

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

3D-конструктор дитячих майданчиків.
Програмно-технічні засоби 3D-конструювання дитячих майданчиків
Назва теми


КВРКІ.190173.19.01.01 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва


Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Виконав: студент III курсу, група КІ2с-19-1


Підпис


О. А. Андрєєв
Ініціали, прізвище

Керівник


Підпис, дата


Т. О. Говорущенко
Ініціали, прізвище

Нормоконтролер


Підпис, дата

С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Т.О. Говорущенко
Ініціали, прізвище

« 26 » травня 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 10 ” 01 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Андрееву Олексію Андрійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) 3D-конструктор дитячих майданчиків. Програмно-технічні засоби 3D-конструювання дитячих майданчиків

Керівник проекту (роботи) Говорущенко Т.О., д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2022 р. № 18

2. Строк подання студентом проекту (роботи) на кафедру 06.06.2022 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Дослідження предметної області та постановка задачі

Аналіз засобів розроблення програмно-технічних засобів 3D-конструювання дитячих майданчиків

Проектування та реалізація програмно-технічних засобів 3D-конструювання дитячих майданчиків

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____





Техніко-економічне обґрунтування доцільності проектування програмно-технічних засобів

3D-конструювання дитячих майданчиків

Структура та алгоритми роботи модулів програмно-технічних засобів

Приклади функціонування програмно-технічних засобів

6. Консультанти розділів дипломного проекту (роботи)


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагиат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2022	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2022	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2022	виконано
4	Робота над розділом 2 – аналіз засобів розроблення програмно-технічних засобів 3D-конструювання дитячих майданчиків	01.04.2022	виконано
5	Робота над розділом 3 – проектування та реалізація програмно-технічних засобів 3D-конструювання дитячих майданчиків	29.04.2022	виконано
6	Оформлення пояснювальної записки згідно вимог	20.05.2022	виконано
7	Попередній захист ВКР	24.05.2022	виконано
8	Захист ВКР на засіданні ЕК	Червень 2022 року	


Студент


Підпис

О. А. Андреев

Ініціали, прізвище

Керівник роботи


Підпис

Т. О. Говорущенко

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «3D-конструктор дитячих майданчиків. Програмно-технічні засоби 3D-конструювання дитячих майданчиків».

Автор роботи: Андреев Олексій Андрійович.

Керівник роботи: Говорущенко Тетяна Олександрівна.

Пояснювальна записка: 69 с., 18 рис., 2 табл., 3 дод., 40 джерел.

Графічна частина: 7 презентаційних слайдів.

ПРОГРАМНО-ТЕХНІЧНІ ЗАСОБИ, 3D-КОНСТРУКТОР ДИТЯЧИХ МАЙДАНЧИКІВ, ГРАФІЧНІ РЕДАКТОРИ, ДИЗАЙН, МОДУЛІ.

Метою роботи є проектування та розроблення програмно-технічних засобів 3D-конструювання дитячих майданчиків.

Об'єктом дослідження є процес 3D-конструювання дитячих майданчиків.

Предметом дослідження є програмно-технічні засоби 3D-конструювання дитячих майданчиків.

Практичне значення має проведений комплекс розрахунків для визначення витрат, доходів та собівартості програмно-технічних засобів 3D-конструювання дитячих майданчиків, а також розроблені програмно-технічні засоби 3D-конструювання дитячих майданчиків.







Підпис студента

26.05.2022

Дата

ЗМІСТ

ВСТУП.....	3
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ..	5
1.1 Техніко-економічне обґрунтування доцільності проектування.....	5
1.2 Існуючі способи вирішення технічної проблеми, їх аналіз і недоліки....	7
1.3 Прогнозування собівартості нової розробки.....	9
1.4 Висновки. Постановка задачі.....	20
2 АНАЛІЗ ЗАСОБІВ РОЗРОБЛЕННЯ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ 3D-КОНСТРУЮВАННЯ ДИТЯЧИХ МАЙДАНЧИКІВ.....	21
2.1 Характеристика та вимоги до конструкторів дитячих майданчиків	21
2.2 Дослідження редакторів растрової та векторної графіки	23
2.3 Аналіз існуючих мов та середовищ програмування.....	29
2.4 Висновки	37
3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ 3D-КОНСТРУЮВАННЯ ДИТЯЧИХ МАЙДАНЧИКІВ.....	39
3.1 Етапи розроблення програмно-технічних засобів	39
3.2 Дизайн 3D-конструктора дитячих майданчиків	41
3.3 Основні модулі 3D-конструктора дитячих майданчиків	47
3.4 Висновки	60
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	63
ДОДАТОК А Копія креслення «Техніко-економічне обґрунтування доцільності проектування програмно-технічних засобів 3D- конструювання дитячих майданчиків».....	67
ДОДАТОК Б Копія креслення «Структура та алгоритми роботи модулів програмно-технічних засобів».....	67
ДОДАТОК В Копія креслення «Приклади функціонування програмно- технічних засобів».....	69

					КвРКІ 190173.19.01.01 ПЗ			
Зм.	Арк.	№докум.	Підпис	Дата		Літера	Арквш	Арквшів
Виконав		Андрєєв О.А.			3D-конструктор дитячих майданчиків. Програмно-технічні засоби 3D-конструювання дитячих майданчиків.	у		
Перевір.		Говорущенко Т.О.					2	68
Н.контр.		Лисенко С.М.			Пояснювальна записка	ХНУ КІ2с-19-1		
Затвер.		Говорущенко Т.О.						

ВСТУП

При облаштуванні інфраструктури сучасних міст чималу увагу варто приділяти дизайну та облаштуванню дитячих майданчиків. З метою забезпечення зміцнення організму дітей, умов перебування дітей на свіжому повітрі, проведення дозвілля активно, в колективі своїх однолітків, а також фізичного розвитку дітей, варто в кожному мікрорайоні міста передбачити дитячі ігрові майданчики.

Ігрові майданчики мають забезпечувати фізичний і духовний розвиток дітей, стимулювати ріст здібностей дитини, задовольняти їх запити, сприяти активній творчій грі, допомагати набуті дітям нових корисних для життя навичок, розвивати свої схильності та таланти.

Дитячі ігрові майданчики повинні створювати умови для прояву найбільшої самостійності та максимальної ініціативи дітей.

Задача побудови ескізного представлення проектів дитячих майданчиків уже давно не є забаганкою. Від естетичного вигляду, можливості оцінити розташування комплексів, залежить доля майбутнього проекту.

На даний час дану проблему можна вирішити за допомогою спеціалізованих програм, таких як 3DS Max, Cinema 4D та інші. Але робота з цими програмами вимагає певних навичок, високопродуктивного комп'ютерного обладнання а також бази моделей для роботи.

З недавнього часу з'являються 3D конструктори дитячих майданчиків. Їх перевагою являється можливість оцінити майбутній проект з різних ракурсів. Але великим недоліком являються високі вимоги до ресурсів комп'ютера, а також моделі для таких конструкторів повинні бути низько полігональні. Це призводить до втрати реалізму.

Метою кваліфікаційної роботи є проектування та розроблення програмно-технічних засобів 3D-конструювання дитячих майданчиків.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

В даній кваліфікаційній роботі розроблені програмно-технічні засоби, які дають можливість побудови дитячих майданчиків з високим рівнем реалізму, мають велику базу комплексів та інструменти збереження і правки готових проектів. Ці засоби працюють без встановлення додаткових програм, не вимагають особливих навичок для роботи та наявності високопродуктивного апаратного забезпечення.

Об'єктом дослідження є процес 3D-конструювання дитячих майданчиків.

Предметом дослідження є програмно-технічні засоби 3D-конструювання дитячих майданчиків.

Для досягнення поставленої мети використовувались наступні методи дослідження: принципи системного аналізу, методи синтезу, аналізу та моделювання процесів, теоретико-множинні підходи.

Практичне значення має проведений комплекс розрахунків для визначення витрат, доходів та собівартості програмно-технічних засобів 3D-конструювання дитячих майданчиків, а також розроблені програмно-технічні засоби 3D-конструювання дитячих майданчиків.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Техніко-економічне обґрунтування доцільності проектування

Наразі в сучасному інформаційному суспільстві динамічно розвиваються технології 3D друку деталей різноманітних геометричних форм та технології доповненої реальності, засоби рендеру та інші інноваційні напрямки. Компанії удосконалюють алгоритми відображення та опрацювання віртуальних моделей, спрощують доступ до тривимірного контенту в мережі Інтернет. Внаслідок таких удосконалень сфери застосування 3D графіки розширюються і припиняють обмежуватись індустрією ігор, системами автоматизованого проектування або створенням візуальних ефектів у телевізійному виробництві та кінематографі, ландшафті, архітектурі, тощо. [1]

Продуманий дизайн майданчика необхідний для комфортного, цікавого, а найголовніше безпечного проведення часу дітей. При плануванні дитячого майданчика важливо врахувати наступні параметри.

Дитячий майданчик рекомендується будувати далеко від автомобільних стоянок, каналізаційних люків, басейну або крутих схилів і захистити його по периметру [2].

Майданчик повинен бути освітлений сонячним промінням максимально вдень і електричним освітленням - у вечірній.

Простір ігрового майданчика слід розташувати так, щоб його було видно з більшості вікон будинку [3].

Всі ігрові елементи майданчика повинні знаходитися на безпечній відстані.

Деревина, яка використовується на дитячому майданчику, повинна бути добре оброблена, щоб убезпечити дітей від скал. Найкраще покривати її прозорим лаком, а не фарбою. А якщо без фарби не обійтись, не можна вживати лаки та фарби, які містять свинець чи інші отруйні речовини [4].

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

Земля для м'якості має бути покрита травою або чистим крупнозернистим піском. Щебінь і будівельний пісок із домішкою вапна використовувати не можна, оскільки вони шкідливі для шкіри дітей. З гігієнічного погляду не рекомендується влаштовувати майданчик на звичайній землі, краще, якщо це буде піщана місцевість. Як покриття для дитячого майданчика ідеально підходить кора коркового дерева, але це матеріал рідкісний і дорогий. Є ще один варіант – спеціальне гумове покриття для спортмайданчиків, воно убереже від травм після падіння [5].

Ще не варто забувати, що в ідеалі дитячі майданчики планують так, щоб за кілька років їх можна було переробити на корисні споруди. Наприклад, наземний будиночок переобладнати на сарай, турнікети — на перекладини для сушіння білизни, а пісочницю — на клумбу.

Коли розпочинати облаштування? Травень — найкращий для цього час. Адже попереду ще ціле літо! Але воно, на жаль, швидкоплинно, і треба постаратися не витратити його повністю на підготовку ділянки, вибір, покупку і установку ігрових конструкцій, що сподобалися [6].

Безумовно, все залежить, в першу чергу, від віку дитини, а також від її фізичного розвитку та характеру. Ще необхідно врахувати наявність вільного місця на дачній ділянці, його ландшафтні характеристики, розташування. Для малюків добре підійде ігровий куточок, що складається з гірки, пісочниці, гойдалки та ігрового будиночка, в якому дитина із задоволенням може «прокопатися» цілий день [7].

Для старших дітей добре підійде спортивний майданчик: вуличні шведські стінки, турники, ігрові комплекси, тунелі, батут, баскетбольне кільце. Хоча цілком можливо, що ігровий куточок можна буде поєднати зі спортивним майданчиком для повного спектру дитячих інтересів [8-10].

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Існуючі способи вирішення технічної проблеми, їх аналіз і недоліки

Проаналізуємо сучасне спеціалізоване програмне забезпечення, серед якого варто відзначити чималу кількість програм для редагування і створення 3D графіки. Найбільш поширеними пакетами для роботи з тривимірними моделями є продукти компанії AUTODESK (3DS MAX, Maya, Mudbox, AutoCAD). Всі вони схожі за своїми принципами, проте відрізняються структурою інтерфейсу та можливостями. Компанія дозволяє використовувати ліцензійне програмне забезпечення 3 роки для навчання, що робить її продукцію більш доступною, наприклад, для студентів у порівнянні із іншими розробниками. Крім цього, є програми CINEMA 4D, MODO, LightWave, Blender 3D, які також відрізняються структурою інтерфейсу та можливостями.

Дітки від 3 до 6 років дуже цікаві та активні, тому елементи ігрового комплексу мають бути максимально безпечними: не мати гострих країв, витримувати великі навантаження, бути дуже стійкими. Для дітей такого віку можна поставити пісочниці, ігрові будиночки, маленькі гірки, пружинні гойдалки. Можна використовувати високі гірки, сітки для лазіння, ліани, гойдалки, різні турніки [11-12].

Дітям старше 7 років необхідно максимум руху. Адже для цього віку характерна «змагальність», схильність до ризику, що потребує фізичної сили та спритності, відмінної координації. Тому варто додати додаткові елементи для виконання фізичних вправ. Тут вже знадобляться атракціони з елементами лазіння, подолання перешкод, стрибання, атракціони з метанням кілець, м'ячів, всілякі тири з повітряними гарматами [13-15].

Якщо діти різновікові, необхідно захистити малюків від «ризикованих» елементів майданчика. Зробити технічно це не дуже складно, наприклад, підняти нижню сходинку біля гірок, нижню перекладину на турниках і т.д. [16]

В цілому, планування дитячого майданчика дуже кропітка справа, потрібно мати технічні навички і тому його краще довірити професіоналам [17].

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

«Родзинкою» майданчика може стати будиночок на деревах. Його основу складає дерев'яний каркас, з усіх боків оббитий водостійкою фанерою. Два дерева служать палями. При необхідності можна принести і встановити сухі стволи, але найкраще виглядає такий будиночок на живих деревах. Коли потрібно встановити сухі дерева, викопують дві ями глибиною і діаметром 60 см, заповнюють їх бетоном на 20 см. У бетон, що злегка закріпився, вставляють стовбури і заливають ями бетоном доверху [18-20].

Будівництво ведуть знизу нагору. Спочатку роблять несучу конструкцію підлоги. Вона має розміщуватися на висоті 1,5–1,7 м від землі. Підлогу вистилають щільною дерев'яною плитою. Знизу підлога та стовбури дерев з'єднують силовими підкосами для міцності. Зовні стіни покривають смужками фанери шириною 15 см, прибиваючи їх внахлест. Дах обшивають листами водостійкої фанери. Можна додатково прибити до неї тонкими цвяхами руберойд. Вікна є порожні прорізи. За бажанням їх можна засклити [21].

Пісочниця. Це найпоширеніший і найкорисніший ігровий елемент для будь-якого дитячого містечка. Пісочниця грає роль дитячої дачної ділянки для малюків. Гра у пісочниці добре розвиває моторику, тактильні відчуття. Дитяча фантазія та уява працюють у пісочниці на 100%, втілюючись у тунелі та міста [22].

Гірки. Зазвичай їх поділяють на дві категорії, залежно від віку дитини. До трьох років – це гірки маленькі, легкі та компактні, що сприяє їхньому простому транспортуванню навіть у сімейному автомобілі. Вони невисокі, з пологим спуском, який гладко закінчується в кінці для того, щоб дитина, з'їхавши, не забилася. Гірка повинна бути оснащена зручними широкими сходами і покрита спеціальним неслизьким покриттям, щоб малюк міг безпечно і комфортно залазити на неї. На верхньому майданчику гірки обов'язково потрібні поручні, щоб дитині було за що триматися на висоті [23].

Для дітей від 3-х років висота гірки має бути від 1,5 м до 2,5 м. Обов'язково наявність перил на верхньому майданчику, що захищають від падіння. Гірка може бути не обов'язково з прямим спуском, але також гвинтовим. Край гірки можна

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

направити в басейн, що дуже зручно влітку - дітлахи будуть із задоволенням плюхатися в басейн [24].

Гойдалка. Гойдалки-балансири розраховані як мінімум для двох дітей. Для безпеки в місцях зіткнення із землею найкраще поставити м'які подушки. Гойдалки на пружині - в них можна хитатися абсолютно в будь-якому напрямку. Зазвичай вони виконані у вигляді невеликих яскравих тварин чи автомобілів [25-27].

Важливо дуже міцно закріпити таку гойдалку на землі, щоб унеможливити падіння дитини. Підвісна гойдалка - найпоширеніша модель, її можна використовувати навіть у будинку. Комфортне крісло та гарне кріплення — все, що для них потрібно. Гойдалки на стійках - варіант мобільних підвісних гойдалок. Якщо замість крісла - проста перекладина, тоді це вже буде гойдалка-тарзанка - на ній можна і стрибати, і крутитись у будь-якому напрямку [28].

Ігровий будиночок. Обоєжнуваний усіма дітьми елемент обов'язково має бути на ділянці. Діти дуже люблять «по-дорослому» обживати своє житло. Будиночки можуть бути різної тематики: від лицарських замків до хатини біля моря [29-30].

1.3 Прогнозування собівартості нової розробки

Для визначення доцільності розробки програмно-технічних засобів необхідно спрогнозувати орієнтовну собівартість розробки та можливі експлуатаційні витрати. Така інформація потрібна і розробникам, і потенційним покупцям електронного додатку [31-40].

Для прогнозування собівартості нової розробки скористаємось методом питомої ваги. Даний метод використовується, коли наявна можливість розрахувати пряму статтю витрат і встановити приблизно питому вагу даної статті витрат в собівартості аналогічної продукції.

Собівартість одиниці нової продукції можна спрогнозувати за формулою:

$$S = \frac{B_n \cdot K_H \cdot 100\%}{П\%} \text{ [грн]}, \quad (1.1)$$

де B_n – величина однієї з статей прямих витрат, яка вибрала за основу, грн;

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						9
Зм.	Арк.	№ докum.	Підпис	Дата		

Π – питома вага однієї з статей прямих витрат в собівартості аналога, %;

K_H – коефіцієнт врахування конструктивних та технологічних особливостей нової розробки, $K_H = 1 \div 1,2$.

$$S = \frac{4929 \cdot 1,2 \cdot 100\%}{23} = 225716,52 \text{ (грн)}.$$

Капітальні вкладення (K) та експлуатаційні витрати (E) визначаються приблизно спрощеним способом.

Величина капітальних вкладень (K) може бути розрахована за формулою:

$$K = B \cdot A \cdot S \text{ [грн]}, \quad (1.2)$$

де B – коефіцієнт, який враховує витрати на розробку, придбання, транспортування, вантаж, налагодження нової розробки тощо; $B \approx 1,1 \div 1,8$;

A – коефіцієнт, який враховує прогнозований прибуток, податок, який повинен виплачувати виробник тощо; $A \approx 1,5 \div 2$;

S – собівартість нової розробки, грн, яка оцінюється приблизним способом.

$$K = 1,2 \cdot 1,8 \cdot 225716,52 = 55547,68 \text{ (грн)}.$$

Експлуатаційними є витрати, які забезпечують стандартне функціонування цього вибору в період експлуатації.

В найбільш спрощеному вигляді величина експлуатаційних витрат в розрахунку за 1 рік може бути спрогнозована за формулою:

$$E = k \cdot A \cdot S \cdot \beta \text{ [грн/рік]}, \quad (1.3)$$

де k – коефіцієнт, який ураховує витрати на амортизацію, електроенергії, обслуговування, ремонти тощо. Рекомендується приймати значення $k =$

					КВРКІ 190173.19.01.01 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

(0,2 ÷ 0,4). Для обчислювальної техніки значення k рекомендується приймати по максимуму;

A – коефіцієнт, який враховує прогнозований прибуток, податок, який повинен виплачувати виробник тощо; $A \approx 1,5 \div 2$;

S – собівартість нової розробки, грн, яка оцінюється приблизним способом;

β – відношення часу, який працівник витрачає на виконання певних робіт із застосуванням конкретного програмного продукту, до загального часу всієї роботи працівника: $\beta = 0,10 \div 0,18$.

$$E = 0,4 \cdot 1,8 \cdot 225716,52 \cdot 0,12 = 19501,9(\text{грн}).$$

Виходячи із отриманих прогнозних розрахунків, можна зробити висновок, що необхідність створення бази конструктора дитячих майданчиків є актуальним та доцільним, що і підтверджують прогнозні розрахунки.

Фактична або повна собівартість програмного забезпечення розраховується в процесі розрахунку калькуляції собівартості та відображає виробничу собівартість, адміністративні витрати та витрати на збут.

Для визначення трудомісткості програмного забезпечення необхідно побудувати таблицю вхідної інформації (таблиця 1.1), скориставшись наступною інформацією.

Для розрахунку трудомісткості програмного забезпечення необхідно визначити наступну інформацію:

- кількість макетів (набори даних вхідної інформації);
- кількість видів форм інформації (форм інформації на машинних носіях та друкованих документів);
- ступінь новизни групи задач – задачі діляться на 4 групи за цим критерієм:

A – задачі, що передбачають проведення науково-дослідних робіт, використання нових методів розроблення;

					КВРКІ 190173.19.01.01 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

Б – розв’язання оригінальних задач, розроблення типових проектних рішень, та оригінальних систем, що не мають аналогів;

В – розроблення типових практичних рішень з можливістю зміни; розв’язання задач, що мають аналогічні рішення;

Г – розроблення типових проектних рішень без можливості зміни; розв’язання задач, що мають аналогічні рішення;

– складність алгоритму. Розрізняють три групи складності:

а) алгоритми моделювання систем і об’єктів та оптимізації;

б) алгоритми обліку, звітності, статистики, пошуку;

в) алгоритми, які представляють стандартні методи рішень і не передбачають використання складних логічних та чисельних методів;

– складність організації контролю вхідної і результуючої інформації, яка характеризується наступними групами:

а) результуючі дані і документи різного розміру, з різною структурою, контроль здійснюється перехресно з врахуванням зв’язку між різними документами;

б) вхідні дані та документи з одноманітними формою та змістом, здійснюється формальний контроль;

в) друк документів складної багаторівневої структури з різноманітними формою та змістом;

г) друк документів з одноманітними формою та змістом, виведення масивів даних на машинні носії.

– вид використовуваної інформації:

а) змінна (ПІ);

б) нормативно-довідкова (НДІ);

в) банк даних (БД);.

– мову програмування;

– можливість використання стандартних модулів, типових програм, пакетів прикладних програм [31].

					КВРКІ 190173.19.01.01 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

Детальний розрахунок трудомісткості програмного продукту (ПП), комп'ютерної системи, автоматизованої системи чи мережі.

Таблиця 1.1 – Вхідна інформація для визначення трудомісткості бази даних

Найменування	Ступінь новизни	Складність алгоритму	Вид інформації	Кількість макетів вхідної інформації	Кількість макетів вихідної інформації	Мова програмування
Параметр	Б	А	ПП	2	7-8	Середня
Нормативні дані визначені на основі вхідної інформації						
	36		$k_{стан.}$	$N_{час} - 252$		$k_m - 1$
			- 0,6	$k_{скл} - 1,08$		

Загальну трудомісткість можна визначити за формулою:

$$T_{заг} = N_{час} \cdot k_{скл} \cdot k_m \cdot k_{станд} \cdot k_{станд\ ПП} \text{ [людино дні]}, \quad (1.4)$$

де $T_{заг}$ – загальна трудомісткість, людино-дні;

$N_{час}$ – норма часу, людино-дні;

$k_{скл}$ – коефіцієнт поправки щодо складності контролю вхідної та результуючої інформації (рекомендовано 1,08);

k_m – коефіцієнт поправки використання мови конкретного рівня складності (для мови низького рівня потрібна корекція з урахуванням коефіцієнта k_m , рівного 1,15; для мов високого рівня коефіцієнт дорівнює 1);

$k_{станд}$ – коефіцієнт використання типових/стандартних програм (оскільки, в основному, використовуються стандартні модулі, типові програми та/або пакети прикладних програм, тому норму часу варто коригувати за допомогою коефіцієнта $k_{станд} = 0,6 - 0,8$;

$k_{станд.ПП}$ – коефіцієнт розробки типового/стандартного програмного продукту (норму часу варто корегувати за допомогою коефіцієнта, який становить 1,2-1,6).

Для визначення трудомісткості з врахуванням інших характеристик варто використати коефіцієнти поправок.

$$T_{заг} = 252 \cdot 1,08 \cdot 1 \cdot 0,6 \cdot 1,5 = 244 \text{ (людино дня)}.$$

Необхідна чисельність працівників, потрібних для розробки програмно-технічних засобів (системи, мережі) визначається згідно з формулою:

$$Ч = \frac{T_{заг}}{\Phi_{р.ч.} \cdot \frac{t_{розр.}}{12}} \text{ [осіб]}, \quad (1.5)$$

де Ч – необхідна чисельність розробників ПП, осіб;

$T_{заг}$ – загальна трудомісткість, людино-дні;

$\Phi_{р.ч.}$ – річний фонд робочого часу (встановлюється трудовим законодавством на кожен рік, днів;

$t_{розробн}$ – запланований строк розробки, місяці.

$$Ч = \frac{244}{256 \cdot \frac{1}{12}} \approx 1 \text{ (особа)}.$$

Для подальших розрахунків приймаємо 1 особу.

Розрахуємо заробітну плату обслуговуючого персоналу

$$З_{обс} = 12 \cdot М \cdot \beta \text{ [грн/рік]}, \quad (1.6)$$

де М – місячний оклад працівника інженерно-технічного рівня, грн;

					КВРКІ 190173.19.01.01 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

12 – число місяців;

β – відношення часу, який витрачається працівником на обслуговування виробу із застосуванням певної програми, до загального часу його роботи – 10-18%.

$$Z_{\text{обс}} = 12 \cdot 4173 \cdot 0,12 = 6009,12(\text{грн/рік}).$$

Додаткова заробітна плата розраховується як 10-12% основної заробітної плати обслуговуючого персоналу $Z_{\text{д}}$.

$$Z_{\text{д}} = \frac{10}{100} \cdot Z_{\text{обс}}[\text{грн}], \quad (1.7)$$

$$Z_{\text{д}} = \frac{10}{100} \cdot 6009,12 = 600,91(\text{грн}).$$

Нарахування на заробітну плату обслуговуючого персоналу розраховується як 22% від суми основної та додаткової зарплати всього обслуговуючого персоналу:

$$H_{\text{есв}} = (Z_{\text{обс}} + Z_{\text{д}}) \cdot 0,22 [\text{грн}], \quad (1.8)$$

$$H_{\text{есв}} = (6009,12 + 600,91) \cdot 0,22 = 1454,2 (\text{грн}).$$

Витрати на електроенергію можуть бути розраховані за формулою:

$$B_{\text{с}} = B \cdot П \cdot \Phi \cdot K_{\text{п}} \cdot \beta[\text{грн}], \quad (1.9)$$

де B – вартість 1 кВт – години електроенергії. $B=1,68\text{грн/кВт}$;

Φ – кількість годин роботи комп'ютера та інших приладів за рік по факту, годин, $\Phi=1800$;

					КВРКІ 190173.19.01.01 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

P – потужність комп'ютера, принтера та приладів, кВт;

K_n – коефіцієнт використання потужності. $K_n < 1$;

β – відношення часу, який витрачається працівником на конкретні роботи із застосуванням конкретного програмного продукту, до загального часу його роботи: $0,10 \div 0,18$:

$$V_c = 1,68 \cdot 0,4 \cdot 1800 \cdot 0,96 \cdot 0,12 = 139,34 \text{ (грн)}.$$

Амортизаційні відрахування для програмного продукту визначаємо за формулою:

$$A = \frac{C \cdot N_a \cdot \beta}{100} \text{ [грн]}, \quad (1.10)$$

де C – вартість робіт по створенню нового програмного продукту;

N_a – річна норма амортизаційних відрахувань – 20%.

$$A = \frac{10342 \cdot 20 \cdot 12}{100} = 248,2 \text{ (грн)}.$$

Витрати на поточний ремонт комп'ютерної техніки можна розрахувати за формулою:

$$P = [(0,04 \div 0,1) \cdot C + Z_d + Z_{\text{обс}}] \cdot \beta \text{ [грн]}, \quad (1.11)$$

де C – балансова вартість персонального комп'ютера, грн;

$Z_{\text{обс}}$ – заробітна плата обслуговуючого персоналу;

Z_d - додаткова заробітна плата;

					КВРКІ 190173.19.01.01 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

β – відношення часу, який витрачається працівником на виконання даних робіт із застосуванням конкретного програмного продукту, до загального часу його роботи: $\beta = 0,10 \div 0,18$.

$$P = 0,1 \cdot 10342 + (6009,12 + 600,91) \cdot 0,12 = 917,2(\text{грн}).$$

Інші витрати, як правило, становлять 5-10% загальної суми усіх попередніх витрат:

$$I_B = (Z_{\text{обс}} + Z_{\text{д}} + H_{\text{есв}} + B_c + A + P) \cdot 0,07[\text{грн}], \quad (1.12)$$

$$I_B = (6009,12 + 600,91 + 1454,2 + 139,34 + 248,2 + 917,2) \cdot 0,05 = 468,44(\text{грн}).$$

Сума попередніх витрат дає можливість визначити величину витрат, коли коли використовується новий програмний продукт (таблиця 1.2):

$$E_2 = Z_{\text{обс}} + Z_{\text{д}} + H_{\text{есв}} + A + I_B[\text{грн}], \quad (1.13)$$

Наочно структура експлуатаційних витрат наведена на рисунку 1.1.

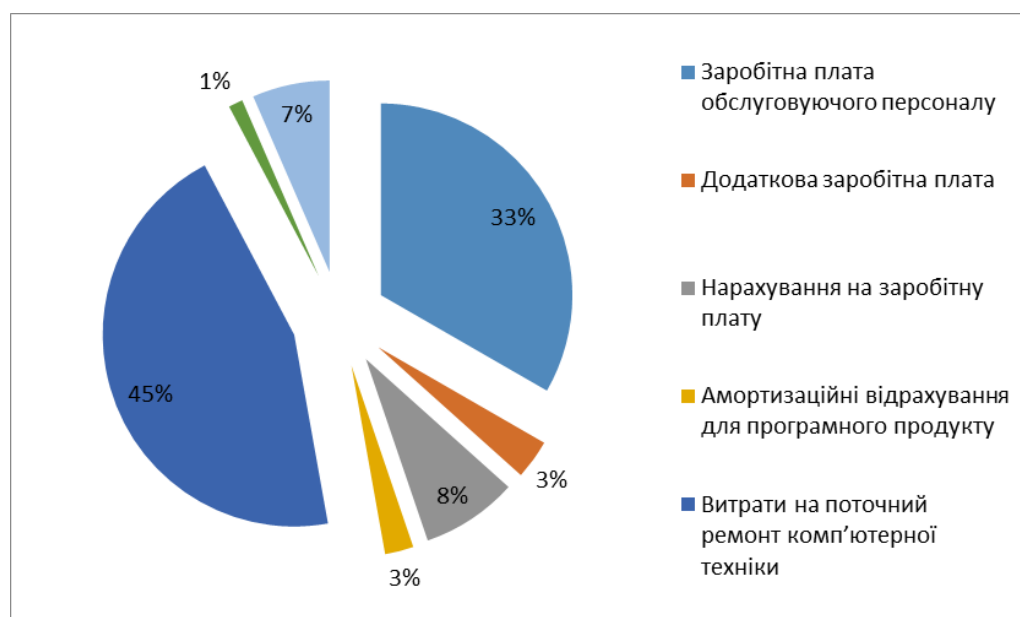


Рисунок 1.1 – Структура експлуатаційних витрат

Зм.	Арк.	№ докум.	Підпис	Дата

Таблиця 1.2 – Кошторис експлуатаційних витрат

Статті витрат	Умовні позначення	Сума грн	Структура, %
Заробітна плата обслуговуючого персоналу	$Z_{\text{обс}}$	6009,12	33,4
Додаткова заробітна плата	$Z_{\text{д}}$	600,91	3,5
Нарахування на заробітну плату	$H_{\text{ссв}}$	1454,2	8,09
Амортизаційні відрахування для програмного продукту	A	248,2	2,4
Витрати на поточний ремонт комп'ютерної техніки	P	917,2	44,8
Витрати на електроенергію	$B_{\text{с}}$	139,34	1,2
Інші витрати	$I_{\text{в}}$	468,44	6,5
Разом	E_2	9837,4	100

Розраховуємо умовний обсяг робіт Q за формулами:

$$Q_1 = \frac{F \cdot 60 \cdot \beta}{t_1} [\text{ум. од.}], \quad (1.14)$$

$$Q_2 = \frac{F \cdot 60 \cdot \beta}{t_2} [\text{ум. од.}], \quad (1.15)$$

де Q_1 – обсяг робіт із застосуванням існуючого програмного продукту, умовні одиниці;

Q_2 – обсяг робіт із застосуванням нового програмного продукту, умовні одиниці;

β – відношення часу, який витрачається працівником на виконання певних робіт із застосуванням даного програмного продукту, до загального часу його роботи – 15%;

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

t_1 та t_2 – відповідні часи виконання функції або роботи при застосуванні існуючого та нового програмного продукту, хв.

$$Q_1 = \frac{1800 \cdot 60 \cdot 0,12}{16} = 810 \text{ (ум. од.)},$$

$$Q_2 = \frac{1800 \cdot 60 \cdot 0,12}{2} = 6370,88 \text{ (ум. од.)}.$$

Розрахуємо річний економічний ефект, який можна отримати при впровадженні нового програмного продукту:

$$\Delta E = \left(\frac{E_1}{Q_1} - \frac{E_2}{Q_2} \right) \cdot Q_2 [\text{грн/рік}], \quad (1.16)$$

де E_1 – експлуатаційні витрати використання існуючого програмного продукту, грн/рік;

E_2 – експлуатаційні витрати використання нового програмного продукту, грн/рік;

Q_1 – обсяг, що отримується за рік за умови використання існуючого програмного продукту;

Q_2 – обсяг, що отримується за рік за умови використання нового програмного продукту.

$$\Delta E = \left(\frac{4320,5}{810} - \frac{9837,4}{6370} \right) \cdot 6370 = 24136,33 \text{ (грн/рік)}.$$

Розрахуємо термін окупності витрат, що використовувались на розроблення нового програмного продукту:

$$T_0 = \frac{B}{\Delta E} [\text{років}], \quad (1.17)$$

де B – загальна сума капіталовкладень;

					КВРКІ 190173.19.01.01 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

ΔE – річний економічний ефект програмного продукту, грн.

$$T_0 = \frac{9837,4}{24136,33} = 0,5 \text{ (року)}.$$

Отже, зробивши всі необхідні розрахунки, встановлено, що технологія розробки програмного продукту відповідає оптимальному рівню витрат, і, в кінцевому підсумку, розроблений продукт є економічно доцільним та конкурентоспроможним.

1.4 Висновки. Постановка задачі

Продуманий дизайн майданчика необхідний для комфортного, цікавого, а найголовніше безпечного проведення часу дітей.

У сучасному інформаційному суспільстві динамічно розвиваються технології 3D друку деталей різних геометричних форм та технології доповненої реальності, засоби рендеру та інші інноваційні напрямки.

Аналіз відомого сучасного спеціалізованого програмного забезпечення показав досить велику кількість програм для редагування і створення 3D графіки.

Виходячи із отриманих прогнозних розрахунків, можна зробити висновок, що необхідність створення бази конструктора дитячих майданчиків є актуальним та доцільним, що і підтверджують прогнозні розрахунки.

В кваліфікаційній роботі слід розв'язати наступну задачу: розробити програмно-технічні засоби 3D-конструювання дитячих майданчиків, які дають можливість побудови дитячих майданчиків з високим рівнем реалізму, мають велику базу комплексів та інструменти збереження і правки готових проектів. Ці засоби працюють без встановлення додаткових програм, не вимагають особливих навичок для роботи та наявності високопродуктивного апаратного забезпечення.

					КВРКІ 190173.19.01.01 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

2 АНАЛІЗ ЗАСОБІВ РОЗРОБЛЕННЯ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ 3D-КОНСТРУЮВАННЯ ДИТЯЧИХ МАЙДАНЧИКІВ

2.1 Характеристика та вимоги до конструкторів дитячих майданчиків

Конструктор дитячих майданчиків повинен мати ряд характеристик та вимог:

- простий, зрозумілий інтерфейс;
- процес створення проекту не повинен вимагати спеціальних навичок;
- можливість задавати розмір і форму майданчика;
- розміщувати виконані в масштабі і потрібній перспективі моделі комплексів;
- швидкий пошук необхідних комплексів;
- інструменти збереження готового проекту;
- можливість переведення проекту у так званий «вид зверху», що дозволить конструкторам працювати без створення проекту в інших програмах;
- збереження проекту в растрових форматах (JPEG, PDF);
- інструмент для відправки заявки.

Дитячий майданчик - це одне з перших місць, де дитина пізнає навколишній світ, розвивається фізично, спілкується з іншими дітьми, вчиться взаємодіяти, знаходити компроміси або відстоювати свої, нехай навіть дитячі, але вже життєві позиції. Дитячий майданчик в першу чергу повинен представляти інтерес для дитини, бути багатофункціональним та безпечним для ігор.

Діти, на відміну від дорослих, довго сидіти без діла не можуть, особливо під час прогулянок на вулиці. Дитина сама знайде собі заняття, але не завжди воно буде безпечним, тому ви як свідомий батько повинні передбачити цей момент. У кожному дворіку чи парку мають бути спеціально обладнані ігрові майданчики для дітей різного віку. Під час ігор дитина розвиває навички, просторове мислення, творчі здібності, моторику.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

Безпека дитячого майданчика регламентується єдиними нормами для виробників ігрових та спортивних майданчиків, затвердженими в далекому 1998 році, але це в Європі, а ось в Україні досі подібних законів немає, тому слід довіряти лише своїй інтуїції, очам та репутації компанії-виробника та установника майданчика.

Вимоги до елементів. Ігровий майданчик. Перше, на що потрібно звернути увагу – всі елементи мають бути спроектовані так, щоб дорослі мали доступ до всіх проміжків пересування дитини. Ігровий майданчик не повинен мати важкодоступних або «мертвих» зон, ви повинні мати можливість вільно пересуватися слідом за дитиною, щоб підстрахувати або допомогти. Майданчик повинен добре переглядатися, щоб ви бачили, де знаходиться ваша дитина. Всі елементи, які піднято над землею, повинні бути оснащені спеціальними поручнями, що захищають. Майданчик не повинен мати травмонебезпечних зон, гострих кутів необробленого металу.

Гірка. Всі дитячі гірки на дитячому майданчику повинні бути оснащені борти стартової ділянки та захисними планками. Висота гірки над рівнем землі – від 20 до 35см. Бортики вздовж гірки – від 10 до 50см. Закриті ділянки гірки не перевищують – 75см. З'їжджаючи з гірки, ваша дитина не повинна приземлятися в пісок.

Гойдалки. Найбезпечніша та зручна для експлуатації висота гойдалки становить 40 – 60см. Рекомендують фарбувати рухомі частини гойдалок у яскраві кольори.

Матеріали та фарби. Всі елементи дитячого майданчика мають бути виготовлені з екологічно безпечних матеріалів, а також фарбовані якісними лаками та фарбами. Обов'язково переконайтесь у наявності у компанії-виробника відповідних сертифікатів. Якщо ігровий комплекс якісний, то кожен його елемент може кілька сезонів відстояти як новий, якщо матеріали неякісні, то з часом ви в цьому переконаєтесь.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

Покриття. Прийнятним покриттям ігрових майданчиків є пісок (товщина якого не менше 30см), традиційний трав'яний газон або гумове покриття. Бетон, гравій та камінь – це неприпустиме покриття для дитячих майданчиків. У разі падіння дитина буде травмована.

Особливості монтажу. Правила розташування окремих елементів біля ігрового майданчика і специфіка їх монтажу здаються очевидними. При цьому серйозні порушення елементарних норм, що відповідають здоровому глузду, трапляються на кожному кроці.

З найпоширеніших порушень: надто близько один до одного розташовані окремі елементи, неякісне зварювання, про яке дряпається дитина, і згодом стає місцем утворення іржі, на гірках немає бортиків у стартовій зоні або вздовж самого полотна тощо, до нескінченності.

Деякі правила монтажу важливі для безпеки дитячого майданчика:

1) безпечною відстанню від крайніх точок одного елемента дитячого майданчика до іншого вважається 2 м.

2) опорні елементи конструкції мають бути зацементовані та вкопані в землю на глибину – 50 см над шаром цементу шар піску (не менше 20 см).

3) усі болти мають бути закриті пластиковими заглушками.

2.2 Дослідження редакторів растрової та векторної графіки

Редактор растрової графіки Adobe Photoshop CS6. Графічний редактор Adobe Photoshop CS6 розроблений фірмою Adobe Systems. Є лідером ринку комерційних засобів редагування растрових зображень. Є найвідомішим продуктом фірми Adobe. Доступний на платформах Mac OS і Microsoft Windows. Photoshop широко використовується у веб-дизайні.

Можливості Adobe Photoshop:

- зручність і простота експлуатації;
- висока якість обробки графічних зображень;

					КВРКІ 190173.19.01.01 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

- широкі можливості автоматизації обробки растрових зображень, що базуються на використанні сценаріїв;
- широкі можливості, що дозволяють виконувати операції створення і обробки зображень;
- чималий набір команд фільтрації, які дозволяють створювати найрізноманітніші художні ефекти;
- сучасний механізм роботи з кольоровими профілями, що допускають їх впровадження у файли зображень для автоматичної корекції кольорових параметрів при виведенні на друк для різних пристроїв.

Інтерфейс програми Adobe Photoshop CS6 наведено на рисунку 2.1.

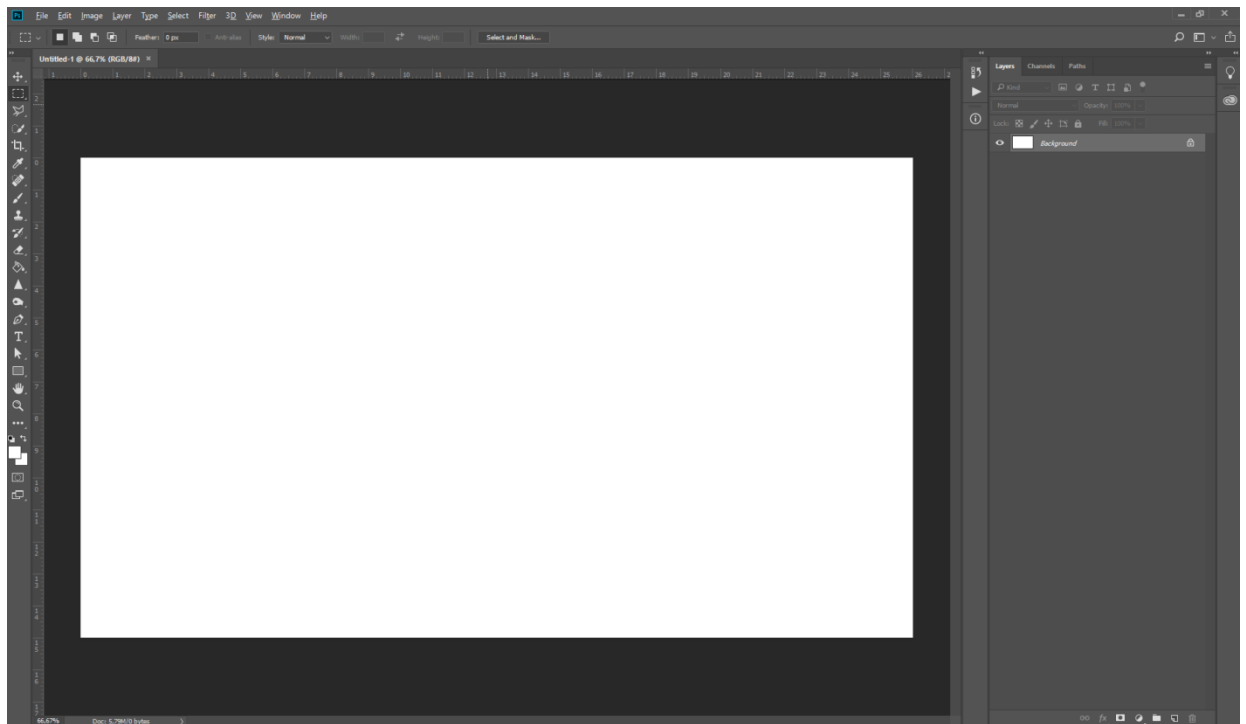


Рисунок 2.1 – Інтерфейс програми Adobe Photoshop CS6

Photoshop підтримує наступні колірні моделі (способи опису кольорів зображення):

- RGB;
- LAB;
- CMYK;

- Grayscale;
- Bitmap;
- Duotone;
- Indexed;
- Multichannel.

Підтримує обробку зображень як з традиційною глибиною кольору (8 біт, 256 градацій яскравості на канал), так і з підвищеною (16 біт, 65536 відтінків в кожному каналі). Дозволяє збереження у файлі додаткових елементів: направляючих (Guide), каналів (Alpha channel), шляхів протравки (Clipping path), шарів з векторними і текстовими об'єктами, колірних профілів (ICC), функцій перетворення кольору (transfer functions).

Редактор векторної графіки Adobe Illustrator CS. Професійний графічний редактор для редагування та створення векторної графіки від компанії Adobe. Дизайнери використовують Adobe Illustrator в самих різних цілях, у тому числі й у для роботи з ілюстраціями. Зручний для оперативної розмітки сторінки, яка містить логотип і графіку, простого односторінкового документа.

Adobe Illustrator CS має інтуїтивно зрозумілий інтерфейс, легкий доступ до багатьох функцій, широкий набір інструментів для малювання і великі можливості управління кольором, текстом, які уможливають створення векторних зображень будь-якого рівня складності.

Інтерфейс програми Adobe Illustrator CS наведено на рисунку 2.2.

					КВРКІ 190173.19.01.01 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

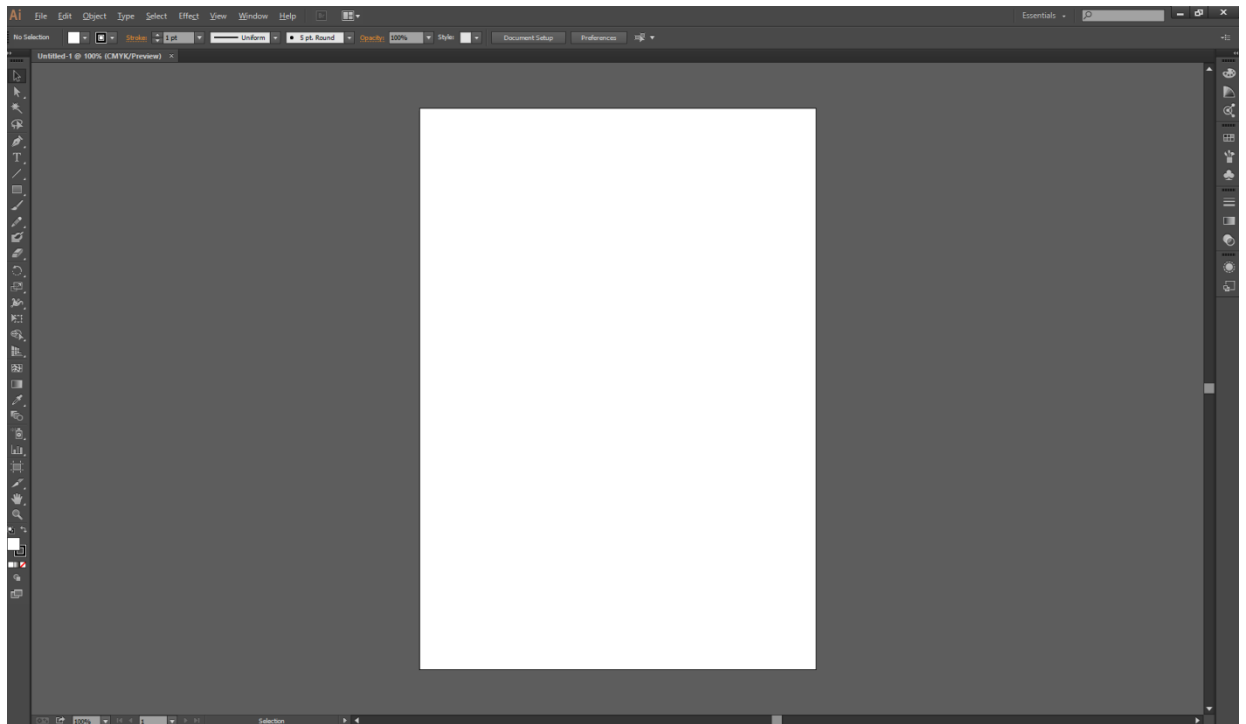


Рисунок 2.2 – Інтерфейс програми Adobe Illustrator CC

Можливості Adobe Illustrator CC:

- велика бібліотека графічних інструментів;
- інтеграція з іншими рішеннями Adobe;
- ефекти, що не залежать від розрішення екрану;
- підтримка композицій слоїв Photoshop;
- ефективне виконання буденних задач і вибраних функцій;
- ефективний інтерфейс, що легко налаштовується;
- універсальні інструменти для розділення документа на фрагменти;
- інтуїтивно зрозумілі і точні інструменти розробки;
- 3D-ефекти;
- розпилення символів;
- інструменти для побудови графіків;
- чіткість контурів в проектах для веб-додатків і мобільних пристроїв;
- інструменти для створення файлів Adobe PDF;
- підтримка графічних файлових форматів промислового стандарту.

Нові можливості:

- інструменти малювання кривих;
- інтеграція програм для мобільних пристроїв та Illustrator;
- робоче середовище, оптимізоване для сенсорного керування;
- інструменти об'єднання;
- автоматичне масштабування тексту в області;
- динамічні фігури: прямокутники та заокруглені прямокутники;
- точне замикання контурів;
- досконала робота з опорними точками;
- прив'язка до пікселя, точки або сітки;
- прискорення обробки графічним процесором у Windows;
- попередній перегляд під час роботи;
- динамічні кути;
- експорт в адаптивний формат SVG;
- власні панелі інструментів;
- імпорт і експорт налаштувань;
- інтеграція Turckit;
- синхронізація кольорів;
- синхронізація налаштувань;
- пакування файлу;
- автоматичне генерування кутів для візерункових пензлів;
- пошук шрифту;
- інструмент вільного трансформування;
- генерування CSS;
- панель Exchange.

Adobe Animate. Мультимедійна платформа компанії Adobe для створення мультимедійних презентацій або веб-додатків. Використовується для створення анімації, ігор, рекламних банерів, відтворення відео-і аудіо записів на веб-сторінках.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

Платформа містить різні засоби розробки – Adobe Flash Professional, Adobe Flash Builder, програму для відтворення flash-контенту – Adobe Flash Player, хоча flash-контент може бути відтворений і плеєрами сторонніх виробників. Наприклад, SWF-файли можуть бути переглянуті за допомогою плеєрів Gnash або swfdec, а FLV-файли – за допомогою мультимедійних програвачів Quicktime, Windows Media Player та різних програвачів в UNIX-системах за наявності відповідних плагінів.

Adobe Animate забезпечує можливість повноцінної роботи з растровою, векторною графікою і можливість обмеженої роботи з тривимірною графікою, а також забезпечує підтримку двонаправленої потокової трансляції відео та аудіо. Для мобільних пристроїв була випущена спеціальна «полегшена» версія – Flash Lite платформа, функціональність якої обмежується можливостями мобільних пристроїв та їх операційних систем.

Стандартним розширенням для flash-файлів є .SWF (Shockwave Flash або Small Web Format). Відеоролики у Flash форматі є файлами з розширенням FLV (при цьому Flash використовується лише як контейнер для відеозапису). Розширення FLA відповідає формату робочих файлів в середовищі розроблення.

Інтерфейс програми Adobe Animate наведено на рисунку 2.3.

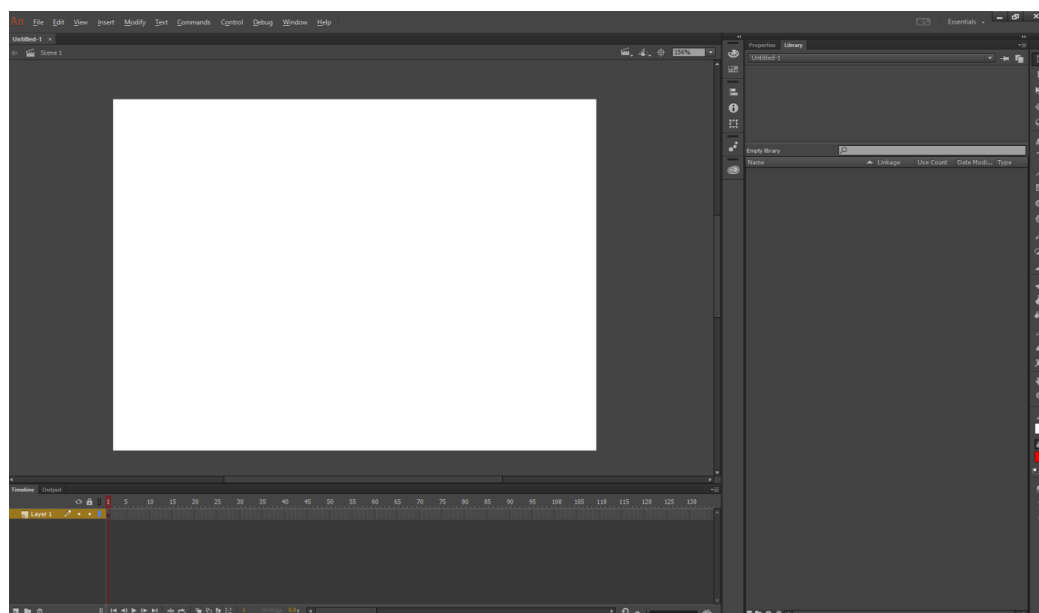


Рисунок 2.3 – Інтерфейс програми Adobe Animate

Flash Player - це віртуальна машина, на якій виконується код flash-програми. Основою анімації у Flash є векторний морфінг – плавний перехід одного ключового кадру в наступний. Це уможлиблює підготовку складних мультиплікаційних сцен із використанням лише кількох ключових кадрів. Продуктивність Flash Player при відтворенні анімації в кілька разів перевищує продуктивність віртуальної машини Javascript в браузерях, що підтримують попередній стандарт HTML5, хоча в багато разів поступається додаткам, які працюють взагалі без використання віртуальних машин.

Flash Player адаптований до мобільної платформи Android, наявні мобільні пристрої з апаратним прискоренням flash-додатків (в тому числі AIR-додатків).

2.3 Аналіз існуючих мов та середовищ програмування

Мова програмування C++. Універсальна мова програмування високого рівня, що підтримує різні парадигми програмування: як об'єктно-орієнтовану, так і процедурну.

Переваги мови C++:

- 1) продуктивність;
- 2) масштабованість;
- 3) можливість роботи на низькому рівні з пам'яттю, адресами, портами;
- 4) можливість створення узагальнених алгоритмів для різних типів даних, їх спеціалізація, і обчислення на етапі компіляції, з використанням шаблонів.

Недоліки мови C++:

- 1) наявність безлічі можливостей, що порушують принципи безпеки;
- 2) у C++ програмі може виникнути важковловима помилка;
- 3) розробники вимушені дотримуватися нетривіальних правил кодування
- 4) неможливість використовувати автоматичне управління пам'яттю (наприклад, збір «сміття»), що призводить до виникнення вразливостей типу "переповнювання буфера";

- 5) погана підтримка модульності;
- 6) серйозно уповільнення компіляції при підключенні великої кількості модулів;
- 7) нестача інформації про типи даних під час компіляції;
- 8) складність для вивчення і для компіляції;
- 9) неінтуїтивність перетворення типів;
- 10) примітивність успадкованого препроцесора C++;
- 11) неможливість або складність здійснювати деякі завдання метапрограмування;
- 12) необхідність великої кількості дій з обходу потенційних проблем.

Інші мови програмування (наприклад, Scheme і Nemerle) мають значно безпечніші і потужніші системи мета програмування.

Мова програмування C#. Об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Синтаксис C# подібний до мов C++ і Java. Мова має статичну типізацію, підтримує перевантаження операторів, підтримує поліморфізм, підтримує вказівники на функції-члени класів, події, властивості, атрибути, винятки, коментарі у форматі XML. C# не підтримує множинне спадкування класів або виведення типів.

C# - це мова компонентного програмування, що забезпечує можливість повторного використання створених компонентів. Мова C# створювалась паралельно з каркасом Framework Net, тому повністю враховує всі його можливості – як FCL, так й CLR. Мова C# - це об'єктно-орієнтована мова, де навіть вбудовані типи представлені класами, наявні можливості спадкування й універсалізації. Завдяки каркасу Framework .Net, що є надбудовою над операційною системою, програмісти C# одержують такі ж переваги роботи з віртуальною машиною, що й програмісти Java. Ефективність коду підвищується завдяки тому, що виконавче середовище CLR представляє собою компілятор проміжної мови, а бібліотека каркасів підтримує можливість побудови різних

типів додатків, дозволяючи легко будувати Web-служби, просто зберігати й одержувати інформацію з бази даних та сховищ даних.

Мова програмування Java Script. Це мова програмування, що дозволяє зробити сторінку інтерактивною, реагуючою на дії користувача.

Послідовність інструкцій виконується інтерпретатором, вбудованим в звичайний Web-браузер. Тобто, код програми вбудовується в HTML-документ і виконується на стороні клієнта. Для виконання програми не потрібне навіть перезавантаження Web-сторінки, всі програми виконуються у відповідь на будь-яку подію.

Це об'єктно-орієнтована скриптова мова програмування. В основному використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить у браузерах як мова сценаріїв для надання інтерактивності веб-сторінкам.

Основні архітектурні риси:

- динамічна типізація;
- автоматичне керування пам'яттю;
- прототипне програмування;
- функції як об'єкти першого класу.

JavaScript має властивості об'єктно-орієнтованої мови, натомість реалізоване прототипування обумовлює відмінності в роботі з об'єктами (в порівнянні з традиційними об'єктно-орієнтованими мовами). Крім того, JavaScript має ряд властивостей функціональних мов: об'єкти як списки, функції як об'єкти першого класу, анонімні функції, каррінг, замикання.

Мова програмування Action Script 3. ActionScript – це об'єктно-орієнтована мова програмування flash. В його основі лежить модель мов сценаріїв Європейської Асоціації Виробників Комп'ютерів (EMCA – European Computer Manufacturers Association).

ActionScript додає інтерактивність, обробку даних і багато іншого в вміст Flash-додатків. Виконується віртуальною машиною (ActionScript Virtual Machine),

що є складовою частиною Flash Player. Скрипти ActionScript компілюється в байткод, який включається в SWF-файл.

SWF-файли виконуються Flash Player-ом. Flash Player існує як самостійний додаток (standalone) або у вигляді плагіна до веб-браузера. В першому випадку можливе створення виконуваних exe-файлів (projector), коли swf-файл включається у Flash Player.

За допомогою ActionScript можна створювати інтерактивні мультимедіа-додатки, ігри, веб-сайти та багато іншого.

Переваги ActionScript 3:

– програмування на AS3 нічим принципово не відрізняється від програмування на мовах Java, C++, тому більшість методологій, підходів та рішень будуть співпадати;

– ActionScript 3 був створений з фокусом на ефективність та продуктивність. І не просто ефективність розробки (хоча зручність AS3 для великих проектів важко поставити під сумнів), а також на продуктивність при відтворенні;

– ActionScript 3 наділений значною кількістю нових функцій та якостей, недоступних у попередніх версіях, як от: робота над помилками, визначення типу змінних під час прогону, замкнення методу, регулярні вирази, нові типи даних, які дозволяють використовувати швидші математичні формули та багато-багато іншого.

Проведено аналіз існуючих мов програмування, оптимальним варіантом для задачі створення конструктора дитячих майданчиків є мова програмування Action Script 3.

Середовище розробки Powerflasher FDT. FDT Powerflasher – пропріетарне середовище розробки (плагін для IDE Eclipse) додатків Flex / ActionScript3, сумісний з Android, iOS, HTML5, JS, PHP. Найближчий конкурент – Adobe Flash Builder. На відміну від нього FDT не дозволяє розробляти на Flex в візуально-компонентній манері, змушуючи працювати руками. Зате багато речей, наприклад

					КВРКІ 190173.19.01.01 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

ту ж перевірку коду, вона робить набагато краще. Крім того в самий найближчий час FDT буде підтримувати розробку на мові haXe.

Інтерфейс програми FDT Powerflasher наведено на рисунку 2.4.

У загальному і цілому це середовище, вельми і вельми заслуговує на увагу кожного, хто пише або збирається писати на flash. Особливо тих, хто не використовує і не збирається використовувати Flex (для роботи з Flex краще Adobe Flash Builder).

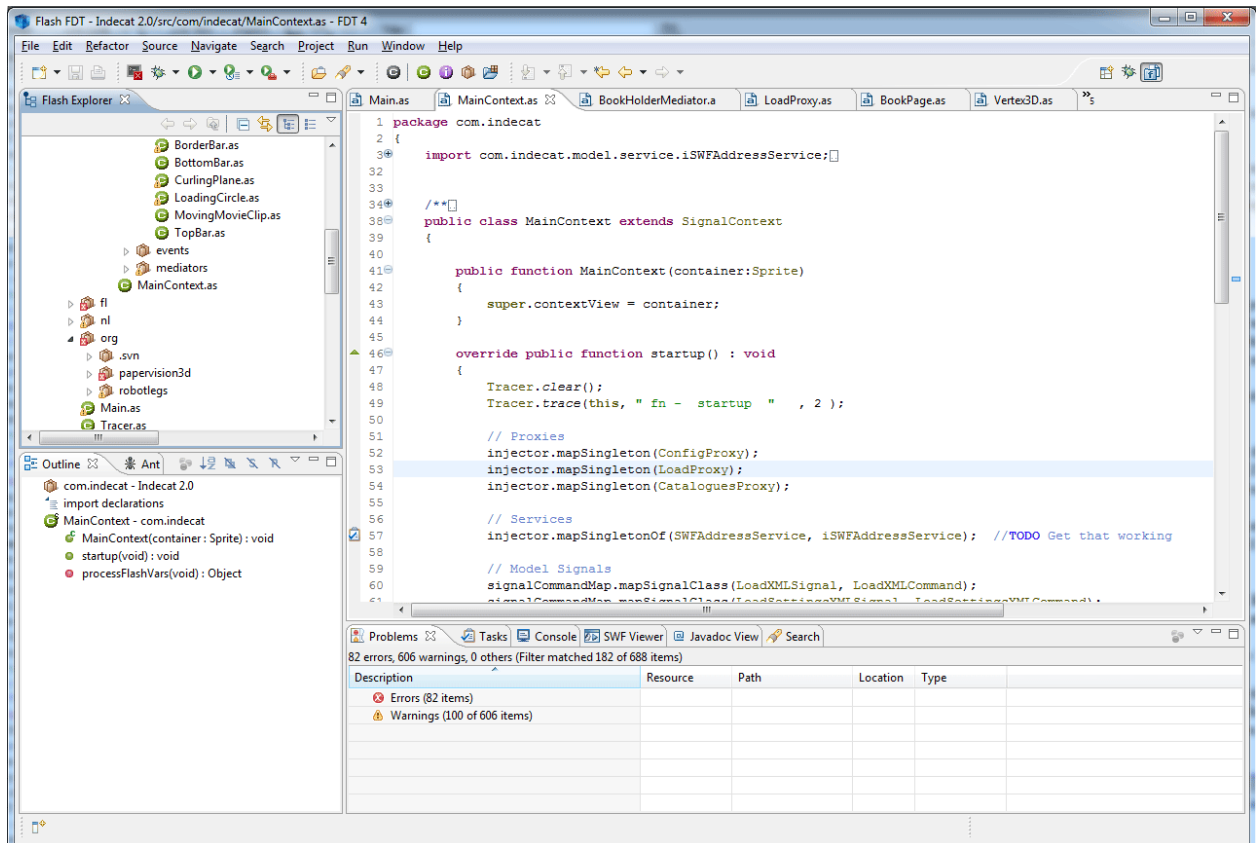


Рисунок 2.4 – Інтерфейс програми FDT Powerflasher

Середовище розробки FlashDevelop. FlashDevelop – вільне середовище розробки і редактор, написаний на C#, який використовує для виведення тексту компонент Scintilla, що дозволяє створювати Flash-додатки за допомогою Flex SDK, MTASC або haXe.

Є однією з альтернатив Adobe Flash Builder для розробки веб-додатків (завдяки своїй безкоштовності і помітно більшій швидкодії) хоча і не володіє всіма можливостями останнього. Підтримує профілювання, налагодження Adobe

Flex і має розумне автодоповнення при написанні коду на мові ActionScript. Також при використанні Adobe Flex SDK 4.5.1 можна писати програми для платформ Android, iOS.

Для установки FlashDevelop потрібно Microsoft .NET 2.0 framework. JRE потрібно для компіляції Flex і AIR додатків. За допомогою haXe можна скомпілювати для платформ Flash, JavaScript, NekoVM, також можна скомпілювати бінарні виконувані додатки для Linux і Windows. Для компіляції ActionScript 2 до складу дистрибутива включено MTASC.

Інтерфейс програми FlashDevelop наведено на рисунку 2.5.

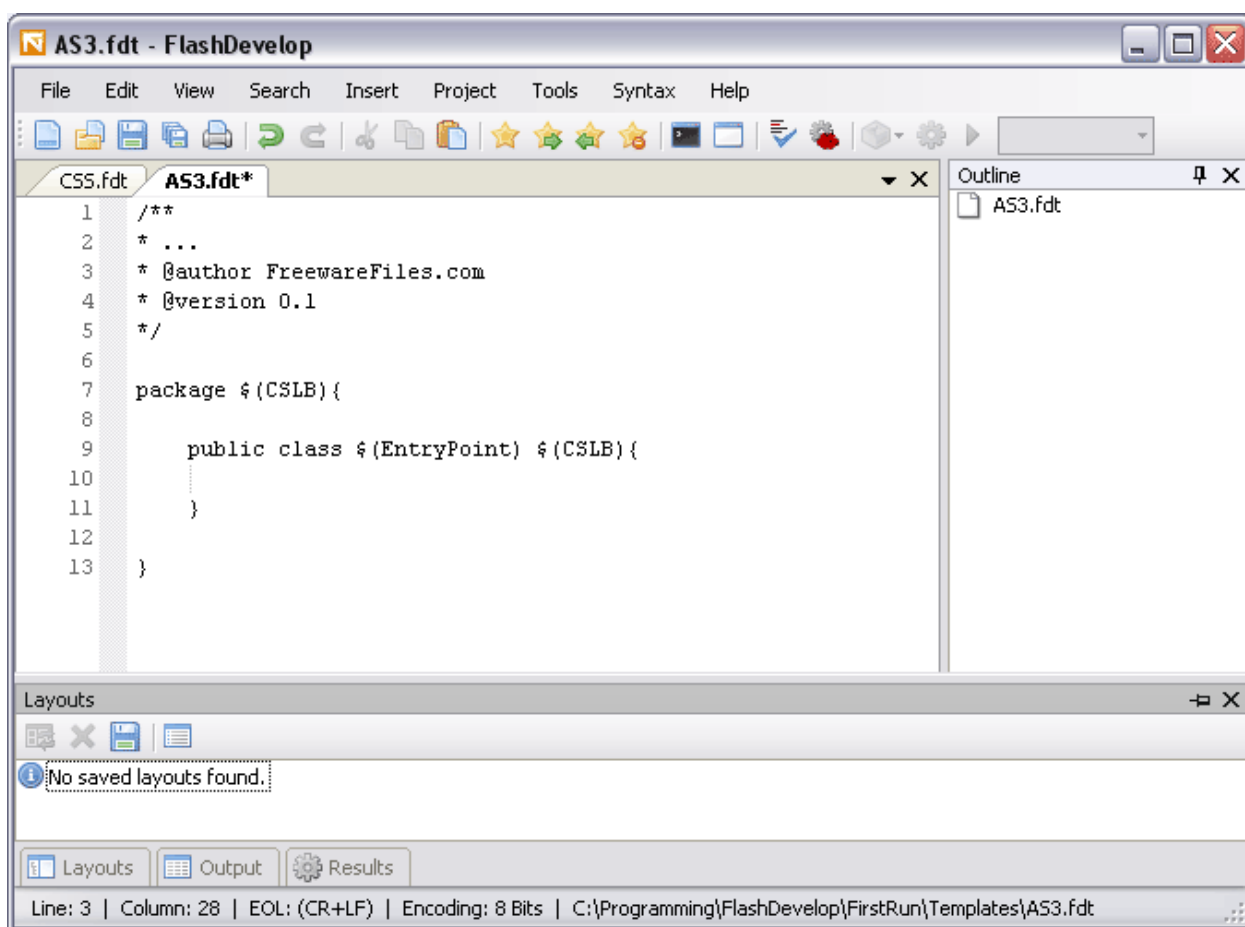


Рисунок 2.5 – Інтерфейс програми FlashDevelop

Можливості та особливості:

- підсвічування синтаксису для ActionScript, Python, HTML, XML, PHP, CSS, HaXe;

- автодоповнення для ActionScript 2, ActionScript 3 + MXML і HaXe;
- можливість розширення додаванням (написанням) плагінів;
- можливість включення / відключення окремих плагінів для підвищення швидкодії;

- фолдінг коду.

Середовище розробки Adobe Flash Builder. Програмне забезпечення Adobe Flash Builder (раніше – Flex Builder) – це інтегроване середовище розробки багатофункціональних міжплатформених інтернет-додатків (RIA), яка використовує відкрите середовище Flex. Рішення Adobe Flash Builder, створене на базі Eclipse, дозволяє створювати настільні, мобільні і web-додатки з більш високою продуктивністю і скорочує час розробки. Інтерфейс наведено на рисунку 2.6.

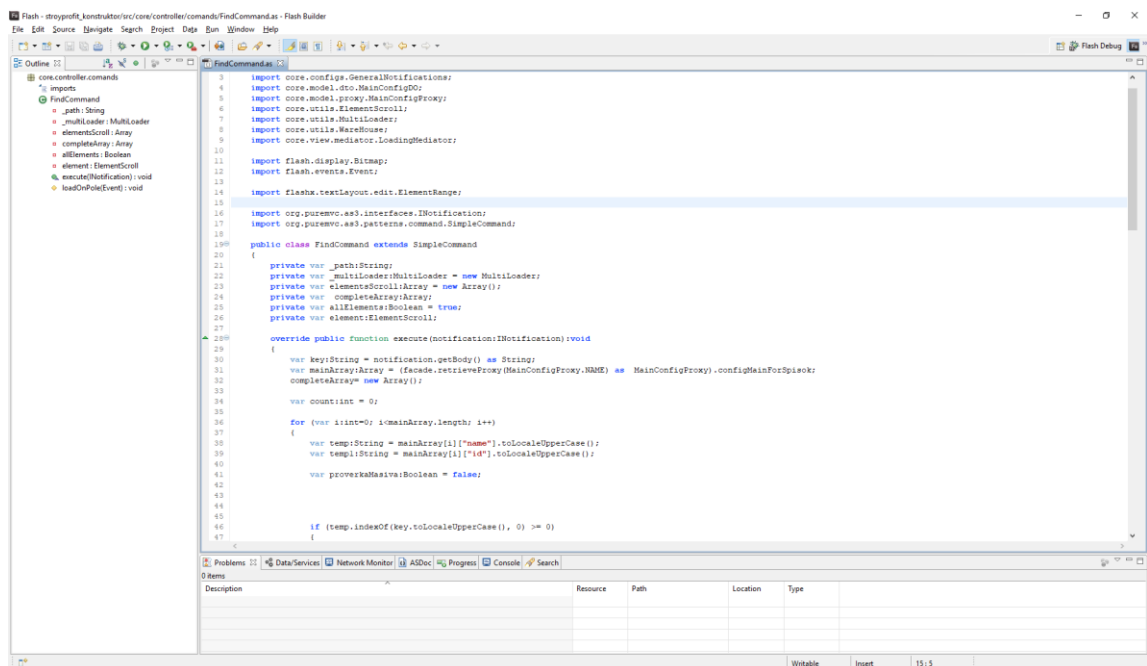


Рисунок 2.6 – Інтерфейс програми Adobe Flash Builder

Основні можливості Flash Builder:

- потужні інструменти створення коду – високопродуктивне середовище IDE на основі Eclipse містить редактори для MXML, Adobe ActionScript і CSS, підтримує підсвічування синтаксису, автозаповнення, згортання коду, інтерактивне покрокове налагодження та автоматичне створення типового коду;

Зм.	Арк.	№ докум.	Підпис	Дата

- багатofункціональна візуальна компоновка – дизайн створюється в візуальному режимі з використанням великої кількості встановлених компонентів і попереднім переглядом макета призначеного для користувача інтерфейсу, оформлення та поведінки;
- розробка додатків, орієнтованих на дані – сервіси Java, PHP, Adobe ColdFusion, REST і SOAP використовуються для відображення методів і властивостей в новому вікні Data / Service Explorer (Оглядач даних / сервісів);
- візуалізація інтерактивних даних – є можливість створювати панелі управління даними і проводити інтерактивний аналіз даних, розміщуючи обраний тип діаграми і прив'язуючи її до джерела даних з використанням бібліотеки діаграм Flex;
- змінні оболонки і стилі – налаштовується зовнішній вигляд програми за допомогою редакторів графічних властивостей CSS. Задаються типові властивості і видно результат в режимі «Дизайн»;
- інтеграція із засобами проектування Adobe Creative Suite – імпортуються елементи проекту, створені в програмах Adobe Flash Professional, Illustrator, Photoshop і Fireworks, або цілі прикладні інтерфейси користувача, створені в програмі Flash Catalyst;
- вбудована підтримка Adobe AIR – Flash Builder містить всі необхідні інструменти для створення, налагодження, об'єднання в пакети і підписування додатків AIR. Adobe AIR дозволяє розробляти додатки RIA для настільних комп'ютерів, застосовуючи напрацювання і базу кодів, що використовуються при створенні RIA для браузерів;
- рефакторинг коду – Flash Builder підтримує додаткові можливості зміни внутрішньої структури коду: переміщення і зміни структури;
- потужні засоби тестування – у програмному забезпеченні Flash Builder реалізована підтримка інструментів автоматичного функціонального тестування (наприклад, HP QuickTest Professional). За допомогою засобів управління профілями пам'яті і продуктивності відстежується і аналізується споживання

пам'яті та цикли роботи процесора;

- монітор – створюються докладні контрольні записи всіх даних, що передаються між локальним додатком Flex і сервером, для поліпшення процесу налагодження і підвищення продуктивності робочого процесу;

- зручні сервіси для роботи з даними – для двійкової, високопродуктивної передачі даних по протоколу HTTP, використовується технологія BlazeDS з відкритим кодом. Для доставки даних в додатки в режимі реального часу і публікації повідомлень і підписки на них додаються модулі Adobe LiveCycle Data Services ES2;

- компонування додатків в командному рядку – є можливість автоматизованого виконання завдань за допомогою функції компонування додатків в командному рядку;

- інтеграція із засобами модульного тестування Flex – автоматизовано функціональне тестування за допомогою середовища модульного тестування Flex;

- підтримка ASDoc – за допомогою інструменту ASDoc виводяться коментарі в редакторах MXML і ActionScript.

Проведено аналіз середовищ програмування, вибрано Adobe Flash Builder. Завдяки зрозумілому інтерфейсу, вбудованому редактору Adobe ActionScript і CSS, рефакторингу коду і потужних засобів для тестування, Adobe Flash Builder є оптимальним варіантом для виконання поставленої задачі.

2.4 Висновки

В другому розділі виконано аналіз засобів розроблення програмно-технічних засобів 3D-конструювання дитячих майданчиків.

Встановлено, що продуктивність Flash Player при відтворенні анімації в кілька разів перевищує продуктивність віртуальної машини Javascript в браузерях, що підтримують попередній стандарт HTML5. Flash Player портований на мобільну платформу Android.

					КВРКІ 190173.19.01.01 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

Проведено аналіз існуючих мов програмування, оптимальним варіантом для задачі створення конструктора дитячих майданчиків є мова програмування Action Script 3.

Проведено аналіз середовищ програмування, вибрано Adobe Flash Builder. Завдяки зрозумілому інтерфейсу, вбудованому редактору Adobe ActionScript і CSS, рефакторингу коду і потужних засобів для тестування, Adobe Flash Builder є оптимальним варіантом для виконання поставленої задачі.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНО-ТЕХНІЧНИХ ЗАСОБІВ 3D-КОНСТРУЮВАННЯ ДИТЯЧИХ МАЙДАНЧИКІВ

3.1 Етапи розроблення програмно-технічних засобів

Для створення масштабованих і легко підтримуваних додатків з високим пріоритетом потрібен поділ інтересів, а так само герметичність і напрямок зв'язків між рівнями додатку. Тому було прийнято рішення про використання фреймворку PureMVC, а також сформовано ряд етапів:

- розробка технічного завдання;
- створення концепції дизайну;
- підключення фреймворку PureMVC ;
- реалізація модуля завантаження ресурсів;
- створення модуля головного поля;
- створення модуля зміни серій моделей;
- створення модуля управління моделями;
- тестування;
- відладка.

На етапі розробки технічного завдання першочергово були встановлені потреби користувачів, а також сформульовано його основні задачі. Були встановлені вимоги до дизайну, функціоналу та наповнення. Цьому етапу надавалася особлива увага: від завдання залежать технічні особливості майбутнього продукту. Упустивши навіть незначну деталь та не заклавши її в архітектуру додатку, можна зіткнутися із необхідністю починати всю роботу спочатку.

На наступному етапі проведено розробку дизайну головного поля та допоміжних панелей.

Потім створено необхідні модулі, а саме модуль завантаження ресурсів, модуль головного поля, модуль зміни серій моделей, модуль управління моделями.

					КвРКІ 190173.19.01.01 ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

Далі відбулося тестування, після якого було зафіксовано всі помилки для обов'язкового виправлення. Після виправлення повторно проведено тестування.

Завершальною стадією розробки є відладка. Оскільки при проектуванні не можливо спрогнозувати та передбачити всі особливості реального використання, то поява помилок на першому етапі роботи дуже імовірна. Для усунення помилок і здійснюється відладка, яка займає половину всього часу розробки.

Модулі конструктора розроблені в даному проекті наведені на рисунку 3.1.

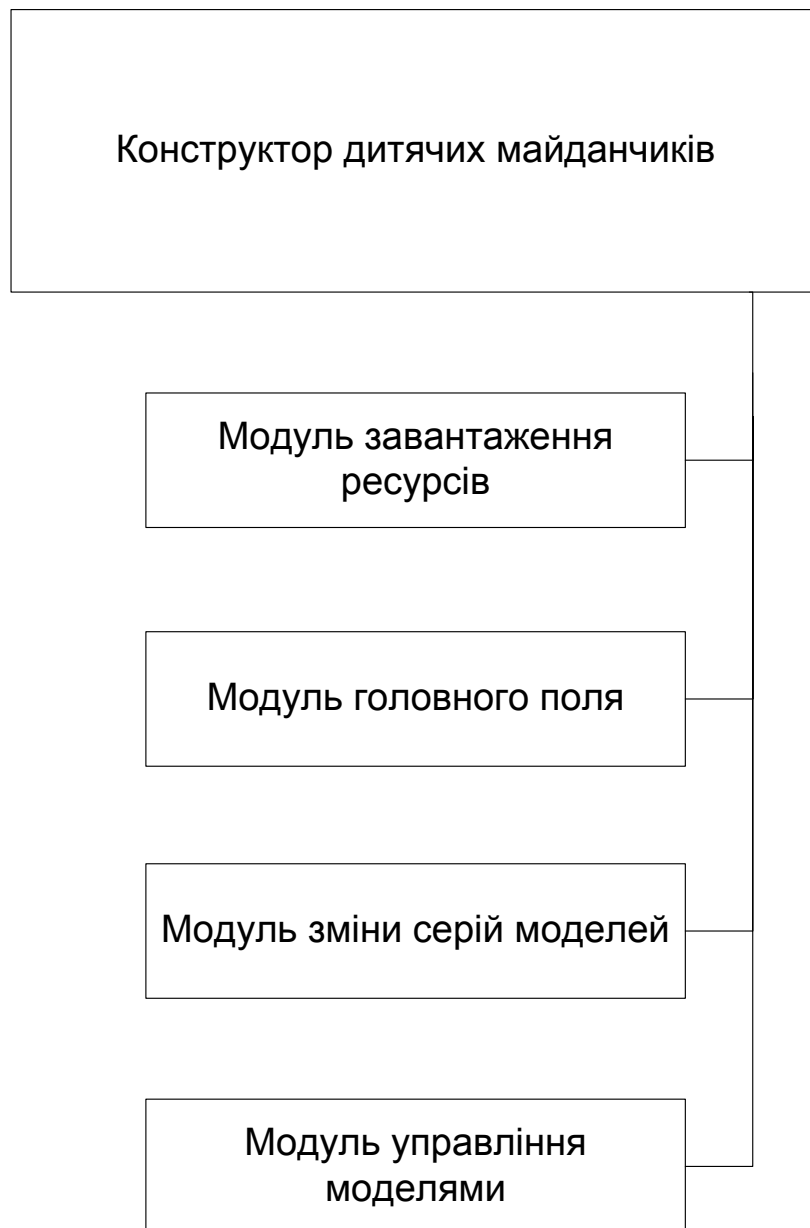


Рисунок 3.1 – Загальна структура конструктора

У даній кваліфікаційній роботі було розроблено частину від загальної структури проекту, при розробці відповідних модулів, створено інтерфейси для зв'язку з іншими модулями конструктора.

Мінімальні технічні вимоги до апаратного забезпечення користувача для роботи з конструктором дитячих майданчиків:

- одноядерний процесор з частотою 2GHz;
- об'єм оперативної пам'яті не менше 1G;
- об'єм вільного місця на жорсткому диску не менше 2,5 Gb;
- відеокарта з об'ємом пам'яті не менше 128 Mb;
- будь-який браузер з підтримкою флеш плеєра, або самостійний додаток з підтримкою флеш-контенту.

3.2 Дизайн 3D-конструктора дитячих майданчиків

У відповідності з розробленою структурою було спроектовано головне поле конструктора, та ряд панелей, панель налаштування параметрів поля наведено на рисунку 3.2.



Рисунок 3.2 – Панель налаштування параметрів поля

Як і планувалось, в центрі розміщене основне поле, на якому будуть розміщені комплекси. Зліва розташована панель для зміни серій комплексів, вона включає в себе форму вибору серії та елемент скролінгу. З правої сторони знаходиться панель, яка відповідає за розмір та вид основного поля.

Графічний дизайн було розроблено в редакторі растрової графіки Adobe Photoshop CS6. Окремі елементи були створені засобами редактора векторної графіки Adobe Illustrator CC. Анімації та кнопки створені за допомогою Adobe Animate.

Фреймворк PureMVC має дуже вузьку мету. Вона полягає в тому, щоб допомогти розділити інтереси коду програми на три окремі рівні: Модель (Model), Представлення (View) і Контролер (Controller).

У даній реалізації класичного мета-патерну MVC ці три рівня програми управляються трьома Сінглтонами (класами, для яких можливе існування одного і тільки одного примірника), які називають: Модель (Model), Представлення (View) і Контролер (Controller). Разом вони називаються Базовими класами (Core actors).

Четвертий Сінглтон, Фасад (Facade), спрощує розробку, надаючи єдиний інтерфейс для повідомлення з Базовими класами.

Модель просто кешує іменовані посилання до Проксі. Код Проксі маніпулює моделлю даних, зв'язуючись з віддаленими сервісами, якщо потрібно зберегти або зробити запит.

Таким чином, Модель даних ізольована від Контролеру та Представлення. Це призводить до легко переносимого коду Моделі. Приклад наведено нижче:

```
package core.model.dto
{
    public class MainConfigDO
    {
        public var id:String;
        public var type:String;
        public var kategory:String;
        public var name:String;
        public var numRazdel:int;
        public var nameRazdel:String;
        public var widht:int;
        public var height:int;
    }
}
```

					КВРКІ 190173.19.01.01 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

Представлення в першу чергу кешує іменовані посилання на Медіатори. Код Медіатора обслуговує компонент(и) Представлення, додаючи до них Слухачів подій і від їх імені надсилаючи та приймаючи оповіщення до і від решти системи, при цьому безпосередньо керуючи їх станом.

Приклад виконання основного Медіатора наведено нижче, усі інші були створені на його основі.

```
package core.view.mediator
{
    import core.configs.GeneralNotifications;
    import core.view.components.ViewLogic;
    import org.puremvc.as3.patterns.mediator.Mediator;

    public class UIMediator extends Mediator
    {
        public function UIMediator(NAME:String = null,
viewComponent:ViewLogic = null)
        {
            super(NAME, viewComponent);
        }
        public function get viewLogic():ViewLogic
        {
            return viewComponent as ViewLogic;
        }
        override public function onRegister():void
        {
            super.onRegister();
sendNotification(GeneralNotifications.ADD_CHILD_TO_ROOT,
(viewComponent as ViewLogic).content);
        }
        override public function onRemove():void
        {
            sendNotification(GeneralNotifications.REMOVE_CHILD_FROM_ROOT,
(viewComponent as ViewLogic).content);
            super.onRemove();
        }
    }
}
```

Приклад виконання основного Представлення наведено нижче, усі інші були створені на його основі.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

```

package core.view.components
{
    import core.utils.WareHouse;
    import flash.display.DisplayObject;
    import flash.display.DisplayObjectContainer;
    import flash.display.SimpleButton;
    import flash.events.Event;
    import flash.events.EventDispatcher;
    import flash.events.MouseEvent;

    public class ViewLogic extends EventDispatcher
    {
        private var _content:DisplayObject;
        public function ViewLogic(NameAsset:String, x:int=0,
y:int=0)
        {
            _content =
WareHouse.getInstance().getAsset(NameAsset);
            _content.x = x;
            _content.y = y;
        }
        public function get content():DisplayObjectContainer
        {return _content as DisplayObjectContainer;}
        private function onRemove():void
        {
            var i:int = 0;
            while (i < content.numChildren)
            {
                var buttonElement:DisplayObject =
content.getChildAt(i) as DisplayObject;
                i++;
            }
            if( _content && _content.parent )
                _content.parent.removeChild( _content
);
        }
    }
}

```

Контролер кешує посилання на класи Команд, створюючи екземпляри класу команди в той момент, коли виникає необхідність виконання цієї команди і знищуючи після виконання. Приклад виконання команди наведено нижче:

```

package core.controller.comands
{
    import core.configs.GeneralNotifications;
    import core.model.proxy.MainConfigProxy;
    import core.view.mediator.LeftPanelMediator;
    import core.view.mediator.PoleMediator;
    import core.view.mediator.RightPanelMediator;

```

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

```

import core.view.mediator.TopPanelMediator;
import core.view.mediator.VerhPanelMediator;
import org.puremvc.as3.interfaces.INotification;
import org.puremvc.as3.patterns.command.SimpleCommand;

public class StartRegistersPanelsCommand extends
SimpleCommand
{
    override public function
execute(notification:INotification):void
    {
        sendNotification(GeneralNotifications.CLOSE_LOADING);
        facade.registerMediator(new PoleMediator());
        sendNotification(GeneralNotifications.LOAD_NEW_RAZDEL_COMMAND,
"nordika");

        var arr:Array =
(facade.retrieveProxy(MainConfigProxy.NAME) as
MainConfigProxy).configMainRazdel;
        facade.registerMediator(new
LeftPanelMediator(arr));
        facade.registerMediator(new
RightPanelMediator());
        facade.registerMediator(new
VerhPanelMediator());
    }
}

```

Команди можуть запитувати Проксі і взаємодіяти з ними, відправляти Сповідження, виконувати інші команди, і часто використовуються для диригування складними, що охоплюють всю або майже всю систему діями, такими, як запуск або зупинка програми. Це альма-матер бізнес-логіки програми.

Фасад (Facade), ще один Сінглтон, ініціалізує Базові класи (Модель, Представлення і Контролер) і надає єдину точку доступу до всіх їх публічних методів. Код Фасаду наведено нижче:

```

package core
{
    import core.configs.GeneralNotifications;
    import core.controller.comands.StartupCommand;
    import core.model.dto.ParamsDTO;
    import flash.events.MouseEvent;
    import flash.external.ExternalInterface;
    import org.puremvc.as3.patterns.facade.Facade;

    public class GameFacade extends Facade
    {

```

					КВРКІ 190173.19.01.01 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        public static function getInstance(): GameFacade
        {
            if (instance == null)
            {instance = new GameFacade();}
            return instance as GameFacade;
        }

        public function startup(command:Class,
root:Sprite):void
        {
            registerCommand(GeneralNotifications.STARTUP,
command);
sendNotification(GeneralNotifications.STARTUP, root);
            root.addEventListener(MouseEvent.MOUSE_UP,
onUp);

            root.stage.addEventListener
(MouseEvent.MOUSE_UP, onUp);
            var square:Shape = new Shape();
            square.graphics.beginFill(0xff0000);
            square.graphics.drawRect(0, 0, 1280, 800);
            square.graphics.endFill();
        }
    }
}

```

Розширюючи Фасад, додаток отримує в розпорядження весь функціонал Базових класів без необхідності їх імпорту і прямої роботи з ними. Реалізується конкретний Фасад тільки один раз, і це робиться дуже просто.

Проксі, Медіатори і Команди можуть використовувати конкретний Фасад додатку для того, щоб отримувати доступ і зв'язуватися один з одним.

Виходячи зі стратегії даного фреймворку було сформовано таку структуру каталогів і класів проекту, яка зображена на рисунку 3.3.

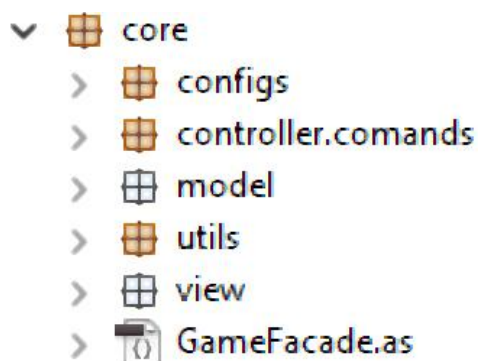


Рисунок 3.3 – Структура каталогів і класів проекту

PureMVC-додатки можуть виконуватися в середовищах, відмінних від Flash Player, не маючи доступу до його подій (класи Event і EventDispatcher), так що фреймворк реалізує схему сповіщень Спостерігач (Observer) для сполучення між Базовими класами MVC і іншими частинами системи в виді слабкого зв'язування.

3.3 Основні модулі 3D-конструктора дитячих майданчиків

Модуль завантаження ресурсів. У зв'язку з великою базою моделей, виникла необхідність в модулі завантаження ресурсів. Алгоритм модуля такий – під час запуску завантажуються файл ресурсів необхідних для дизайну конструктора, а саме файл main.swf, файл першої серії комплексів (nordika.swf), а також файл з налаштуваннями (main_xml.xml). Всі інші ресурси завантажуються за необхідністю. Деякі серії розділені на декілька файлів ресурсів, у зв'язку з їх великим розміром. Обсяг усіх файлів ресурсів близько 2 Gb.

Алгоритм роботи модуля завантаження показано на рисунку 3.4.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

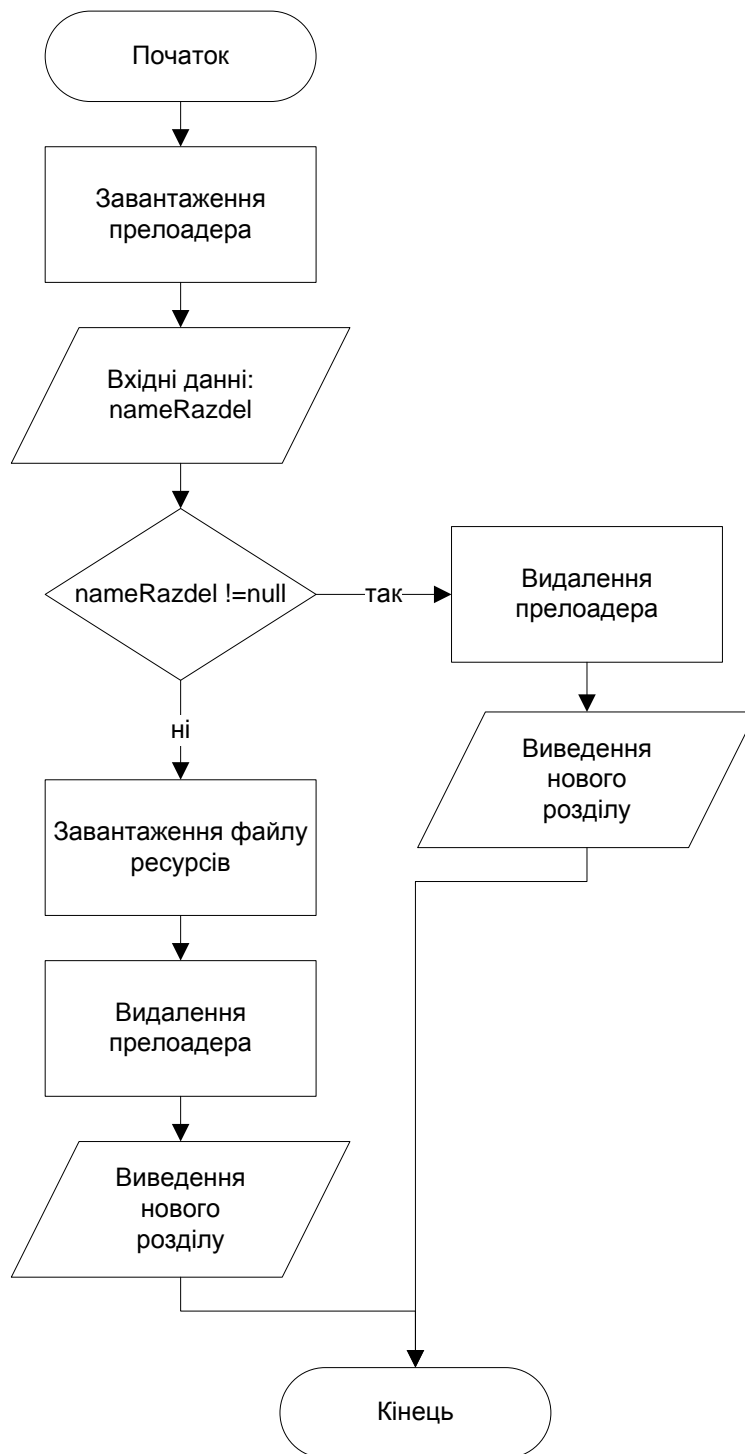


Рисунок 3.4 – Алгоритм роботи модуля завантаження ресурсів

За допомогою класу LoadNewRazdelCommand виконується перевірка на наявність завантаженого розділу моделей, якщо його ще не завантажено виконується команда «LOAD_NEW_RAZDEL». Ця команда відповідає за процес завантаження.

```

public class LoadNewRazdelCommand extends SimpleCommand
{
    private var _multiLoader:MultiLoader = new
MultiLoader;
    private var _url:String;
    private var nameRasdel:String;

    override public function
execute(notification:INotification):void
    {facade.registerMediator(new
LoadingMediator());
    nameRasdel = notification.getBody() as
String;
    if (WareHouse.getInstance().
_arrWareH[nameRasdel] != null)
    {
        sendNotification(
GeneralNotifications.CLOSE_LOADING);
        sendNotification(GeneralNotifications.
LOAD_NEW_RAZDEL, nameRasdel);
    }
    else
    {_url = "res/";
_multiLoader.addTask(_url+nameRasdel+".swf",
nameRasdel, "movie");
_multiLoader.start();
_multiLoader.addEventListener(Event.COMPLETE,
assetsCompleteLoadHandler);}
    protected function
assetsCompleteLoadHandler(event:Event):void
    {
        WareHouse.getInstance().setAsset(nameRasdel,
_multiLoader.getContentByRegistrationName(nameRasdel) as
DisplayObject);
_multiLoader.removeEventListener(Event.COMPLETE,
assetsCompleteLoadHandler);
sendNotification(GeneralNotifications.CLOSE_LOADING);
sendNotification(GeneralNotifications.LOAD_NEW_RAZDEL, nameRasdel);
    }
}
}

```

Доступ до усіх ресурсів виконується за допомогою класу WareHouse, який має ряд методів:

- перевірка на наявність завантаженого розділу;
- метод завантаження ресурсів;
- метод отримання класу конкретного комплексу;

Реалізацію перевірки на наявність завантаженого розділу наведено нижче:

					КВРКІ 190173.19.01.01 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

```

public function hasAsset(className:String):Boolean
{
    var bool:Boolean = false;
    for each (var item:Loader in _arrWareH)
    {
        if
(item.contentLoaderInfo.applicationDomain.hasDefinition(className))
        {
            bool =
item.contentLoaderInfo.applicationDomain.hasDefinition(className);
            break;
        }
    }
    return bool;
}

```

Реалізацію завантаження ресурсів наведено нижче:

```

public function setAsset(Name:String, content:DisplayObject):void
{
    _arrWareH[Name] = content;
}

```

Реалізацію отримання класу конкретного комплексу наведено нижче:

```

public function getAssetClass(className:String):Class
{
    var MyClass:Class = null;
    for each (var item:Loader in _arrWareH)
    {
        if (item.contentLoaderInfo.applicationDomain.
hasDefinition(className))
        {
            MyClass = (item.contentLoaderInfo.
applicationDomain.getDefinition(className)) as Class;
            break;
        }
    }
    return MyClass;
}

```

Клас Warehouse створено для зручного доступу до завантажених ресурсів, вся графіка зберігається в ньому.

Модуль головного поля. Модуль головного поля виконує такі функції:

– змінна розміру поля;

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

- змінна типу покриття поля;
- змінна типу поля (форми поля);
- змінна масштабу поля.

Алгоритм роботи модуля головного поля виконаний за допомогою кнопок і перемикачів, у разі їх зміни проходить зміна основного поля.

Приклад організації зміни типу покриття наведено нижче.

```
switch (ParamsDTO.getInstance().typePokritiya){
case "pesok":
    res1 = (WareHouse.getInstance().getAsset("res_pesok") as
MovieClip).getChildAt(0) as Shape;
    bd_res1 = new BitmapData( res1.width, res1.height );
    bd_res1.draw( res1 );
    poleSprite.graphics.beginBitmapFill(bd_res1);
    break;
}
```

Змінна розмірів поля здійснюється за допомогою клавіатури. Для організації введення було створено окремий клас `KeyboardProxy`, який слідкує за діями з клавіатурою, код наведено нижче:

```
package core.model.proxy
{
    import core.configs.GeneralNotifications;
    import flash.display.Sprite;
    import flash.events.KeyboardEvent;
    import org.puremvc.as3.patterns.proxy.Proxy;

    public class KeyboardProxy extends Proxy
    {
        static public const NAME:String = "KeyboardProxy";
        public var turnList:Array;
        private var _key:String;

        public function KeyboardProxy(root:Sprite)
        {
            super(NAME, data);

            root.stage.addEventListener(KeyboardEvent.KEY_DOWN, onKey);
            root.stage.addEventListener(KeyboardEvent.KEY_UP, onKeyOff);
        }
    }
}
```

```

        protected function onKeyOff(event:KeyboardEvent):void
        {
            if(event.keyCode==32)
            {
                sendNotification(GeneralNotifications.KEY_PRESS_PROBEL_OFF);
            }
        }

        protected function onKey(event:KeyboardEvent):void
        {
            if(event.keyCode==13)
            {
                sendNotification(GeneralNotifications.KEY_PRESS_ENTER);
            }
            if(event.keyCode==32)
            {
                sendNotification(GeneralNotifications.KEY_PRESS_PROBEL);
            }
        }
    }
}

```

Організовано правило для введення тільки цифр в поля для зміни розмірів і обмеження на кількість символів, код наведено нижче:

```

txtheight.restrict = "0-9.";
txtwidht.restrict = "0-9.";
txtheight.maxChars = 4;
txtwidht.maxChars = 4;

```

Також реалізовано довільну форму поля. Це являє собою поле нестандартного розміру і форми за допомогою кривих Безьє, приклад створення такої форми показано рисунку 3.5.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.5 – Процес створення довільної форми поля

Крива Безьє – параметрично задана крива, яка використовується в комп'ютерній графіці та суміжних областях.

У векторній графіці, криві Безьє використовуються для моделювання гладких кривих, які можна масштабувати до нескінченності. «Шляхи», як їх зазвичай називають у програмах для роботи з зображеннями, є комбінаціями з'єднаних кривих Безьє. «Шляхи» не обмежуються розмірами растрових зображень, і їх редагування є інтуїтивно зрозумілим. Криві Безьє також використовуються в анімації в якості інструмента для управління рухом та в системах автоматизованого проектування.

Фрагмент коду, що відповідає за створення шейпу з довільною кількістю точок наведено нижче.

```
var star_commands2:Vector.<int> = new Vector.<int>(newVect.length,
true);
for (var i:int=0; i<newVect.length/2; i++)
{
    star_commands2[i] = 2;
}
star_commands2[0] = 1;
dopLinii.graphics.clear();
```

```
dopLinii.graphics.lineStyle(1,0xcedece,1);
dopLinii.graphics.drawPath(star_commands2, newVect, "nonZero");
dopLinii.graphics.endFill();
```

Також організовано можливість додавання доріжок на поле (рис. 3.6).



Рисунок 3.6 – Процес додавання доріжок

Доріжки можуть бути двох типів, тип і розмір задається за допомогою контекстного меню. Контекстне меню налаштування наведено на рисунку 3.7.



Рисунок 3.7 – Контекстне меню налаштування доріжок

Реалізацію додавання доріжок наведено нижче:

```
public function create (widthDorogka:Number,
highDorogka:Number, x:Number, y:Number, material:String):void
{
    materialDorogki = material;
    var square:Sprite = new Sprite();
    var dorogka:Shape =
(WareHouse.getInstance().getAsset(materialDorogki) as
MovieClip).getChildAt(0) as Shape;
    var bd:BitmapData = new BitmapData( dorogka.width,
dorogka.height );
    bd.draw( dorogka );
    square.graphics.beginBitmapFill(bd);
    square.graphics.lineStyle(5,
ParamsDTO.getInstance().colorKant);
    var drawCoords:Vector.<Number> = new
Vector.<Number>();
    var drawCommands:Vector.<int> = new Vector.<int>();
    var star_commands:Vector.<int> = new Vector.<int>(5,
true);

    star_commands[0] = 1;
    star_commands[1] = 2;
    star_commands[2] = 2;
    star_commands[3] = 2;
    star_commands[4] = 2;
    var star_coord:Vector.<Number> = new
Vector.<Number>(10, true);
    star_coord[0] = 0;
    star_coord[1] = 0;
    star_coord[2] = widthDorogka*50;
    star_coord[3] = 0;
    star_coord[4] =widthDorogka*50;
    star_coord[5] = highDorogka*42.5;
    star_coord[6] = 0;
    star_coord[7] = highDorogka*42.5;
    star_coord[8] = 0;
    star_coord[9] = 0;
    square.graphics.drawPath(star_commands, star_coord,
"nonZero");

    square.graphics.endFill();
    konteiner.addChildAt(square, 0);
    square.x = x;
    square.y = y;
```

Модуль головного поля надає ряд інструментів редагування, а також для переміщення, повороту, дублювання та видалення комплексів та доріжок.

					КВРКІ 190173.19.01.01 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

Модуль зміни серій моделей. Модуль зміни серій моделей виконує такі функції:

- зміну серії моделей;
- завантаження ресурсів серії;

Елемент скролінгу створений за допомогою коду та векторної графіки.

Алгоритм роботи модуля зміни серій показано на рисунку 3.8.

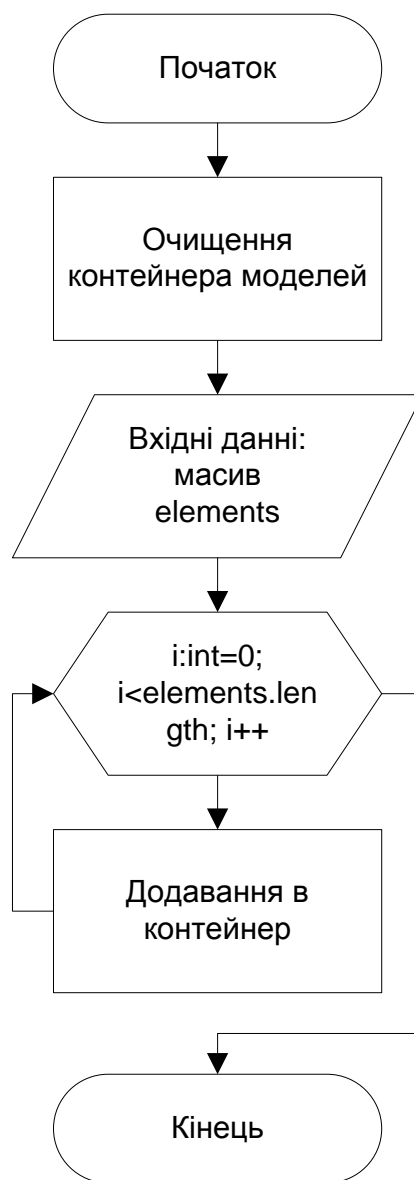


Рисунок 3.8 – Алгоритм роботи модуля зміни серій

За допомогою редактора векторної графіки було створено контейнер для зберігання прев'ю картинок кожного комплексу, з можливістю прокрутки (рис. 3.9).



Рисунок 3.9 – Панель для зберігання прев'ю картинок з можливістю прокрутки

За допомогою програмного коду, панель заповнюється раніше підготовленими картинками комплексів відповідно до файлу конфігурації. Тобто, автоматично вибираються комплекси, які відносяться до вибраної серії.

Модуль управління моделями. Алгоритм роботи модуля управління простий і логічний. Установка моделі на поле відбувається за допомогою перетягування елемента с панелі розділів.

Модуль управління моделями виконує такі функції:

- установку моделей на поле;
- повороти моделей;

- зміну дизайну моделі;
- видалення моделі;
- копіювання моделі;
- переміщення моделі.

Кожен елемент поля має можливість переміщення, це організовано за рахунок коду, який наведено нижче:

```
if(dosvilDrag == true)
{
content.startDrag(false);
drag=true;
sX = content.x;
sY = content.y;
}
```

Також кожен елемент має можливість виклику додаткового меню параметрів, за допомогою подвійного кліку мишкою, приклад виклику меню наведено на рисунку 3.10.



Рисунок 3.10 – Меню параметрів елементів поля

Кожна кнопка даного меню запрограмована на певну дію, код програмування кнопок наведено нижче. Нетиповою є кнопка зміни кольору моделі, таку модель зображено на рисунку 3.11. Вона доступна лише для тих елементів, які мають декілька дизайнерських рішень, код наведено нижче:

```
(popUp["btn_r_r"] as MovieClip).addEventListener(MouseEvent.CLICK,
onLeftRotateClick);
(popUp["btn_r_l"] as MovieClip).addEventListener(MouseEvent.CLICK,
onRightRotateClick);
(popUp["btn_close"] as MovieClip).addEventListener(MouseEvent.CLICK,
onClosePopUp);
(popUp["btn_minus"] as MovieClip).addEventListener(MouseEvent.CLICK,
onDelete);
(popUp["btn_plus"] as MovieClip).addEventListener(MouseEvent.CLICK,
onPlus);
```

```
(popUp["btn_color"] as MovieClip).addEventListener(MouseEvent.CLICK,
onColorChange);
(popUp["btn_color"] as MovieClip).visible = false;
if (numberColor >0){
(popUp["btn_color"] as MovieClip).visible = true;
}
```



Рисунок 3.11 – Модель з різним дизайном

Додаткове меню параметрів прийнято називати «Попуп» – спливаюче вікно, що відкривається на екрані комп'ютера в результаті виконання будь-якої операції. Спливаючі вікна використовуються для реалізації призначеного для користувача інтерфейсу в додатках. В даному випадку, для управління комплексами на полі.

У даному розділі було показано створення концепції дизайну, реалізовано підключення фреймворку PureMVC, реалізовано модулі завантаження ресурсів, управління моделями, зміни серій моделей і модуль головного поля. Також проведено тестування та відладку.

3.4 Висновки

В третьому розділі кваліфікаційної роботи виконано проектування та реалізацію програмно-технічних засобів 3D-конструювання дитячих майданчиків.

У даній кваліфікаційній роботі було розроблено частину від загальної структури проекту, при розробці відповідних модулів, створено інтерфейси для зв'язку з іншими модулями конструктора.

Графічний дизайн було розроблено в редакторі растрової графіки Adobe Photoshop CS6. Окремі елементи були створені засобами редактора векторної графіки Adobe Illustrator CC. Анімації та кнопки створені за допомогою Adobe Animate.

У даному розділі було показано створення концепції дизайну, реалізовано підключення фреймворку PureMVC, реалізовано модулі завантаження ресурсів, управління моделями, зміни серій моделей і модуль головного поля. Також проведено тестування та відладку.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Дизайн дитячих майданчиків повинен створити умови для прояву максимальної ініціативи дітей і їх найбільшої самостійності.

Задача побудови ескізного представлення проектів дитячих майданчиків уже давно не є забаганкою. Від естетичного вигляду, можливості оцінити розташування комплексів, залежить доля майбутнього проекту.

В даній дипломній роботі було розглянуто різноманітні технології для створення додатків. Проаналізовані найпопулярніші мови програмування та наведено їх позитивні та негативні якості. Розглянуто середовища програмування, наведені їх переваги та недоліки.

Проаналізовано редактори растрової та векторної графіки, а саме: Adobe Photoshop CS6, Adobe Illustrator CC та Adobe Animate.

Розглянуто характеристики та вимоги до конструкторів дитячих майданчиків. Досліджено які технологій є більш зручними для створення конструктора. Розроблено структуру та модулі проекту.

Розроблено основний дизайн додатку. Створено модулі головного поля, завантаження ресурсів, зміни серій моделей та управління моделями.

Конструктор дитячих майданчиків є базовою частиною проекту, передбачено підключення додаткових модулів, за рахунок використання незалежних інтерфейсів.

Метою кваліфікаційної роботи є проектування та розроблення програмно-технічних засобів 3D-конструювання дитячих майданчиків.

Об'єктом дослідження є процес 3D-конструювання дитячих майданчиків.

Предметом дослідження є програмно-технічні засоби 3D-конструювання дитячих майданчиків.

Для досягнення поставленої мети використовувались принципи системного аналізу, методи синтезу, аналізу та моделювання процесів, теоретико-множинні підходи.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підпис	Дата		

В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проведено техніко-економічне обґрунтування доцільності проектування програмно-технічних засобів, поаналізовано існуючі способи вирішення технічної проблеми, їх аналіз і недоліки, а також виконано прогнозування собівартості нової розробки) та виконано постановку задачі дослідження.

В другому розділі виконано аналіз засобів розроблення програмно-технічних засобів 3D-конструювання дитячих майданчиків. Встановлено, що продуктивність Flash Player при відтворенні анімації в кілька разів перевищує продуктивність віртуальної машини Javascript в браузерях, що підтримують попередній стандарт HTML5. Flash Player портований на мобільну платформу Android. Проведено аналіз існуючих мов програмування, оптимальним варіантом для задачі створення конструктора дитячих майданчиків є мова програмування Action Script 3. Проведено аналіз середовищ програмування, вибрано Adobe Flash Builder. Завдяки зрозумілому інтерфейсу, вбудованому редактору Adobe ActionScript і CSS, рефакторингу коду і потужних засобів для тестування, Adobe Flash Builder є оптимальним варіантом для виконання поставленої задачі.

В третьому розділі кваліфікаційної роботи виконано проектування та реалізацію програмно-технічних засобів 3D-конструювання дитячих майданчиків. У даній кваліфікаційній роботі було розроблено частину від загальної структури проекту, при розробці відповідних модулів, створено інтерфейси для зв'язку з іншими модулями конструктора. Графічний дизайн було розроблено в редакторі растрової графіки Adobe Photoshop CS6. Окремі елементи були створені засобами редактора векторної графіки Adobe Illustrator CC. Анімації та кнопки створені за допомогою Adobe Animate. У третьому розділі було показано створення концепції дизайну, реалізовано підключення фреймворку PureMVC, реалізовано модулі завантаження ресурсів, управління моделями, зміни серій моделей і модуль головного поля. Також проведено тестування та відладку.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Chang W. Building a Five Sense Experience into Children's Playgrounds Using Renewable Energy. *Proceedings of International Conference on Energy, Environment and Materials Engineering (EEME 2014)*. 2014. Pp. 36-40.
2. Menconi M., Grohmann D. Participatory retrofitting of school playgrounds: Collaboration between children and university students to develop a vision. *Thinking Skills and Creativity*. 2018. Vol. 29. Pp. 71-86.
3. Beristain J., Arenaza X., Prieto I. X3D Based Playground Configurator. *Proceedings of WEB3D 2013: 18TH International Conference on 3D WEB Technology*. 2013. Pp. 87-90.
4. Acher M., Baudry B., Barais O., Jezequel J. Customization and 3D Printing: A Challenging Playground for Software Product Lines. *Proceedings of 18th International Software Product Line Conference*. 2014. Vol. 1. Pp. 142-146.
5. Menconi M., Grohmann D. Participatory retrofitting of school playgrounds: Collaboration between children and university students to develop a vision. *Thinking Skills and Creativity*. 2018. Vol. 29. Pp. 71-86.
6. Jerald J., Mlyniec P., Yoganandan A., Rubin A., Paullus D., Solotko S. MakeVR: A 3D World-Building Interface. *Proceedings of 2013 IEEE Symposium on 3D User Interfaces*. 2013. Pp. 197-198.
7. Blanchard M., Reisch M., Yepes V. Swing: 2D and 3D Animation in Virtual Reality. *Proceedings of 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops*. 2021. Pp. 774.
8. Ali A., Puthusserypady S. A 3D Learning Playground for Potential Attention Training in ADHD: A Brain Computer Interface Approach. *Proceedings of 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2015. Pp. 67-70.
9. Kalinke L., Cangussu D., Mon M., Bruno R., Tiburcio E., Lloret F., Armentano D., Pardo E., Ferrando-Soria J. Metal-Organic Frameworks as Playgrounds

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

for Reticulate Single-Molecule Magnets. *Inorganic Chemistry*. 2019. Vol. 58. Issue 21. Pp. 14498-14506.

10. Thomas M., Benin S., Schneider C., Rainbow C., Can T., Simsek I., Kose Biber S., Philp H., Zwart N., Turchetta B., Benedetti F., Garista P., Cinganotto L. Digital Game-Based Language Learning in 3D Immersive Environments: The GUINEVERE Project. *Proceedings of The 11th Innovation in Language Learning International Conference*. 2018. Pp. 103-106.

11. Allen E., Hill A., Tranter E., Sheehan K. Playground Safety and Quality in Chicago. *Pediatrics*. 2013. Vol. 131. Issue 2. Pp. 233-241.

12. Flaes S., Chinapaw M., Koolhaas C., Van Mechelen W., Verhagen E. More children more active: Tailored playgrounds positively affect physical activity levels amongst youth. *Journal of Science and Medicine in Sport*. 2016. Vol. 19. Issue 3. Pp. 250-254.

13. Butcher J. Socialization of Childrens Playground Skill. *Perceptual and Motor Skills*. 2013. Vol. 77. Issue 3. Pp. 731-738.

14. Stanton-Chapman T., Schmidt E. Building Playgrounds for Children of All Abilities: Legal Requirements and Professional Recommendations. *Early Childhood Education Journal*. 2009. Vol. 47. Issue 5. Pp. 509-517.

15. Hug D., Kemper M., Panitz K., Franinovic K. Sonic Playgrounds - Exploring Principles and Tools for Outdoor Sonic Interaction. *Proceedings of Audio Mostly 2016 - A Conference On Interaction with Sound in Cooperation with ACM*. 2016. Pp. 139-146.

16. Czalczyńska-Podolska M. The impact of playground spatial features on children's play and activity forms: An evaluation of contemporary playgrounds' play and social value. *Journal of Environmental Psychology*. 2014. Vol. 38. Pp. 132-142.

17. Jansson M. Attractive Playgrounds: Some Factors Affecting User Interest and Visiting Patterns. *Landscape Research*. 2010. Vol. 35. Issue 1. Pp. 63-81.

18. Suminski R., Presley T., Wasserman J., Mayfield C., McClain E., Johnson M. Playground Safety is Associated with Playground, Park, and Neighborhood Characteristics. *Journal of Physical Activity & Health*. 2015. Vol. 12. Issue 3. Pp. 402-408.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

19. Reimers A., Knapp G. Playground usage and physical activity levels of children based on playground spatial features. *Journal of Public Health-Heidelberg*. 2017. Vol. 25. Issue 6. Pp. 661-669.

20. Olsen H., Kennedy E. Safety of School Playgrounds: Field Analysis from a Randomized Sample. *Journal of School Nursing*. 2020. Vol. 36. Issue 5. Pp. 369-375.

21. Talarowski M., Cohen D., Williamson S., Han B. Innovative playgrounds: use, physical activity, and implications for health. *Public Health*. 2019. Vol. 174. Pp. 102-109.

22. Olsen H., Hudson S., Thompson D. Developing a Playground Injury Prevention Plan. *Journal of School Nursing*. 2008. Vol. 24. Issue 3. Pp. 131-137.

23. Wenger I., Schulze C., Lundstrom U., Prellwitz M. Children's perceptions of playing on inclusive playgrounds: A qualitative study. *Scandinavian Journal of Occupational Therapy*. 2020. Vol. 28. Issue 2. Pp. 136-146.

24. McCarthy S., Hughey S., Kaczynski A. Examining Sociodemographic Differences in Playground Availability and Quality and Associations with Childhood Obesity. *Childhood Obesity*. 2017. Vol. 13. Issue 4. Pp. 324-331.

25. Arroyo-Johnson C., Woodward K., Milam L., Ackermann N., Komaie G., Goodman M., Hipp J. Still Separate, Still Unequal: Social Determinants of Playground Safety and Proximity Disparities in St. Louis. *Journal of Urban Health-Bulletin of The New York Academy of Medicine*. 2016. Vol. 93. Issue 4. Pp. 627-638.

26. Fuselli P., Yanchar N. Preventing playground injuries. *Paediatrics & Child Health*. 2012. Vol. 17. Issue 6. Pp. 328.

27. Stanton-Chapman T., Schmidt E. How Do the Children Play? The Influence of Playground Type on Children's Play Styles. *Frontiers in Psychology*. 2021. Vol. 12. Article Number 703940.

28. Nasar J., Holloman C. Playground Characteristics to Encourage Children to Visit and Play. *Journal of Physical Activity & Health*. 2014. Vol. 10. Issue 8. Pp. 1201-1208.

29. Hirose M., Taniguchi M., Nakagaki Y., Nihei K. Virtual Playground and Communication Environments for Children. *IEICE Transactions On Information and Systems*. 2014. Vol. E77D. Issue 12. Pp. 1330-1334.

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

30. Markovic K., Milcic L., Lanc D. Analysis of The Quality Children's Playgrounds to Be Used for Learning Basic Gymnastic Content. *Proceedings of 7th International Scientific Conference On Kinesiology: Fundamental and Applied Kinesiology - Steps Forward*. 2014. Pp. 322-326.

31. Гриньов А. В. Організація та управління на підприємстві. Харків: Вид. дім "ИНЖЕК", 2004. 329 с.

32. Ворона Т. Стартап на мільйон. Як українці заробляють статки на технологіях. К.: Віват, 2017. 224 с.

33. Бачинський Т. В., Радейко Р. І., Харитоновна О. І. Основи ІТ-права: навчальний посібник. Київ: Юрінком Інтер, 2017. 208 с.

34. T. DeMarco. *Peopleware: Productive Projects and Teams*. Addison-Wesley Professional, 2013. 272 p.

35. Elon Musk: Tesla, SpaceX, and the Quest for a Fantastic Future. 2017. 416 p.

36. Москаленко В. В, Шматко О. В. Методичні вказівки до виконання економічного обґрунтування проекту щодо розробки програмного забезпечення дипломної роботи освітньо-кваліфікаційного рівня «бакалавр» для студентів спеціальності 122 – Комп’ютерні науки галузі знань 12 – Інформаційні технології. Харків: НТУ «ХП». 2020. 34 с.

37. Коваленко О. С., Добровська Л. М. Проектування інформаційних систем: Загальні питання теорії проектування ІС: навч. посіб. для студ. спеціальності 122 «Комп’ютерні науки». Київ : КПІ ім. Ігоря Сікорського, 2020. 192 с.

38. Кудлаєва Н.В., Круць С.В. Особливості калькулювання собівартості програмного продукту. *Причорноморські економічні студії*. 2019. В. 47-2. С. 138-144.

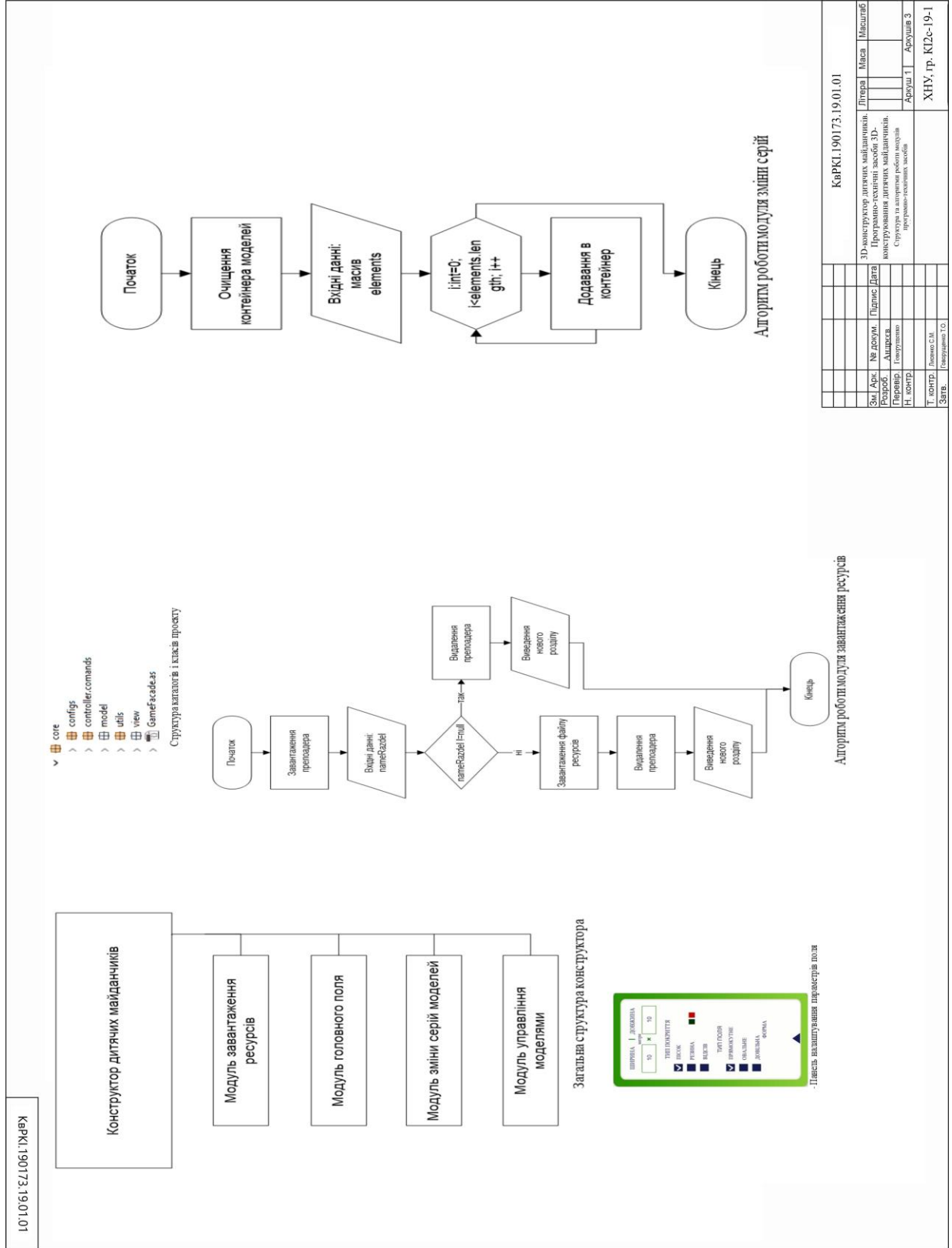
39. Бузак Н.І. Інформаційні технології як об’єкт обліку та калькулювання собівартості їх упровадження. *Вісник Національного університету «Львівська політехніка»*. 2017. Вип. 576. С. 378–384.

40. Гороховатський В. Методика визначення собівартості програмного забезпечення. URL: http://www.hups.mil.gov.ua/periodic-app/article/11566/soi_2014_4_21.pdf (дата звернення: 01.03.2022).

					КВРКІ 190173.19.01.01 ПЗ	Арк.
						66
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Б (обов'язковий)

Копія креслення «Структура та алгоритми роботи модулів програмно-технічних засобів»



КерКІ.190173.19.01.01		Літера	Місяць
См. Арх.	№ докум.	Підпис	Дата
3D-конструктор дитячих майданчиків. Програмно-технічні засоби 3D-модулювання дитячих майданчиків. Структура програмно-технічних засобів.			
Розроб.	Людськ.	Господарство	Архив 1
Н. контр.			Архив 3
Т. контр.	Ковале С.М.		ХНУ, гр. КІ2С-19-1
Затв.	Гончаренко І.О.		

Ім'я користувача:
Кафедра КІ

Дата перевірки:
25.05.2022 16:54:41 EEST

Дата звіту:
25.05.2022 16:58:04 EEST

ID перевірки:
1011337402

Тип перевірки:
Doc vs Internet + Library

ID користувача:
10005591

Назва документа: Андреев_2_3D-конструктор дитячих найданчиків. Програмно-технічні засоби 3D-конструюв...

Кількість сторінок: 63 Кількість слів: 9410 Кількість символів: 74176 Розмір файлу: 2.26 MB ID файлу: 1011223083

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

14.8% Схожість

Найбільша схожість: 8.59% з джерелом з Бібліотеки (ID файлу: 1011215018)

6.32% Джерела з Інтернету 80 Сторінка 65

9.05% Джерела з Бібліотеки 67 Сторінка 66

2.09% Цитат

Цитати 5 Сторінка 67

Не знайдено жодних посилань

66.6% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

Немає вилучених Інтернет-джерел

66.6% Вилученого тексту з Бібліотеки 1 Сторінка 67

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 37

Підозріле форматування 10 сторінок

Ім'я користувача:
Кафедра КІ

Дата перевірки:
25.05.2022 08:04:51 EEST

Дата звіту:
25.05.2022 08:09:39 EEST

ID перевірки:
1011328879

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Андреев 3D-конструктор дитячих майданчиків. Програмно-технічні засоби 3D-конструюван...

Кількість сторінок: 66 Кількість слів: 10114 Кількість символів: 79015 Розмір файлу: 2.26 MB ID файлу: 1011215017

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

27.6% Схожість

Найбільша схожість: 3.76% з Інтернет-джерелом (<https://lektsii.org/3-116021.html>)

27.6% Джерела з Інтернету 104

Сторінка 68

1.51% Джерела з Бібліотеки 7

Сторінка 70

2.14% Цитат

Цитати 6

Сторінка 71

Не знайдено жодних посилань

8.22% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

0.21% Вилучення з Інтернету 2

Сторінка 72

8.22% Вилученого тексту з Бібліотеки 69

Сторінка 72

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 37

Підозріле форматування 11 сторінок

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 3.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 9%**

ID: 103920 Название: 3D-конструктор дитячих майданчиків. Програмно-технічні засоби 3D-конструювання дитячих майданчиків Добавлено в БД: 2022-05-25 Авторы: О. А. Андреев Руководители: Т. О. Говорущенко Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	65366	596	10225 (16%)	98 (16%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Андреев Олексій Андрійович

Тема: 3D-конструктор дитячих майданчиків. Програмно-технічні засоби 3D-конструювання дитячих майданчиків

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 68

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є програмно-технічних засобів 3D-конструювання дитячих майданчиків.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (виконано техніко-економічне обґрунтування доцільності проектування, проаналізовано існуючі способи вирішення технічної проблеми, виконано прогнозування собівартості нової розробки) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено аналіз засобів розроблення програмно-технічних засобів 3D-конструювання дитячих майданчиків, а саме: описано характеристики та вимоги до конструкторів дитячих майданчиків, досліджено редактори растрової та векторної графіки, проаналізовано існуючі мови та середовища розроблення. В третьому розділі спроектовано та реалізовано програмно-технічні засоби 3D-конструювання дитячих майданчиків, а саме: описано етапи розроблення програмно-технічних засобів, дизайн 3D-конструктора дитячих майданчиків, основні модулі 3D- конструктора дитячих майданчиків.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи:

6. Оцінка графічного оформлення та пояснювальної записки роботи:
Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному інженерно-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре (4.25/В)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Мартинюк Валерія Володимирівна,
зав. кат. АКІТ, д.т.н., проф.

"26" 05 2022 р.

 (підпис)

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорущенко Т. О.

Андрєєва Олексія Андрійовича

ІІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

24 травня 2022 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: 3D-конструктор дитячих майданчиків. Програмно-технічні засоби 3D-конструювання дитячих майданчиків

Автор: Андреев Олексій Андрійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Говорущенко Тетяна Олександрівна, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріплення запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) найбільшу схожість встановлено з одним документом і становить вона 8.59% в частині загальноприйнятої термінології та огляду відомих засобів розробки, а також автоматично генерованого програмного коду;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів з україномовними скороченнями індексів в формулах, що не є модифікацією тексту;
- 5) вилучення відбулись внаслідок того, що це комплексна кваліфікаційна робота, і були збіги в двох студентах саме за термінологією, а також внаслідок того, що знадобилось доопрацювання і повторна перевірка роботи.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 14.8% і адресується до 37 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОПП

Завідувач кафедри КІС

 Т. О. Говорущенко

 С. М. Лисенко

 Т. О. Говорущенко