

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Розробка ігрового додатку «ToMZ» на базі LibGDX

Виконав студент 4 курсу групи К-41
Напряму 6.050101 2 комп'ютерні науки,
Козаченко Георгій Юрійович

Керівник асистент
Штефан Наталія Зінов'ївна

Консультант: к.т.н., доцент
Великодний Станіслав Сергійович

Рецензент: к.георг.н., доцент
Бояринцев Євген Львович

ЗМІСТ

Скорочення та умовні позначки	7
Вступ	9
1 Аналіз предметної області	11
1.1 Характеристика об'єкта розробки	11
1.2 Опис предметної області.....	11
1.3 Аналіз існуючих аналогів	12
1.4 Аналіз засобів розробки.....	19
1.4.1 Середовище розробки AndroidStudio від JetBrains	19
1.4.2 Мова програмування Java	20
1.4.3 XML для розмітки GUI компонентів.....	22
1.4.4 Додаткові інструментальні засоби.....	23
2 Проектування системи	25
2.1 Діаграми діяльності.....	25
2.2 Діаграми класів.....	29
3 Розробка ігрового додатку	33
3.1 Основні компоненти ігрової моделі	33
3.1.1 Меню	33
3.1.2 Клас «Button»	33
3.1.3 Ігровий процес	35
3.1.4 Ячейка ігрового поля.....	36
3.1.5 Елемент поля.....	37
3.1.7 Progress bar	40
3.1.8 Партікли	41
3.2 Алгоритм MATCH 3.....	42
3.2.1 Бонус	43
3.2.2 Генератор рівня.....	47
Висновки.....	50
Перелік джерел посилання	52

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЕП – елемент поля гри

ОС – Операційна система

GUI – Graphical User Interface

HTML – HyperText Markup Language

JVM – Java Virtual Machine

ToMZ – Type of Montezuma

UI – User Interface

UML – Unified Modeling Language

XML – Extensible Markup Language

Терміни

Геймплей – ігровий процес

Квести – (англ. Quest), або пригодницька гра

Токен – абстрактна точка на діаграмі використання

Adobe Photoshop – багатофункціональний редактор для роботи з графікою

Action – дія на діаграмі UML

Android Studio – Середовище розробки під Андроїд

Applets – аплети

Control flow token – маркер управління на діаграмі

C++ – компільована, статично типізована мова програмування загального призначення

Data flow token – маркером даних

IntelliJ IDEA – інтегроване середовище розробки програмного забезпечення для багатьох мов програмування, розроблена компанією JetBrains

Java – Мова програмування

JetBrains – міжнародна компанія, яка створює засоби для розробки програмних продуктів різними мовами програмування

Match three – категорія гри-головоломки

Private – закритий атрибут класу

Protected – захищений атрибут

Public – відкритий атрибут класу

Spring – бібліотека IntelliJ IDEA

Sun Microsystems – Американська компанія, виробник програмного і апаратного забезпечення

Visibility – видимість атрибутів класу

Use Case – варіанти використання в UML

ВСТУП

Сьогодні серед людей самого різного віку все більш популярними стають гри, в які можливо грати прямо на сайті, без завантаження на свій персональний комп'ютер. Існує маса різновидів таких ігор для дорослих і дітей, всі вони мають досить простий геймплей і яскраву привабливу графіку.

Якщо зазирнути в кожен смартфон і планшет на земній кулі, то обов'язково можна знайти в більшості з них хоча б одну встановлену гру «Match three». Можливо «Bejeweled», «Zuma» або вічний хіт «Candy Crush Saga» [1]¹⁾.

Особливість такого типу ігор полягає в тому, що всі представники жанру використовують практично однакову ігрову механіку. Основне завдання гравця – скласти комбінації з 3-х однакових об'єктів (цукерок, дорогоцінних каменів, фруктів і т.д.), хаотично розташованих на ігровому полі, і утворювати різні комбінації для того, щоб заробити більше очок.

Незважаючи на те, що світові хіти казуальне геймінга не володіють жодним з властивостей «AAA-ігор» (закручений сюжет, кастомізація персонажів, що відкривається контент і т.д.), мільйони людей у всьому світі щодня годинами грають в ці ігри. Секрет їхнього успіху в інтуїтивної, нехитрій механіці [2]²⁾.

Різноманітність онлайн ігор дає можливість будь-якому вибрати те, що йому буде до смаку, і одним з найпоширеніших жанрів таких ігор можна вважати «три в ряд» або «Match three». Сама назва вже пояснює основний принцип механізму гри. Від гравця потрібно збирати в ряд ланцюжка

¹⁾ [1] Игры Match 3: в чем секрет успеха и привлекательности?. URL: <http://renatus.com/ru/match-3-games-what-makes-fun-and-addictive>. (дата звернення 03.02.2019).

²⁾ [2] Что собой представляют игры жанра три-в-ряд или матч-3. URL: <http://smages.com/stati/chto-soboj-predstavlyayut-igry-zhanra-tri-v-ryad-ili-match-3/>. (дата звернення 03.02.2019).

однакових елементів – по три і більше, після чого вони зникають, і їх тут же змінюють нові.

Цей жанр – один з найпростіших, він прекрасно підходить тим, кому потрібно просто розслабитися на деякий час, не дуже замислюючись. Однак певні гри «три в ряд» не надто просто проходити на завершальних рівнях. Як правило, в подібних іграх присутній тимчасове обмеження, але нерідко передбачений режим «без часу» – для початківців гравців або любителів спокійної гри [3]¹⁾.

Метою роботи є створення розважальної гри стилю «Match three» для мобільних додатків.

Загальні характеристики кваліфікаційної роботи:

- повний обсяг сторінок пояснювальної записки – 51;
- кількість рисунків – 20;
- кількість таблиць – 0;
- кількість посилань – 18.

¹⁾ [3] Топ-10 лучших игр на андроид 2017-2018 года для вашего смартфона. URL: <https://www.fly-phone.ru/notes/android/luchshie-igry-dlya-android/>. (дата звернення 03.02.2019).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Характеристика об'єкта розробки

Найменування програмного продукту – комп'ютерна гра «ТоМЗ». Жанр гри «Match three». Сама гра представлена у вигляді головоломки, де треба поєднувати три або більше елементів поля однаково типу в ряд. Головоломка не доладна зовсім, вона призначена не для глибокого мислення, а більш для відпочинку. Гра підійде як для малят, так й для дорослих.

Жанр цей став популярний завдяки бентежній простоті, але до таких ігор швидко звикаєш, вони дивним чином затягують, незважаючи на примітивізм. Різноманітні головоломки популярні у широкої аудиторії. Ці головоломки не вимагають вникання в суть: все лежить на поверхні. Можна відкрити на десять хвилин, повернутися через тиждень або місяць – і процес буде таким же захоплюючим [2]¹⁾.

Суть всього жанру «Match three» – зібрати в один ряд три предмети одного виду. Однакові фігурки, колір, форма – ознака може бути будь-яким. Та й об'єктів не обов'язково три, просто це найбільш популярне значення.

1.2 Опис предметної області

Жанр «Match three» дуже динамічний і постійно розвивається. Топові розробники вдосконалюють свої ігри, додають нові фічі і івенти. Якісь зміни з'являються у всіх успішних проектах і стають стандартом жанру.

На початку гри всі повинно бути просто, без складнощів, тому що потрібно навчати гравця. Розробник повинен його навчити, не треба відразу на нього вивалювати складний рівень, але і довго в тримати в стороні від

¹⁾ [2] Что собой представляют игры жанра три-в-ряд или матч-3. URL: <http://smages.com/stati/chto-soboj-predstavlyayut-igry-zhanra-tri-v-ryad-ili-match-3/>. (дата звернення 03.02.2019).

складнощів – не варто. Рекомендується на першу сесію гри викладати 5-8 рівнів складності, щоб у гравця не склалося думка, що гра дуже проста.

Якщо спочатку зробити всі рівні простими, в результаті вийде, що гравець наче «пролітає» їх дуже швидко, вони не викликають у нього ніяких емоцій, ніяких відчуттів. Тому потрібно намагатися прості рівні робити приблизно в діапазоні 50-60% складності. Це не так складно для гравця, це кілька спроб, але він відчуває, що переміг не просто так, хоч і не застряг надовго на цих рівнях [3]¹⁾.

Видів ігрового механізму також передбачено кілька. Найбільш поширений такий варіант, як зміна місцями сусідніх фішок, або «обмін». При цьому створюються вертикальні, горизонтальні або навіть діагональні ряди. Інший спосіб – протягування ланцюжків, що складаються з однакових елементів, в різних напрямках. Третій метод – виділення груп фішок і видалення їх з поля гри. Четвертий варіант найбільш рідкісний, він полягає в перетягуванні стовпців елементів, що мають однаковий колір, які тут же зникають [4]²⁾.

1.3 Аналіз існуючих аналогів

На початку проектування необхідно оглянути аналоги, порівняти їх, розібратися, як це зроблено у конкурентів і як це буде реалізовано у себе. Гравці, для яких буде реалізована ідея об'єкту розробки, пізніше будуть самі порівнювати гру з конкурентами.

¹⁾ [3] Топ-10 лучших игр на андроид 2017-2018 года для вашего смартфона. URL: <https://www.fly-phone.ru/notes/android/luchshie-igr-dlya-android/>. (дата звернення 03.02.2019).

²⁾ [4] Жанровый анализ рынка мобильных игр. URL: <https://dtf.ru/gameindustry/23993-zhanrovyy-analiz-rynka-mobilnyh-igr>. (дата звернення 03.02.2019).

Розглянемо декілька популярних ігор даного жанру для подальшої чіткої постановки завдань і вимог до об'єкту розробки. Першим аналогом виступає гра під назвою «Jewel Quest» (рис. 1):

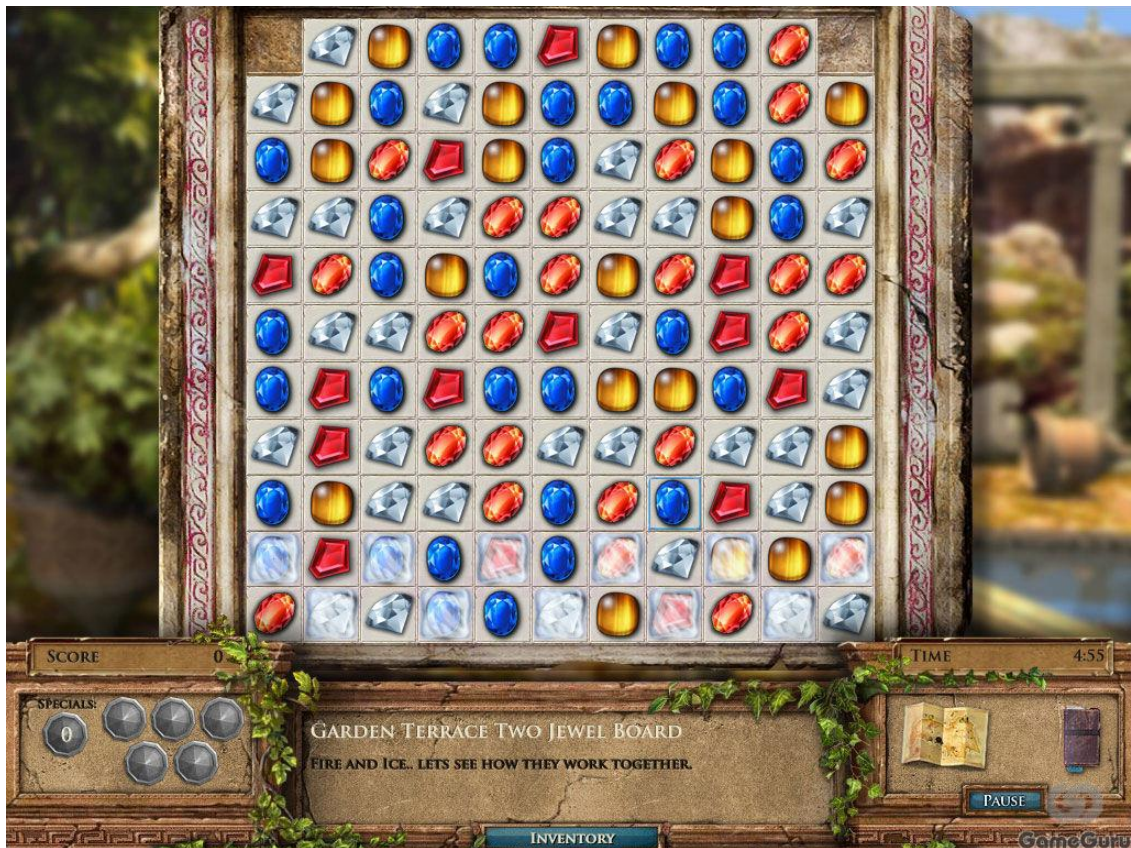


Рисунок 1 – Скріншот гри «Jewel Quest»

Безкоштовна онлайн гра відрізняється гарною промальовуванням фігурок і навіть непоганим фоновим малюнком. Бекграунд відсилає до стародавніх скарбів і далеких подорожей: шматки паперу, щоденник, класичні годинник.

Основна тематика – цивілізація майя. У грі 180 рівнів. Сюжет розкривається через замітки головного героя. У наступних частинах шукач скарбів обзавівся ім'ям (Руперт). За черговими коштовностями він відправився в спекотну Африку [5]¹⁾.

¹⁾ [5] Топ игр Match3. URL: https://playliner.com/ru/report_match3_market_review_2.php (дата звернення 05.02.2019).

Другим аналогом виступає гра під назвою «Treasures of Montezuma» (рис. 2), яка є головним натхненням створення цього дипломного проекту (назва навіть базується :ToMZ – Type of Montezuma).

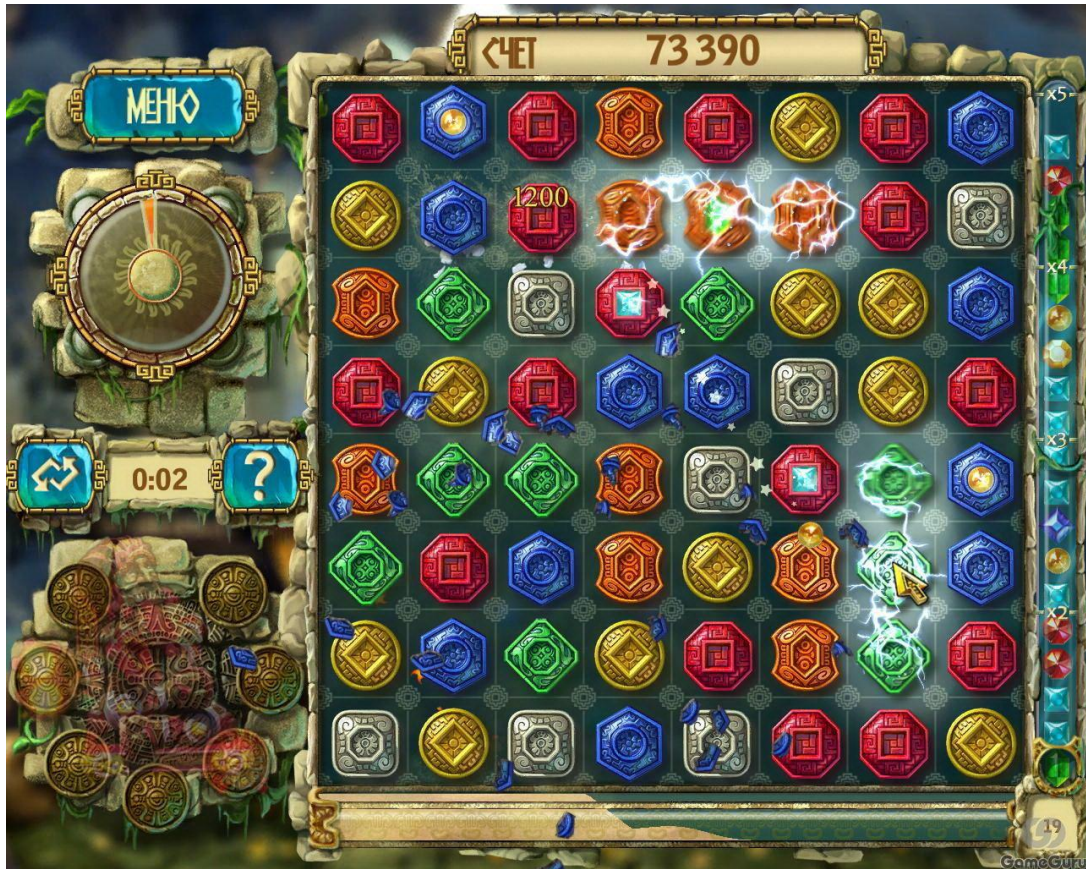


Рисунок 2 – Скріншот гри «Treasures of Montezuma»

Ця гра, як і «Jewel Quest», зав'язана на пошуку скарбів. Тепер в центрі історії, що датується одна тисяча дев'яцот тридцять п'ять роком, виявляється студентка Емілі Джонс. Вона проходить практику в іспанському музеї і натикається на оригінальну кам'яну експозицію.

У знайдених стародавніх рукописах повідомляється, що це карта, яка веде до долини, яка дає багатство і незвичайну здоров'я. Власне, заради таких

речей будь-який поважаючи себе шукач скарбів навіть вухом не поведе. Але Емілі вирішує знайти реліквію [6]¹⁾.

Звичайно, для цього доведеться довго складати фігурки і вибивати окуляри.

Особливе значення надається бонусів – предмети випадають таким чином, щоб гравець міг заробити побільше балів і красиво пройти рівень, а не нудно вибивати по три об'єкти.

Переваги:

- гарне графічне оформлення і спецефекти, тобто динамічність гри – наскільки барвисто і яскраво змінюється картинка;
- наявність різних бонусів;
- підказки.

До недоліків слід віднести наступне:

- кінцівка рівнів і статичність поля;
- рівні з тимчасовими обмеженнями, що відводить гру від релаксатної;
- музика атмосферна, немов перед тобою говорять кристали, але ось набір її обмежений і швидко приїдається.

Наступний аналог це гра «Puzzle Quest» (рис. 3):

¹⁾ [6] Обзор The Treasures of Montezuma 4. Складываем кристаллы. URL: <https://megaobzor.com/obzor-The-Treasures-of-Montezuma-4-skladyvaem-kristally.html> (дата звернення 12.02.2019).



Рисунок 3 – Інтерфейс гри «Puzzle Quest»

За великим рахунком, «Puzzle Quest» – рольова гра, в якій бої побудовані на механіці «Match 3». Тут ідеально поєднуються рольові елементи і складова головоломка, є стандартні класи героя, сюжетні і додаткові квести, печери з монстрами і величезні боси.

Потрібно враховувати рівень противника і об'єктивно оцінювати свої сили. Незважаючи на гадану простоту, «Puzzle Quest» часом буває дуже складна.

Замість того щоб в черговий раз намагатися вбити дрібного боса, краще піти в іншу сторону і набити рівні на рибці подрібніше – це самий розумний варіант.

У цій серії вийшло навіть відгалуження про космос, «Puzzle Quest: Galactrix». Гра рекомендується всім, благо доступна тепер на будь-яких платформах, включаючи мобільні [6]¹⁾.

¹⁾ [6] Обзор The Treasures of Montezuma 4. Складываем кристаллы. URL: <https://megaobzor.com/obzor-The-Treasures-of-Montezuma-4-skladyvaem-kristally.html> (дата звернення 12.02.2019).

Четвертим аналогом розглянемо не менш популярну гру по типу «Match three», яка носить назву «Jewels Worlds» (рис. 4):

Необхідно підключити три або більше дорогоцінних каменів в рядку, щоб заробити більше очок. Ця програма є в основному подібна гра в «Jewel» матч 3 гри. Якщо ви любите коштовності відповідності гри, це відмінний приклад, один на Android.

До переваг слід віднести:

- наявність бонусів. У порівнянні з попереднім аналогом, тут бонуси сконцентровані на поле, а не окремо від нього;
- велике поле.

Недоліки:

- відсутність підказок;
- статичні рівні;
- стара графічна стилізація, підбрані кольори, які тиснуть на очі.



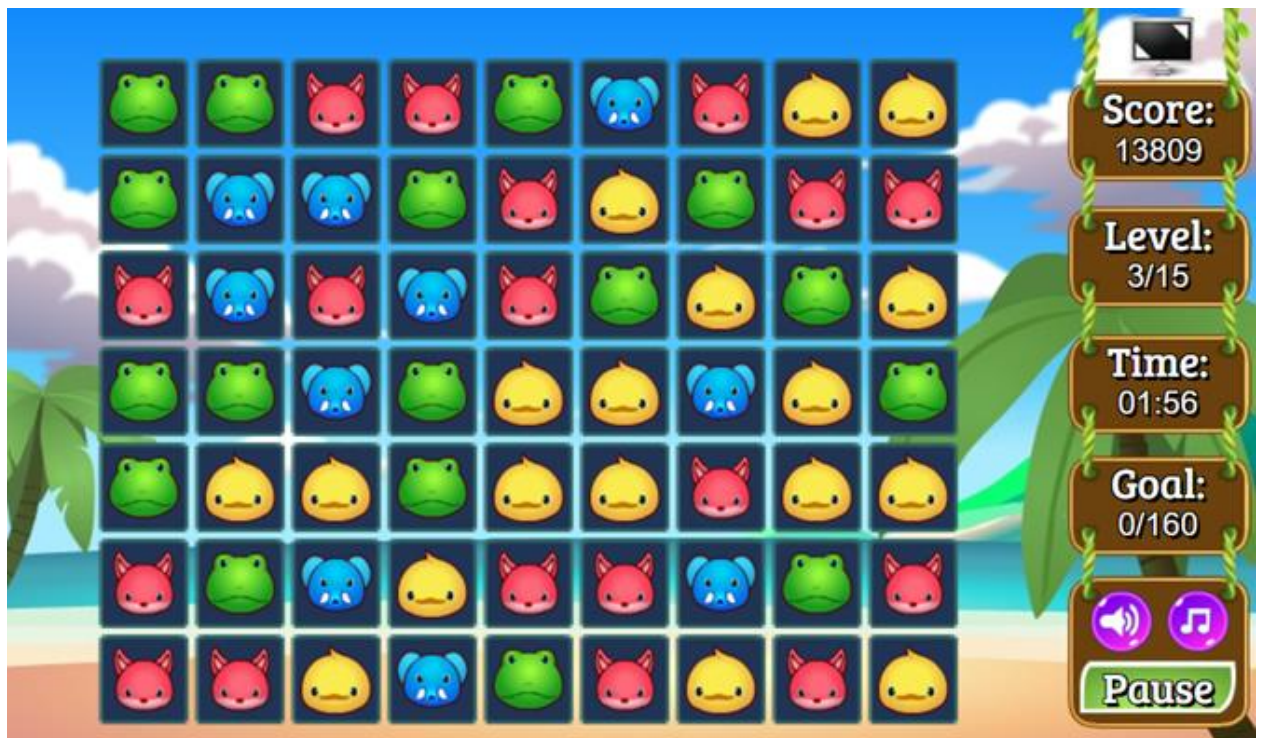
Рисунок 4 – Рівень гри «Jewels Worlds»

Останнім аналогом буде розглянуто гру «Zoobies match» (рис. 5). Забавні різнокольорові тварини запрошують вас в гру «Zoobies Match». Це гарна гра для проведення часу, вдала захоплююча головоломка. Тут на гравця чекає п'ятнадцять цікавих поступово ускладнених рівнів.

Щоб пройти кожен, необхідно набрати певну кількість очок. Необхідно міняти місцями звіряток, виставляючи їх по три і більше однакових в ряди або стовпці [5]¹⁾.

При вдалих комбінаціях з великим числом елементів, утворюються бонусні об'єкти, вони відрізняються за розмірами або мерехтіння.

Якщо їх вбудувати в лінію, від вибуху знищуються цілі горизонтальні ряди, що дозволяє швидше завершити рівень. Час обмежений.



¹⁾ [5] [Топ игр Match3. URL: https://playliner.com/ru/report_match3_market_review_2.php.](https://playliner.com/ru/report_match3_market_review_2.php) (дата звернення 05.02.2019).

Рисунок 5 – Інтерфейс гри «Zoobies match»

За результатами аналізу топових аналогів гри для дипломного проектування основна ідея створити проект, який буде максимально мати все вище перераховані переваги, а також по максимуму позбутися від недоліків. Тому гра буде в більшій мірі копіювати другий аналог, так як він найбільш перспективний з усіх, але і особливості інших так само заслужать увагу [4]¹⁾.

По-цілому, це стиль графіки, спецефекти, динамічність гри, розмірність поля, нескінченність рівнів, підтримка ОС Android, наявність бонусів і різноманітність музики.

1.4 Аналіз засобів розробки

1.4.1 Середовище розробки AndroidStudio від JetBrains

Платформа IntelliJ, розроблена JetBrains, стала основою для нового продукту Google – Android Studio, середовища розробки під Андроїд.

Це вже другий реліз з того моменту, як рік тому компанія-розробник прийняла рішення відкрити вихідний код проекту. IntelliJ IDEA 10 позиціонується як сама інтелектуальне середовище розробки для Java.

У новій версії додані поліпшення користувальницького інтерфейсу, досягнуто дворазове прискорення індексування коду при відкритті проекту, перероблена система автодоповнення коду.

Також в числі найбільш суттєвих доповнень – включення в IntelliJ IDEA засобів розробки мобільних додатків під ОС Android, які раніше були доступні тільки в платній версії. Покращена підтримка технологій програмування і бібліотек.

¹⁾ [4] Жанровый анализ рынка мобильных игр. URL: <https://dtf.ru/gameindustry/23993-zhanrovyy-analiz-rynka-mobilnyh-igr>. (дата звернення 03.02.2019).

В цілому, JetBrains, середа IntelliJ IDEA краще розуміє контекст і дозволяє розробляти додатки швидше і простіше, ніж в інших середовищах. IntelliJ IDEA (рис. 6) індексує весь проект, аналізує все, що він містить, і будує синтаксичне дерево.

Завдяки цьому в будь-який час, куди б розробник ні помістив курсор, середа знає, де він знаходиться, і що можна в цьому місці зробити з кодом [7]¹⁾.

Наприклад, IntelliJ IDEA вміє автоматично завершувати вираження, які розробник починає або продовжує набирати з середини. Середовище включає засоби для тестування сумісності додатків з різними версіями платформи і редактор призначеного для користувача графічного інтерфейсу.

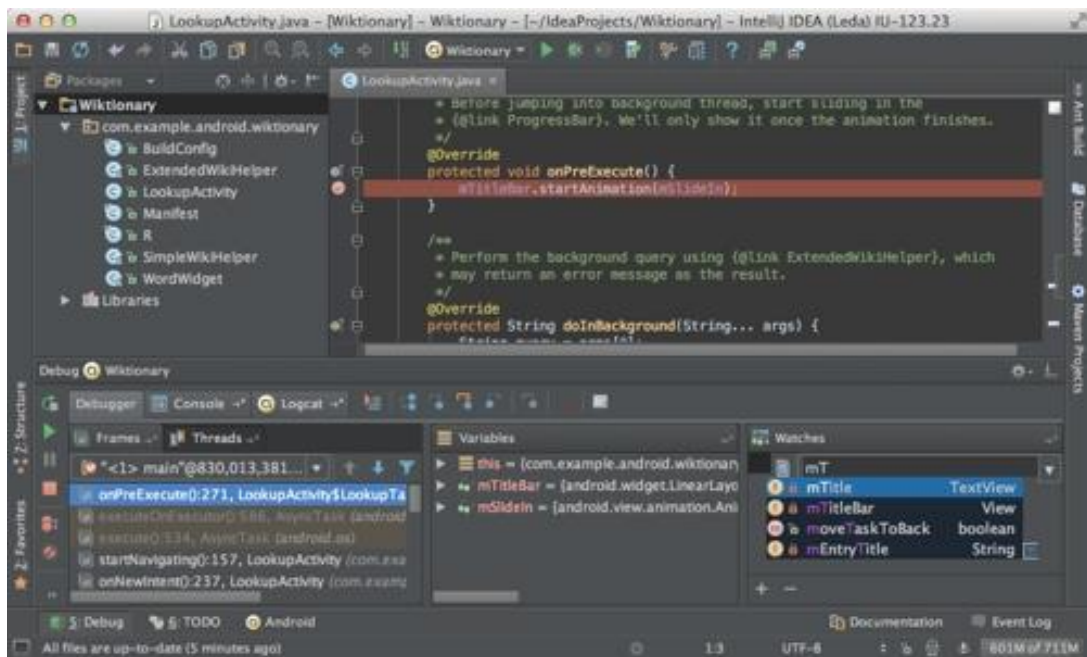


Рисунок 6 – Середа розробки IntelliJ IDEA

1.4.2 Мова програмування Java

¹⁾ [7] Навчальні матеріали з інформатики. Програмування мовою Java. URL: <https://www.ua5.org/java/>. (дата звернення 20.03.2019).

Java – об'єктно-орієнтована мова програмування, що розробляється компанією Sun Microsystems з 1991 року і офіційно випущений 23 травня 1995 року. Спочатку нову мову програмування називався Oak (James Gosling) і розроблявся для побутової електроніки, але згодом був перейменований в Java і став використовуватися для написання аплетів, додатків і серверного програмного забезпечення [7]¹⁾.

Програми на Java можуть бути трансльовані в байт-код, що виконується на віртуальній java-машині (JVM) – програмою, обробній байт-код і передавальній інструкції обладнанню, як інтерпретатор, але з тією відмінністю, що байт-код, на відміну від тексту, обробляється значно швидше.

Мова Java зародився як частина проекту створення передового програмного забезпечення для різних побутових приладів.

Реалізація проекту була почата на мові C ++, але незабаром виник ряд проблем, найкращим засобом боротьби з якими була зміна самого інструмента – мови програмування. Стало очевидним, що необхідний платформи-незалежний мова програмування, що дозволяє створювати програми, які не доводилося б компілювати окремо для кожної архітектури і можна було б використовувати на різних процесорах під різними операційними системами.

Мова Java потрібна для створення інтерактивних продуктів для мережі Internet. Фактично, більшість архітектурних рішень, прийнятих при створенні Java, було продиктовано бажанням надати синтаксис, схожий з C і C ++. В Java використовуються практично ідентичні угоди для оголошення змінних, передачі параметрів, операторів і для управління потоком виконання коду. В Java додані всі хороші риси C ++ [8]¹⁾.

¹⁾ [7] Навчальні матеріали з інформатики. Програмування мовою Java. URL: <https://www.ua5.org/java/>. (дата звернення 20.03.2019)..

¹⁾ [8] Плюсы и минусы программирования на Java. URL: <https://medium.com/nuances-of-programming/%D0%BF%D0%BB%D1%8E%D1%81%D1%8B-%D0%B8->

Три ключові елементи об'єдналися в технології мови Java:

- 1) Java надає для широкого використання свої аплети (applets) – невеликі, надійні, динамічні, які не залежать від платформи активні мережеві додатки, що вбудовуються в сторінки Web. Аплети Java можуть налаштовуватися і поширюватися споживачам з такою ж легкістю, як будь-які документи HTML;
- 2) Java вивільняє міць об'єктно-орієнтованої розробки додатків, поєднуючи простий і знайомий синтаксис з надійним і зручним в роботі середовищем розробки. Це дозволяє широкому колу програмістів швидко створювати нові програми і нові аплети;
- 3) Java надає програмісту багатий набір класів об'єктів для ясного абстрагування багатьох системних функцій, використовуваних при роботі з вікнами, мережею і для введення-виведення. Ключова риса цих класів полягає в тому, що вони забезпечують створення незалежних від використовуваної платформи абстракцій для широкого спектра системних інтерфейсів [9]¹⁾.

1.4.3 XML для розмітки GUI компонентів

Розробка графічного інтерфейсу може являти собою досить трудомістку задачу. Для її полегшення служать різні інфраструктури GUI, проте вони не завжди добре задокументовані і здатні звільнити розробника від необхідності написання великого обсягу коду, що уповільнює процес створення програми в цілому [7]²⁾.

%D0%BC%D0%B8%D0%BD%D1%83%D1%81%D1%8B-
%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1
%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%BD%D0%B0-java-
2861f4c2a0d5. (дата звернення 21.03.2019).

¹⁾ [9] Самоучитель по Java с нуля. URL: <https://vertex-academy.com/tutorials/ru/samouchitel-po-java-s-nulya/>. (дата звернення 24.03.2019).

²⁾ [7] Навчальні матеріали з інформатики. Програмування мовою Java. URL: <https://www.ua5.org/java/>. (дата звернення 20.03.2019).

Крім того, використання інтегрованих середовищ розробки (IDE) і інших графічних засобів, що підтримують технології створення GUI, часто стає причиною появи погано читається і важкого в підтримці коду. Це, в свою чергу, призводить до змішування програмних шарів, що реалізують бізнес-логіку і призначений для користувача інтерфейс, ускладнюючи подальший розвиток програми.

Один з підходів до вирішення цієї проблеми полягає в декларативному описі графічного інтерфейсу. Декларативність означає, що описується бажаний результат, а не послідовність дій для його досягнення.

Прикладом такого підходу є HTML, який служить для опису контенту, а не функцій, за допомогою яких він повинен генеруватися. Таким чином, декларативні мови не дозволяють повністю контролювати процес фактичного створення елементів інтерфейсу [10]³⁾.

Це може здатися недоліком, однак існують і переваги, зокрема, усунення побічних ефектів, таких як модифікація глобального стану (наприклад, глобальних змінних), а також виклик сторонніх функцій і методів.

Крім того, використання декларативної мови опису UI допомагає відокремити інтерфейс від решти коду програми. Подібне розділення має ряд вкрай важливих переваг, зокрема, воно дозволяє розмежувати ролі розробників і, як наслідок, скоротити витрати на інтеграцію бізнес-логіки з різними технологіями представлення даних в додатку [7]¹⁾.

1.4.4 Додаткові інструментальні засоби

³⁾ [10] Платформа Java – виртуальна машина Java, как и зачем она используется. URL: <https://hightech.in.ua/content/art-java-platform>. (дата звернення 24.03.2019).

¹⁾ [7] Навчальні матеріали з інформатики. Програмування мовою Java. URL: <https://www.ua5.org/java/>. (дата звернення 20.03.2019).

Adobe Photoshop – багатфункціональний редактор для роботи з фото та відеофайлами (растрові зображення і кілька векторних інструментів). Програмний продукт працює на ПК з операційними системами macOS, Windows і в мобільних версіях iOS, Windows Phone і Android. Використовується графічний редактор для комерційних цілей (телебачення, кінематограф, реклама, ігри, ретуш і так далі) [11]²⁾.

Для роботи поліграфістів, дизайнерів, фотографів, мультиплікаторів, архітекторів та інших творчих професій важливо мати повний набір інструментів для корекції мультимедійних файлів. Adobe Photoshop – незвичайний редактор, в якому можна не тільки коригувати і ретушувати зображення, а й перетворювати форми, зберігати GIF-анімації, використовувати шари і ефекти. Ці функції дозволяють легко трансформувати, клонувати, покращувати зображення, застосовувати різні фільтри для створення живописної, що продає картинки [12]³⁾.

Основні можливості Adobe Photoshop:

- ліцензована версія Adobe Photoshop оснащена великою кількістю функціональних можливостей для отримання високоякісних знімків;
- поліпшення якості відсканованих, віддрукованих знімків, усунення дефектів кольору під час зйомки;
- створення багатшарового зображення з можливістю редагування кожного окремо для передачі кольору, роботи зі складною графікою;
- монтування фотографій і картинок, окремих елементів в колажі;
- відновлення старих знімків, ретуш фотографій;
- робота з ескізами і кресленнями;

²⁾ [11] Інформаційний ресурс Ixbt. URL: <https://www.ixbt.com/soft/photoshop.shtml>. (дата звернення 12.12.2018).

³⁾ [12] Отличительные преимущества редактора Adobe Photoshop. URL: <https://soft-file.ru/adobe-photoshop/>. (дата звернення 26.03.2019).

- великий вибір інструментів для обробки текстових елементів (шрифти, ефекти, фільтри);
- обробка 2D і 3D моделей, додавання до них текстур, робота з формами;
- створення цікавих графічних елементів для поліграфії, дизайну, оформлення сайтів і так далі;
- підготовка до друку на принтері або в інтернеті і миттєва публікація файлів;
- робота зі стандартними форматами для растрових і векторних зображень;
- розфарбовування чорно-білих файлів [11]¹⁾.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Діаграми діяльності

Одним з основних етапів у розробці ігрового додатку є опис основної думки (стратегії чи правил), а також формування послідовності поведінки основних елементів. Створення інформаційної системи – складний процес, який можна уявити як поетапний спуск від загальної концепції майбутньої ІС, через розуміння її логічної структури до найбільш детальним моделям, що описує фізичну реалізацію [13]¹⁾.

¹⁾ [11] Photoshop. Інформаційний ресурс Ixbt. URL: <https://www.ixbt.com/soft/photoshop.shtml> (дата звернення 12.12.2018).

¹⁾ [13] Диаграмма деятельности UML. URL: <https://planerka.info/item/diagrammy-deyatelnosti-uml/>. (дата звернення 10.01.2019).

Діаграма діяльності належить до логічної моделі. В якості графічного уявлення для виділення основних функцій Системи ми застосовуємо діаграму варіантів використання (use case). Діаграма варіантів використання дає нам уявлення що повинна робити система.

Найбільша від'ємність діаграм діяльності полягає в тому, що вони підтримують і стимулюють застосування паралельних процесів. Саме завдяки цьому вони являють собою могутній засіб моделювання потоків робіт. Безліч імпульсів до розвитку UML 2 прийшло від людей, залучених в ці потоки робіт.

Можна також застосовувати діаграму діяльності в якості сумісної з мовою UML блок-схеми. Хоча це дозволяє розробляти блок-схеми, близькі до UML, але навряд чи це дуже захоплюючий процес.

В принципі, можна скористатися перевагами, наданими розгалуженням і об'єднанням, для опису паралельних алгоритмів одночасно виконуються програм.

Якщо застосовувати UML як мову програмування, то тут діаграми діяльності є цінним інструментом для подання логіки поведінки систем [14]²⁾.

Тобто якщо варіанти використання ставлять перед Системою мета, то діаграма діяльності показує послідовність дій, необхідних для її досягнення. Дії (action) це елементарні кроки, які не передбачають подальшу декомпозицію. Діяльність може містити входять і/або вихідні дуги діяльності, що показують потоки управління і потоки даних. Якщо потік з'єднує дві діяльності, він є потоком управління. Якщо потік закінчується об'єктом, він є потоком даних.

Діяльність виконується, тільки тоді, коли готові всі його «входи», після виконання, діяльність передає управління і (або) дані на свої «виходи». Саму діаграму діяльності прийнято розташовувати таким чином, щоб дії слідували зліва направо або зверху вниз.

²⁾ [14] Ведення в UML 2.0. URL: <http://bourabai.kz/dbt/uml/index.htm>. (дата звернення 11.03.2019).

Щоб вказати, де саме знаходиться процес, використовується абстрактна точка «маркер» (або «токен»). Візуально на діаграмі маркер не відображається, дане поняття вводиться тільки для зручності опису динамічного процесу.

Перехід маркера здійснюється між вузлами. Маркер може не містити ніякої додаткової інформації (порожній маркер) і тоді він називається маркером управління (control flow token), або ж може містити посилання на об'єкт або структуру даних, і тоді маркер називається маркером даних (data flow token) [15]¹⁾.

Взаємодія з будь-якою грою починається саме з меню. Для даної гри складається з двох кнопок – "New Game / Continue" і "Close" відповідно. Логіка роботи головного меню представлена за допомогою діаграми активності (рис.7).

Після старту нової гри спочатку йде перевірка на збереження минулого процесу гри, завдяки чому гравець має можливість продовжити попередній рівень. Для кожного рівня використовується рандомна генерація елементів поля, тобто у даному додатку відсутні однакові рівні.

¹⁾ [15] Проектирование программного обеспечения. Разработка веб-сайтов. UML Design. URL: <https://habr.com/ru/post/74330/>. (дата звернення 01.03.2019).

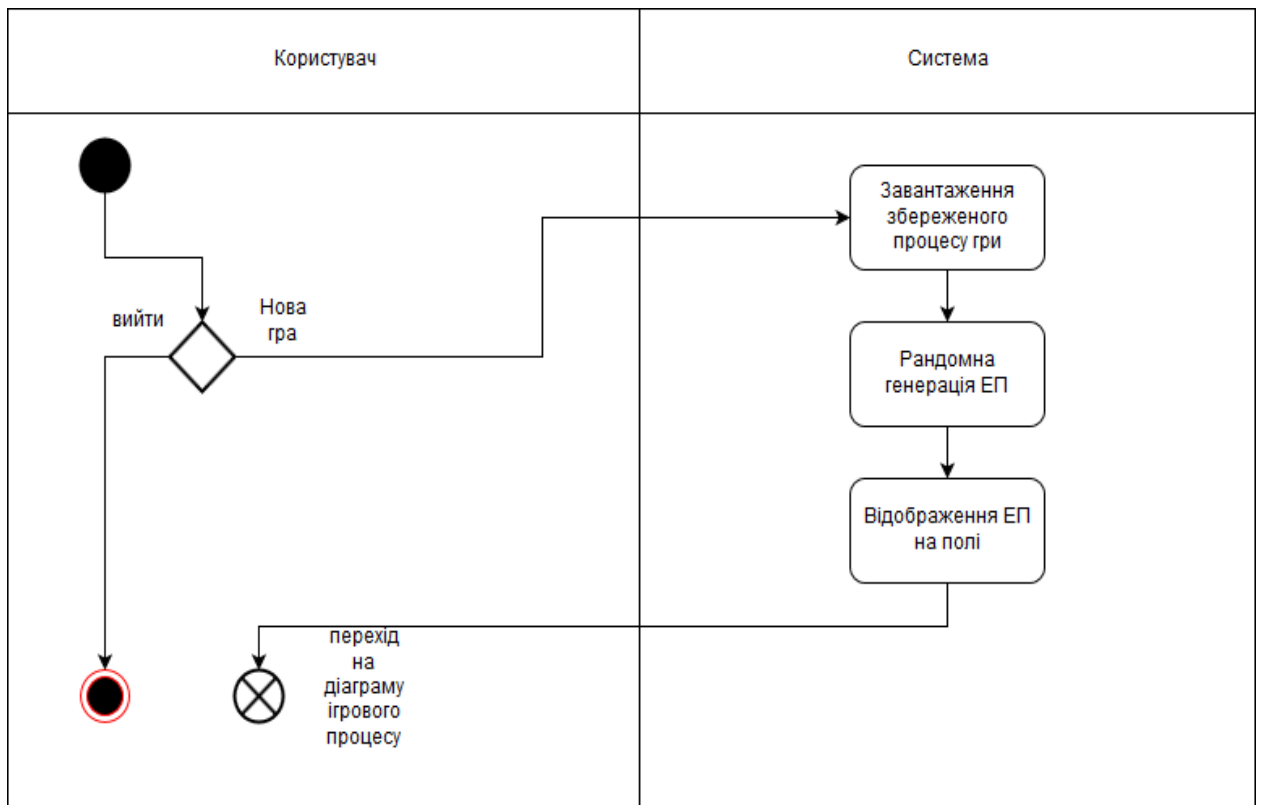


Рисунок 7 – Логіка роботи головного меню гри

Після цього система відображає генеровану послідовність елементів на полі гри. Подальші дії системи і гравець відображені на діаграмі ігрового процесу (рис. 8).

Якщо гравець натиснув кнопку миші на полі гри, система запускає процес заміни елементів поля на іншу комбінацію. Після цього йде перевірка на поєднані елементи поля.

У разі істинного результату система виконує поєднання елементів і виконує генерацію та з'єднання нового елементу на полі. Якщо умова переходу не виконана, система повертається на перший крок.

При натисканні кнопки «Clear» система виконує пошук елементів, які найбільш всього в кількості на полі. Після чого, йде генерація снаряда і відображення його на полі. При цьому, елементи поля, в які попали снаряди, знищуються.

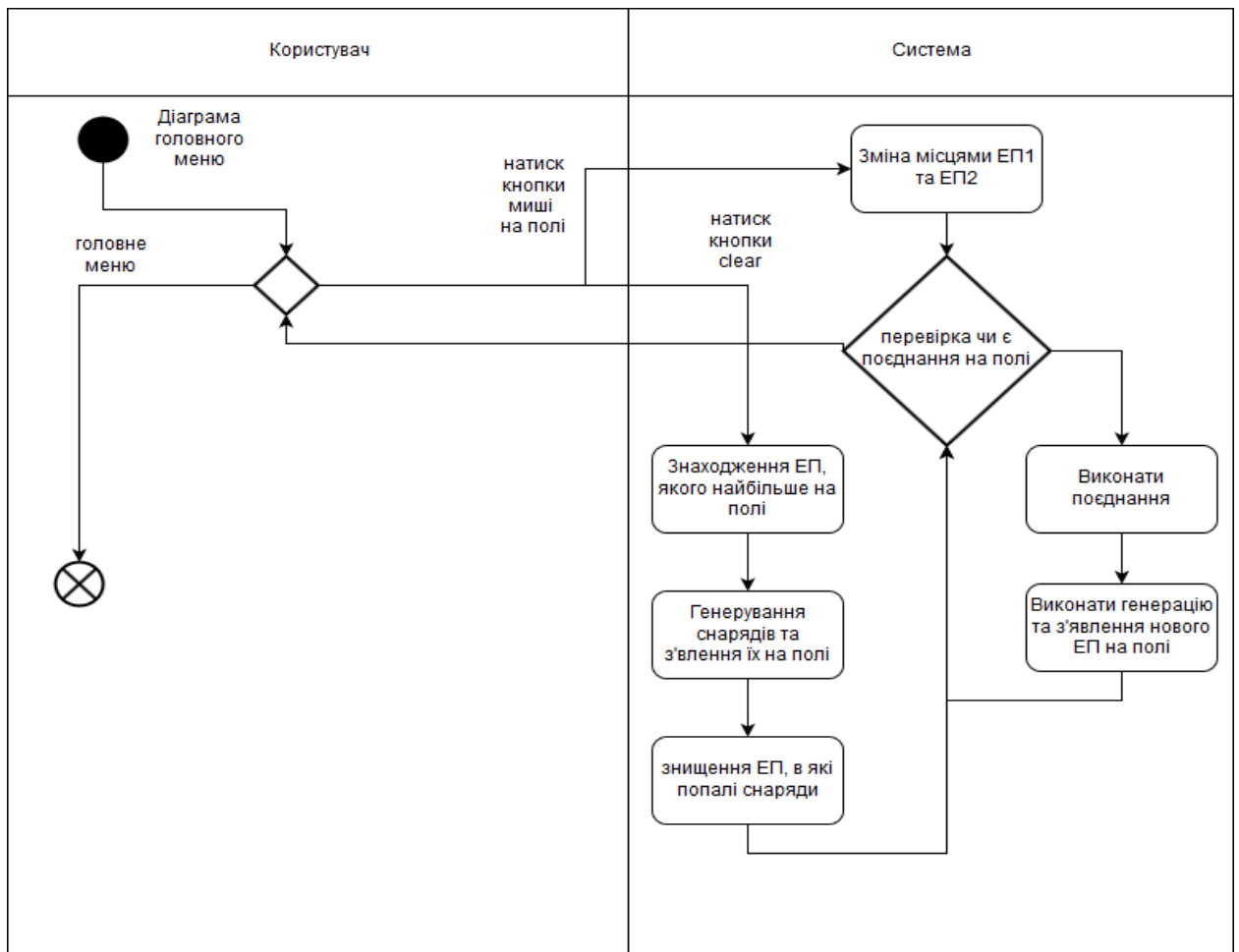


Рисунок 8 – Діаграма активності схеми роботи ігрового процесу

2.2 Діаграми класів

Діаграма класів використовуються при моделюванні системи найбільш часто. Вони є однією з форм статичного опису системи з точки зору її проектування, показуючи її структуру. Діаграма класів не відображує динамічну поведінку об'єктів зображених на ній класів. На діаграмах класів показуються класи, інтерфейси і відносини між ними [14]¹⁾.

Клас – це основний будівельний блок проектування системи. Це поняття є і в мовах програмування, тобто між класами UML і програмними

¹⁾ [14] Введення в UML 2.0. URL: <http://bourabai.kz/dbt/uml/index.htm>. (дата звернення 11.03.2019).

класами є відповідність, що є основою для автоматичної генерації програмних кодів або для виконання реінжинірингу.

Кожен клас має назву, атрибути і операції. Клас на діаграмі показується у вигляді прямокутника, розділеного на 3 області. У верхній міститься назва класу, в середній – опис атрибутів (властивостей), в нижній – назви операцій- послуг, що надаються об'єктами цього класу.

Атрибути класу визначають склад і структуру даних, що зберігаються в об'єктах цього класу. Кожен атрибут має ім'я і тип, який визначає, які дані він представляє. При реалізації об'єкта в програмному коді для атрибутів буде виділена пам'ять, необхідна для зберігання всіх атрибутів, і кожен атрибут матиме конкретне значення в будь-який момент часу роботи програми.

Об'єктів одного класу в програмі може бути як завгодно багато, всі вони мають однаковий набір атрибутів, описаний в класі, але значення атрибутів у кожного об'єкта свої і можуть змінюватися в ході виконання програми [15]¹⁾.

Для кожного атрибута класу можна задати видимість (visibility). Ця характеристика показує, чи доступний атрибут для інших класів. В UML визначені наступні рівні видимості атрибутів:

- відкритий (public) – атрибут видно для будь-якого іншого класу (об'єкта);
- захищений (protected) – атрибут видно для нащадків даного класу;
- закритий (private) – атрибут не видний зовнішніми класами (об'єктами) і може використовуватися тільки об'єктом, його містить.

Останнє значення дозволяє реалізувати властивість інкапсуляції даних. Наприклад, оголосивши всі атрибути класу закритими, можна повністю приховати від зовнішнього світу його дані, гарантуючи відсутність несанкціонованого доступу до них. Це дозволяє скоротити число помилок в програмі. При цьому будь-які зміни в складі атрибутів класу ніяк не

¹⁾ [15] Проектирование программного обеспечения Разработка веб-сайтов, UML Design. URL: <https://habr.com/ru/post/74330/>. (дата звернення 01.03.2019).

позначаються на решті частини системи. Клас містить оголошення операцій, що представляють собою визначення запитів, які повинні виконувати об'єкти даного класу. Кожна операція має сигнатуру, яка містить ім'я операції, тип значення і список параметрів, який може бути порожнім [14]¹⁾.

Реалізація операції у вигляді процедури – це метод, що належить класу. Для операцій, як і для атрибутів класу, визначено поняття «видимість». Закриті операції є внутрішніми для об'єктів класу і недоступні з інших об'єктів. Решта утворюють інтерфейсну частину класу і є засобом інтеграції класу в систему. На діаграмах класів зазвичай показуються асоціації та узагальнення.

Крім назви, асоціація може мати таку характеристику, як множинність. Вона показує, скільки об'єктів кожного класу може брати участь в асоціації. Множинність вказується у кожного кінця асоціації (полюса) і задається конкретним числом або діапазоном чисел. Множинність, зазначена у вигляді зірочки, передбачає будь-яку кількість (в тому числі, і нуль).

Узагальнення на діаграмах класів використовується, щоб показати зв'язок між класом-батьком і класом-нащадком. Воно вводиться на діаграму, коли виникає різновид будь-якого класу [15]²⁾.

Діаграма класів створюються при логічному моделюванні системи і служать для наступних цілей:

- для моделювання даних. Аналіз предметної області дозволяє виявити основні характерні для неї сутності і зв'язку між ними. Ці діаграми є основою для побудови концептуальної схеми бази даних;
- для уявлення архітектури системи;
- для моделювання навігації екранів. На таких діаграмах показуються прикордонні класи і їх логічний взаємозв'язок;
- для моделювання логіки програмних;

¹⁾ [14] Ведення в UML 2.0. URL: <http://bourabai.kz/dbt/uml/index.htm>. (дата звернення 11.03.2019).

²⁾ [15] Проектирование программного обеспечения Разработка веб-сайтов, UML Design. URL: <https://habr.com/ru/post/74330/>. (дата звернення 01.03.2019).

3 РОЗРОБКА ІГРОВОГО ДОДАТКУ

3.1 Основні компоненти ігрової моделі

3.1.1 Меню

Спочатку необхідно визначити базові класи меню, а саме:

- «Lcunte» використовується для ініціалізації параметрів розміру вікна, текстур, а також подальший виклик інших методів;
- «InputPrsc» реалізовує відклик програми на натиск кнопки любого пристрою (миша, клавіатура, геймпад тощо);
- «Enterante» клас відповідає за обчислення в грі розміру всіх текстур з урахуванням розміру вікна, перевірка поєднання елементів поля;
- «Condition» являє собою абстрактний клас для перемикання меню та ігрового процесу;
- «Menu» реалізує ігрове меню, що включає три кнопки («Нова гра», «Налаштування» та «Вихід»), наявність фону та музики.

3.1.2 Клас «Button»

Для опису функціоналу даного класу початку необхідно створити клас «TextComponent», який являє собою UI(User Interface) компонент, тобто він має програмно заданий текст. Цей компонент буде використовуватися і в інших далі описаних компонентах.

В цілому, цей клас має настройки, пов'язані з текстом, а саме:

- розмір тексту;
- координати розміщення;
- зрушення щодо координат (padding);
- колір тексту.

Наступним кроком є створення класу «Button», який і реалізує кнопку в грі. «Button» є спадкоємцем класу «TextComponent», що на відміну від

більшості ігор і вище перерахованих аналогів, кнопки тепер мають динамічний текст. Це дозволяє значно простіше в розробці (не треба перемальовувати текстуру кнопки заново), а так під час гри змінювати текст.

Крім налаштувань, які перейняті від parent-класу, «Button» має додатково масив текстур або кадрів, які в цілому утворюють анімацію натискання миші на кнопку. Ще так же, об'єкт класу «Sound» для програвання звуку, натискання на кнопку і об'єкт класу «ActionClass», який визначає дію для кнопки.

Додатково, крім тексту було зrealізовано програмне задаєвання кольору кнопки і кольору її анімації (рис. 10):

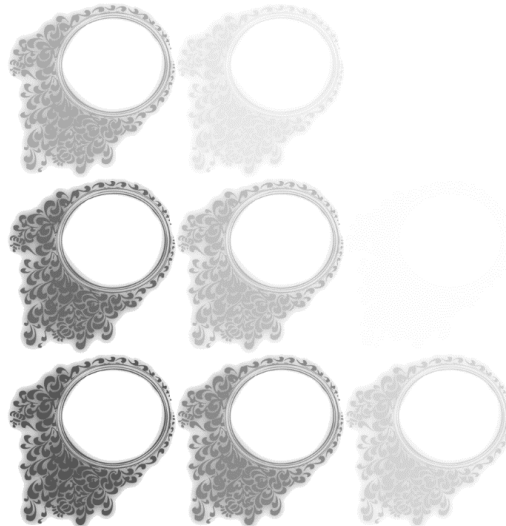


Рисунок 10 – Колірна маска для кадрів анімації кнопки

Останнім етапом в реалізації кнопки є її реагування на натискання миші. Наступний код демонструє принцип роботи окружного типу колізії:

```
if (x > Menu.buttonNewGame.x && x < TextComponent.size +
Menu.buttonNewGame.x &&
y > Menu.buttonNewGame.y && y < Menu.buttonNewGame.y + But-
ton_M.height)

if (activeComponent != null) {
```

```

if (activeComponent == Menu.buttonNewGame)
    return true;
activeComponent.animation = 2;
}

```

3.1.3 Ігровий процес

Наступним кроком у розробці ігрового додатку є створення ігрового процесу. Для цього необхідні такі класи, як:

- «Cell» – клас, що реалізує плитку поля та его анімацію, на який знаходиться один ЕП (Елемент поля);
- «Element» – клас, що реалізує самий ЕП. Тут Указані всі его характеристики та методи поведінки (рух, знищення);
- «ProgressBar» – клас реалізує методи відображення Лінії прогресу на сам показчик з урахуванням анімації;
- «Spark» – абстрактний клас, що реалізує Загальну поведінку – переміщення;
- «SparkBomb» – це реалізація спалахи, що знищує ЕП. Генерується при натисканні на кнопку «CLEAR»;
- «SparkClearCount» – це спалах, що добувається через швидке поєднання ЕП. Він вилітає зі статус бару та влітає в ClearCount-показчик;
- «SparkList» – спеціально розроблення список для спалахів;
- «SparkProgressBar» – клас для поведінки спалахів, що утворюються з поєднаніх ЕП;
- «Sparky» – відповідає за поведінку спалахів, що кружлять вокруг миші;
- «GameProcess» – великий клас, що реалізовує загальну поведінку ігрового режиму, він налічує 4 Паралельні потоки та всі методи викликаються з нього;

- «TSEx» – доповнення до «TextComponent», що забезпечує наявністю ще одного текстового рядка.

3.1.4 Ячейка ігрового поля

Перше, з чого варто було почати розробку ігрового процесу, це ячейка поля, на якій і будуть розміщуватися ЕП (рис. 11):

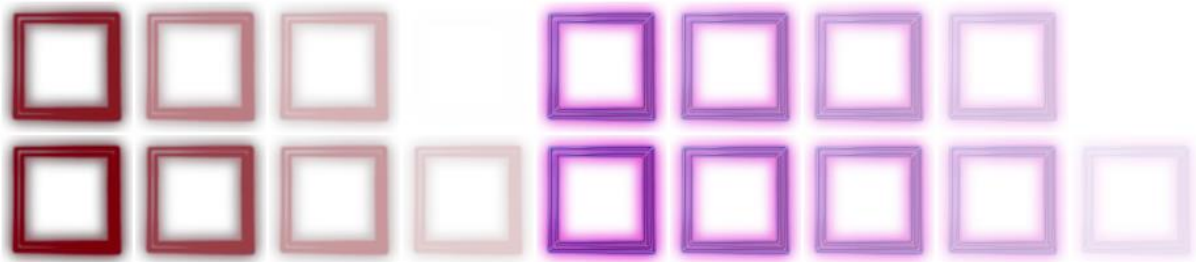


Рисунок 11 – Атлас двох типів кадрів ячейки

Даний клас «Cell» представляє не тільки осередок поля будучи об'єктом, а й є класом-менеджером всіх осередків в статичному вигляді. Тому перше властивість класу це статичний масив об'єктів типу Cell. Об'єктними властивостями ж класу «Cell» наступні:

- координати;
- відносний розмір;
- посилання на об'єкт типу Element;
- масив кадрів з текстур.

Відносний розмір – це параметр, який відповідає за розмірність осередки. Справа в тому, зі збільшенням рівня збільшується і розмір поля, але за рахунок цього зменшується розмірність осередки. І так як розмірність величина не статична, необхідно поле, яке описує її.

Посилання на об'єкт «Element» використовується для того, щоб зв'язати чітко осередок з елементом на ній. Це пов'язано з раціоналізацією визначення

натиснутого ЕП. Справа в тому, що ЕП знаходяться в динамічній структурі «Список», так як життєвий цикл кожного ЕП занадто динамічний – вони переміщуються, з'являються, видаляються, інше.

Складність визначення натиснутого ЕП за списком одно $O(\log(n))$, в той час як по осередку $O(1)$, адже відомо статичні координати кожного осередку і координатами миші можна миттєво визначити натиснуту осередок, а через посилання на «Element» є доступ на натиснутий елемент.

Масив кадрів призначений суто для барвистості. Коли гравець просто наводить курсором миші на будь-яку з осередків, вона починає підсвічуватися. Ще одним важливим досягненням було те, що в ході розробки було визначено, що швидкість підсвічування повинна бути значно вищою за швидкість погашення світіння осередку (рис. 12):

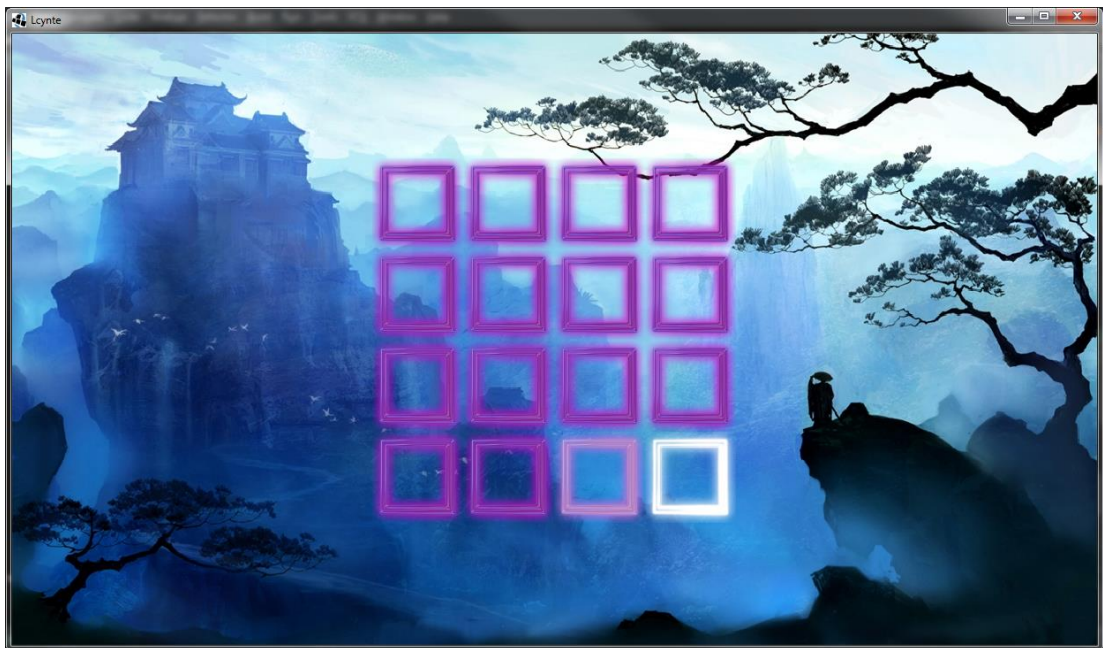


Рисунок 12 – Поле з ячеек

3.1.5 Елемент поля

Найголовнішим і інтерактивним елементом гри є Елемент Поля (ЕП).

Хоч він і приймає найбільше кліків в грі, ЕП не має обробника колізії з мишею. Як вище було згадано, обчислення натиснутого ЕП проводиться через осередок.

ЕП є найбільш динамічним об'єктами – вони з'являються, переміщуються і видаляються. Атлас текстур елементів ігрового поля представлено на рис. 13:



Рисунок 13 – Атлас текстур елементів

Слід розглянути наступний алгоритм, який відображає принцип переміщення ЕП. Для початку треба визначити, чи потрібно рухатися ЕП, для цього перевіряємо, якщо поточна комірка не остання по вертикалі і нижче її осередок порожня, то ініціювати рух:

```
if (cell.yسق != 0 && cell.sqcCell[cell.xسق][cell.yسق - 1].elem == null)
    isDying = 3;
```

Після ініціювання руху ЕП робимо зміщення вниз і намагаємося визначити осередок, на якій після зсуву опинився ЕП:

```
y -= velocity;
```

```
hor = (short)((x - InputPrCs.ofsX) / Cell.size);
ver = (short)((y + elemSize - InputPrCs.ofsY) / Cell.size);
```

Потім ЕП звільняється від осередку, на якій він раніше був, і прив'язується до нової:

```
cell.elem = null;
cell = Cell.sqcCell[hor][ver];
cell.elem = this;
```

І останнім кроком визначаємо, чи можна далі пересуватися. На це вказує наявність непорожній осередку далі по ходу вниз, або тільки що прив'язана осередок є останньою:

```
if (ver == 0 || Cell.sqcCell[hor][ver - 1].elem != null) {
    isDying = 0;
    short ofs = (short)((Cell.size - Element.size) >> 1);
    x = ofs + cell.x;
    y = ofs + cell.y;
}
```

Так як елементів поля аж 22, то робити кадрову анімацію для кожного це не раціонально, як за часом, так і за обсягом. Тому було прийнято рішення анімувати через зміну розміру.

Тобто коли ЕП віддалявся його розмір зменшувався, і коли він досягав величини менше 1, ЕП віддалявся зі списку. І навпаки – коли створювався новий ЕП, його розмір спочатку дорівнює 0, а потім поступово раз до нормальної величини ЕП.

Ну і завершальній родзинкою по відношенню до ЕП це те, що їх існує два типи. Вище згадувалося, що за все ЕП існує 22, але насправді їх 11, але для більшої барвистості, і щоб одні й ті ж ЕП не набридає, було зроблено дві текстури одного і того ж елемента. Тепер, за рахунок генератора, можна чергувати рівень за рівнем різні типи текстур.

3.1.7 Progress bar

Для завершення рівня потрібен якийсь прогрес, і природно, він повинен якось відобразитися і показувати загальний прогрес рівня. В іграх даного напрямку особливо не витрачають на це час і створюють часто числовий прогрес.

Наприклад, 51/100, де показано 51% рівня, або просту реалізацію Line Progress Bar, що представляє собою просту лінію, заповнює якимось кольором.

За основу для прогрес бару був узятий саме Line Progress Bar, але сам по собі він дуже простий, тому було вдатися до модифікацій. Перша ідея це змінювати колірний відтінок прогресу щодо вибитих ЕП.

Тобто якщо вибити три в ряд червоних ЕП, то прогрес бар ставав червоного відтінку. Але миттєво перефарбовувати прогрес бар в колір вибитого ЕП занадто різко, тому до основного кольору прогресу просто додавалося 10% від кольору ЕП.

Таким чином перефарбування відбувається більш плавно. Але і цього виявилось мало, і було прийнято рішення розбити прогрес на частини, при чому чим вище рівень, тим частин більше.

Кожна частина змінювала свою висоту незалежно від інших і на загальній картині це виглядало як міні хвилі різного кольору.

Ну і родзинкою прогрес бару є сам індикатор рівня. Справа в тому, що під час останньої реалізації прогресу за рахунок хвиль не було зрозуміло який саме відсоток рівня. Для цього було додано ще й індикатор, який брав відтінок того ділянки прогресу, на якому він стоїть (рис. 13):



Рисунок 13 – Progress Bar гри

3.1.8 Партікли

Для створення насиченості необхідно було створити якісь візуальні ефекти, але це дуже довго і робиться великою командою дизайнерів, але без цього гра здавалася б неповноцінною. Тому було вдатися до альтернативного рішення, яке давало досить збалансований результат, і це партікли. Партікли представляють собою кульки зі свіченням, які залишають хвіст при польоті, як комета.

Партікли потрібні були в двох місцях:

- як інструмент, який з'єднує прогрес бар і вибиті ЕП;
- бонус, який знищує ЕП.

Партікл як клас має двовимірний масив типу float. Перший рівень масиву відноситься до ланки хвоста партікла і кількість елементів масиву дорівнює довжині хвоста. Другий же завжди має розмір на три елементи, і це координата x, координата y і розмір ланки.

Так само у партікла є поле текстури, щоб можна було поставити партікл будь-якого кольору. Кожен партікл має на меті, це те, куди він прагне (переміщається) і відповідно має бути визначено дію, яке активується по досягненню мети.

Так, оскільки чіткого зв'язку між прогрес баром і вибитими ЕП не спостерігається, то якимось не зрозуміло, чому ж колір прогресу змінюється, візуально це ніяк не пов'язано. Тому перше застосування партікли знайшли тут. Для цього було створено близько 20 текстур для партіклів (рис. 14), які відображали відтінки відповідних ЕП. І їх не 22, що здається логічним, адже текстур для ЕП 22. А все тому що деякі відтінки ЕП приблизно однакові, і

візуально не відрізняються, тому було раціонально визначити загальні партікли для таких.



Рисунок 14 – Атлас текстур різних партіклів

3.2 Алгоритм MATCH 3

Алгоритм MATCH 3 забезпечує вибивання елементів поля. Кожну ітерацію гри відбувається обхід поля на перевірку ряду з 3-х або більше однотипних ЕП, і в разі, якщо такі знайдені, дати їм команду на вибивання.

Ще однією важливою властивістю це динамічність поля для вводів. Справа в тому, що у аналогів, коли якийсь ряд пішов проводиться дозаповнення поля, тобто створення нових ЕП, які падають згори і заповнюють тільки що спорожнілі осередки, і в цей час не можна зрушувати інші ЕП, які вже стоять на полі. Багато гравців згадували про це в негативному ключі, тому в дипломній роботі цей недолік був виправлений.

Ось приклад коду, який демонструє принцип визначення однотипних ЕП в кількості більше 3-х. Для початку йде перевірка горизонтального рядка:

```
if (cell1.xsqc + 2 < horis) {
```

```
if (cell.sqcCell[cell1.xsqc + 1][cell1.ysqc].elem != null &&
cell.sqcCell[cell1.xsqc + 2][cell1.ysqc].elem != null && e1-
```



```

em1.imageIndex == Cell.sqcCell[cell1.xsqc + 1][cell1.ysqc].elem.imageIndex && elem1.imageIndex == Cell.sqcCell[cell1.xsqc + 2][cell1.ysqc].elem.imageIndex)
return true;

}

```

Перевірка вертикального рядку реалізована наступним чином:

```

if (cell1.ysqc + 2 < vertis) {

if(Cell.sqcCell[cell1.xsqc][cell1.ysqc + 1].elem != null && Cell.sqcCell[cell1.xsqc][cell1.ysqc + 2].elem != null && elem1.imageIndex == Cell.sqcCell[cell1.xsqc][cell1.ysqc + 1].elem.imageIndex && elem1.imageIndex == Cell.sqcCell[cell1.xsqc][cell1.ysqc + 2].elem.imageIndex )

return true;
}

```

3.2.1 Бонус

У грі присутній бонус, який руйнує на поле ЕП якогось кольору. Колір визначається той, яких ЕП більше. Але знову таки, якщо натиснути на кнопку і елементи просто пішли б, то це виглядало б якось не щось не завершене.

Тому були створені партікли, які починали шлях з джерела і розліталися кожен до свого елементу поля (рис. 15):

Якщо елемент в ряду обставин почав переміщатися, то партікл коригував свою траєкторію польоту, а якщо ЕП пішов, то партікл відразу ж вибухав.



Рисунок 15 – Відображення руху спалахів Status Bar

На відміну від попередніх партіклов (ті, які були реалізовані для прогресу), тут була додана особливість – іскри (рис. 16-17).

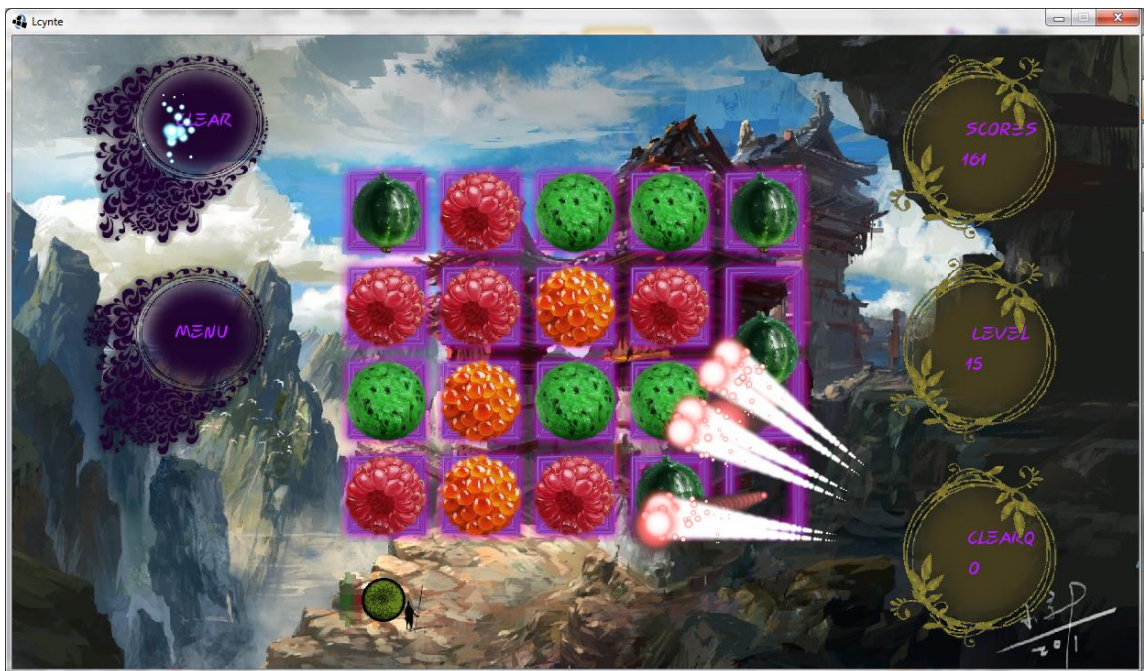


Рисунок 16 – Анімація бонусного спалаху



Рисунок 17 – Фінальна анімація бонусного спалаху

Так як для бонуса партікл повинен був якимось схожий на вогненну кулю були додані іскри, які додатково розліталися по ходу руху.

Для даної гри був розроблений алгоритм статичної генерації для спалахів миші. Так як вони доволі часто з'являються та зникають кожен раз викликати метод рандомізації не є оптимально, тому для цього було створено спеціально масив, що при старті гри рандомно заповнюється та надалі використовується для з'явлення кожного спалаху:

```
public Sparcly(float x, float y, float vector, Texture texture)
{
    this.x = x;
    this.y = y;
    this.texture = texture;
    if (vector==0)
        aride = sqcArideStep[globalArideStepIndex];
    else
        aride = (float)Math.acos(vectorX);
        arideStepIndex = globalArideStepIndex ++;
    if (globalArideStepIndex == sqcArideStep.length)
        globalArideStepIndex =0; }
}
```

Активація спалахів реалізована наступним чином:

```
this.size = size;
float originalSparkSize = 0.8f*size;
```

визначення максимальної ділини траєкторії:

```
sparclsizeStep = vel* originalSparkSize;
sqcSparkXY = new float[20][3];
short itera = 0;
```

```
do{
    sqcSparkXY[itera][0] = x;
    sqcSparkXY[itera][1] = y;

    sqcSparkXY[itera][2] = itera* original-
    SparkSize/sqcSparkXY.length;
}
```

```
while (++itera<sqcSparkXY.length);
}
```

Слід також визначити можливість поповнювати (заряджати) бонус. Зарядка бонусу проводиться шляхом розкручування індикатора прогрес бару (рис. 18). Коли вибиваються ЕП вони додають швидкості обертання індикатора, але сама по собі швидкість постійно падає і в кінці кінців, коли швидкість дорівнює 0, індикатор відповідно не обертається.

Але коли швидкість досягає певного значення, індикатор зупиняється і викидає партікл, який, долетівши до точки появи вогненних, інкрементує кількість зарядів бонусу.

Цей партікл так само має особливість, адже коли він долітає до кінцевої мети він вибухає, викидаючи маленькі частинки, які кружляють і зникають, що виглядає досить приємно.



Рисунок 18 – Партікл зарядки бонусу

3.2.2 Генератор рівня

Генератор забезпечує в першу чергу динамічність рівнів і нескінченний ігровий процес. Він займається не тільки генерацією поля, в його обов'язки входить також підбір музичної композиції, фон, стилістика індикаторів очок і рівня.

Щодо рівня за формулою визначається кількість осередків на поле по ширині і висоті, обсяг прогресу, а також якого типу елементи поля будуть згенеровані, адже вони відкриватися в міру збільшення рівня.

```
horis = (byte)(11*(level - 1) / 99 + 4);
vertis = (byte)(4*(level - 1) / 99 + 4);
ProgressBar.maxAccum = (short)(90*(level - 1) / 99 + 10);
ranQ = (byte)((8*(level - 1)) / 99 + 3);
```

Обчисливши розміри поля, генератор визначає розмір комірки і ЕП відповідно:

```
cell.size = (short)((Lcynte.screenHeight - (InputPrCs.ofsY << 1)) / vertis);
```

```
Element.size = (short)(0.9f*cell.size);
```

Останній генератор визначає самі Елементи поля і розміщує їх на ячейки:


```

hor = 0; ver = (short)(vertis - 1);

do{
    Cell.sqcCell[hor][ver].elem = (new Element
        (Cell.sqcCell[hor][ver], GameProcess.ran.nextInt(ranQ)));
}

while(++hor < horis);

```

Кінцевий результат формування стартового ігрового поля і його елементів представлено на рис.19:



Рисунок 19 – Сформоване поле гри

Якщо гравець вдало проходить перші рівні гри, він має змогу збільшення кількості ячеек у ігровому полі (рис. 20):

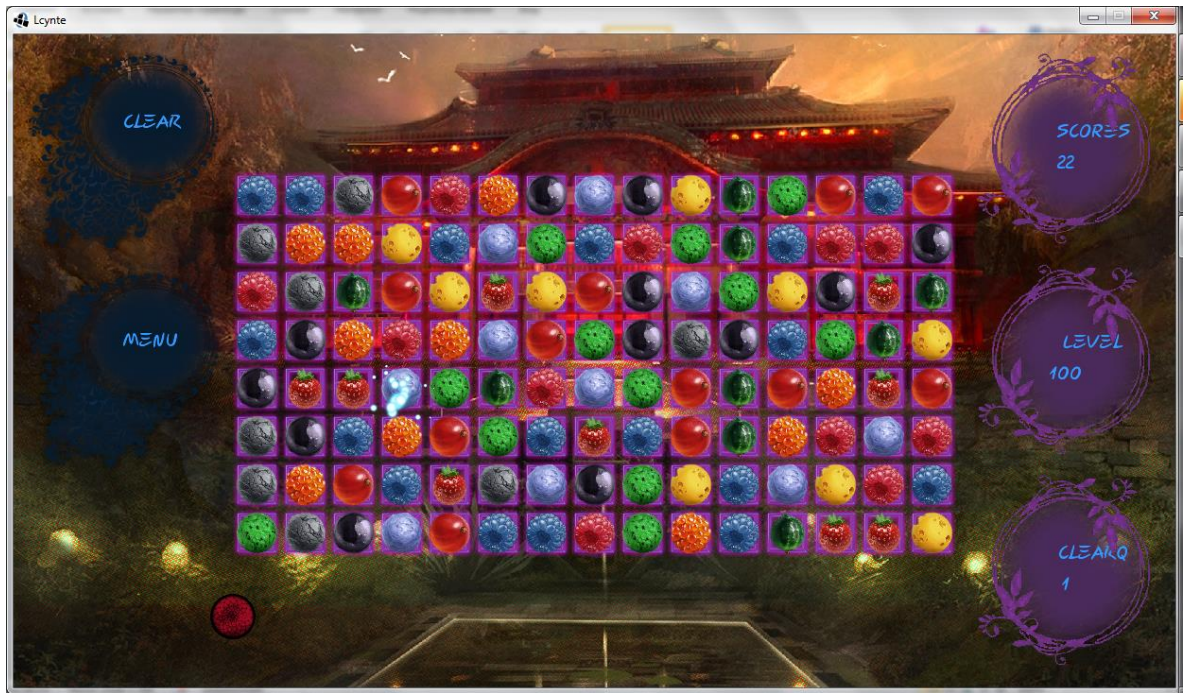


Рисунок 20 – Збільшене поле гри

ВИСНОВКИ

На сьогодні Java являє собою одну з найпоширеніших мов, та все дедалі більше «захоплює» світ. Жанр гри у дипломній роботі – це «Match three». Сама гра представлена у виді головоломки, де треба поєднувати три або більше елементів поля однаково типу в ряд. Ця гра призначена не для глибокого мислення, а більш для відпочинку.

Для дипломної роботи було обрано саме таку тематику, бо саме гра являє собою різноманітний проект, де одночасно присутній аналітичний підхід та творчий.

Творчий концепт становить графіка, музикальне супроводження та безпосередньо сюжет гри, а аналітичний – проектування основних дій на основі сюжету та програмування.

Під час роботи над проектом було виконано:

- аналітичний огляд предметної області, а саме створення ігрових додатків;
- зроблено аналіз топових аналогів об'єкту розробки, проаналізовано їх переваги і недоліки, на основі яких поставлені вимоги і сформовані мета роботи;
- розглянута мова UML, зокрема діаграма класів та діаграма діяльностей;
- обрані програмні засоби для реалізації проекту.

Результатом практичної частини дипломної роботи є:

- проектування алгоритму меню ігрового додатку за допомогою діаграм діяльностей;
- проектування структури проекту на основі діаграми класів;
- створення ігрового двигуна;
- реалізовано основні ігрові об'єкти, їх характеристики та взаємозв'язок;
- реалізовані додаткові спецефекти, на відміну від існуючих аналогів.

В рамках подальшого розвитку даної дипломної роботи планується:

- зареєструвати та продавати гру на Google Play Market, AppStore та Steam Greenlight;
- доповнювати графічним та звуковим контентом задля урізноманітнення відеогри.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Игры Match 3: в чем секрет успеха и привлекательности? URL: <http://renatus.com/ru/match-3-games-what-makes-fun-and-addictive>. (дата звернення 03.02.2019).
2. Что собой представляют игры жанра три-в-ряд или матч-3. URL: <http://smages.com/stati/chto-soboj-predstavlyayut-igry-zhanra-tri-v-ryad-ili-match-3/>. (дата звернення 03.02.2019).
3. Топ-10 лучших игр на андроид 2017-2018 года для вашего смартфона. URL: <https://www.fly-phone.ru/notes/android/luchshie-igry-dlya-android/>. (дата звернення 03.02.2019).
4. Жанровый анализ рынка мобильных игр. URL: <https://dtf.ru/gameindustry/23993-zhanrovyy-analiz-rynka-mobilnyh-igr>. (дата звернення 03.02.2019).
5. Топ игр Match3. URL: https://playliner.com/ru/report_match3_market_review_2.php (дата звернення 05.02.2019).
6. Обзор The Treasures of Montezuma 4. Складываем кристаллы. URL: <https://megaobzor.com/obzor-The-Treasures-of-Montezuma-4-skladyvaem-kristally.html> (дата звернення 12.02.2019).
7. Навчальні матеріали з інформатики. Програмування мовою Java. URL: <https://www.ua5.org/java/>. (дата звернення 20.03.2019).
8. Плюсы и минусы программирования на Java. URL: <https://medium.com/nuances-of-programming/%D0%BF%D0%BB%D1%8E%D1%81%D1%8B-%D0%B8-%D0%BC%D0%B8%D0%BD%D1%83%D1%81%D1%8B-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%BD%D0%B0-java-2861f4c2a0d5>. (дата звернення 21.03.2019).

9. Самоучитель по Java с нуля. URL: <https://vertex-academy.com/tutorials/ru/samouchitel-po-java-s-nulya/>. (дата звернення 24.03.2019).
10. Платформа Java – виртуальная машина Java, как и зачем она используется. URL: <https://hightech.in.ua/content/art-java-platform>. (дата звернення 24.03.2019).
11. Photoshop. Інформаційний ресурс Ixbt. URL: <https://www.ixbt.com/soft/photoshop.shtml> (дата звернення 12.12.2018).
12. Отличительные преимущества редактора Adobe Photoshop. URL: <https://soft-file.ru/adobe-photoshop/>. (дата звернення 26.03.2019).
13. Диаграмма деятельности UML. URL: <https://planerka.info/item/diagrammy-deyatelnosti-uml/>. (дата звернення 10.01.2019).
14. Ведення в UML 2.0. URL: <http://bourabai.kz/dbt/uml/index.htm>. (дата звернення 11.03.2019).
15. Проектирование программного обеспечения Разработка веб-сайтов, UML Design. URL: <https://habr.com/ru/post/74330/>. (дата звернення 01.03.2019).
16. Диаграммы классов UML. Логическое моделирование. URL: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=3>. (дата звернення 10.01.2019).