

Міністерство освіти і науки України
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавра

з напрямку підготовки 121 «Інженерія програмного забезпечення»

На тему: Розробка програмного забезпечення навчальної комп'ютерної гри з кібербезпеки

*Засвідчую, що в цій
кваліфікаційній роботі немає
запозичень із праць інших
авторів без відповідних посилань.
Студентка гр. ПЗ-20-2
_____ Пилипець С. М.*

Керівник кваліфікаційної
роботи

/ Стрюк А. М. /

Завідувач кафедри

/ Стрюк А. М. /

Кривий Ріг
2024

Криворізький національний університет
Факультет: Інформаційних технологій
Кафедра: Моделювання та програмного забезпечення
Освітньо-кваліфікаційний рівень: бакалавр
Спеціальність: 121 "Інженерія програмного забезпечення"

ЗАТВЕРДЖУЮ
Зав. кафедри
Стрюк А. М.
«__» _____ 2024__ р.

ЗАВДАННЯ

на кваліфікаційну роботу

студентці групи ІПЗ-20-2 Пилипець Софії Миколаївні

1. Тема: Розробка програмного забезпечення навчальної комп'ютерної гри з кібербезпеки затверджено наказом по КНУ №__ від «__» лютого 2024 р.
2. Термін подання студентом закінченого проекту «10» червня 2024 р.
3. Вихідні дані по роботі: Пояснювальна записка: 83 сторінки, 18 рисунків, 1 додаток, 14 використаних у роботі джерел
4. Зміст пояснювальної записки (перелік питань, що їх треба розробити): .
5. Перелік графічного демонстраційного матеріалу: Приклади існуючих програм. Функціональна схема. Блок-схема основного алгоритму. Структура бази даних. Приклади роботи розробленої програми.

Календарний план:

№	Найменування етапів кваліфікаційної роботи	Термін виконання етапів роботи
1	<i>Формулювання мети та задач роботи</i>	13.02.2024-20.02.2024
2	<i>Аналіз інформаційних джерел</i>	21.02.2024-09.03.2024
3	<i>Визначення вимог до програмного забезпечення</i>	10.03.2024-18.03.2024
4	<i>Розробка функціональної схеми</i>	19.03.2024-23.03.2024
5	<i>Розробка алгоритмів</i>	24.03.2024-31.03.2024
6	<i>Розробка бази даних</i>	01.04.2024-14.04.2024
7	<i>Розробка програмного забезпечення</i>	15.04.2024-13.05.2024
8	<i>Тестування програмного забезпечення</i>	14.05.2024-20.05.2024
9	<i>Оформлення пояснювальної записки</i>	21.05.2024-12.06.2024
10	<i>Розробка демонстраційних матеріалів</i>	13.06.2024-17.06.2024

Дата видачі завдання: «__» _____ 2024 р.
Студент _____ Пилипець С. М.
Керівник роботи _____ Стрюк А. М.

РЕФЕРАТ

КІБЕРБЕЗПЕКА, НАВЧАЛЬНА ГРА, PYTHON, REN'PY, VS CODE.

Пояснювальна записка: 83 с., 18 рис., 14 джерел.

Метою кваліфікаційної роботи є проектування та реалізація навчальної комп'ютерної гри з основ кібербезпеки.

В роботі досліджено значення кібербезпеки в контексті сучасних технологічних викликів. Визначено ключові вимоги, які повинна відповідати навчальна гра для вивчення основ кібербезпеки. Спроектовано структуру гри, включно з її інтерфейсом та сюжетом. Структура гри розрахована таким чином, щоб навчальний матеріал подавався логічно та послідовно, забезпечуючи поступове зростання складності завдань.

Для розробки гри ми вибрали рушій Ren'Py, який допоміг з рендерингом графіки, обробкою тексту та управлінням потоком ігрового процесу. Програмування гри було виконано на мові Python, що дозволило нам гнучко імплементувати весь необхідний функціонал та логіку.

В результаті ми отримали повнофункціональний навчальний продукт, що може бути використаний у якості засобу для підвищення обізнаності про кібербезпеку та відпрацювання практичних навичок з цієї дисципліни.

ABSTRACT

CYBERSECURITY, EDUCATIONAL GAME, PYTHON, REN'PY, VS CODE.

Explanatory note: 83 pages, 18 figures, 14 sources.

The purpose of this thesis is the design and implementation of an educational computer game focused on the fundamentals of cybersecurity.

The work explores the significance of cybersecurity in the context of modern technological challenges. It identifies the key requirements that an educational game for learning the basics of cybersecurity must meet. The game's structure, along with its interface and plot, has been designed. The structure is devised in such a way that educational content is presented logically and sequentially, ensuring a gradual increase in the complexity of tasks.

For the game development, we chose the Ren'Py engine, which assisted with graphic rendering, text processing, and game process flow management. The programming of the game was carried out in Python, which allowed us to flexibly implement all necessary functionalities and logic.

As a result, we developed a fully functional educational product that can be used as a tool to raise awareness about cybersecurity and to practice practical skills in this discipline.

ЗМІСТ

ВСТУП	7
1 ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ КОМП'ЮТЕРНОЇ ГРИ З КІБЕРБЕЗПЕКИ	11
1.1 Проблеми кібербезпеки в сучасному суспільстві.....	11
1.2 Засоби гейміфікації процесу навчання основ кібербезпеки	18
1.3 Визначення вимог до програмного забезпечення навчальної комп'ютерної гри з кібербезпеки	19
2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ КОМП'ЮТЕРНОЇ ГРИ З КІБЕРБЕЗПЕКИ	25
2.1 Функціональна схема навчальної комп'ютерної гри з кібербезпеки	25
2.2 Проєктування інтерфейсу навчальної комп'ютерної гри з кібербезпеки.....	27
2.3 Проєктування сценарію навчальної комп'ютерної гри з кібербезпеки.....	31
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ КОМП'ЮТЕРНОЇ ГРИ З КІБЕРБЕЗПЕКИ	33
3.1 Вибір засобів реалізації навчальної комп'ютерної гри з кібербезпеки.....	33
3.2 Програмна реалізація навчальної комп'ютерної гри з кібербезпеки	35
ВИСНОВКИ	43
Перелік посилань	44
Додаток А Текст програми.....	46

ВСТУП

Кіберзагрози є невід'ємною частиною сучасного цифрового світу, і вони можуть зустрічатися в різних аспектах повсякденного життя, як у професійному, так і в особистому. З огляду на велику поширеність інформаційних технологій, фахівці у різних галузях часто стикаються з кіберзагрозами, такими як спам, віруси, фішинг, фармінг, троянські програми, шпигунське ПЗ, рансомвар та несанкціонований доступ до комп'ютерних систем. Це вимагає не лише своєчасного виявлення та реагування на подібні інциденти, а й обізнаності та готовності діяти профілактично, щоб запобігти їх появі.

Особливо актуально це для освітніх установ, де комп'ютерні технології використовуються для вивчення широкого спектру предметів, починаючи від математики і природничих наук до гуманітарних дисциплін і мистецтва. Інтеграція ІТ у навчальний процес надає учням цінні практичні навички, проте водночас збільшує вразливість систем до кібератак.

У зв'язку з цим, важливо вдосконалювати сучасну професійну підготовку не тільки в галузі ІТ, але й у сфері кібербезпеки. Освітні програми повинні включати курси з кібергігієни, розробки безпечних мереж, адміністрування систем та криптографії. Також вони мають акцентувати увагу на сучасних методах захисту даних, на розумінні правових аспектів кібербезпеки, та на розвитку навичок критичного мислення та ризик-орієнтованого підходу до оцінки потенційних загроз і вироблення стратегії їх уникнення.

Таке глибоке і всебічне розуміння кібербезпеки є критично важливим для створення стійкого та безпечного освітнього простору, а також для підготовки фахівців, здатних ефективно протистояти кіберзагрозам в майбутньому.

Кібербезпека включає в себе ряд технічних та адміністративних процедур і засобів, а також політик, які допомагають захистити інформаційні

системи, мережі та дані, що обробляються або зберігаються в електронній формі від неправомірного втручання або атак. Принципи кібербезпеки охоплюють не лише технічні заходи, як-от використання антивірусного програмного забезпечення і файєрволів, але й організаційні заходи, такі як розробка і впровадження політик безпеки, проведення навчання для збільшення обізнаності персоналу з питань безпеки та регулярний аудит та моніторинг систем.

Ефективні стратегії в області кібербезпеки враховують різні аспекти забезпечення захисту, включаючи фізичну безпеку обладнання, безпеку програмного забезпечення та захист мережевої інфраструктури. Забезпечення конфіденційності передбачає захист від несанкціонованого доступу до важливих даних. Захист цілісності зосереджений на запобіганні незаконним змінам інформації. Забезпечення доступності – це гарантія того, що користувачі можуть своєчасно дістатися до своїх даних та систем, коли вони цього потребують.

Окрім розгалужених технічних заходів, сфера кібербезпеки включає в себе також юридичні аспекти, такі як дотримання вимог конфіденційності даних, правил щодо зберігання інформації та відповідальності за порушення в цій області. Розділ соціальних засобів охоплює роботу з персоналом, обізнаність людей про шахрайство та інші зловмисні дії в інтернеті та розвиток корпоративної культури, в якій безпека інформації є високим пріоритетом.

Враховуючи швидкий розвиток технологій та зростання числа кібератак, кібербезпека стає все більш актуальною і важливою дисципліною, вимагаючи постійного адаптування та оновлення заходів захисту для протистояння новим загрозам.

У сучасному світовому контексті важливість кіберпростору та кібербезпеки стрімко зростає. В Україні це визначено у відповідному законі "Про основні засади здійснення кібербезпеки України". Згідно з цим законом, поняття кіберпростору охоплює віртуальне середовище, яке існує завдяки функціонуванню з'єднаних між собою комунікаційних систем. Це середовище

дозволяє людям вести комунікації та реалізовувати суспільні відносини через мережу Інтернет та інші глобальні мережі передачі даних.

В той же час, кібербезпека визначається як захист не тільки технологічного середовища, але й безпечності людей, їх особистих даних, а також забезпечення стабільного розвитку інформаційного суспільства. Ефективна кібербезпека включає своєчасне виявлення, запобігання та нейтралізацію загроз, які можуть виникнути в кіберпросторі і вплинути на національну безпеку.

Актуальність культури безпечної поведінки у кіберпросторі не може бути недооцінена в сучасних умовах. Це не лише стосується корпоративного сектору, де системи кіберзахисту є обов'язковими елементами інфраструктури, але й особистісного аспекту, де кожна людина повинна бути усвідомленою щодо можливих ризиків та вміти захистити себе в онлайн середовищі. Навчання користувачів основам кібергігієни, вмінню розпізнавати фішингові атаки, використовувати надійні паролі та забезпечувати захист персональних даних є ключовими завданнями в боротьбі з кіберзагрозами. Тому значну увагу слід приділяти розвитку культури кібербезпеки серед усіх верств населення, щоб підвищити рівень захищеності індивідуальних та корпоративних користувачів в цифровому світі.

Впровадження гейміфікації у процес навчання основ кібербезпеки значно перетворює традиційний підхід до освіти, роблячи її більш захоплюючою та ефективною. Гейміфікація передбачає використання ігрових принципів та механізмів у неігровому контексті, що допомагає підвищити мотивацію та залученість учнів.

Ігрові елементи, такі як бали, рівні, виклики, віртуальні нагороди та лідерські таблиці, можуть бути інтегровані в навчальний план, даючи студентам відчуття досягнення та прогресу, коли вони освоюють нові поняття та навички. Такий підхід також стимулює здорову конкуренцію серед учасників.

Крім того, застосування сценаріїв та симуляцій у гейміфікованих додатках дозволяє студентам експериментувати та вчитися від помилок у безпечному віртуальному середовищі. Вони можуть моделювати різні типи кібератак, такі як фішинг, програми-вимагачі, або атаки на веб-служби, і вчитися, як ідентифікувати та запобігати потенційним загрозам.

Важливо підкреслити, що гейміфікація також допомагає наближати складні та абстрактні концепти кібербезпеки до реальних життєвих ситуацій, роблячи навчання більш практичним і зрозумілим. Це сприяє кращому засвоєнню інформації та підготовці студентів до стикання з реальними кіберзагрозами в майбутньому.

Загалом, впровадження гейміфікації в освіту з кібербезпеки може значно поліпшити відношення студентів до навчання, роблячи процес засвоєння базових знань про кіберзагрози та методи боротьби з ними більш захоплюючим та продуктивним.

Це робить актуальною обрану нами тему кваліфікаційної роботи: «Розробка програмного забезпечення навчальної комп'ютерної гри з кібербезпеки».

1 ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ КОМП'ЮТЕРНОЇ ГРИ З КІБЕРБЕЗПЕКИ

1.1 Проблеми кібербезпеки в сучасному суспільстві

Кібербезпека є критично важливим аспектом для усіх користувачів інтернету у сучасному цифровому світі, де кіберзагрози та кібератаки відбуваються з алярмуючою частотою. Це питання стосується широкого спектру проблем - від несанкціонованого доступу до особистих соціальних мережеских акаунтів, через які можуть розсилатися шкідливі повідомлення, до фінансових шахрайств, при яких з банківських карток людей незаконно знімаються кошти.

Особисті акаунти в соціальних мережах є однією з поширених мішеней для хакерів, які можуть використовувати їх для розсилки спам-повідомлень або фішингових атак, спрямованих на отримання конфіденційної інформації від контактів жертви. Не менш тривожним є випадки, коли шахраї здійснюють несанкціоновані транзакції за допомогою вкрадених даних банківських карток, залишаючи власників карток без їх власних грошей.

Також зазіхання на електронні поштові скрині стають все більш розповсюдженим явищем. Зловмисники можуть використовувати різноманітні методи, такі як фішинг, малвер або використання вразливостей у програмному забезпеченні, щоб шахраювати або вимагати викуп за відновлення доступу до закодованих даних.

Злочинці не обирають своїх жертв за розміром або значимістю; вони ціляться як на окремих осіб, так і на малі підприємства, а також на великі організації і установи. Методи кіберзлочинців стають все більш витонченими та складними, від соціальної інженерії до високотехнологічних атак, що вимагають від усіх нас бути пильними та інвестувати в засоби і процеси для захисту наших цифрових активів. Це також підкреслює необхідність

регулярного оновлення програмного забезпечення, використання складних паролей та двофакторної аутентифікації, інформування про кращі практики в галузі кібербезпеки та наявність готовності швидко реагувати на будь-які кіберінциденти.

Отже, у сучасному цифровому світі, організації незалежно від того, чи належать вони до ІТ сектору, чи є представниками інших галузей, все більше усвідомлюють невідкладну потребу в кібербезпеці. Вони приділяють значну увагу розробці та впровадженню заходів, направлених на протидію кіберзагрозам. У зв'язку з тим, що сучасні технології дозволяють нам підключати величезну кількість пристроїв до мережі Інтернет, кількість потенційних точок втручання та атак зростає, а разом з тим і ризик виявлення нових вразливостей, порушень безпеки та недоліків.

Не можна більше покладатися винятково на паролі як єдиний метод захисту - ця стратегія є застарілою та не відповідає сучасним реаліям кіберпростору. Складний і багат шаровий підхід до кібербезпеки тепер є нормою. Розуміння того, як захистити конфіденційність, цілісність та доступність інформації, стає ключовим для кожної особи й організації.

Впровадження сучасних технологій безпеки та постійна освіта персоналу, управління ідентичностями та доступом, шифрування даних, використання протоколів безпечного з'єднання, регулярні оновлення та патчі – це лише кілька прикладів компонентів комплексної стратегії кібербезпеки, яка вирішальна для захисту як особистих, так і корпоративних даних.

Кібербезпека, або кіберзахист, таким чином, перетворюється на ключову область знання, яка необхідна кожному, хто бажає ефективно захистити інформацію від зростаючих та постійно еволюціонуючих онлайн-загроз.

Підключення до електронної інформаційної мережі — це ключовий аспект сучасного суспільства, яке стало невід'ємною частиною нашого повсякденного життя. Компанії, установи та освітні заклади використовують Інтернет та інші цифрові засоби зв'язку для забезпечення ефективного функціонування. Ця мережа служить платформою для зібрання,

систематизації, зберігання та розподілу обсягів цифрової інформації, які постійно зростають у нашій цифровій епосі.

Оскільки обсяг цифрових даних збільшується експоненційно, задіяні технології стають дедалі складнішими, і відповідно, ризики та виклики, пов'язані з безпекою цих даних, також зростають. Важливість передачі та спільного використання інформації в електронній формі вимагає високого рівня захисту для запобігання несанкціонованому доступу, втраті даних, кібератакам та іншим загрозам. Відповідно, зміцнення кібербезпеки є критичним елементом, який сприяє захисту національної безпеки, забезпечення приватності громадян та підтримки економічної стабільності в умовах постійно змінюваного цифрового ландшафту. Стратегічні кроки урядів та приватного сектору для захисту інформаційної інфраструктури мають велике значення для забезпечення стійкості цифрового суспільства.

Кібербезпека включає захист інформації та систем, які обробляють, зберігають та передають цю інформацію від численних кіберзагроз, включаючи шкідливе програмне забезпечення, атаки на деніаль сервісу, фішинг, шпигунські програми, рансомварі та витік даних. Це також має справу із захистом фізичних систем, таких як комп'ютерні мережі, сервери, комп'ютери, мобільні пристрої та дані, які індивіди і організації використовують у своїх застосуваннях.

Сукупність технічних заходів охоплює шифрування, безперервне оновлення програмного забезпечення, розробку надійних алгоритмів аутентифікації та авторизації, використання брандмауерів, внутрішніх та зовнішніх мережевих сканерів для виявлення уразливостей, а також систем виявлення і запобігання вторгнень.

Соціальна складова кібербезпеки зосереджується на підвищенні обізнаності людей про кіберзагрози, навчанні правильної поведінки в інтернеті та розробці комплексних політик, які допомагають убезпечити індивідів та організації від кібератак. Значною частиною також є підготовка професіоналів з кібербезпеки, які здатні розпізнати і ефективно реагувати на

інциденти кібербезпеки, а також розробка законодавства, що регулює цю сферу і створює правові рамки для боротьби з кіберзлочинністю.

Загалом, кібербезпека відіграє ключову роль у захисті сучасного суспільства від широкого спектру загроз, що можуть мати серйозні наслідки, починаючи від економічних втрат і завершуючи загрозами національній безпеці.

Кібербезпека – це комплексна дисципліна, яка займається захистом систем, мереж та даних в кіберпросторі. Основу підходу до кібербезпеки складають три ключові принципи, відомі як триада "CIA": конфіденційність, цілісність і доступність.

Конфіденційність (Confidentiality) полягає у захисті інформації від несанкціонованого доступу і розголошення, таким чином забезпечуючи, що інформація доступна лише тим особам, які мають до неї право.

Цілісність (Integrity) вимагає забезпечення правильності та повноти інформації та оброблювальних методів. Це означає, що дані мають бути захищені від несанкціонованих змін, а будь-які модифікації мають проводитися тільки уповноваженими особами.

Доступність (Availability) відноситься до забезпечення стабільного та своєчасного доступу до інформації та ресурсів, які необхідні уповноваженим користувачам. В цьому контексті важливо підтримувати функціонування інформаційних систем, ефективно реагувати на інциденти та відновлення роботи після збоїв.

Ця модель служить фундаментом для розробки та впровадження політик кібербезпеки в організаціях, становлячи основу їх стратегії захисту інформаційних активів від широкого спектру кіберзагроз. Вона є частиною більш широкої дисципліни інформаційної безпеки, яка також включає фізичну безпеку, управління ризиками та інші аспекти захисту інформаційних ресурсів.

Конфіденційність, цілісність і доступність – це запорука ефективного захисту даних та безпеки інфраструктури організації. Ці три поняття – основоположні принципи для впровадження плану InfoSec.

При аналізі безпеки інформаційних систем, важливо виокремити дві основні групи проблем: безпека комп'ютерів і мережева безпека.

Безпека комп'ютерів охоплює заходи, що забезпечують захист окремих комп'ютерів від несанкціонованого доступу, збоїв в системі, шкідливого програмного забезпечення та інших типів кібератак. Це включає встановлення надійних антивірусних програм, налаштування файрволів, використання складних паролів і багатофакторної аутентифікації, регулярні оновлення системного і програмного забезпечення для запобігання вразливостей, а також фізичні засоби захисту, як-от кабінети з обмеженим доступом та засоби для знищення даних.

Мережева безпека включає заходи для захисту даних, які передаються через мережі зв'язку. Основні завдання мережевої безпеки полягають у запобіганні несанкціонованому доступу, перехопленню даних та атакам типу "відмова в обслуговуванні" (DoS). Для захисту інформації, що циркулює в мережах, використовуються шифрування, тунелювання, рішення VPN (Virtual Private Network), розумне управління системним доступом, а також розподіл прав і привілеїв користувачів. Також важливим є моніторинг і аудит мережі з метою своєчасного виявлення і реагування на будь-яку підозрілу активність.

Обидві групи проблем взаємопов'язані, оскільки злом мережевих ресурсів може призвести до уразливостей на окремих комп'ютерах, і навпаки, компрометація окремого комп'ютера може створити загрозу для всієї мережі. Тому комплексний підхід до інформаційної безпеки, який включає і безпеку комп'ютерів, і мережеву безпеку, є ключовим для створення ефективної системи захисту інформації.

Безпечна інформаційна система – це комплекс технічних і програмних засобів, а також процедур і політик, призначених для захисту цифрових даних.

Вона виконує декілька ключових функцій для забезпечення оптимального рівня безпеки інформації.

Першою і головною задачею є захист даних від несанкціонованого доступу. Це досягається через впровадження системи аутентифікації користувачів, використання криптографічних механізмів для шифрування даних, а також встановлення фаєрволів і антивірусного програмного забезпечення для запобігання несанкціонованих втручань.

Друге значуще завдання такої системи – забезпечення доступності даних для авторизованих користувачів. Безпечна система має бути надійною та працездатною, щоб користувачі мали можливість доступу до інформації та інформаційних ресурсів в будь-який необхідний момент. Це передбачає використання надійного обладнання, резервування даних та реалізацію стратегії відновлення після збоїв системи.

Третє завдання – це збереження цілісності інформації, тобто забезпечення захисту від несанкціонованої або випадкової модифікації даних. Для цього використовуються контрольні суми, цифрові підписи та інші механізми, які дозволяють перевіряти автентичність та незмінність інформації.

Безпечна інформаційна система – це комплексна структура, яка об'єднує в собі низку основоположних властивостей, які гарантують захист інформації від неавторизованого доступу, змін, знищення та інших загроз. Серед найважливіших властивостей безпечної інформаційної системи можна виокремити конфіденційність, цілісність, доступність, автентичність, підзвітність та невідторгнення.

Важливо розуміти, що ці властивості є взаємопов'язаними та взаємозалежними. Наприклад, забезпечення конфіденційності без забезпечення цілісності може призвести до того, що хоча дані й будуть утримуватися у секреті, вони можуть бути змінені, тим самим підриваючи довіру до інформації. Також, якщо система є доступною, але не забезпечує достатній рівень автентичності, маємо ризик того, що неавторизовані особи

можуть отримати доступ до чутливих даних. Слабкість у будь-якому із компонентів може мати серйозні наслідки і поставити під загрозу безпеку інформаційної системи в цілому.

В сучасному світі, де технології стрімко розвиваються, а більшість бізнес-процесів та обміну інформацією переходять в цифровий формат, підприємства та організації стають все більш вразливими до кібератак. Це може призводити до втрати конфіденційності даних, порушення їх цілісності та доступності, що може мати серйозні фінансові й репутаційні наслідки для бізнесу. У зв'язку з цим, навчання персоналу з основ кібербезпеки є не лише рекомендацією, але й необхідністю.

Освітні програми з кібербезпеки часто включають тренінги з виявлення та запобігання шахрайським діям, таким як фішинг, розпізнавання малварі (шкідливих програм), а також навички створення сильних паролів і правильного управління ключами доступу. Також важливим аспектом є ознайомлення з політикою інформаційної безпеки власної об'єкта та відповідальністю за дотримання правил захисту інформації на робочому місці.

Додатково, регулярні тренінги підтримують високий рівень обізнаності серед співробітників про постійно змінювані загрози та навички реагування на інциденти безпеки. Співробітники, які знають, як діяти в разі кіберінциденту, можуть швидко вжити заходів для мінімізації шкоди, що позитивно впливає на відновлення систем після атак. Підвищення рівня кіберграмотності персоналу стає стратегічною інвестицією в загальну безпеку організації та її стійкість до кіберзагроз.

Для тренінгів з основ кібербезпеки комп'ютерні навчальні програми слугують важливим інструментом, дозволяючи користувачам в інтерактивній формі засвоювати матеріал. Серед таких програм особливе місце займають навчальні комп'ютерні ігри, які допомагають освоїти складні поняття в галузі кібербезпеки за допомогою гейміфікації.

Ці ігри можуть симулювати різноманітні кібератаки та захисні стратегії, вчачи користувачів розпізнавати фішингові спроби, захищати від вірусів та

шкідливого ПЗ, управляти паролями і застосовувати шифрування. Вони можуть також включати сценарії моделювання для розробки комплексних відповідей на загрози кібербезпеки, навчання навикам реагування на інциденти та відновлення після атак.

Такий підхід до навчання не тільки зробить процес більш захоплюючим і зрозумілим, але й забезпечить практичні навички, які можуть бути використані в реальних ситуаціях. Навчальні комп'ютерні ігри можуть бути ефективно інтегровані в корпоративні навчальні програми або навчальні курси для освітніх установ, підвищуючи рівень інформаційної безпеки і згуртованості колективу в питаннях протидії кіберзагрозам.

1.2 Засоби гейміфікації процесу навчання основ кібербезпеки

Гейміфікація в системі вищої освіти стає все більш популярною тенденцією, оскільки до процесу навчання впроваджуються елементи комп'ютерних ігор. Це особливо актуально при роботі зі студентами-дорослими, які часто потребують додаткової мотивації та залучення до освітнього матеріалу. Застосуванню ігрових технологій в освіті приділяється значна увага з боку дослідницької спільноти, оскільки вони дозволяють перетворити традиційні форми навчання на більш інтерактивний і захоплюючий процес.

З використанням ігрових елементів, таких як рівні складності, сценарії місій, заробіток балів та здобуття значків або нагород, студенти стають активними учасниками свого навчання. Це дозволяє кожному розвиватися власним темпом і водночас стимулює конкуренцію та прагнення до самовдосконалення.

Гейміфікація в освіті – це не просто використання ігор для навчання, а систематичний підхід, який використовує ігрові принципи, щоб залучити студентів, підвищити їх мотивацію до навчання, розвивати критичне мислення

та співпрацю. Вона обіцяє зробити навчальний процес не тільки більш ефективним, але й приємним, підвищуючи в цілому результативність освітнього процесу і готовність студентів до вирішення реальних проблем.

1.3 Визначення вимог до програмного забезпечення навчальної комп'ютерної гри з кібербезпеки

Функціональні вимоги до програмного забезпечення навчальної комп'ютерної гри з кібербезпеки зазвичай визначають основні функції та процеси, які має виконувати програма. До них можна віднести:

1. Інтерактивне навчання: гра повинна забезпечувати користувачам можливість для розвитку навичок в кібербезпеці через інтерактивні завдання та сценарії. Гра повинна пропонувати гравцям захоплюючий та інтерактивний сценарій, який плавно інтегрує основи кібербезпеки в свою розповідь. За допомогою розгалужених діалогів, гра навчить гравців важливим поняттям та практикам, таким як створення надійних паролів, виявлення і запобігання фішинговим атакам, шифрування даних, безпечне використання публічних Wi-Fi мереж, а також поведінка у випадку виявлення вразливостей або шкідливого програмного забезпечення.

Гра може розгортатися у вигляді кіберквесту, де кожен рівень або глава навчає новому аспекту безпеки в цифровому просторі. Гравець виконує роль спеціаліста з кібербезпеки або звичайного користувача інтернету, який стикається з реальними викликами та загрозами. Через серію виборів дій, що диктуються діалогами з іншими персонажами або системними повідомленнями, гравець розвиває навички, які потім можна застосовувати в реальному житті.

Навчальний аспект гри повинен бути збалансований зі збереженням ігрового досвіду захопливим. Це можна досягти, наприклад, через інтеграцію історії у кібер-детективний сюжет, де розкриттями злочинів або вирішенням

технологічних загадок супроводжуються уроки з кібербезпеки. Така гра може використовувати елементи реального часу, а також різноманітні сценарії та кінцівки, які залежать від прийнятих рішень і навичок, здобутих протягом гри, надаючи їй велику перегравальну цінність.

2. Симуляція реальних умов вказує на процес, де гра, зазвичай у вигляді комп'ютерного застосунку чи віртуального тренажеру, детально наслідує поведінку й характер кібератак та методики кібероборони, які можуть виникати у реальному світі. Такий тип навчання розроблено для того, аби користувачі або інформаційні аналітики могли практично застосувати теоретичні знання у безпечному контрольованому середовищі, отримавши цінний досвід у виявленні, реагуванні та запобіганні реальних кіберінцидентів.

В таких імітаційних іграх користувачі можуть зіткнутися з різноманітними випробуваннями, як-от навали вірусів, спроби фішингу, розгорнуті DDoS атаки, проблеми з апаратним та програмним забезпеченням, критичні відмови систем, а також з необхідністю розробки та впровадження комплексних планів кібербезпеки. Важливим аспектом є також розвиток навичок швидкого реагування на інциденти, розробка політик безпеки, аудит систем і навичок комунікації, оскільки ефективна кіберзахист часто потребує командної роботи та міжвідділкової взаємодії.

Така практика підвищує обізнаність користувача про поточні кіберзагрози, дозволяючи розвинути стратегічне мислення та прийняти ефективні рішення для захисту цифрових активів організації чи індивідуального користувача від потенційно шкідливих кіберінцидентів.

3. Для ефективного навчання та підтримки мотивації користувачів, система навчання повинна містити вбудований інструментарій для відстеження прогресу.

Технічні вимоги є важливою частиною розробки та розповсюдження відеоігор. Вони окреслюють специфікації апаратного та програмного забезпечення, які необхідні для забезпечення коректної роботи гри. Ці вимоги поділяються на мінімальні та рекомендовані.

Мінімальні технічні вимоги повідомляють користувачам, яке найменше потужне залізо та версії програмного забезпечення здатні запустити гру хоча б на мінімальних налаштуваннях графіки. Зазвичай до таких вимог входять: тип та частота процесора, обсяг оперативної пам'яті, модель та обсяг відеокарти, необхідний простір на жорсткому диску, а також версії операційної системи та необхідних драйверів.

Рекомендовані технічні вимоги повідомляють користувачам про апаратне та програмне забезпечення, яке рекомендується для найкращого досвіду гри, яке дозволяє використовувати графічні налаштування вищого рівня та отримати більшу продуктивність та якість зображення. Це може включати більш потужний процесор, більший обсяг оперативної пам'яті, високопродуктивну відеокарту та більше місця на диску для інсталяції та запуску гри.

Відповідність технічним вимогам важлива не тільки для запуску гри, але й для забезпечення безперебійної та комфортної гри. Недостатнє апаратне забезпечення може призвести до проблем з продуктивністю, таких як затримки, зависання та довгі часи завантаження. Кожен розробник відеоігор встановлює власні технічні вимоги виходячи з можливостей та особливостей самої гри.

Для забезпечення широкої доступності, гра повинна бути сумісною з різноманітними операційними системами, що охоплює не тільки популярні версії, такі як Windows, macOS і Linux для персональних комп'ютерів, але й мобільні платформи, включаючи Android та iOS для смартфонів і планшетів. Це підвищує гнучкість доступу до гри і розширює потенційну аудиторію, оскільки користувачі зможуть насолоджуватись геймплеєм незалежно від того, який пристрій вони використовують.

Додатково, гра має бути адаптована до екранів різних розмірів і роздільної здатності, від великих моніторів персональних комп'ютерів до невеликих дисплеїв мобільних телефонів, забезпечуючи однаково комфортну взаємодію і чітке відображення графічних елементів. Сенсорне управління

повинно бути інтуїтивним та легко адаптується під тачскрін, тоді як для версій на персональні комп'ютери необхідний гладкий контроль за допомогою клавіатури та миші.

Оптимізація роботи гри на різноманітному обладнанні також важлива, щоб забезпечити плавний геймплей на пристроях з менш потужною обчислювальною здатністю, а також високоякісну графіку на здатних до цього системах. Врахування цих аспектів допоможе розробити високоякісний продукт, сумісний з широким асортиментом технічних пристроїв та операційних систем, максимально задовольняючи потреби різноманітної аудиторії.

Мінімальні системні вимоги для запуску гри традиційно включають наступні компоненти і параметри апаратного забезпечення:

1. Процесор (CPU): Центральний процесор є мозком комп'ютера, який виконує інструкції програмного забезпечення. Для ігор рекомендується вказувати як мінімум модель і тактову частоту. Наприклад, Intel Core i3 2.5 GHz або AMD Phenom II X4 945 3.0 GHz.

2. Оперативна пам'ять (RAM): Пам'ять, яку використовує комп'ютер для зберігання даних, які обробляються в даний момент. Чим більше об'єм оперативної пам'яті, тим краще. Мінімальні вимоги: 4 ГБ.

3. Відеокарта (GPU): Спеціалізований компонент, який обробляє графіку. Важливо вказати не тільки модель відеокарти, але й обсяг вбудованої відеопам'яті. Рекомендується, NVIDIA GTX 660 2 ГБ або AMD Radeon HD 7870 2 ГБ.

4. Місце на жорсткому диску: Вказується необхідний для інсталяції і збереження ігрових файлів обсяг вільного простору. Мінімальні вимоги: 2 ГБ вільного місця.

5. Операційна система (OS): Операційна система, необхідна для запуску гри. Зазначається мінімально підтримувана версія, наприклад, Windows 7 SP1 64-bit.

Це базові компоненти, які розглядаються при визначенні мінімальних вимог до апаратного забезпечення для ігор. Варто зауважити, що мінімальні вимоги забезпечують лише базову производительність, і, можливо, ігри не будуть працювати на високих налаштуваннях графіки чи забезпечувати оптимальну чіткість зображення.

Інтеграція системи: розроблена гра має бути сумісною для інтеграції з різними навчальними платформами та системами управління навчанням (Learning Management Systems, LMS). Це означає, що гра повинна мати відкриті API для взаємодії з навчальними платформами, щоб дані про прогрес і досягнення учнів могли бути легко обмінювані між системами. Також важливою є підтримка стандартів даних, таких як xAPI (Experience API) та SCORM (Sharable Content Object Reference Model), які дозволяють інтегрувати навчальний контент і слідкувати за взаємодією користувачів з цим контентом всередині LMS. Використання цих стандартів гарантує, що ігри можуть бути легко додані до бібліотеки навчальних ресурсів і що результати і статистика їх використання будуть доступними для оцінювання та детального аналізу ефективності навчання.

Інтерфейс користувача повинен бути розроблений таким чином, щоб забезпечити легку та ефективну взаємодію між людиною та комп'ютером. Інтерфейс має бути інтуїтивно зрозумілим, забезпечуючи логічну та зручну організацію елементів управління, які включають кнопки, меню, іконки та інші віджети. Окрім того, він має бути простим у використанні, незалежно від того, чи є користувач досвідченим в обігу з комп'ютерною технікою, чи ні.

Дотримання принципів консистентності у дизайні інтерфейсу забезпечує, що подібні завдання можуть бути виконані однаковими методами, тим самим знижуючи криву навчання для нових користувачів та збільшуючи загальну продуктивність. Також важливими є швидкість реакції інтерфейсу на дії користувача та наявність зворотного зв'язку, який дозволяє користувачу зрозуміти, що його дії були зареєстровані системою.

Додатково, інтерфейс повинен забезпечувати доступ до всебічної допомоги та підтримки, включаючи пояснення функціоналу та засоби навчання користувачів, як-от вбудовані підказки, відеоуроки чи підтримка на форумах. Належне продумування доступності інтерфейсу для людей з обмеженими можливостями, наприклад, через використання альтернативних методів введення та спеціальних адаптацій, також є важливим аспектом, який потрібно врахувати при дизайні інтерфейсу користувача.

На основі визначених вимог розпочнемо процес моделювання та розробки програмного забезпечення для освітньої комп'ютерної гри, яка зосереджена на кібербезпеці. Це включатиме визначення основних функціональних та нефункціональних вимог до гри, створення концепту геймплея, встановлення сценаріїв та завдань, які навчатимуть користувачів аспектів кібербезпеки, таких як безпечна поведінка в інтернеті, розуміння загроз та методи захисту інформаційних систем.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ КОМП'ЮТЕРНОЇ ГРИ З КІБЕРБЕЗПЕКИ

2.1 Функціональна схема навчальної комп'ютерної гри з кібербезпеки

Функціональна схема гри, яку ви описали, є детальним планом, котрий ілюструє різні аспекти геймплею та взаємодії з користувачем. Структура гри напроєктована таким чином, що вона дає можливість гравцям самостійно вирішувати, яким шляхом пройти через гру, роблячи її більш захоплюючою та непередбачуваною.

Основні компоненти гри включають:

1. Сюжетні розділи - секції гри, де розкривається історія через інтерактивні діалоги, завдання та вибори, які впливають на подальший розвиток подій.

2. Інтерактивні діалоги - моменти в грі, де користувачі можуть взаємодіяти з іншими персонажами або елементами сюжету, роблячи вибір, що впливатиме на історію.

3. Міні-ігри, що навчають кібербезпеці - спеціалізовані ігрові моменти, якими передбачено тестування та розвиток умінь користувача в області цифрової безпеки через гейміфікацію навчального процесу.

4. Навчальний режим - частина гри, що забезпечує гравцям теоретичні знання через лекції, інструктажі та тестові завдання для кращого засвоєння інформації.

5. Тести - інструменти для перевірки знань і розуміння навчального матеріалу, що був представлений в грі.

6. Налаштування - розділ, який дозволяє користувачам налаштувати гру відповідно до власних переваг, змінювати різні параметри інтерфейсу, графіки, звуку та управління для максимально комфортного геймплею.

Така структура забезпечує глибоке занурення в ігровий світ, надаючи одночасно і розваги, і освітню цінність, використовуючи інструменти та методики сучасного навчання. Загальна структура навчальної гри з кібербезпеки показан на рисунку 2.1.



Рисунок 2.1 – Загальна структура навчальної гри з кібербезпеки

2.2 Проектування інтерфейсу навчальної комп'ютерної гри з кібербезпеки

Для створення інтерфейсу основних режимів навчальної комп'ютерної гри з кібербезпеки потрібно спочатку визначити ключові режими, які будуть доступні користувачеві, а також зрозуміти цільову аудиторію гри та їх вимоги до інтерфейсу. Розглянемо схематичні макети дизайну основних режимів гри.

Макет дизайну стартової сторінки показано на рисунку 2.2. На цій сторінці користувачу пропонується загальна титульна заставка гри та головне меню, що має пункти:

- Почати гру
- Завантажити
- Налаштування
- Про гру
- Довідка



Рисунок 2.2 – Макет дизайну стартової сторінки

Макет дизайну сторінки діалогового режиму гри показано на рисунку 2.3. В дизайні цієї сторінки передбачено розміщення наступних елементів:

- Тло сцени
- Поле основного діалогу
- Відображення декількох сюжетних персонажів
- Відображення меню з вибором декілької варіантів відповідей в діалогах

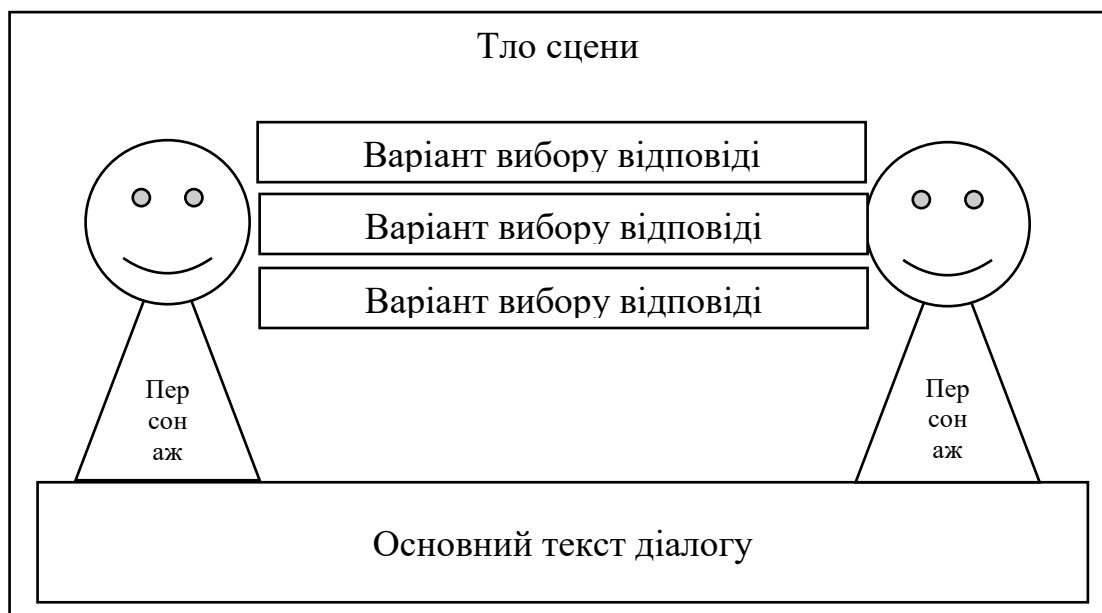


Рисунок 2.3 – Макет дизайну сторінки діалогового режиму гри

Крім відображення сюжетних діалогів та опису сцен, передбачено декілька міні-ігор з кібербезпеки. Макет дизайну сторінки з міні-іграми показано на рисунку 2.4.

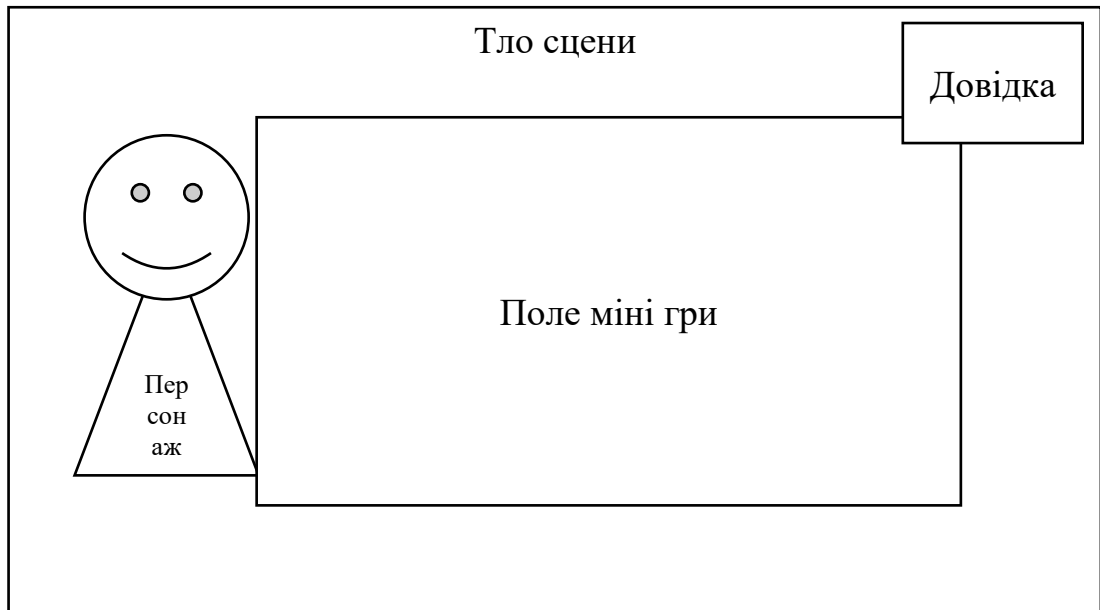


Рисунок 2.4 – Макет дизайну сторінки з міні-іграми

А також перевірка отриманих гравцем знань відбувається у вигляді тестів або вікторини. Макет дизайну сторінки перевірки знань в режимі тестування/вікторини показана на рисунку 2.5.

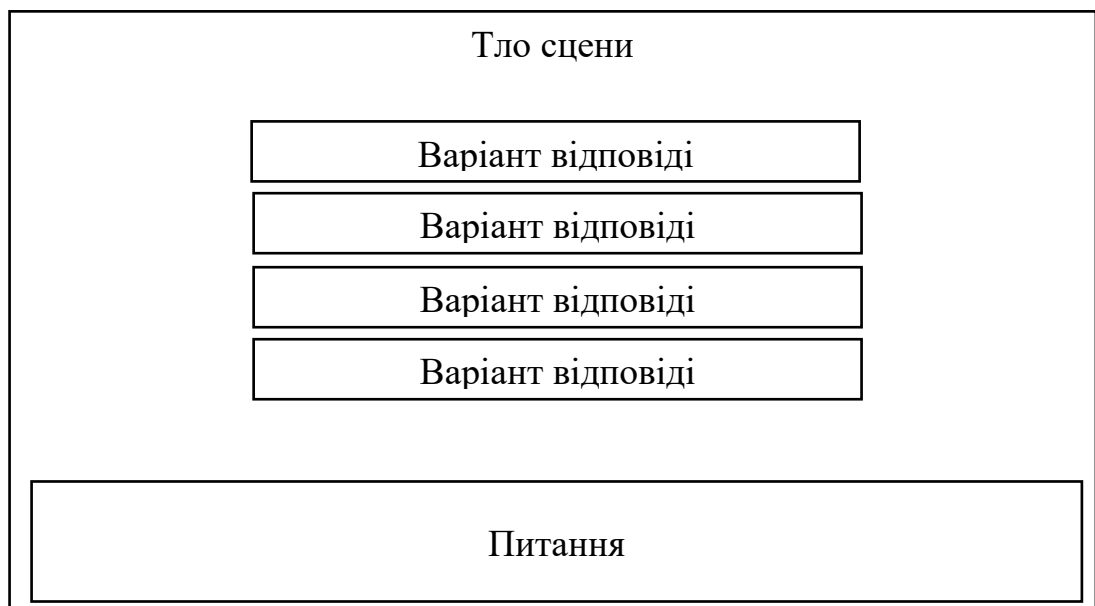


Рисунок 2.5 – Макет дизайну сторінки перевірки знань в режимі тестування/вікторини

Також гравець може зберігати час від часу прогрес гри. Макет дизайну сторінки управління збереженнями показано на рисунку 2.6.

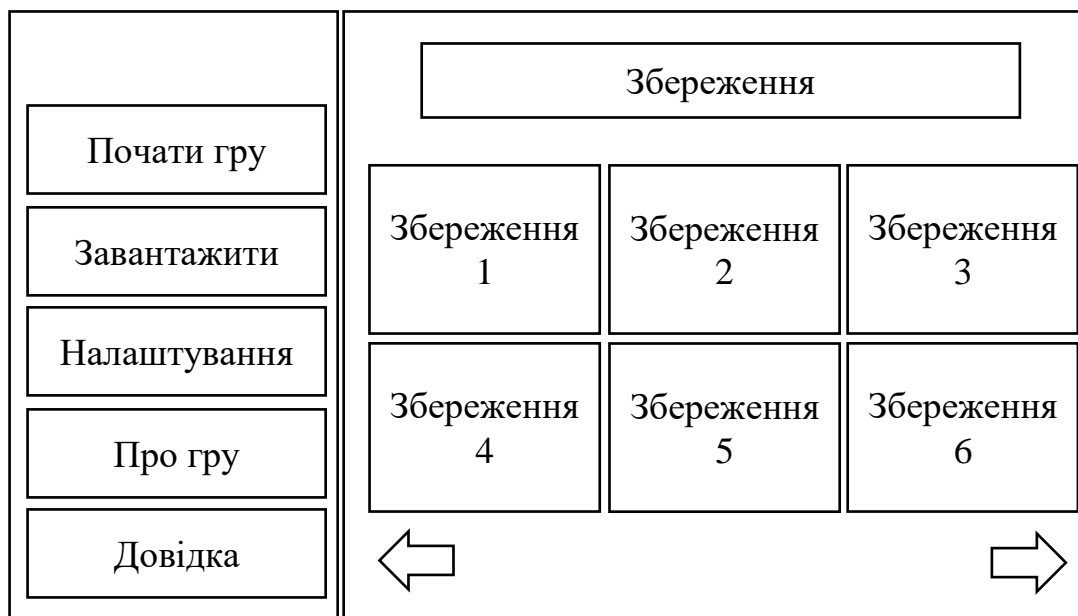


Рисунок 2.6 – Макет дизайну сторінки управління збереженнями

Ну а також користувач може налаштувати відображення гри під себе. Макет дизайну сторінки налаштувань показано на рисунку 2.7.

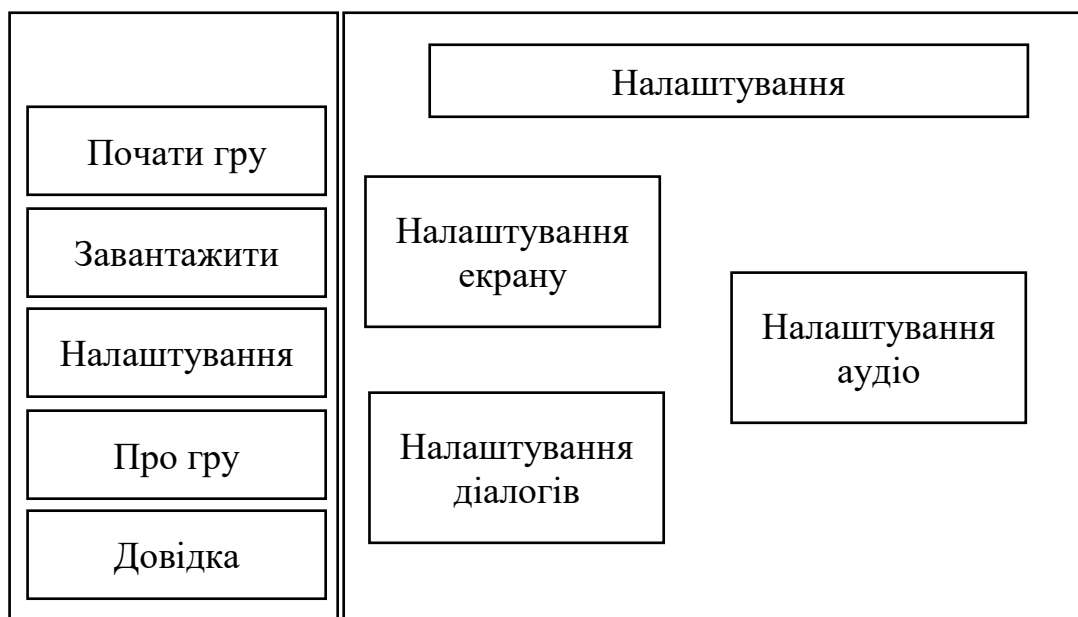


Рисунок 2.7 – Макет дизайну сторінки налаштувань

Передбачено можливість налаштування наступних параметрів:

- Параметри відображення гри на екрані
- Параметри відображення діалогів (швидкість)
- Налаштування аудіо:
 - гучність звуку;
 - гучність аудіо-ефектів;
 - гучність голосу в озвучених діалогах.

2.3 Проектування сценарію навчальної комп'ютерної гри з кібербезпеки

Окреслимо основні сценарні елементи навчальної комп'ютерної гри з кібербезпеки.

Сетинг: Кривий Ріг, 2045 рік. Місто перетворилося на високотехнологічний центр з яскравими неоновими вивісками та розумними будинками, але зі своїм традиційним шармом.

Головний герой: молодий спеціаліст з кібербезпеки, який щойно отримав роботу в престижній фірмі "КіберЩит".

Сюжет: протагоніст починає свій перший день на новій роботі і виявляє, що компанія стала мішенню для серії складних кібератак. Гравці допомагають йому захистити компанію, використовуючи свої знання з кібербезпеки.

Геймплей: Гравець взаємодіє з різними персонажами, збирають підказки та вирішують головоломки, щоб виявити слабкі місця в системі безпеки. Вони також вчаться розпізнавати фішингові атаки, захищати паролі та використовувати шифрування.

Міні-ігри:

- Фішинговий детектив: Гравці аналізують електронні листи, щоб визначити, які з них є фішинговими.
- Парольний пазл: Гравці складають надійні паролі, використовуючи дані підказки.
- Шифрувальний виклик: Гравці використовують базові принципи шифрування, щоб зашифрувати повідомлення.

Завдання:

- Захистити сервери від DDoS-атак.
- Встановити фаєрволи та антивірусне програмне забезпечення.
- Навчити персонал основам кібергігієни.

Кінцівка: Залежно від виборів гравця, герой може успішно захистити компанію від атак, або компанія може зазнати серйозних втрат через кіберзлочини. Історія підкреслює важливість кібербезпеки в сучасному світі.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ КОМП'ЮТЕРНОЇ ГРИ З КІБЕРБЕЗПЕКИ

3.1 Вибір засобів реалізації навчальної комп'ютерної гри з кібербезпеки

У рамках нашого проєкту, що передбачає створення освітньої комп'ютерної гри в жанрі кібербезпеки, ми вирішили використовувати Ren'Py як основний рушій. Ren'Py є відкритим програмним забезпеченням, яке спеціалізується на створенні візуальних новел та інтерактивних історій. Воно пропонує широкі можливості для розробки історій з різноманітними сюжетними розвилками та інтерактивними елементами, що дозволяє занурити гравців у захоплююче навчальне середовище.

Вибір Ren'Py для нашої гри з кібербезпеки обумовлений його перевагами: платформа є безкоштовною та підтримує мультиплатформність, тобто гру можна запускати на різних операційних системах, включаючи Windows, macOS та Linux. Простий у використанні скриптовий мова, заснований на Python, дає нам змогу реалізувати складні ігрові механіки, своєрідні для кібербезпеки, такі як зашифровані повідомлення, системи аутентифікації чи вирішення питань щодо захисту від вірусів.

Крім того, багата спільнота розробників та користувачів Ren'Py забезпечує доступ до множини туторіалів, документації та готових до використання модулів, що значно спрощує процес розробки і скорочує час на вивід продукту на ринок. Можливість інтеграції з іншими програмами та сервісами також відкриває широкі можливості для розширення функціональності нашої освітньої гри.

Таким чином, вибір Ren'Py як движка для нашої проєктної роботи є обдуманим і стратегічним рішенням, яке дозволить нам створити інтерактивний та ефективний навчальний продукт у сфері кібербезпеки, забезпечивши якісне навчання і зацікавлення з боку користувачів.

Ren'Py є потужним двигуном для створення візуальних новел, який вбудовує підтримку скриптів на мові Python. Це дає розробникам величезну гнучкість при проектуванні своїх ігор, оскільки Python є однією з найпопулярніших та найпотужніших мов програмування. З використанням Python у Ren'Py, творці можуть легко інтегрувати різноманітні інтерактивні елементи та міні-ігри, розширюючи традиційну лінійну структуру візуальних новел.

Окрім стандартних опцій, оснащення двигуна Python-скриптингом відкриває двері до таких можливостей, як індивідуальні системи балів, складні розгалуження сюжету, симулятори побачень, пазли, стратегічні ігри, і навіть забезпечує можливість інтегрування штучного інтелекту для більш динамічних діалогів та подій.

Для редагування коду проєкту створеного з використанням ігрового движка Ren'Py, ми обрали редактор коду Visual Studio Code (VS Code) завдяки його високій гнучкості та багатому набору функцій. Visual Studio Code популярний серед розробників через його інтуїтивний інтерфейс, велику кількість доступних розширень і підтримку багатьох мов програмування.

Використання VS Code для розробки ігор на Ren'Py вимагає встановлення певного розширення, яке підтримує мову скриптів Ren'Py і забезпечує такі функції як підсвічування синтаксису, автодоповнення коду та перевірку на помилки у коді. Також зручною є підтримка Git для керування версіями проєкту, що дуже корисно для командної роботи та ефективного управління змінами в коді.

Щоб почати роботу з Ren'Py в VS Code, спочатку потрібно встановити сам Ren'Py SDK (Software Development Kit), потім VS Code, а потім знайти та інсталювати розширення, спеціально призначене для Ren'Py, через вбудований магазин розширень у VS Code. Після налаштування оточення можна приступати до написання сценаріїв для ігор, створення персонажів, діалогів, рішень, що впливають на сюжет, та інших елементів, які входять у розробку

візуальних новел і симуляторів побачень, для чого зазвичай і використовується Ren'Py.

3.2 Програмна реалізація навчальної комп'ютерної гри з кібербезпеки

Завдяки гнучкості движка Ren'Py і можливостям мови програмування Python, ми успішно реалізували навчальну гру з кібербезпеки. Ми створили і оптимізували графічний інтерфейс, забезпечивши його зручністю та естетичністю для кінцевого користувача. Візуальні елементи та анімації були уважно втілені відповідно до задалегідь спроектованої концепції. Це не тільки покращило загальний вигляд гри, але й сприяло більшій занурюваності гравців у ігровий світ. Код на Python був написаний з урахуванням найкращих практик для забезпечення високої продуктивності та надійності гри.

На рисунку 3.1 показано стартову сторінку гри.



Рисунок 3.1 – Стартова сторінка гри

На цій сторінці ми можемо бачити титульне зображення з назвою гри, та головне меню, що надає можливість перейти до самої гри, переглянути завантаження, налаштування, отримати довідку тощо.

Типові сцени гри побудовані за допомогою статичних, анімованих зображень або відео, що відображаються на фоні, та текстового супроводу, що відображаються вниз (рис. 3.2).



Рисунок 3.2 – Типові сцени гри

Крім того на сцені можуть також відображатись персонажі гри (рис. 3.3), що також можуть бути анімовані, інтерактивно взаємодіяти з гравцем. Якщо відбувається діалог з персонажем, його текстовий супровід також відображається внизу.



Рисунок 3.3 – Відображення персонажу

Гра наповнена різноманітними активностями та міні-іграми. Одним з важливих компонентів гри є вікторини або опитування. За їх допомогою можна перевірити, чи засвоїв гравець ті чи інші теоретичні дані. Приклад такої вікторини показано на рисунку 3.4.

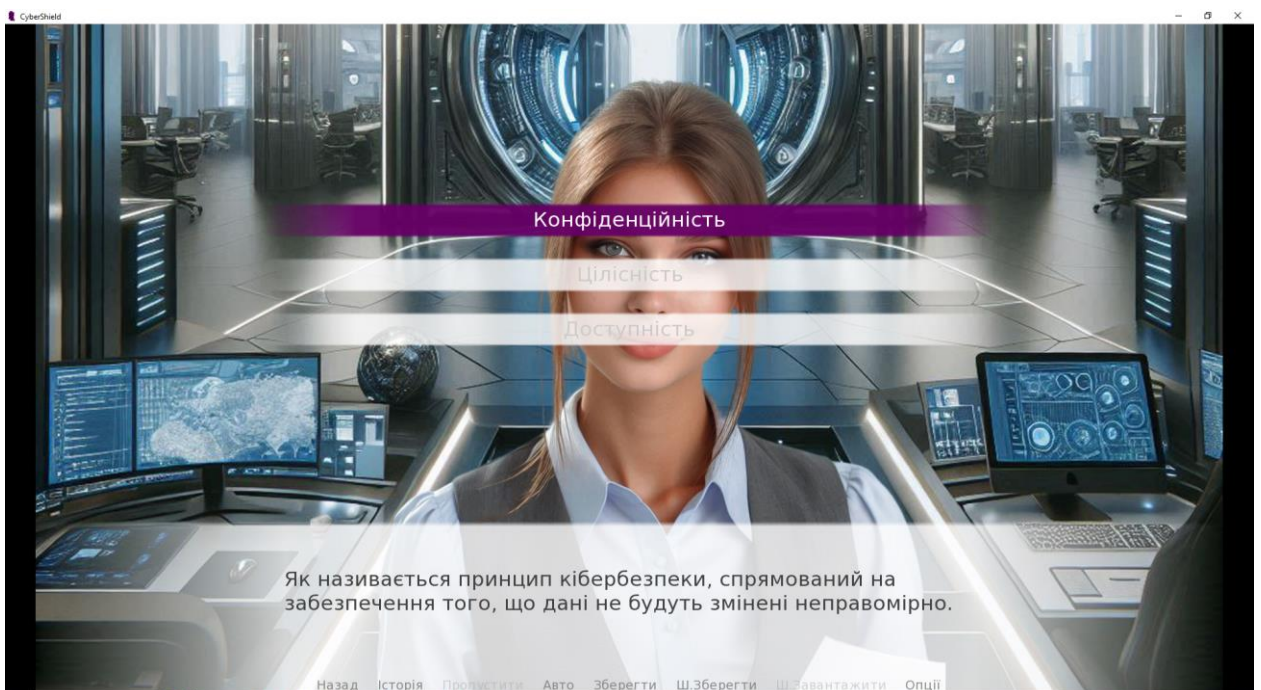


Рисунок 3.4 – Приклад вікторини в грі

Іншим прикладом міні-гри може слугувати гра з пошуком надійного або ненадійного паролю (рис. 3.5).

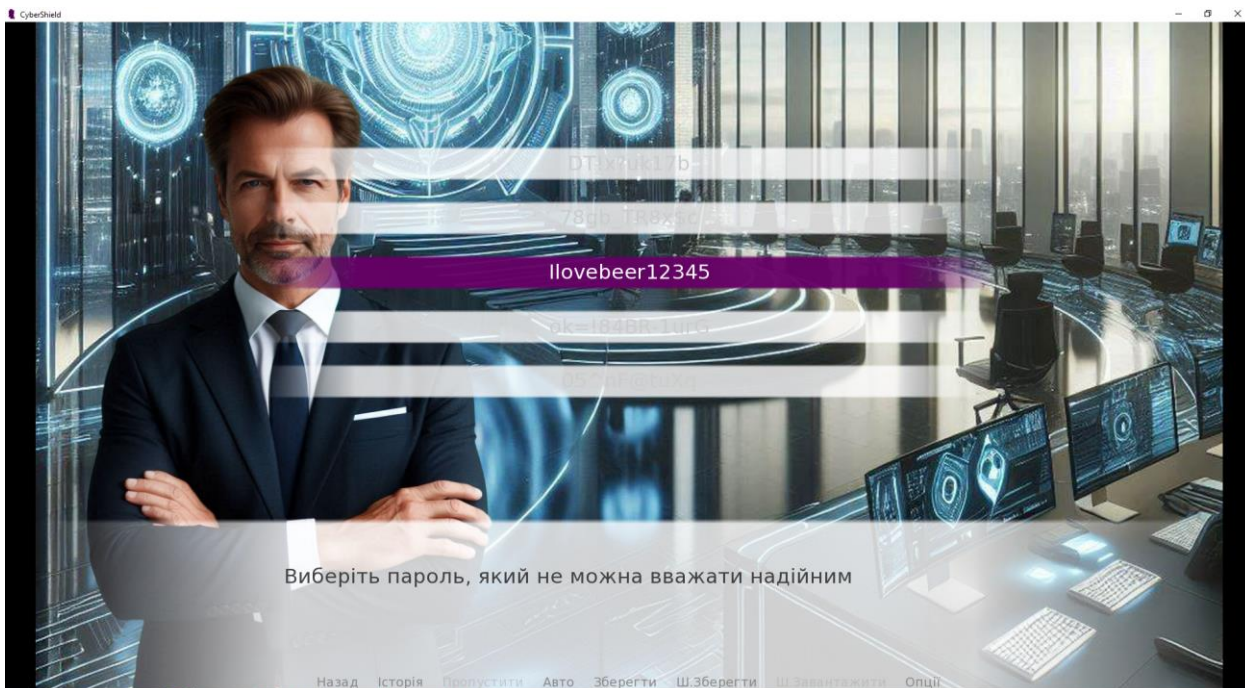


Рисунок 3.5 – Міні-гра з пошуком ненадійного паролю

Або гра з перевіркою листів на наявність фішингових повідомлень (рис. 3.6).

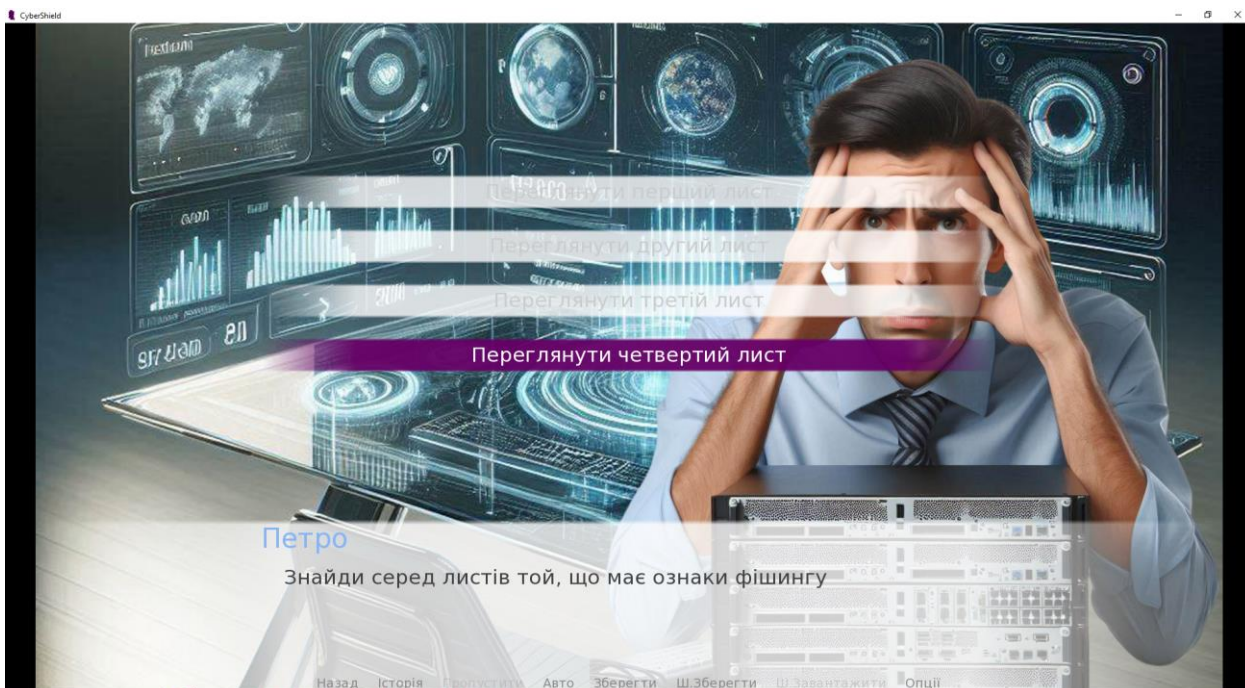


Рисунок 3.6 – Міні-гра з пошуком фішингових повідомлень

В залежності від прогресу гравця і його вибору під час розгортання сюжету, він може дістатись або до хорошої кінцівки гри (рис.3.7) або до поганої (рис. 3.8).



Рисунок 3.7 – Хороша кінцівка гри



Рисунок 3.8 – Погана кінцівка гри

Крім самої гри також реалізовано механізм збереження прогресу та завантаження збережених ігор (рис. 3.9)

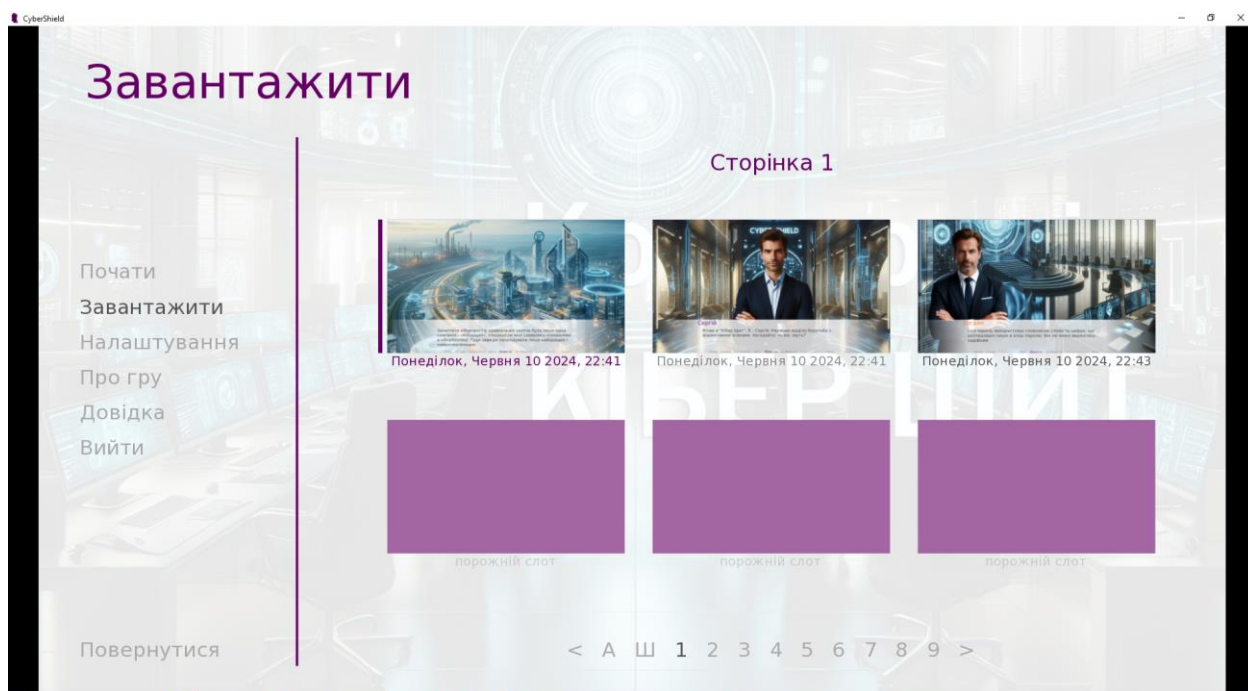


Рисунок 3.9 – Вікно перегляду збережених ігор

Також реалізована сторінка налаштувань гри, що включає налаштування відображення на екрані, параметри відображення тексту та параметри гучності музики, фонових звуків, голосу озвучки.

Вікно налаштувань показано на рисунку 3.10.

На рисунку 3.11 показано вікно довідки, в якому гравець може дізнатись за якими клавішами які дії закріплені, які механіки гри реалізуються за допомогою миші і тому подібне.

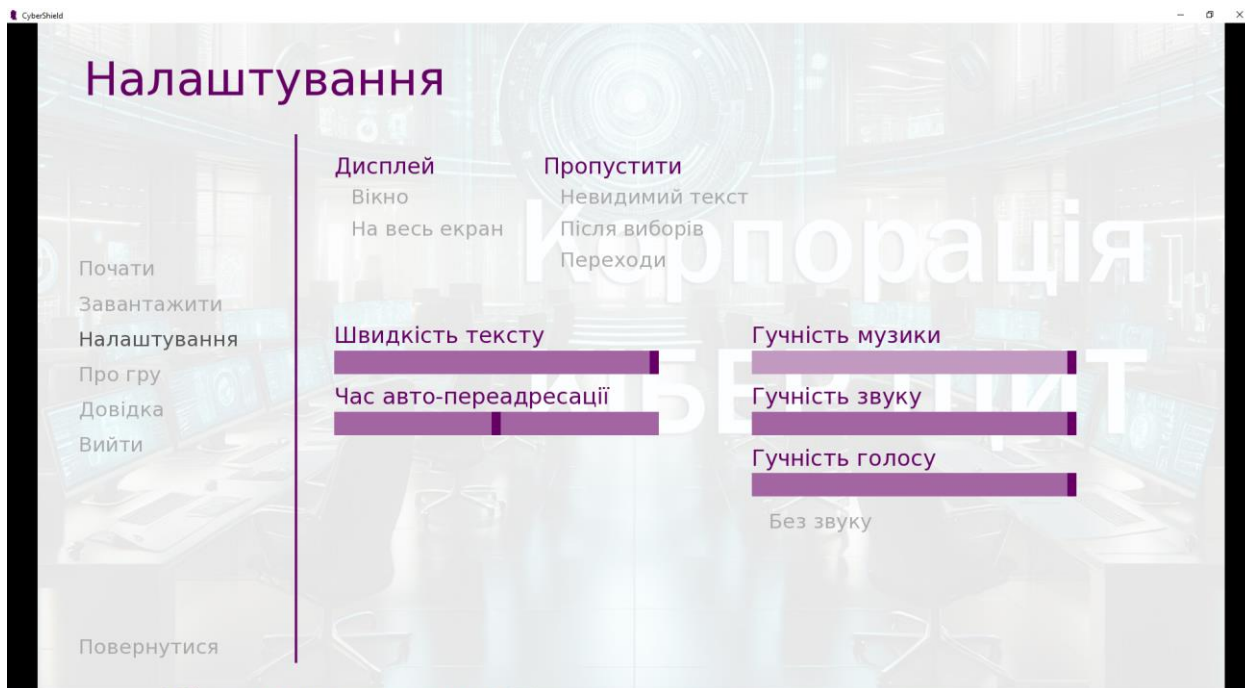


Рисунок 3.10 – Вікно налаштувань

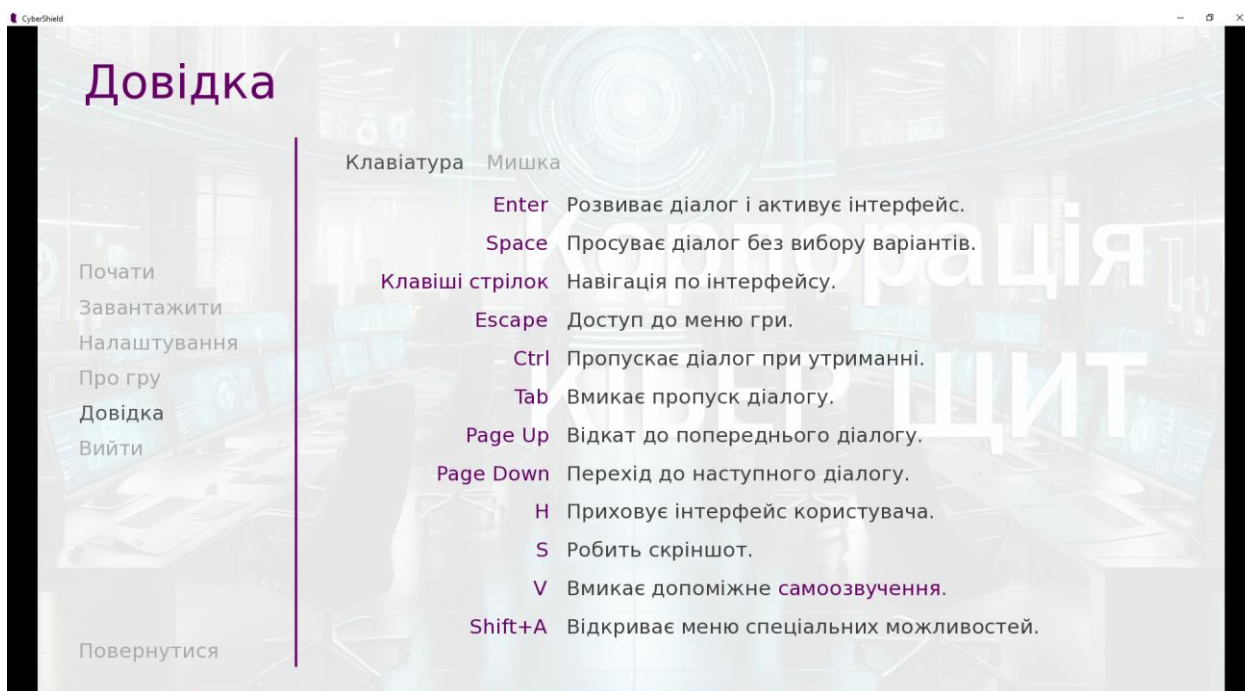


Рисунок 3.11 – Відображення довідки по управлінню грою

Завершивши цей проект, ми представляємо нашу повноцінну навчальну гру з кібербезпеки, яка втілює собою синтез захоплення та навчання. Гра була ретельно розроблена для забезпечення інтерактивного досвіду з глибоким і захоплюючим сюжетом, що тісно переплітається з серією завдань та

головоломок, зорієнтованих на розвиток критичного мислення та навичок у сфері кібербезпеки.

Гравець відправляється в подорож, де зустрічає харизматичних персонажів, кожен з яких може донести унікальні знання та розкрити аспекти кібербезпеки з різних боків. Виконуючи місії, гравець опановує основні концепції та методи захисту інформації, такі як безпечне створення та зберігання паролів, розпізнавання фішингових атак, основи шифрування, а також правила безпечного онлайн поведіння.

Кожен етап гри продумано збалансовано, щоб поєднувати розважальний контент із освітнім матеріалом, гарантуючи, що кожен взаємодіє з матеріалом гри на його власних умовах. В процесі гри залучаються аудіовізуальні елементи та сценарії реального часу, які допомагають підтримувати високий рівень залученості та підвищити засвоюваність матеріалу.

Завдяки цій грі користувачі не просто набувають знання з кібербезпеки – вони також розвивають навички, необхідні для їх застосування в повсякденному житті і професійній діяльності, роблячи цифровий світ безпечнішим. Така інтерактивна форма навчання не тільки сприяє більшій залученості у процес навчання, але й стимулює критичне мислення та вирішення проблем, що становить цінність сама по собі.

ВИСНОВКИ

Ми успішно завершили кваліфікаційну роботу, яка зосереджена на дослідженні значущості кібербезпеки в контексті сучасних технологічних викликів. В ході роботи ми детально визначили ключові вимоги, які повинна відповідати навчальна гра для вивчення основ кібербезпеки. Основною метою гри є надання користувачам практичних навичок та знань, необхідних для захисту інформаційних систем і даних.

На наступному етапі, ми спроектували структуру гри, включно з її інтерфейсом та сюжетом, звертаючи особливу увагу на інтерактивність та залученість гравця. Структура гри розрахована таким чином, щоб навчальний матеріал подавався логічно та послідовно, забезпечуючи поступове зростання складності завдань.

Для розробки гри ми вибрали рушій Ren'Py, який є ідеальним інструментом для створення візуальних новел та інтерактивних розповідей, завдяки своїм властивостям орієнтації на сюжет і легкості використання. Рендеринг графіки, обробка тексту та управління потоком ігрового процесу — усе це ми зреалізували завдяки можливостям, які надає цей рушій.

Програмування гри було виконано на мові Python, що дозволило нам гнучко імплементувати весь необхідний функціонал та логіку. Python виявився вкрай зручним у використанні для розробки ігрових механік і аналізу даних, також надавши змогу легко інтегрувати додаткові бібліотеки та інструменти, необхідні для розширення можливостей гри.

Завдяки цілісному підходу до розробки, ми створили повнофункціональний навчальний продукт, що може бути використаний у якості засобу для підвищення обізнаності про кібербезпеку та відпрацювання практичних навичок з цієї дисципліни.

ПЕРЕЛІК ПОСИЛАНЬ

1. Даник Ю.Г. основи кібербезпеки та кібероборони / Даник Ю.Г., Воробієнко П.П., Чернега В.М. – Одеса: ОНАЗ ім. О.С. Попова, 2019. – 320 с.
2. Козубцова Л. М. Концепція навчально-тренувального комплексу підготовки військових спеціалістів інформаційної та кібербезпеки на засадах комп'ютерної гри (гейміфікації) / Козубцова Леся Михайлівна, Козубцов Ігор Миколайович, Ліщина Валерій Олександрович, Штаненко Сергій Станіславович // Кібербезпека: освіта, наука, техніка. – №2 (18), 2022. – С. 49-60.
3. Основи розробки комп'ютерних ігор: електронний навчальний посібник для підготовки студентів на першому (бакалаврському) рівні вищої освіти, галузі знань 12 «Інформаційні технології», спеціальності 121 «Інженерія програмного забезпечення» / Укладач: В.Г. Шерстюк. – Херсон: видавництво ФОП Вишемирський В.С., 2018. – 210 с.
4. Лугова Т.А., Блажко О.А. Проектування комп'ютерних ігор для навчання : навчальний підручник / Т.А. Лугова, О.А. Блажко. – Одеса : ФОП «Побута». – 2018. – 212 с.
5. Радайтис Б. Навчальні комп'ютерні ігри: правила розробки та етапи створення / Радайтис Богдан, Яценко Оксана // The Aspects of Contemporary Scientific Research that Encompass Both Theoretical and Practical Components. – С. 113-117.
6. Гончарова І.П. Кібербезпека в цифровому освітньому середовищі закладів професійної освіти: електронний навчальний курс / І.П. Гончарова, Біла Церква, БІНПО ДЗВО «УМО» НАПН УКРАЇНИ, 2022. 80 с.

7. Основи кібергігієни. Як держслужбовцям захиститися від хакерських атак [Електронний ресурс]. – Режим доступу: <https://osvita.diia.gov.ua/courses/cyber-hygiene>
8. Кібербезпека [Електронний ресурс]. – Режим доступу: <https://moz.gov.ua/kiberbezpeka>
9. Закон України Про основні засади забезпечення кібербезпеки України [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2163-19#Text>
10. Створення візуальної новели на движку Ren'py [Електронний ресурс]. – Режим доступу: <https://conf.ztu.edu.ua/wp-content/uploads/2022/04/98.pdf>
- 11.5 движків для створення ігор у Python [Електронний ресурс]. – Режим доступу: <https://itproger.com/ua/news/5-dvizhkov-dlya-sozdaniya-igr-v-python>
12. Основи створення персонажу для комп'ютерної 2d гри [Електронний ресурс]. – Режим доступу: http://eprints.zu.edu.ua/35620/1/%D0%B7%D0%B1%D1%96%D1%80%D0%BD%D0%B8%D0%BA_22-247-251%20%281%29.pdf
13. Gamedev – все про напрям розробки ігор [Електронний ресурс]. – Режим доступу: <https://goiteens.com/blog/gamedev-vse-pro-napriam/>
14. Особливості розробки ігор жанру візуальної новели [Електронний ресурс]. – Режим доступу: <https://conf.ztu.edu.ua/wp-content/uploads/2023/06/89.pdf>

Додаток А

Текст програми

```
# У цьому файлі міститься скрипт гри.

# Оголошення символів, які використовуються в цій грі. Колірний
аргумент розфарбовує
# ім'я персонажа.

define n = Character("[name]", color="#fcf976")
define s = Character("Сергій", color="#6f208f")
define ok = Character("Оксана", color="#6f208f")
define mk = Character("Микола", color="#87b4f7")
define bg = Character("Богдан", color="#fa802e")
define pr = Character("Петро", color="#87b4f7")

# Гра починається тут.

label start:

    scene kr01
    with dissolve

    "2045 рік. Кривий Ріг перетворився на високотехнологічний
центр"
    "Тепер він славиться не лише рудою та металургійними
комбінатами, а й розвиненим машинобудуванням, нано- та біо-
технологіями"
    "Ну і розвиненим tech-сектором."
    scene kr02
    with dissolve
    "Але такий шалений успіх неминуче викликав заздрість у
недобросовісних конкурентів."
    "І несподівано на підприємства Кривого Рогу почались потужні
кібератаки."
    scene kr03
    with dissolve
    "Захистити кіберпростір криворіжжя здатна була лише одна
компанія - «Кібершит», спеціалісти якої славились пізнаннями в
кібербезпеці. Туди завжди запрошували лише найкращих і
найдосвідченіших."
    "Але цій компанії стало просто не вистачати рук. І тоді вона
вирішила відкрити свої двері новачкам."
    "І ви - саме такий новачок."

    scene cs01
    with dissolve

    "Ось ви на порозі компанії «Кібершит». Робота в цій компанії
- чудовий шанс зробити стрімку кар'єру."
```

"Але для цього потрібно в короткий час опанувати всі секрети кібербезпеки. Довести, що ви – кращий."

"Впоравшись з хвилюванням, ви робите перший крок."

"І тепер все залежить від вас."

```
scene cs02
with dissolve
# Це репліки діалогу.
show serg
with dissolve
s "Вітаю в "Кібер Щит". Я – Сергій. Керівник відділу боротьби
з фішинговими атаками. Нагадайте, як вас звать?"
python:
    name = renpy.input("Ваше ім'я:")
    name = name.strip() or "Анонім"
s "Радий знайомству, [name]. Мені приємно бачити такий молодий
талант в нашій команді"
n "Дякую, пане Сергію. Я дуже радий бути тут і готовий
навчатися"
s "Отже, твої основні обов'язки будуть полягати в моніторингу
наших систем на предмет будь-яких ознак вторгнення та
вразливостей"
s "Ти будеш працювати з нашими найновішими інструментами для
виявлення та запобігання атакам."
n "Це звучить захоплююче! Чи є у нас специфічні інструменти
або методи, які я повинен вивчити?"
s "Так, ми використовуємо кастомізовані рішення, які були
розроблені спеціально для наших клієнтів. "
s "Ти почнеш з тренінгу, щоб ознайомитися з ними. Також, ти
будеш працювати над розробкою нових стратегій захисту."
n "Чудово! Я завжди хотів працювати над чимось, що має реальний
вплив. Які перші кроки?"
s "На перших порах, ти будеш спостерігати за роботою наших
старших аналітиків."
s "Важливо розуміти контекст та специфіку нашої роботи. "
s "Після цього, ми дамо тобі невеликий проект, щоб ти міг
показати свої навички."
n "Я готовий до викликів. Дякую за можливість!"
s "Ми теж раді, що ти з нами. Ласкаво просимо до команди!"
```

```
scene cs03
with dissolve
show oksana
with dissolve
ok "Вітаю на борту, [name]. Я Оксана. В мої обов'язки входить
навчання персоналу"
ok "Перш ніж ми почнемо, давай обговоримо основи кібербезпеки,
які є життєво важливими для нашої роботи тут, в "Кібер Щит""
n "Дякую. Я готовий вчитися"
ok "Добре. Почнемо з трьох основних принципів кібербезпеки:
конфіденційність, цілісність та доступність. "
ok "Конфіденційність означає захист інформації від
несанкціонованого доступу. "
```

ок "Цілісність - це забезпечення того, що дані не були змінені неправомірно."

ок "А доступність означає, що інформація завжди доступна для тих, хто має до неї право."

п "Розумію. Як ми можемо забезпечити ці принципи в нашій роботі?"

ок "Ми використовуємо різноманітні інструменти та методики. Наприклад, для забезпечення конфіденційності ми використовуємо шифрування. "

ок "Цілісність даних можна забезпечити за допомогою контрольних сум та цифрових підписів. "

ок "Що стосується доступності, то ми розробляємо резервні та відновлювальні стратегії для систем, щоб вони могли швидко відновитися після будь-яких інцидентів"

п "Це здається складним. Чи є якісь конкретні технології або інструменти, які я повинен знати?"

ок "Так, ти повинен ознайомитися з брандмауерами, системами виявлення та запобігання вторгненням, а також з антивірусним програмним забезпеченням."

ок "Також важливо розуміти основи криптографії та мережевої безпеки."

п "Дякую за інформацію. Я вже не можу дочекатися, коли почну працювати з цими інструментами"

ок ": І ми раді, що ти з нами. Твоя готовність до навчання - це саме те, що робить "Кібер Щит" сильнішим"

ок "Але тепер давай перевіримо, як добре ти все зрозумів"

menu:

"Як називається принцип кібербезпеки, спрямований на забезпечення того, що дані не будуть змінені неправомірно."

"Конфіденційність":

ок "Не правильно!"

ок "Конфіденційність означає захист інформації від несанкціонованого доступу."

"Цілісність":

ок "Молодець! Все правильно!"

ок "Цілісність - це забезпечення того, що дані не були змінені неправомірно"

"Доступність":

ок "Не правильно!"

ок "Доступність означає, що інформація завжди доступна для тих, хто має до неї право"

menu:

"Як називається принцип кібербезпеки, спрямований на захист інформації від несанкціонованого доступу."

"Конфіденційність":

ок "Абсолютно правильно!"

ок "Конфіденційність означає захист інформації від несанкціонованого доступу."

"Цілісність":

ок "Не правильно!"

```
        ok "Цілісність - це забезпечення того, що дані не були
змінені неправомірно"
        "Доступність":
            ok "Не правильно!"
            ok "Доступність означає, що інформація завжди доступна
для тих, хто має до неї право"
```

```
menu:
```

```
    "Як називається принцип кібербезпеки, спрямований на те,
що інформація завжди доступна для тих, хто має до неї право"
```

```
    "Конфіденційність":
```

```
        ok "Не правильно!"
```

```
        ok "Конфіденційність означає захист інформації від
несанкціонованого доступу."
```

```
    "Цілісність":
```

```
        ok "Не правильно!"
```

```
        ok "Цілісність - це забезпечення того, що дані не були
змінені неправомірно"
```

```
    "Доступність":
```

```
        ok "Неперевершено! Ти - молодець!"
```

```
        ok "Доступність дійсно означає, що інформація завжди
доступна для тих, хто має до неї право"
```

```
    ok "Що ж, маємо рухатись далі"
```

```
scene cs04
```

```
with dissolve
```

```
show bogdan at left
```

```
with dissolve
```

```
menu:
```

```
    "Виберіть пароль, який не можна вважати надійним"
```

```
    "DT!x?uk17b":
```

```
        bg "Не правильно!"
```

```
        bg "Цей пароль цілком надійний"
```

```
    "78gb_TR8x$c":
```

```
        bg "Не правильно!"
```

```
        bg "Цей пароль цілком надійний"
```

```
    "Ilovebeer12345":
```

```
        bg "Неперевершено! Ти - молодець!"
```

```
        bg "Цей пароль використовує словникові слова та цифри,
що розташовані лише в кінці паролю. Він не може вважатись надійним"
```

```
    "ok=!84BR-1urG":
```

```
        bg "Не правильно!"
```

```
        bg "Цей пароль цілком надійний"
```

```
    "05^nF@tuXq":
```

```
        bg "Не правильно!"
```

```
        bg "Цей пароль цілком надійний"
```

```
python:
```

```
    import re
```

```
    def check_password_strength(password):
```



```

    # Перевірка на мінімальну довжину, наявність цифр, великих та
    малих літер та спецсимволів
    if (len(password) >= 8 and re.search(r"\d", password) and
        re.search(r"[A-Z]", password) and
        re.search(r"[a-z]", password) and re.search(r"\W",
password)):
        return True
    else:
        return False

def game():
    score = 0
    print("Ласкаво просимо до гри 'Майстер Паролів'!")
    print("Спробуйте визначити, чи є пароль надійним. За кожну
правильну відповідь ви отримуєте бал.")
    print("Гра триває до трьох раундів.\n")

    for i in range(3):
        password = input(f"Раунд {i+1}. Введіть пароль: ")
        is_strong = check_password_strength(password)
        answer = input("Чи є цей пароль надійним? (так/ні):
").lower()

        if (is_strong and answer == "так") or (not is_strong and
answer == "ні"):
            print("Вірно! Ви отримуєте бал.")
            score += 1
        else:
            print("Невірно. Не засмучуйтеся, спробуйте ще раз!")

    print(f"Ваш поточний рахунок: {score}\n")

print(f"Гра завершена! Ваш фінальний рахунок: {score}")

scene cs05
with dissolve
show petro at right
with dissolve
pr "Знайди серед листів той, що має ознаки фішингу"
menu:
    pr "Знайди серед листів той, що має ознаки фішингу"
    "Переглянути перший лист":
        pr "Не правильно!"

    "Переглянути другий лист":
        pr "Не правильно!"

    "Переглянути третій лист":
        pr "Неперевершено! Ти - молодець!"

    "Переглянути четвертий лист":
        pr "Не правильно!"

```

```

Python:
    import random

# Приклади фішингових та безпечних повідомлень
phishing_examples = [
    "Вітаємо! Ви виграли автомобіль. Надішліть свої дані для отримання призу.",
    "Ваш обліковий запис потребує перевірки. Будь ласка, введіть свій пароль.",
    "Це повідомлення від служби підтримки. Ми помітили підозрілу активність на вашому обліковому записі."
]
safe_examples = [
    "Не забудьте завтра на зустріч о 10:00.",
    "Ваша квитанція про оплату додається до цього листа.",
    "Оновлення безпеки буде встановлено автоматично."
]

def is_phishing(message):
    # Функція для визначення, чи є повідомлення фішинговим
    return message in phishing_examples

def play_game():
    score = 0
    print("Ласкаво просимо до гри 'Фішинг Детектор'!")
    print("Вам будуть показані різні повідомлення, і ви повинні визначити, чи є вони фішинговими.")
    print("За кожну правильну відповідь ви отримаєте бал.\n")

    for i in range(5):
        if random.choice([True, False]):
            message = random.choice(phishing_examples)
        else:
            message = random.choice(safe_examples)

        print(f"Повідомлення: {message}")
        answer = input("Чи є це фішинговим повідомленням? (так/ні): ").lower()

        if is_phishing(message) == (answer == "так"):
            print("Вірно! Ви отримуєте бал.")
            score += 1
        else:
            print("Невірно. Будьте уважніші!")

    print(f"Ваш поточний рахунок: {score}\n")

    print(f"Гра завершена! Ваш фінальний рахунок: {score}")

label goodfinal:
    scene cs06
    with dissolve
    show serg at left

```

```

with dissolve
show dmytro at right
with dissolve
show olena
with dissolve
s "Вітаємо! Ти успішно пройшов всі випробування!"

label badfinal:
scene cs06
with dissolve
show serg at left
with dissolve
show mykola at right
with dissolve
show dmytro
with dissolve
s "Нажаль, ти не впорався. Ми програли цю кібервійну"
s "До побачення!"

# На цьому гра закінчується.

return

#####
#####
## Ініціалізація
#####
#####

init offset = -1

#####
#####
## Стилi
#####
#####

style default:
properties gui.text_properties()
language gui.language

style input:
properties gui.text_properties("input", accent=True)
adjust_spacing False

style hyperlink_text:
properties gui.text_properties("hyperlink", accent=True)
hover_underline True

style gui_text:
properties gui.text_properties("interface")

```

```

style button:
    properties gui.button_properties("button")

style button_text is gui_text:
    properties gui.text_properties("button")
    yalign 0.5

style label_text is gui_text:
    properties gui.text_properties("label", accent=True)

style prompt_text is gui_text:
    properties gui.text_properties("prompt")

style bar:
    ysize gui.bar_size
    left_bar      Frame("gui/bar/left.png",      gui.bar_borders,
tile=gui.bar_tile)
    right_bar     Frame("gui/bar/right.png",     gui.bar_borders,
tile=gui.bar_tile)

style vbar:
    xsize gui.bar_size
    top_bar       Frame("gui/bar/top.png",       gui.vbar_borders,
tile=gui.bar_tile)
    bottom_bar    Frame("gui/bar/bottom.png",    gui.vbar_borders,
tile=gui.bar_tile)

style scrollbar:
    ysize gui.scrollbar_size
    base_bar      Frame("gui/scrollbar/horizontal_[prefix_]bar.png",
gui.scrollbar_borders, tile=gui.scrollbar_tile)
    thumb         Frame("gui/scrollbar/horizontal_[prefix_]thumb.png",
gui.scrollbar_borders, tile=gui.scrollbar_tile)

style vscrollbar:
    xsize gui.scrollbar_size
    base_bar      Frame("gui/scrollbar/vertical_[prefix_]bar.png",
gui.vscrollbar_borders, tile=gui.scrollbar_tile)
    thumb         Frame("gui/scrollbar/vertical_[prefix_]thumb.png",
gui.vscrollbar_borders, tile=gui.scrollbar_tile)

style slider:
    ysize gui.slider_size
    base_bar      Frame("gui/slider/horizontal_[prefix_]bar.png",
gui.slider_borders, tile=gui.slider_tile)
    thumb         "gui/slider/horizontal_[prefix_]thumb.png"

style vslider:
    xsize gui.slider_size

```

```
    base_bar      Frame("gui/slider/vertical_[prefix_]bar.png",
gui.vslider_borders, tile=gui.slider_tile)
    thumb "gui/slider/vertical_[prefix_]thumb.png"
```

```
style frame:
    padding gui.frame_borders.padding
    background      Frame("gui/frame.png",      gui.frame_borders,
tile=gui.frame_tile)
```

```
#####
#####
##  Екрани в гри
#####
#####
```

```
##                               Екран                               розмови
#####
##
##  Екран say використовується для відображення діалогу гравцеві.
Він приймає
##  два параметри, хто і що, ім'я персонажа, що говорить, і текст,
який буде
##  відобразитися відповідно. (Параметр who може мати значення
None, якщо ім'я не
##  вказано.)
##
##  Цей екран має створювати текст, який можна відобразити з
ідентифікатором
##  "what", оскільки Ren'Py використовує це для керування
відображенням тексту.
##  Він також може створювати відображувані елементи з
ідентифікаторами "who" та
##  ідентифікаторами "window" для застосування властивостей стилю.
##
##  https://www.renpy.org/doc/html/screen\_special.html#say
```

```
screen say(who, what):
    style_prefix "say"
```

```
    window:
        id "window"
```

```
        if who is not None:
```

```
            window:
                id "namebox"
                style "namebox"
                text who id "who"
```

```

text what id "what"

## Якщо є зображення збоку, відобразить його над текстом. Не
відображати на
## телефоні як варіант - немає місця.
if not renpy.variant("small"):
    add SideImage() xalign 0.0 yalign 1.0

## Робить поле імен доступним для стилізації через об'єкт
Character.
init python:
    config.character_id_prefixes.append('namebox')

style window is default
style say_label is default
style say_dialogue is default
style say_thought is say_dialogue

style namebox is default
style namebox_label is say_label

style window:
    xalign 0.5
    xfill True
    yalign gui.textbox_yalign
    ysize gui.textbox_height

    background Image("gui/textbox.png", xalign=0.5, yalign=1.0)

style namebox:
    xpos gui.name_xpos
    xanchor gui.name_xalign
    xsize gui.namebox_width
    ypos gui.name_ypos
    ysize gui.namebox_height

    background Frame("gui/namebox.png", gui.namebox_borders,
tile=gui.namebox_tile, xalign=gui.name_xalign)
    padding gui.namebox_borders.padding

style say_label:
    properties gui.text_properties("name", accent=True)
    xalign gui.name_xalign
    yalign 0.5

style say_dialogue:
    properties gui.text_properties("dialogue")

    xpos gui.dialogue_xpos
    xsize gui.dialogue_width

```

```

ypos gui.dialogue_ypos

adjust_spacing False

##                                     Екран                                     введення
#####
##
## Цей екран використовується для відображення renpy.input.
Параметр prompt
## використовується для передачі текстової підказки.
##
## Цей екран має створити відображуваний вхід з ідентифікатором
"input" для
## прийняття різних вхідних параметрів.
##
## https://www.renpy.org/doc/html/screen\_special.html#input

screen input(prompt):
    style_prefix "input"

    window:

        vbox:
            xanchor gui.dialogue_text_xalign
            xpos gui.dialogue_xpos
            xsize gui.dialogue_width
            ypos gui.dialogue_ypos

            text prompt style "input_prompt"
            input id "input"

style input_prompt is default

style input_prompt:
    xalign gui.dialogue_text_xalign
    properties gui.text_properties("input_prompt")

style input:
    xalign gui.dialogue_text_xalign
    xmaximum gui.dialogue_width

##                                     Екран                                     вибору
#####
##
## Цей екран використовується для відображення вибору в грі,
представленого
## командою 'menu'. Один параметр, 'items', – це список об'єктів,
кожен із
## полями заголовка та дії.
##
## https://www.renpy.org/doc/html/screen\_special.html#choice

```

```

screen choice(items):
    style_prefix "choice"

    vbox:
        for i in items:
            textbutton i.caption action i.action

style choice_vbox is vbox
style choice_button is button
style choice_button_text is button_text

style choice_vbox:
    xalign 0.5
    ypos 405
    yanchor 0.5

    spacing gui.choice_spacing

style choice_button is default:
    properties gui.button_properties("choice_button")

style choice_button_text is default:
    properties gui.button_text_properties("choice_button")

##                               Экран                               ШВИДКОГО                               МЕНЮ
#####
##
## Швидке меню відображається в грі, щоб забезпечити легкий доступ
до меню поза
## грою.

screen quick_menu():

    ## Переконайтеся, що це відображається поверх інших екранів.
    zorder 100

    if quick_menu:

        hbox:
            style_prefix "quick"

            xalign 0.5
            yalign 1.0

            textbutton _("Назад") action Rollback()
            textbutton _("Історія") action ShowMenu('history')
            textbutton _("Пропустити") action Skip() alternate
Skip(fast=True, confirm=True)
            textbutton _("Авто") action Preference("auto-
forward", "toggle")
            textbutton _("Зберегти") action ShowMenu('save')

```



```

        textbutton _("Ш.Зберегти") action QuickSave()
        textbutton _("Ш.Завантажити") action QuickLoad()
        textbutton _("Опції") action ShowMenu('preferences')

## Цей код гарантує, що екран quick_menu відображається в грі,
якщо гравець явно
## не приховав інтерфейс.
init python:
    config.overlay_screens.append("quick_menu")

default quick_menu = True

style quick_button is default
style quick_button_text is button_text

style quick_button:
    properties gui.button_properties("quick_button")

style quick_button_text:
    properties gui.button_text_properties("quick_button")

#####
#####
## Екрани головного та ігрового меню
#####
#####

##                               Екран                               навігації
#####
##
## Цей екран включено в головне та ігрове меню та забезпечує
навігацію до інших
## меню та початок гри.

screen navigation():

    vbox:
        style_prefix "navigation"

        xpos gui.navigation_xpos
        yalign 0.5

        spacing gui.navigation_spacing

    if main_menu:

        textbutton _("Почати") action Start()

    else:

        textbutton _("Історія") action ShowMenu("history")

```

```

        textbutton _("Зберегти") action ShowMenu("save")

        textbutton _("Завантажити") action ShowMenu("load")

        textbutton          _("Налаштування")          action
ShowMenu("preferences")

        if _in_replay:

            textbutton          _("Кінець          повтору")          action
EndReplay(confirm=True)

        elif not main_menu:

            textbutton _("Головне меню") action MainMenu()

            textbutton _("Про гру") action ShowMenu("about")

            if renpy.variant("pc") or (renpy.variant("web") and not
renpy.variant("mobile")):

                ## Допомога не є необхідною або актуальною для
мобільних пристроїв.
                textbutton _("Довідка") action ShowMenu("help")

            if renpy.variant("pc"):

                ## Кнопка виходу заборонена на iOS і непотрібна на
Android і в Веб.
                textbutton          _("Вийти")          action          Quit(confirm=not
main_menu)

style navigation_button is gui_button
style navigation_button_text is gui_button_text

style navigation_button:
    size_group "navigation"
    properties gui.button_properties("navigation_button")

style navigation_button_text:
    properties gui.button_text_properties("navigation_button")

##          Екран          ГОЛОВНОГО          МЕНЮ
#####
##
## Використовується для відображення головного меню під час
запуску Ren'Py.
##
## https://www.renpy.org/doc/html/screen\_special.html#main-menu

```

```

screen main_menu():

    ## Це гарантує, що будь-який інший екран меню буде замінено.
    tag menu

    add gui.main_menu_background

    ## Ця порожня рамка затемнює головне меню.
    frame:
        style "main_menu_frame"

    ## Інструкція використання включає інший екран всередині
    цього. Фактичний
    ## вміст головного меню знаходиться на екрані навігації.
    use navigation

    if gui.show_name:

        vbox:
            style "main_menu_vbox"

            text "[config.name!t]":
                style "main_menu_title"

            text "[config.version]":
                style "main_menu_version"

style main_menu_frame is empty
style main_menu_vbox is vbox
style main_menu_text is gui_text
style main_menu_title is main_menu_text
style main_menu_version is main_menu_text

style main_menu_frame:
    xsize 420
    yfill True

    background "gui/overlay/main_menu.png"

style main_menu_vbox:
    xalign 1.0
    xoffset -30
    xmaximum 1200
    yalign 1.0
    yoffset -30

style main_menu_text:
    properties gui.text_properties("main_menu", accent=True)

style main_menu_title:
    properties gui.text_properties("title")

```

```

style main_menu_version:
    properties gui.text_properties("version")

##                               Екран                               меню                               гри
#####
##
## Це описує базову структуру екрана меню гри. Він викликається за
## допомогою
## заголовка екрана та відображає фон, заголовок і навігацію.
##
## Параметр прокручування може бути None або одним із "viewport"
## чи "vpgrid".
## Коли цей екран призначений для використання з одним або
## декількома об'єктами,
## які включені (розміщені) всередині нього.

screen game_menu(title, scroll=None, yinitial=0.0):

    style_prefix "game_menu"

    if main_menu:
        add gui.main_menu_background
    else:
        add gui.game_menu_background

    frame:
        style "game_menu_outer_frame"

    hbox:

        ## Зарезервуйте місце для розділу навігації.
        frame:
            style "game_menu_navigation_frame"

        frame:
            style "game_menu_content_frame"

            if scroll == "viewport":

                viewport:
                    yinitial yinitial
                    scrollbars "vertical"
                    mousewheel True
                    draggable True
                    pagekeys True

                    side_yfill True

                vbox:
                    transclude

            elif scroll == "vpgrid":

```

```

        vpgrid:
            cols 1
            yinitial yinitial

            scrollbars "vertical"
            mousewheel True
            draggable True
            pagekeys True

            side_yfill True

            transclude

    else:

        transclude

    use navigation

    textbutton _("Повернуться"):
        style "return_button"

        action Return()

    label title

    if main_menu:
        key "game_menu" action ShowMenu("main_menu")

style game_menu_outer_frame is empty
style game_menu_navigation_frame is empty
style game_menu_content_frame is empty
style game_menu_viewport is gui_viewport
style game_menu_side is gui_side
style game_menu_scrollbar is gui_vscrollbar

style game_menu_label is gui_label
style game_menu_label_text is gui_label_text

style return_button is navigation_button
style return_button_text is navigation_button_text

style game_menu_outer_frame:
    bottom_padding 45
    top_padding 180

    background "gui/overlay/game_menu.png"

style game_menu_navigation_frame:
    xsize 420
    yfill True

```

```

style game_menu_content_frame:
    left_margin 60
    right_margin 30
    top_margin 15

style game_menu_viewport:
    xsize 1380

style game_menu_vscrollbar:
    unscrollable gui.unscrollable

style game_menu_side:
    spacing 15

style game_menu_label:
    xpos 75
    ysize 180

style game_menu_label_text:
    size gui.title_text_size
    color gui.accent_color
    yalign 0.5

style return_button:
    xpos gui.navigation_xpos
    yalign 1.0
    yoffset -45

##                                     Про                                     екран
#####
###
##
## Цей екран містить інформацію про авторство та авторські права
щодо гри та
## Ren'Py.
##
## У цьому екрані немає нічого особливого, тому він також служить
прикладом
## того, як створити власний екран.

screen about():

    tag menu

    ## Цей оператор використання включає екран game_menu всередині
цього. Потім
    ## дочірній елемент vbox включається в область перегляду
всередині екрана
    ## game_menu.
    use game_menu(_("Про гру"), scroll="viewport"):

```

```

style_prefix "about"

vbox:

    label "[config.name!t]"
    text _("Версія [config.version!t]\n")

    ## gui.about зазвичай встановлюється в options.rpy.
    if gui.about:
        text "[gui.about!t]\n"

        text _("Створено за допомогою
{a=https://www.renpy.org/}Ren'Py{/a}
[renpy.version_only].\n\n[renpy.license!t]")

style about_label is gui_label
style about_label_text is gui_label_text
style about_text is gui_text

style about_label_text:
    size gui.label_text_size

## Екрани завантаження та збереження
#####
##
## Ці екрани дозволяють гравцеві зберегти гру та завантажити її
знову. Оскільки
## вони мають майже все спільне, обидва реалізовані в термінах
третього екрана,
## file_slots.
##
## https://www.renpy.org/doc/html/screen_special.html#save
https://
## www.renpy.org/doc/html/screen_special.html#load

screen save():

    tag menu

    use file_slots(_("Зберегти"))

screen load():

    tag menu

    use file_slots(_("Завантажити"))

screen file_slots(title):

```

```

    default                                page_name_value                                =
FilePageNameInputValue(pattern=_("Сторінка {}"), auto=_("Авто
збереження"), quick=_("Швидкі збереження"))

    use game_menu(title):

        fixed:

            ## Це гарантує, що введення отримає подію enter
раніше, ніж будь-яка
            ## кнопка.
            order_reverse True

            ## Назва сторінки, яку можна редагувати, натиснувши
кнопку.
            button:
                style "page_label"

                key_events True
                xalign 0.5
                action page_name_value.Toggle()

            input:
                style "page_label_text"
                value page_name_value

            ## Сітка слотів для файлів.
            grid gui.file_slot_cols gui.file_slot_rows:
                style_prefix "slot"

                xalign 0.5
                yalign 0.5

                spacing gui.slot_spacing

                for i in range(gui.file_slot_cols *
gui.file_slot_rows):

                    $ slot = i + 1

                    button:
                        action FileAction(slot)

                        has vbox

                        add FileScreenshot(slot) xalign 0.5

                        text FileTime(slot,
format=_("{#file_time}%A, %B %d %Y, %H:%M"), empty=_("порожній
слот")):
                            style "slot_time_text"

                        text FileSaveName(slot):

```



```

        style "slot_name_text"

        key "save_delete" action FileDelete(slot)

    ## Кнопки для доступу до інших сторінок.
    hbox:
        style_prefix "page"

        xalign 0.5
        yalign 1.0

        spacing gui.page_spacing

        textbutton _("<") action FilePagePrevious()

        if config.has_autosave:
            textbutton _("#{auto_page}A") action
FilePage("auto")

        if config.has_quicksave:
            textbutton _("#{quick_page}III") action
FilePage("quick")

        ## діапазон(1, 10) дає числа від 1 до 9.
        for page in range(1, 10):
            textbutton "[page]" action FilePage(page)

        textbutton _(">") action FilePageNext()

style page_label is gui_label
style page_label_text is gui_label_text
style page_button is gui_button
style page_button_text is gui_button_text

style slot_button is gui_button
style slot_button_text is gui_button_text
style slot_time_text is slot_button_text
style slot_name_text is slot_button_text

style page_label:
    xpadding 75
    ypadding 5

style page_label_text:
    text_align 0.5
    layout "subtitle"
    hover_color gui.hover_color

style page_button:
    properties gui.button_properties("page_button")

style page_button_text:

```

```

        properties gui.button_text_properties("page_button")

style slot_button:
    properties gui.button_properties("slot_button")

style slot_button_text:
    properties gui.button_text_properties("slot_button")

##                               Екран                               налаштувань
#####
##
## Екран налаштувань дозволяє гравцеві налаштувати гру на свій
смак.
##
## https://www.renpy.org/doc/html/screen\_special.html#preferences

screen preferences():

    tag menu

    use game_menu(_("Налаштування"), scroll="viewport"):

        vbox:

            hbox:
                box_wrap True

                if renpy.variant("pc") or renpy.variant("web"):

                    vbox:
                        style_prefix "radio"
                        label _("Дисплей")
                        textbutton _("Вікно") action
Preference("display", "window")
                        textbutton _("На весь екран") action
Preference("display", "fullscreen")

                    vbox:
                        style_prefix "check"
                        label _("Пропустити")
                        textbutton _("Невидимий текст") action
Preference("skip", "toggle")
                        textbutton _("Після виборів") action
Preference("after choices", "toggle")
                        textbutton _("Переходи") action
InvertSelected(Preference("transitions", "toggle"))

            ## Сюди можна долучити додаткові вікна vbox типу
"radio_pref"
            ## або "check_pref", щоб долучити додаткові
параметри, визначені
            ## творцем.

```

```

null height (4 * gui.pref_spacing)

hbox:
    style_prefix "slider"
    box_wrap True

    vbox:

        label _("Швидкість тексту")

        bar value Preference("text speed")

        label _("Час авто-переадресації")

        bar value Preference("auto-forward time")

    vbox:

        if config.has_music:
            label _("Гучність музики")

            hbox:
                bar value Preference("music volume")

        if config.has_sound:
            label _("Гучність звуку")

            hbox:
                bar value Preference("sound volume")

                if config.sample_sound:
                    textbutton _("Тест") action
Play("sound", config.sample_sound)

            if config.has_voice:
                label _("Гучність голосу")

                hbox:
                    bar value Preference("voice volume")

                    if config.sample_voice:
                        textbutton _("Тест") action
Play("voice", config.sample_voice)

        if config.has_music or config.has_sound or
config.has_voice:
            null height gui.pref_spacing

            textbutton _("Без звуку"):

```

```

        action      Preference("all      mute",
"toggle")
        style "mute_all_button"

style pref_label is gui_label
style pref_label_text is gui_label_text
style pref_vbox is vbox

style radio_label is pref_label
style radio_label_text is pref_label_text
style radio_button is gui_button
style radio_button_text is gui_button_text
style radio_vbox is pref_vbox

style check_label is pref_label
style check_label_text is pref_label_text
style check_button is gui_button
style check_button_text is gui_button_text
style check_vbox is pref_vbox

style slider_label is pref_label
style slider_label_text is pref_label_text
style slider_slider is gui_slider
style slider_button is gui_button
style slider_button_text is gui_button_text
style slider_pref_vbox is pref_vbox

style mute_all_button is check_button
style mute_all_button_text is check_button_text

style pref_label:
    top_margin gui.pref_spacing
    bottom_margin 3

style pref_label_text:
    yalign 1.0

style pref_vbox:
    xsize 338

style radio_vbox:
    spacing gui.pref_button_spacing

style radio_button:
    properties gui.button_properties("radio_button")
    foreground "gui/button/radio_[prefix_]foreground.png"

style radio_button_text:
    properties gui.button_text_properties("radio_button")

style check_vbox:
    spacing gui.pref_button_spacing

```

```

style check_button:
    properties gui.button_properties("check_button")
    foreground "gui/button/check_[prefix_]foreground.png"

style check_button_text:
    properties gui.button_text_properties("check_button")

style slider_slider:
    xsize 525

style slider_button:
    properties gui.button_properties("slider_button")
    yalign 0.5
    left_margin 15

style slider_button_text:
    properties gui.button_text_properties("slider_button")

style slider_vbox:
    xsize 675

##                                     Екран                                     історії
#####
##
## Це екран, на якому гравцеві відображається історія діалогів.
Хоча в цьому
## екрані немає нічого особливого, він має доступ до історії
діалогів, що
## зберігається в _history_list.
##
## https://www.renpy.org/doc/html/history.html

screen history():

    tag menu

    ## Уникайте прогнозування цього екрана, оскільки він може бути
    дуже великим.
    predict False

    use      game_menu(_("Історія"),      scroll=("vpgrid"      if
gui.history_height else "viewport"), yinitial=1.0):

        style_prefix "history"

        for h in _history_list:

            window:

                ## Це правильно розкладає речі, якщо
history_height дорівнює

```

```

## None.
has fixed:
    yfit True

if h.who:

    label h.who:
        style "history_name"
        substitute False

        ## Взяти колір тексту що від персонажа,
якщо
        ## встановлено.
        if "color" in h.who_args:
            text_color h.who_args["color"]

    $ what = renpy.filter_text_tags(h.what,
allow=gui.history_allow_tags)
    text what:
        substitute False

    if not _history_list:
        label _("Історія діалогу порожня.")

## Це визначає теги, які можна відображати на екрані історії.
define gui.history_allow_tags = { "alt", "noalt", "rt", "rb",
"art" }

style history_window is empty

style history_name is gui_label
style history_name_text is gui_label_text
style history_text is gui_text

style history_label is gui_label
style history_label_text is gui_label_text

style history_window:
    xfill True
    ysize gui.history_height

style history_name:
    xpos gui.history_name_xpos
    xanchor gui.history_name_xalign
    ypos gui.history_name_ypos
    xsize gui.history_name_width

style history_name_text:
    min_width gui.history_name_width
    text_align gui.history_name_xalign

```

```

style history_text:
    xpos gui.history_text_xpos
    ypos gui.history_text_ypos
    xanchor gui.history_text_xalign
    xsize gui.history_text_width
    min_width gui.history_text_width
    text_align gui.history_text_xalign
    layout ("subtitle" if gui.history_text_xalign else "tex")

style history_label:
    xfill True

style history_label_text:
    xalign 0.5

##                                     Екран                                     довідки
#####
##
## Екран, що надає інформацію про прив'язку клавіш і миші. Він
використовує інші
## екрани (keyboard_help, mouse_help і gamepad_help), щоб
відобразити фактичну
## довідку.

screen help():

    tag menu

    default device = "keyboard"

    use game_menu(_("Довідка"), scroll="viewport"):

        style_prefix "help"

        vbox:
            spacing 23

            hbox:

                textbutton          _("Клавіатура")          action
                SetScreenVariable("device", "keyboard")
                textbutton          _("Мишка")                action
                SetScreenVariable("device", "mouse")

                if GamepadExists():
                    textbutton          _("Геймпад")          action
                    SetScreenVariable("device", "gamepad")

                if device == "keyboard":
                    use keyboard_help
                elif device == "mouse":

```

```

        use mouse_help
    elif device == "gamepad":
        use gamepad_help

screen keyboard_help():

    hbox:
        label _("Enter")
        text _("Розвиває діалог і активує інтерфейс.")

    hbox:
        label _("Space")
        text _("Просуває діалог без вибору варіантів.")

    hbox:
        label _("Клавіші стрілок")
        text _("Навігація по інтерфейсу.")

    hbox:
        label _("Escape")
        text _("Доступ до меню гри.")

    hbox:
        label _("Ctrl")
        text _("Пропускає діалог при утриманні.")

    hbox:
        label _("Tab")
        text _("Вмикає пропуск діалогу.")

    hbox:
        label _("Page Up")
        text _("Відкат до попереднього діалогу.")

    hbox:
        label _("Page Down")
        text _("Перехід до наступного діалогу.")

    hbox:
        label "H"
        text _("Приховує інтерфейс користувача.")

    hbox:
        label "S"
        text _("Робить скріншот.")

    hbox:
        label "V"
        text _("Вмикає допоміжне
{a=https://www.renpy.org/l/voicing}самоозвучення{/a}.").")

    hbox:

```



```
label "Shift+A"  
text _("Відкриває меню спеціальних можливостей.")
```

```
screen mouse_help():
```

```
hbox:  
    label _("Лівий клік")  
    text _("Розвиває діалог і активує інтерфейс.")  
  
hbox:  
    label _("Середній клік")  
    text _("Приховує інтерфейс користувача.")  
  
hbox:  
    label _("Правий клік")  
    text _("Доступ до меню гри.")  
  
hbox:  
    label _("Колесо миші вгору\nбік повернення назад")  
    text _("Відкат до попереднього діалогу.")  
  
hbox:  
    label _("Колесо миші вниз")  
    text _("Перехід до наступного діалогу.")
```

```
screen gamepad_help():
```

```
hbox:  
    label _("Правий тригер\nA/нижня кнопка")  
    text _("Розвиває діалог і активує інтерфейс.")  
  
hbox:  
    label _("Лівий тригер\nЛіве плече")  
    text _("Відкат до попереднього діалогу.")  
  
hbox:  
    label _("Праве плече")  
    text _("Перехід до наступного діалогу.")  
  
hbox:  
    label _("D-Pad, стіки")  
    text _("Навігація по інтерфейсу.")  
  
hbox:  
    label _("Старт, Гайд")  
    text _("Доступ до меню гри.")  
  
hbox:  
    label _("Кнопка Y/Тор")  
    text _("Приховує інтерфейс користувача.")
```

```

textbutton _("Калібрувати") action GamepadCalibrate()

style help_button is gui_button
style help_button_text is gui_button_text
style help_label is gui_label
style help_label_text is gui_label_text
style help_text is gui_text

style help_button:
    properties gui.button_properties("help_button")
    xmargin 12

style help_button_text:
    properties gui.button_text_properties("help_button")

style help_label:
    xsize 375
    right_padding 30

style help_label_text:
    size gui.text_size
    xalign 1.0
    text_align 1.0

#####
#####
## Додаткові екрани
#####
#####

##                               Екран                               підтвердження
#####
##
## Екран підтвердження викликається, коли Ren'Py хоче поставити
гравцеві
## запитання «так» або «ні».
##
## https://www.renpy.org/doc/html/screen\_special.html#confirm

screen confirm(message, yes_action, no_action):

    ## Переконайтеся, що інші екрани не отримують введення, поки
    відображається
    ## цей екран.
    modal True

    zorder 200

```

```

style_prefix "confirm"

add "gui/overlay/confirm.png"

frame:

    vbox:
        xalign .5
        yalign .5
        spacing 45

        label _(message):
            style "confirm_prompt"
            xalign 0.5

        hbox:
            xalign 0.5
            spacing 150

            textbutton _("Так") action yes_action
            textbutton _("Ні") action no_action

    ## Клацніть правою кнопкою миші та вийдіть із відповіді "но".
    key "game_menu" action no_action

style confirm_frame is gui_frame
style confirm_prompt is gui_prompt
style confirm_prompt_text is gui_prompt_text
style confirm_button is gui_medium_button
style confirm_button_text is gui_medium_button_text

style confirm_frame:
    background Frame([ "gui/confirm_frame.png", "gui/frame.png"],
gui.confirm_frame_borders, tile=gui.frame_tile)
    padding gui.confirm_frame_borders.padding
    xalign .5
    yalign .5

style confirm_prompt_text:
    text_align 0.5
    layout "subtitle"

style confirm_button:
    properties gui.button_properties("confirm_button")

style confirm_button_text:
    properties gui.button_text_properties("confirm_button")

##                               Екран                               індикатора                               пропуску
#####
##

```

```

## Екран skip_indicator відображається, щоб вказати, що триває пропуск.
##
##      https://www.renpy.org/doc/html/screen_special.html#skip-indicator

screen skip_indicator():

    zorder 100
    style_prefix "skip"

    frame:

        hbox:
            spacing 9

            text _("Пропуск")

            text "▶" at delayed_blink(0.0, 1.0) style "skip_triangle"
            text "▶" at delayed_blink(0.2, 1.0) style "skip_triangle"
            text "▶" at delayed_blink(0.4, 1.0) style "skip_triangle"

## Це перетворення використовується для миготіння стрілок одна за одною.
transform delayed_blink(delay, cycle):
    alpha .5

    pause delay

    block:
        linear .2 alpha 1.0
        pause .2
        linear .2 alpha 0.5
        pause (cycle - .4)
        repeat

style skip_frame is empty
style skip_text is gui_text
style skip_triangle is skip_text

style skip_frame:
    ypos gui.skip_ypos
    background Frame("gui/skip.png", gui.skip_frame_borders,
tile=gui.frame_tile)
    padding gui.skip_frame_borders.padding

style skip_text:
    size gui.notify_text_size

```

```

style skip_triangle:
    ## Ми повинні використовувати шрифт із гліфом МАЛЕНЬКИЙ ЧОРНИЙ
    ТРИКУТНИК
    ## ВПРАВО.
    font "DejaVuSans.ttf"

##                                     Екран                                     сповіщень
#####
##
## Екран сповіщень використовується, щоб показати гравцеві
повідомлення.
## (Наприклад, коли гра швидко збережена або зроблено знімок
екрана.)
##
##      https://www.renpy.org/doc/html/screen\_special.html#notify-
screen

screen notify(message):

    zorder 100
    style_prefix "notify"

    frame at notify_appear:
        text "[message!tq]"

    timer 3.25 action Hide('notify')

transform notify_appear:
    on show:
        alpha 0
        linear .25 alpha 1.0
    on hide:
        linear .5 alpha 0.0

style notify_frame is empty
style notify_text is gui_text

style notify_frame:
    ypos gui.notify_ypos

    background Frame("gui/notify.png", gui.notify_frame_borders,
tile=gui.frame_tile)
    padding gui.notify_frame_borders.padding

style notify_text:
    properties gui.text_properties("notify")

```

```

##                                     Екран                                     NVL
#####
###
##
## Відображає діалог у vpggrid або vbox.
##
## https://www.renpy.org/doc/html/screen\_special.html#nvl

```

```

screen nvl(dialogue, items=None):

```

```

    window:

```

```

        style "nvl_window"

```

```

    has vbox:

```

```

        spacing gui.nvl_spacing

```

```

    ## Відображає діалог у vpggrid або vbox.

```

```

    if gui.nvl_height:

```

```

        vpggrid:

```

```

            cols 1

```

```

            yinitial 1.0

```

```

            use nvl_dialogue(dialogue)

```

```

    else:

```

```

        use nvl_dialogue(dialogue)

```

```

    ## Відображає меню, якщо є. Меню може відобразитися
    неправильно, якщо

```

```

    ## для config.narrator_menu встановлено значення True.

```

```

    for i in items:

```

```

        textbutton i.caption:

```

```

            action i.action

```

```

            style "nvl_button"

```

```

    add SideImage() xalign 0.0 yalign 1.0

```

```

screen nvl_dialogue(dialogue):

```

```

    for d in dialogue:

```

```

        window:

```

```

            id d.window_id

```

```

        fixed:

```

```

            yfit gui.nvl_height is None

```

```

            if d.who is not None:

```

```

        text d.who:
            id d.who_id

        text d.what:
            id d.what_id

## Це визначає максимальну кількість записів у режимі NVL, які
## можуть бути
## відображені одночасно.
define config.nvl_list_length = gui.nvl_list_length

style nvl_window is default
style nvl_entry is default

style nvl_label is say_label
style nvl_dialogue is say_dialogue

style nvl_button is button
style nvl_button_text is button_text

style nvl_window:
    xfill True
    yfill True

    background "gui/nvl.png"
    padding gui.nvl_borders.padding

style nvl_entry:
    xfill True
    ysize gui.nvl_height

style nvl_label:
    xpos gui.nvl_name_xpos
    xanchor gui.nvl_name_xalign
    ypos gui.nvl_name_ypos
    yanchor 0.0
    xsize gui.nvl_name_width
    min_width gui.nvl_name_width
    text_align gui.nvl_name_xalign

style nvl_dialogue:
    xpos gui.nvl_text_xpos
    xanchor gui.nvl_text_xalign
    ypos gui.nvl_text_ypos
    xsize gui.nvl_text_width
    min_width gui.nvl_text_width
    text_align gui.nvl_text_xalign
    layout ("subtitle" if gui.nvl_text_xalign else "tex")

style nvl_thought:
    xpos gui.nvl_thought_xpos

```

```

xanchor gui.nvl_thought_xalign
ypos gui.nvl_thought_ypos
xsize gui.nvl_thought_width
min_width gui.nvl_thought_width
text_align gui.nvl_thought_xalign
layout ("subtitle" if gui.nvl_text_xalign else "tex")

style nvl_button:
    properties gui.button_properties("nvl_button")
    xpos gui.nvl_button_xpos
    xanchor gui.nvl_button_xalign

style nvl_button_text:
    properties gui.button_text_properties("nvl_button")

#####
#####
## Мобільні варіанти
#####
#####

style pref_vbox:
    variant "medium"
    xsize 675

## Оскільки миша може бути відсутня, ми замінюємо швидке меню
версією, яка
## використовує менше та більші кнопки, які легше торкатися.
screen quick_menu():
    variant "touch"

    zorder 100

    if quick_menu:
        hbox:
            style_prefix "quick"

            xalign 0.5
            yalign 1.0

            textbutton _("Назад") action Rollback()
            textbutton _("Пропустити") action Skip() alternate
            Skip(fast=True, confirm=True)
            textbutton _("Авто") action Preference("auto-
forward", "toggle")
            textbutton _("Меню") action ShowMenu()

style window:
    variant "small"

```



```

        background "gui/phone/textbox.png"

style radio_button:
    variant "small"
    foreground "gui/phone/button/radio_[prefix_]foreground.png"

style check_button:
    variant "small"
    foreground "gui/phone/button/check_[prefix_]foreground.png"

style nvl_window:
    variant "small"
    background "gui/phone/nvl.png"

style main_menu_frame:
    variant "small"
    background "gui/phone/overlay/main_menu.png"

style game_menu_outer_frame:
    variant "small"
    background "gui/phone/overlay/game_menu.png"

style game_menu_navigation_frame:
    variant "small"
    xsize 510

style game_menu_content_frame:
    variant "small"
    top_margin 0

style pref_vbox:
    variant "small"
    xsize 600

style bar:
    variant "small"
    ysize gui.bar_size
    left_bar    Frame("gui/phone/bar/left.png",    gui.bar_borders,
tile=gui.bar_tile)
    right_bar   Frame("gui/phone/bar/right.png",   gui.bar_borders,
tile=gui.bar_tile)

style vbar:
    variant "small"
    xsize gui.bar_size
    top_bar     Frame("gui/phone/bar/top.png",     gui.vbar_borders,
tile=gui.bar_tile)
    bottom_bar  Frame("gui/phone/bar/bottom.png",  gui.vbar_borders, tile=gui.bar_tile)

style scrollbar:
    variant "small"
    ysize gui.scrollbar_size

```

```

    base_bar
Frame("gui/phone/scrollbar/horizontal_[prefix_]bar.png",
gui.scrollbar_borders, tile=gui.scrollbar_tile)
    thumb
Frame("gui/phone/scrollbar/horizontal_[prefix_]thumb.png",
gui.scrollbar_borders, tile=gui.scrollbar_tile)

style vscrollbar:
    variant "small"
    xsize gui.scrollbar_size
    base_bar
Frame("gui/phone/scrollbar/vertical_[prefix_]bar.png",
gui.vscrollbar_borders, tile=gui.scrollbar_tile)
    thumb
Frame("gui/phone/scrollbar/vertical_[prefix_]thumb.png",
gui.vscrollbar_borders, tile=gui.scrollbar_tile)

style slider:
    variant "small"
    ysize gui.slider_size
    base_bar
Frame("gui/phone/slider/horizontal_[prefix_]bar.png",
gui.slider_borders, tile=gui.slider_tile)
    thumb "gui/phone/slider/horizontal_[prefix_]thumb.png"

style vslider:
    variant "small"
    xsize gui.slider_size
    base_bar Frame("gui/phone/slider/vertical_[prefix_]bar.png",
gui.vslider_borders, tile=gui.slider_tile)
    thumb "gui/phone/slider/vertical_[prefix_]thumb.png"

style slider_vbox:
    variant "small"
    xsize None

style slider_slider:
    variant "small"
    xsize 900

```