

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ	7
1.1. Аналіз існуючого програмного продукту.....	7
1.2 Концепт документ.....	10
1.3 Аналіз задачі, засобів та методів її вирішення.....	11
Висновки до розділу	12
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	13
2.1 Основні механіки оточення.....	13
2.2 Прототипи рівнів.....	16
2.3 Розробка штучного інтелекту	17
2.4 Розробка бойової системи.....	19
2.5 Розробка інтерфейсу	20
Висновки до розділу	22
РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ.....	23
3.1 Графіка оточення	23
3.2 Тестування рівнів.....	25
3.3 Тестування штучного інтелекту	27
Висновки до розділу	28
ВИСНОВКИ.....	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	30
ДОДАТКИ.....	31

ВСТУП

Актуальність теми. Комп'ютерні та консольні ігри є одним з найпоширеніших видів розваг в наші часи. Великою перевагою ігор перед іншими видами розваг є те, що існують ігри на будь-який смак і кожен гравець може знайти щось до душі. Онлайн ігри дозволяють спілкуватися з друзями дистанційно і бути залученими до спільної діяльності, або ж знайомитися з гравцями з інших країн світу, що розвиває гравця як особистість. Ігри розвивають багато якостей таких як координація очей та рук, моторика, стратегічне мислення або роботу в команді. І хоча ігри мають свої недоліки: навантаження на зір та інтенсивні розумові процеси, якщо все робити в міру ігри можуть бути корисними, позбавити стресу та допомогти відволіктися від щоденної рутини. Тому актуальність ігрової індустрії зараз як ніколи висока.

Одним із жанрів, який набуває популярність є соулслайк (souls-like, з англ. souls-подібні). Назва жанру походить від серії ігор Dark Souls, яка започаткувала жанр і визначила головні його характеристики. Соулслайк ігри захоплюють гравця своє високою складністю та грамотно побудованим світом. Хоча, як правило, соулслайк ігри дуже складні, вони справедливі, тобто гру можливо пройти від початку до кінця без особливих навичок. Смерть головного героя - це велика частина гри для ігор жанру соулслайк, гравцю потрібно вивчати усі можливі атаки ворогів і вміти на них реагувати. Така структура ігрового процесу дозволяє гравцю постійно випробовувати свої навички і отримувати моральне задоволення від кожної перемоги.

Мета дослідження – отримати практичні навички з розробки простої гри, навчитися працювати в команді над спільним проектом.

Об'єктом дослідження даної курсової роботи є технологія створення ігрових додатків та дизайну рівнів.

Предметом дослідження – побудова рівнів для гри платформеру, розробка бойової системи, штучного інтелекту для ворогів та розробка інтерфейсу.

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1. Аналіз існуючого програмного продукту

Соулслайк ігри стали досить поширеними за останні роки, від невеликих інді ігор до AAA проектів відомих студій. Проте найбільшим джерелом натхнення для цієї курсової роботи є гра Salt and Sanctuary. Salt and Sanctuary була революційною грою в свій час, це була перша суміш соулслайку і платформеру, що призвело до чималої популярності гри.



Рис. 1.1. Обкладинка Salt and Sanctuary
(<https://store.steampowered.com/agecheck/app/283640/>).

Окрім захоплюючої системи бою та цікавої історії Salt and Sanctuary відзначає ще одну рису соулслайків дуже добре - продуманий дизайн рівнів. Соулслайк ігри славляться своїми розумно побудованими рівнями, що винагороджують гравця за дослідження та час від часу відкривають зрізи для швидшого переміщення між локаціями. Salt and Sanctuary також вдало впорався з цим завданням, але відмінність від інших соулслайк ігор в тому, що рівні побудовані у виді двовимірного платформеру. Не дивлячись на обмеження в вимірах Salt and Sanctuary має велику, продуману мапу і налічує 23 локації, кожна з своїм унікальним візуальним стилем та своїм набором ворогів. Також розробник використовував елементи метроїдванії: кожна локація має місця до яких потрібно повернутися потім, після отримання нової здібності чи спорядження. Завдяки

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

цьому гравець відвідує кожену локацію декілька разів, що дозволяє обійтися меншим розміром локацій не втрачаючи захоплення від дослідження світу.

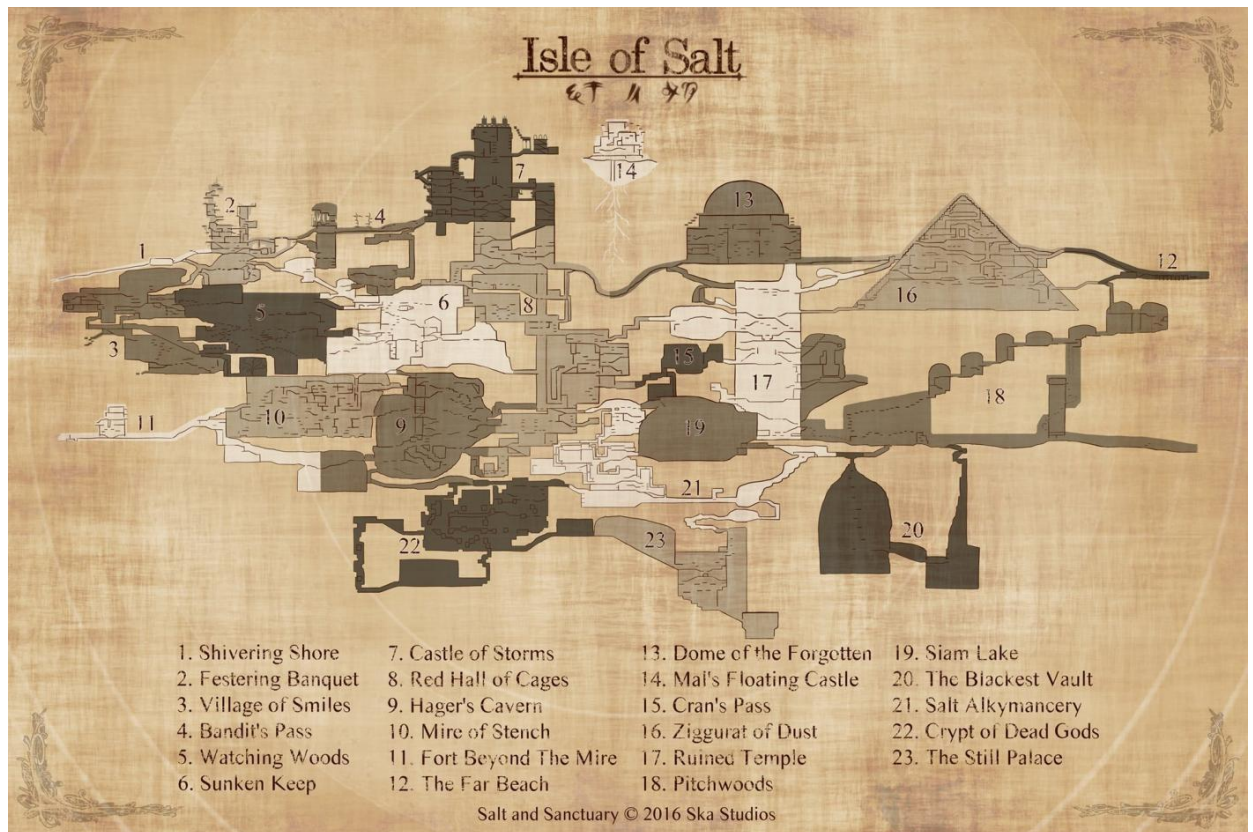


Рис. 1.2. Мапа усіх локацій Salt and Sanctuary.

Як можна побачити на мапі (рис. 1.2.) усі локації пов'язані між собою. Це також є одною з характеристик соулслайк ігор - безліч зрізів, які допомагають гравцю зручно переміщатися між локаціями. Це також допомагає спрямувати гравця в потрібні місця, якщо наприклад двері відкриваються тільки з однієї сторони чи гравець ще не здобув потрібної здібності для подолання перешкоди. Таким чином чим більше гравець досліджує мапу, тим більш вільно він може по ній переміщатися. Це досить корисно, якщо враховувати ще одну характеристику соулслайків - рідкісні точки відродження. Коли гравець помирає, а це трапляється часто, він відроджується в останньому святилищі, яке активував. Ці святилища зустрічаються досить рідко і складність соулслайк ігор частково в тому, щоб пережити сутички з ворогами до наступної точки збереження. Тому також важливо враховувати точки відродження при побудові рівнів і розрахувати чи не надто легко дійти від однієї такої точки до іншої.

Звичайно не можна говорити про соулслайки не згадавши серію ігор Dark Souls. Іноваційна на свій час формула здобула багато фанатів і тримає Dark Souls в ряді культових ігор. Є багато аспектів які зробили серію Dark Souls такою популярною, але для цієї курсової роботи сфокусуємося на основній частині гри - бойовій системі.



Рис. 1.3. Dark Souls III

(https://store.steampowered.com/app/374320/DARK_SOULS_III/).

В центрі бойової системи знаходиться так звана стаміна (stamina, з англ. витривалість). Усі бойові дії: атаки, блокування, ухиляння, магичні заклинання витрачають стаміну і не відтворюються якщо стаміни немає. Це змушує гравця не бездумно розмахувати своєю зброєю на право і на ліво, а вивчати атаки ворогів та уважно підбирати моменти для контратаки. Стаміна повністю відновлюється з деякою затримкою і ця затримка стає більшою, якщо гравець витратив всю стаміну.

Іще одна велика характеристика бойової системи соулслайків це незначна живучість головного героя. Навіть найслабкіші вороги можуть з легкістю вбити головного героя за декілька ударів. Це ще більше доповнює ідею з стаміною, оскільки будь-яка помилка може коштувати гравцю життя.

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

Цікава бойова система тримається не лише на можливостях гравця, а і на ворогах та їх штучному інтелекту. Підходящим аналогом ворогів для цієї курсової роботи є вороги з соулслайк гри Blasphemous. Вороги в цій грі не мають дуже складного штучного інтелекту, кожен ворог має 1-2 основні здібності і способи переміщення, але це не робить їх менш цікавими. В соулслайк іграх головне не що можуть робити вороги, а як вони це роблять.



Рис. 1.4. Blasphemous (<https://store.steampowered.com/app/774361/Blasphemous/>).

1.2 Концепт документ

Цільова аудиторія: 16+ хардкорні гравці на ПК, що люблять суміш платформерів та соулслайків.

Основні особливості гри: гра переносить гравця у потойбічний світ наповнений жахливими чудовиськами, чудернацькими будівлями та екзотичними рослинами.

Геймплей: Гравцю доведеться досліджувати карту наповнену міфічними ворогами та старовинною зброєю. Вороги заважатимуть спокійному просуванню по карті. Свя-тилища слугують точкою збереження та відновлення здоров'я, проте відпочивши, усі чудовиська воскресають. У ході дослідження ігрової карти гравець знаходить покращення. Одна зброя матиме різні удари, що мають різний обсяг

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

шкоди, швид-кість та дальність атаки. Також можна комбінувати атаки послідовно натискаючи відповідні клавіші. На шляху гравця також будуть ставати перешкоди, які він не зможе подолати відразу, потрібно бути віднайти ключ чи нову здібність аби її по-долати. Головний герой матиме перелік характеристик, які безпосередньо вплива-тимуть на його бойові здібності: сила чи швидкість атаки, кількість здоров'я, тощо. Ці характеристики можна знайти у точках інтересу, що дозволить легше пе-ремагати монстрів і використовувати ефективніше зброю.

Сюжет: У ході підготовки до війни, табір головного героя піддається нападу ворогів. Молодий солдат Лютомисл зазнає смертельного поранення. Прийшовши до тями, він бачить, що опинився в невідомому місці чи як стане відомо згодом, в невідомому іншому вимірі. В світі наповненому чудовиськами, чудернацькими будівлями та екзотичними рослинами Лютомислу доведеться знаходити нове спорядження та ставати сильнішим аби вижити. У кінці буде нелегкий вибір: залишитись володарем потойбічного світу або все ж отримати другий шанс на життя і повернутися до людського світу.

1.3 Аналіз задачі, засобів та методів її вирішення

Головною задачею для дизайнера рівнів в соулслайк жанрі є добре продумати рівні та точки інтересу на них. Також потрібно випробовувати навички гравця не лише в бою з ворогами, а й в подоланні перешкод розміщених на рівні. Гравець повинен бути зацікавлений в дослідженні світу. Для побудови рівнів буде використовуватися редактор рівнів вбудований в Unity та для механік оточення, бойової системи будуть створенні C# скрипти за допомогою Microsoft Visual Studio 2022.

Окрім цього, оточення гравця повинно чіпляти око і передавати атмосферу потойбічного, невідомого фентезі світу. Для створення спрайтів буде використовуватися Aseprite, а також будуть використані системи партіклів вбудовані в Unity.

		Бомблюк М. О.			Житомирська політехніка.20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

Оскільки курсова робота виконується в команді, для зручної комунікації та розподілення завдань буде використовуватися Trello та різні сервіси від Google, такі як Google Документи чи Google Meet.

Висновки до першого розділу

Отже, результатом курсової роботи повинна бути гра жанру соулслайк з елементами платформеру. Гра повинна мати цікавий та продуманий дизайн рівнів, що використовує традиційні механіки платформер. Також гра повинна мати приємний візуальний стиль, який буде передавати атмосферу потойбічного світу. Для виконання завдання буде використовуватися Unity, Aseprite, Trello, Google Документи, Google Meet, тощо.

		Бомблюк М. О.			Житомирська політехніка.20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Основні механіки оточення

Якісні платформери мають спільні механіки, які роблять платформінг більш різноманітним і відкривають нові можливості для левелдизайнера. Деякі такі механіки також мають бути реалізовані в цій курсовій роботі:

- Платформи - гравець проходить крізь них знизу і по бокам, але може стояти на верху платформи. Також важливо щоб можна було провалитися крізь платформу коли гравець стоїть на ній.
- Платформи, що ламаються - коли гравець стає на таку платформу через деякий час вона ламається і гравець падає.
- Шипи - спосіб наносити шкоду гравцю елементами карти.

На щастя, платформи вбудовані в редактор Unity, тому все що треба це поставити на потрібний об'єкт компонент "Platform Effector 2D", та налаштувати його:

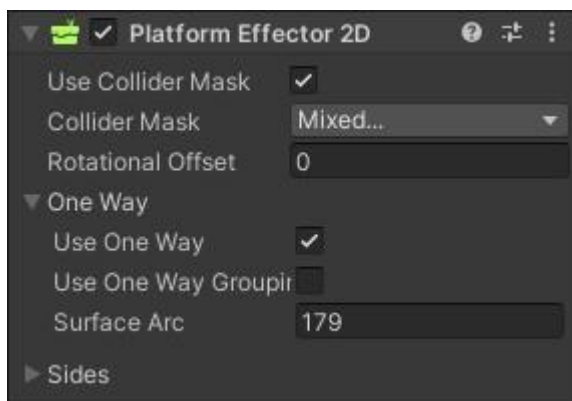


Рис. 2.1. Platform Effector 2D.

Далі виставляємо Surface Arc 179, щоб гравець проходив крізь платформу з обох боків та Collider Mask де вказуємо потрібні шари. Реалізувати ламання платформи можна простим скриптом:

```
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.tag == "Player")
    {
        StartCoroutine(startBreaking());
    }
}
```

```

private IEnumerator startBreaking()
{
    particleBreaking.Play();// активуємо партікли для більш ефектного ламання
    animator.SetBool("Breaking", true);// активуємо анімацію для кращого розуміння, що платформа ось ось зламється
    yield return new WaitForSeconds(1);

    particleBreaking.Stop();
    StartCoroutine(Destruct());
}

private IEnumerator Destruct()
{
    particleBroke.Play();
    spriteRenderer.enabled = false;
    boxCollider.enabled = false;

    yield return new WaitForSeconds(particleBroke.main.startLifetime.constantMax);
    // знищуємо об'єкт коли партікли закінчують програватися
    Destroy(gameObject);
}

```

Шипи реалізуються схожим чином, але перед тим як зламатися вони наносять шкоду гравцю:

```

private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.tag == "Player")
    {
        var player = collision.GetComponent<Player>();
        if (player.StateMachine.CurrentState == player.inAirState)
        {
            player.GetDamaged(damage, box_collider);
            StartCoroutine(Destruct());
        }
    }
}

private IEnumerator Destruct()
{
    particle.Play();
    spriteRenderer.enabled = false;
    box_collider.enabled = false;
    yield return new WaitForSeconds(particle.main.startLifetime.constantMax);

    Destroy(gameObject);
}

```

Одною з механік рівнів в жанрі соулслайк є зрізи, які зазвичай реалізуються дверима, що відкриваються лише з однієї сторони чи лише певним ключем. Тому створимо переліки, які дозволять зручно вибирати призначення дверей на рівні інспектора:

```

enum OpenFrom { Both, Left, Right }
enum Keys { None, Prison, Lift, Mason, Boss }

```

Далі простими “if”, “else” виразами будемо перевіряти які умови відкриття дверей і чи гравець виконує ці умови при взаємодії з ними.

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

Якщо дверям можуть бути потрібні ключі, також треба створити спосіб отримання цих ключів, що підводить нас до наступної механіки оточення - предмети та скрині. Досліджуючи мапу, гравець матиме змогу знайти предмети та скрині, які можуть містити в собі ключ чи покращення характеристик. Так само додаємо потрібні переліки:

```
enum Keys { None, Prison, Lift, Mason, Boss }
enum Upgrade { None, HP, SP, Damage, AtkSpd }
```

І тоді при взаємодії предмет знищиться, а скриня програє анімацію відкриття і стане не активною. В залежності від обраної нагороди, гравець отримає покращення чи ключ.

Також, будь-який соулслайк повинен мати рідкісні точки збереження, де гравець відновлює все здоров'я та може перепочити. Відповідно, потрібно розробити систему збереження і для цього буде зручно використати шаблон проектування сінглтон. Сінглтон гарантує, що клас матиме лише 1 екземпляр і це дозволить викликати з нього методи, не маючи посилання на об'єкт чи якогось прямого зв'язку до нього. Дані будуть зберігатися на машині гравця у вигляді JSON файлу. Для серйознішого проекту, варто було б зашифрувати дані в файлі, щоб гравець не міг їх легко змінити, але для цієї курсової роботи JSON формату буде достатньо. Гра буде зберігати усі підняті покращення, ключі та відкриті двері, але точка відродження зберігатиметься лише в точках збереження - святилищах.

Остання механіка, яку буде додано до оточення це об'єкти, які гравець може зламати. Це додасть різноманітності та дозволить створювати головоломки пов'язані з ламанням того чи іншого об'єкту. Також це не складно зробити, адже потрібно лише модифікувати вже існуючі скрипти. Коли гравець б'є по об'єкту буде викликатися метод GetDamaged():

```
public void GetDamaged()
{
    health -= 1;

    if (health <= 0)
    {
        StartCoroutine(Destruct());
        return;
    }

    hitParticle.Play();
    animator.SetTrigger("gotHit");
}
```

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

```

}

private IEnumerator Destruct()
{
    boxCollider.enabled = false;
    spriteRenderer.enabled = false;
    breakParticle.Play();

    yield return new WaitForSeconds(breakParticle.main.startLifetime.constantMax);

    Destroy(gameObject);
}

```

2.2 Прототипи рівнів

Визначившись та реалізувавши механіки оточення можна переходити до побудови карти. Варто враховувати розроблені механіки та використати їх на повну.

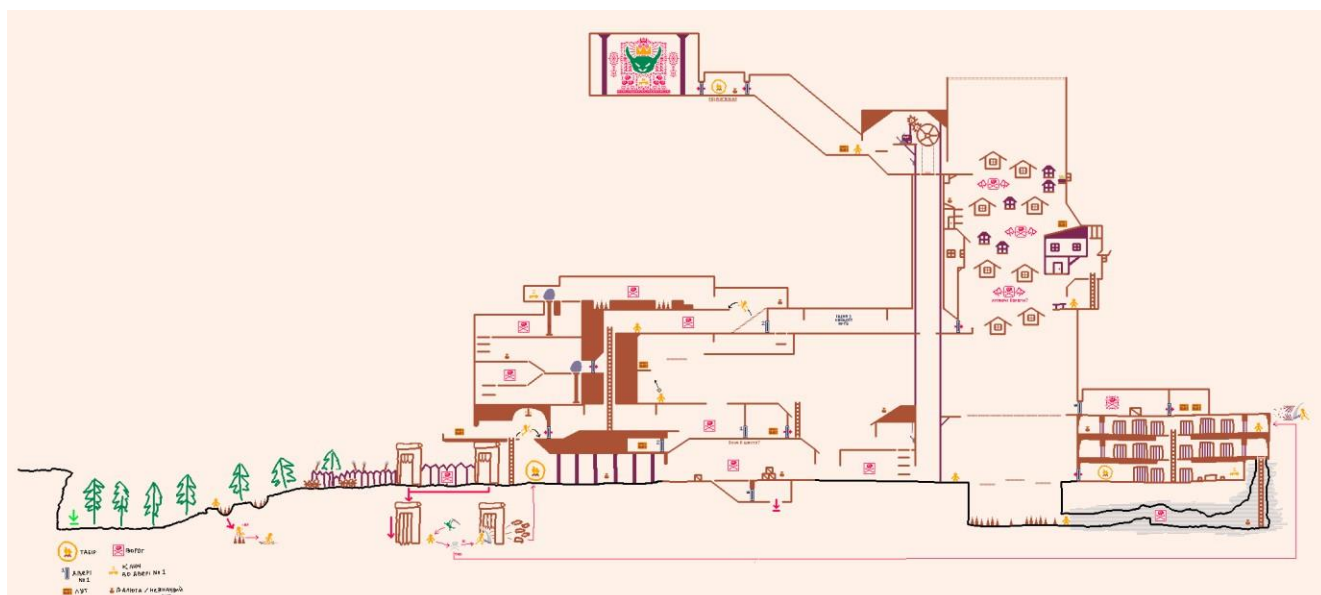


Рис. 2.2. Ескіз карти.

На розробленому ескізі (рис. 2.2) використовуються усі механіки, а також відзначені місця для ворогів. Щоб дійти до першої точки збереження гравцю доведеться перестрибнути ями з шипами та зламати ворота у місто гоблінів, де він також знайде свої перші покращення. Також в цю кімнату з точкою збереження ведуть ще одні двері, які відкриваються з іншої сторони і дозволять гравцю далі по грі відкрити цей зріз і зцілитися в точці збереження.

В правій нижній частині карти гравцю потрібно перестрибнути велику яму з шипами використовуючи крихкі платформи щоб дістатися в'язниці, де він отримає ключ тюремника, який відкриває нагороду в центральній частині карти. По такому

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

ж принципу буде побудована вся карта і кожен куточок ігрового світу буде чимось наповнений.

Геймплей виходить різноманітний, бо гравець не лише б'ється з ворогами, а також випробовуються його навички платформінгу та вміння вирішувати головоломки.

2.3 Розробка штучного інтелекту

Карту не достатньо наповнити лише власними механіками та декораціями, потрібно також додати елемент небезпеки - ворогів. Для штучного інтелекту ворогів буде використовуватися стейт машина - паттерн, що передбачає так звані "стани" об'єкту, між якими здійснюються переходи в ході реакції на дії гравця чи оточення.

Для цієї курсової роботи буде розроблено 3 типи ворогів та кожен з них матиме свій набір станів:

- Спільні для усіх стани:
 - Спокій
 - Переслідування гравця
 - Повернення на початкову точку
 - Отримання шкоди від гравця
 - Смерть
- Звичайний, слабкий гоблін:
 - Блукання - випадково блукає в заданій зоні
 - Атака
- Гоблін шаман, який стріляє в гравця вогняними кулями
 - Постріл
- Елітний гоблін, який блокує атаки гравця
 - Патрулювання - ходить від точки А до точки Б
 - Атака
 - Блок

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

За основу стейт машини ворогів буде взято стейт машину гравця, розроблену членом команди Богданом Дзюбанчуком. Роботу стейт машини легше зрозуміти на блок схемі (рис. 2.3). Також в якості приклада розглянемо стан переслідування (рис. 2.4).



Рис. 2.3. Загальний алгоритм стейт машини.

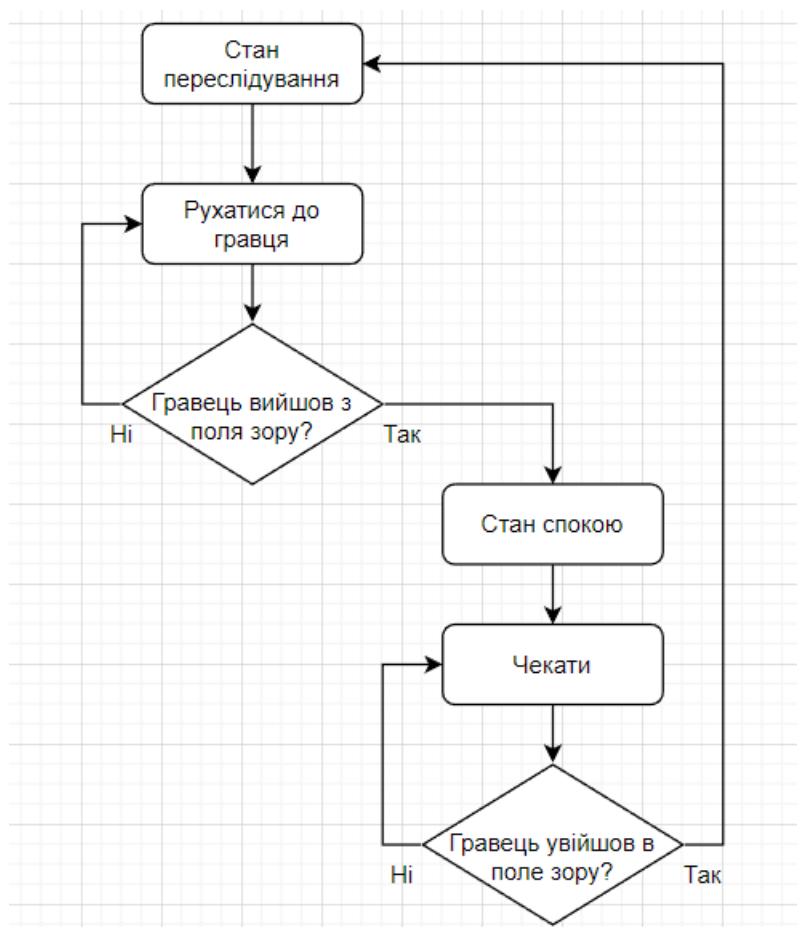


Рис. 2.4. Спрощений алгоритм стану переслідування.

Кожен стан виконує дії задані розробником на етапі входу, вихода з стану, кожен кадр в цьому стані та дії вказані в анімаційних тригерах. Це дозволяє робити багато різновидів поведінок для ворога і є досить простим в реалізації для проектів малого масштабу.

2.4 Розробка бойової системи

Тепер, коли вороги мають усі інструменти для знищення гравця, треба дати гравцю можливість відбиватися. Окрім механік бойової системи, вказаних в першому розділі, також буде реалізовано систему комбо атак, схожу до системи комбо Dark Souls ігор.

Комбо атаки в серії Dark Souls можна зобразити так:

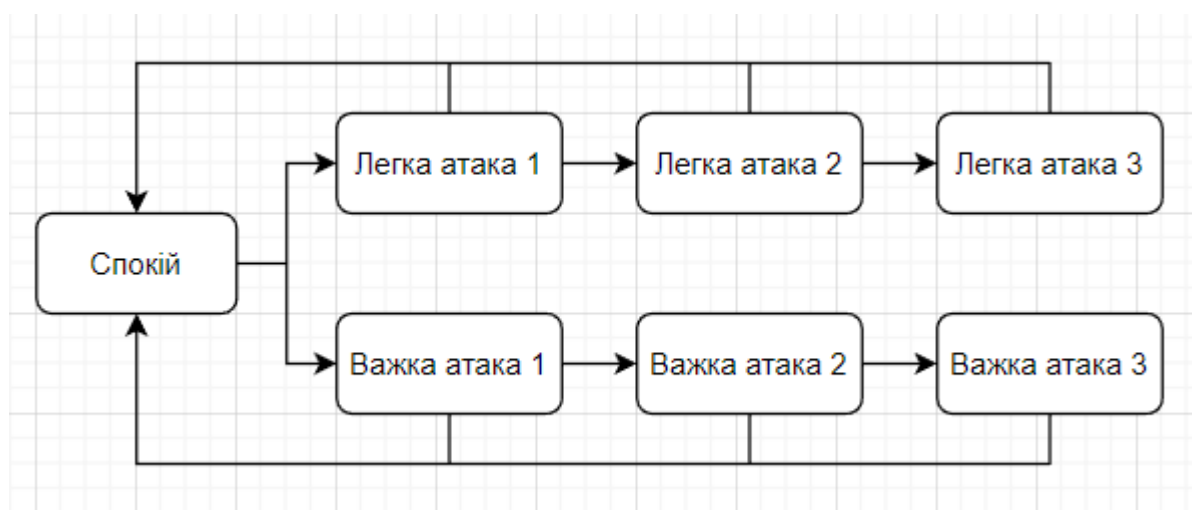


Рис. 2.5. Комбо атаки в серії Dark Souls.

Гравець має 2 атаки: легку та важку і натискаючи відповідні для них кнопки він може комбінувати легкі з легкими та важкі з важкими. Після будь-якої атаки можна перервати комбо і почати нове. Проте для цієї курсової роботи буде використовуватися інша схема:

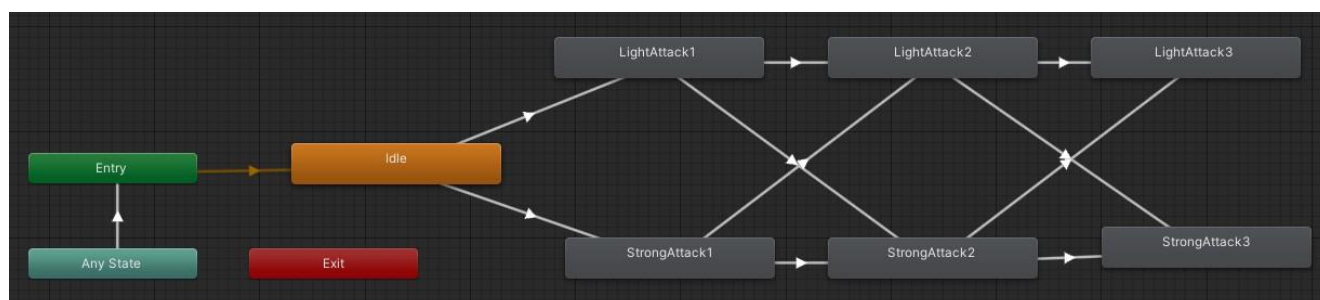


Рис. 2.6. Розроблені комбо атаки.

Кожним натиском на будь-яку атаку гравець збільшує рахунок комбо на 1 і тоді програвється анімація натиснутої атаки відповідного рахунку комбо. Це дозволяє маючи таку саму кількість анімацій створити набагато більше варіацій комбо. Так само якщо не продовжити комбо в потрібний момент, гравець перейде в стан спокою (Any State > Entry > Idle).

Також, як і в серії Dark Souls, кожна атака супроводжується невеликим кроком вперед, що додає динамічності та більшої дальності атакам.

2.5 Розробка інтерфейсу

Гравцю також потрібно візуальна репрезентація здоров'я та стаміни, коли вони грають таку важливу роль в бойовій системі. Хоча Unity має досить потужні вбудованні UI ассети, він не має progressbar елемент, який часто використовують для відображення даних у виді смужки. Проте Unity має елемент slider, який може виконувати ту саму функцію, прибравши можливість гравцю змінювати значення власноруч. Таким чином були створені такі елементи UI:



Рис. 2.7. Смужка здоров'я та стаміни.



Рис. 2.8. Відображення характеристик персонажу (сила, швидкість атаки).

Також використовуючи можливості UI в Unity, було створено меню паузи (рис. 2.9), де також відображається скільки покращень зібрав гравець, що стимулює досліджувати карту, аби знайти усі покращення.

Будь-яка гра повинна мати головне меню, яке зустрічає гравця при запуску гри. Використовуючи схожий до меню паузи принцип, було створено головне меню (рис. 2.10), з можливістю створювати новий файл збереження, якщо файл збереження присутній в системі (рис. 2.11).

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		20



Рис. 2.9. Меню паузи.



Рис. 2.10. Головне меню.

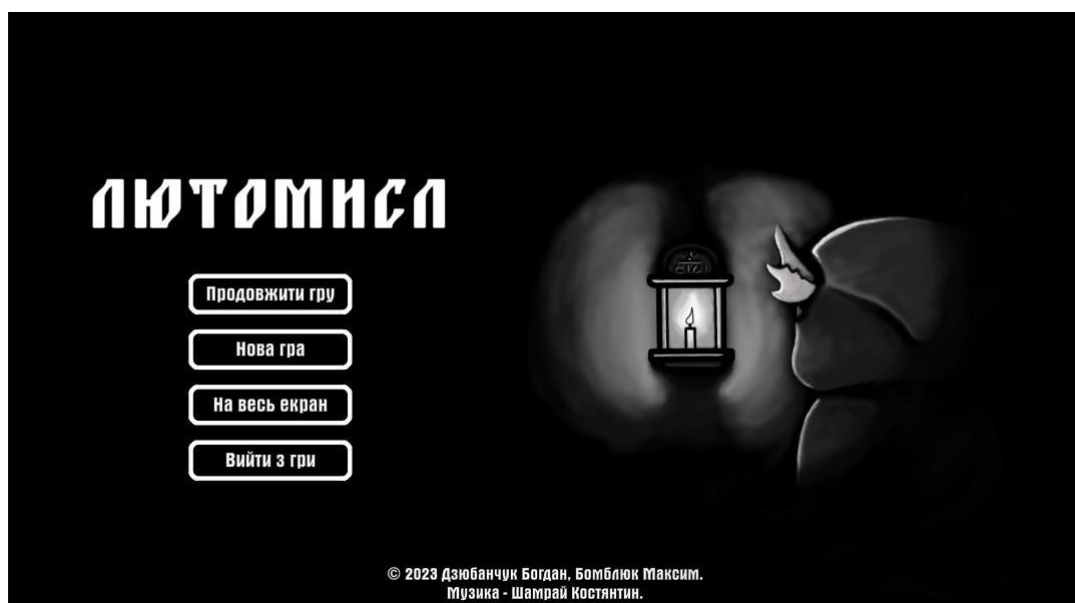


Рис. 2.11. Можливість продовжити гру або почати спочатку.

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

Висновки до розділу

Отже, в ході виконання другого розділу було створено усі основні системи для рівнів, ворогів, бойової системи, збереження та користувацького інтерфейсу. Бойова система вийшла схожою на бойову систему з серії Dark Souls, але не є абсолютною її копією. Стейт машина дуже зручна для створення штучного інтелекту ворогів, які не мають багато можливих дій. Паттерн сінглтон також виявився досить зручним в використанні, бо з ним не потрібно в кожен елемент, який активує збереження гри додавати посилання на скрипт, який зберігає гру.

		Бомблюк М. О.			Житомирська політехніка.20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

		Бомблюк М. О.			Житомирська політехніка.20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Тестування рівнів

Використовуючи графіку з попереднього пункту було побудовано досить велику карту, яка відповідає початковій ідеї (рис. 2.2), але має деякі зміни.



Рис. 3.5. Побудована карта.



Рис. 3.6. Локації з шахтою, другим поверхом.



Рис. 3.7. В'язниця і шлях до неї.

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				25
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 3.8. Гоблінська чудирнацька архітектура.

Під час тестування карти було усунено всі знайдені баги, які в більшій мірі були пов'язані з фізикою: гравець ковзає по нахилених поверхнях, гравець застрягає в платформах чи в підлозі, тощо. Ці проблеми було усунено зміною фізичних матеріалів та корегуванням стейт машини гравця.



Рис. 3.9. Один із багів - гравець застряг на краю платформи.

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

3.3 Тестування штучного інтелекту

Штучний інтелект ворогів працював добре, доки ворогів не було переміщено на ігрову карту. Першою помилкою було те, що поточне здоров'я ворогів зберігалось в спільних scriptable object. Таким чином, якщо гравець вбиває одного гобліна, вмирають усі гобліни цього ж типу. Поточне здоров'я та інші індивідуальні змінні було перенесено з scriptable object на самі об'єкти гоблінів і тепер задаються через інспектор.

Ще один баг пов'язаний з штучним інтелектом викликав помилку Stack Overflow, що означає одна й та сама частина коду виконується нескінченну кількість разів. Проблема була в гобліні шамані, а саме в його стані пострілу. Шаман переходив з стану переслідування в стан пострілу, якщо гравець знаходиться в радіусі атаки. В свою чергу стан пострілу перевіряє, якщо гравець не в радіусі атаки або не в полі зору, змінити стан на переслідування. Таким чином коли гравець знаходився в радіусі атаки, але не в полі зору шаман нескінченно переходив з пострілу в переслідування і з переслідування в постріл.

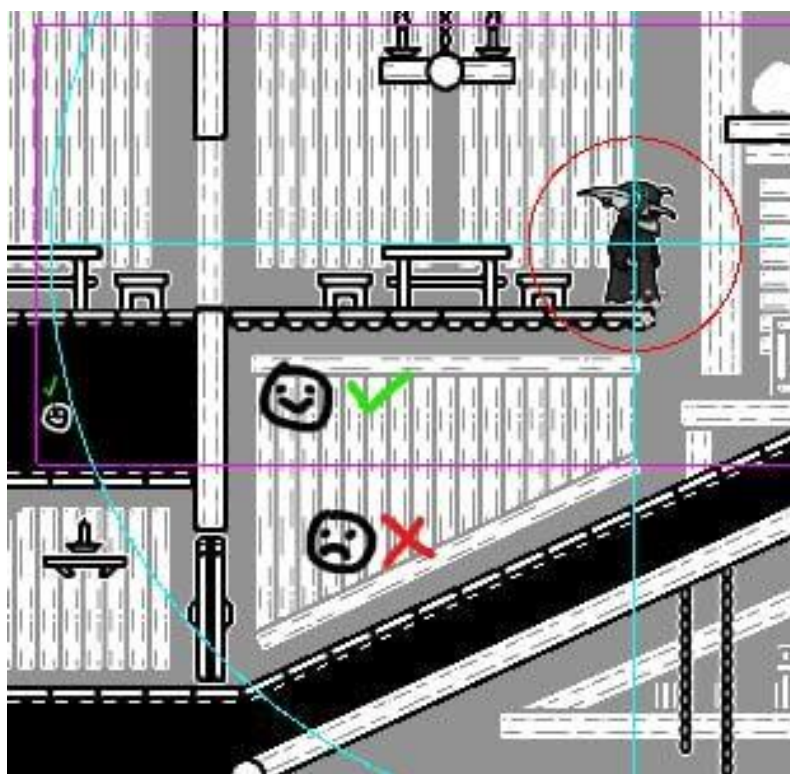


Рис. 3.10. Графічне зображення багу. Рожевий прямокутник - поле зору, блакитне коло - радіус атаки.

		Бомблюк М. О.			Житомирська політехніка. 20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Це було складно виявити, але легко виправити, додавши перевірку, чи гравець знаходиться в полі зору для переходу з переслідування в постріл .

Висновки до розділу

Таким чином, усі виявлені баги та проблеми було усунено, усі системи розроблено та відполіровано і гра готова до випуску. Також в ході тестування гри було додано додаткові повідомлення, які навчають гравця певним механікам та підкореговано початкові характеристики персонажу для більш приємного проходження гри.

		Бомблюк М. О.			Житомирська політехніка.20.122.3.000 - ПЗ	Арк.
		Терещук С. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		28