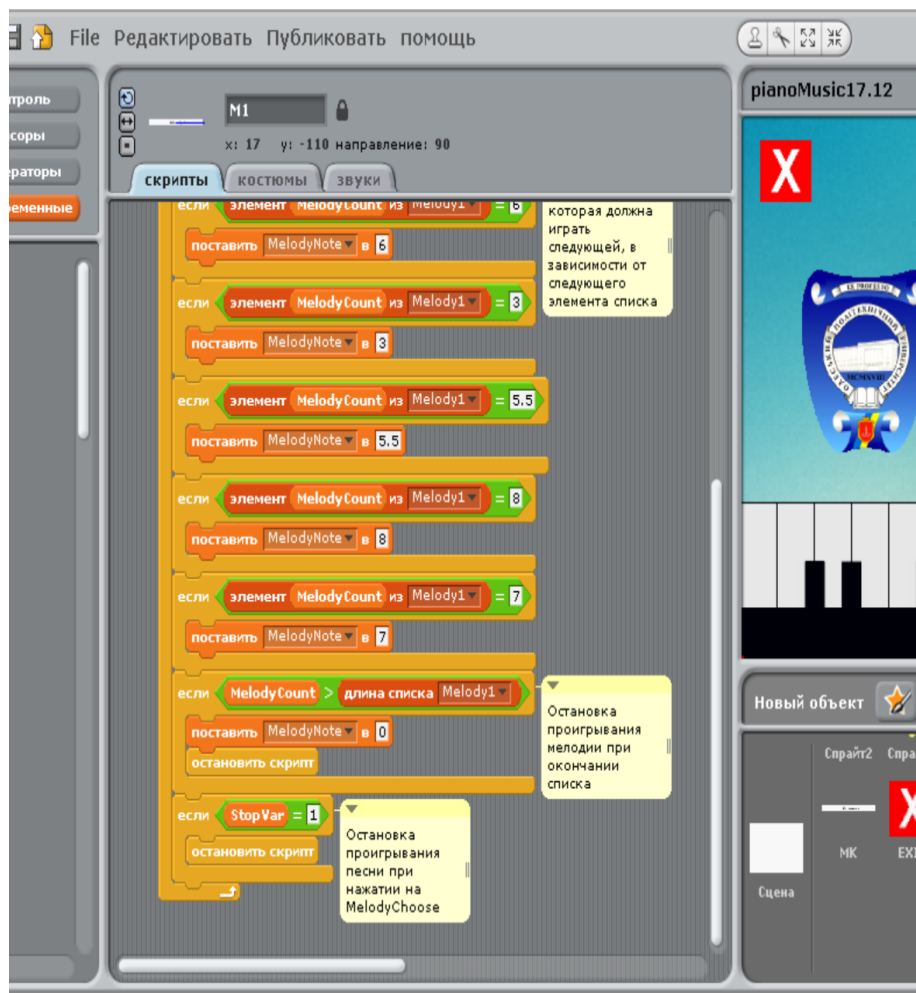


Рисунок Б.4 – Скрипт готовой мелодии

