

ЛАБОРАТОРНА РОБОТА 3

Аналіз і формалізація вимог до програмного продукту. Розробка технічного завдання на створення програмного продукту

Мета роботи: навчитися аналізувати й формалізувати вимоги замовника з використанням UML, розраховувати витрати на створення програмного продукту, виконувати планування робіт, розробляти та оформлювати технічне завдання на створення програмного продукту відповідно до ДСТУ.

Завдання:

1. Вивчити теоретичні відомості.
2. Виконати аналіз і формалізацію вимог замовника на розробку програмного продукту відповідно до індивідуального завдання.
3. Розробити діаграму прецедентів використання й виконати опис прецедентів.
4. Виконати розрахунок витрат на створення програмного продукту.
5. Виконати планування робіт зі створення програмного продукту.
6. Розробити технічне завдання на створення програмного продукту.
7. Зробити висновки про вибір моделі створення програмного продукту.

Хід роботи:

1) Сформулюємо індивідуальне завдання:

Тема: розробка однокористувацької двовимірної гри "Тетріс" з використанням принципів ООП.

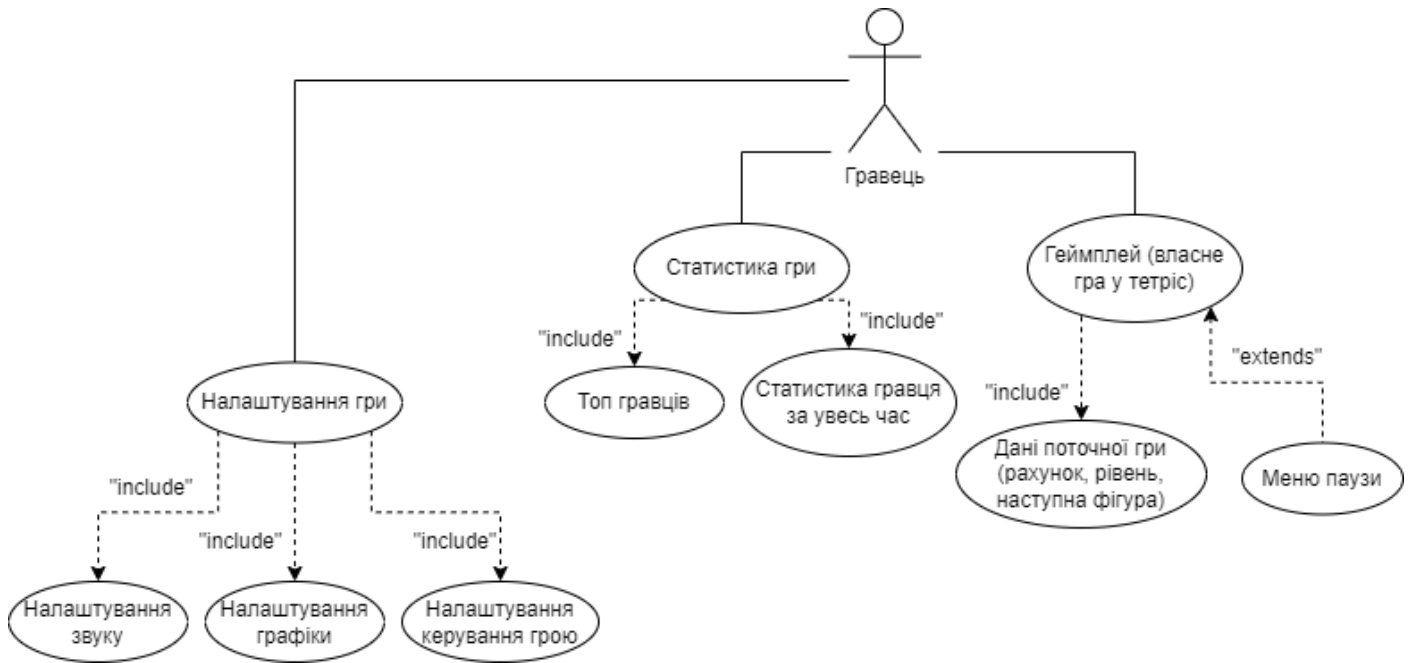
"Тетріс" є головоломкою, побудованою на використанні геометричних фігур "тетраміно" - різновиду поліміно, що складаються з чотирьох квадратів.

Випадкові фігурки тетраміно падають зверху в прямокутний стакан шириною 10 і висотою 20 клітин. У польоті гравець може повертати фігурку та рухати її по горизонталі. Також можна «скидати» фігурку, тобто прискорювати її падіння, коли вже вирішено, куди фігурка повинна впасти. Фігурка летить, поки не наткнеться на іншу фігурку або на дно склянки. Якщо при цьому заповнився горизонтальний ряд з 10 кліток, він пропадає і все, що вище нього, опускається на одну клітку.

У спеціальному полі гравець бачить фігурку, яка буде слідувати після поточної — ця підказка дозволяє планувати свої дії. Темп гри поступово збільшується. Назва гри походить від кількості клітин, з яких складається кожна фігура. Гра закінчується, коли нова фігурка не може поміститися в стакан. Гравець отримує бали за кожен фігурку,

тому його задача — заповнювати ряди, не заповнюючи саму склянку якомога довше, щоб таким чином отримати якомога більше балів.

2) Розробимо діаграму прецедентів:



Опис прецеденту «Статистика гри»

Основний виконавець – Гра Зацікавлені особи – Гравець
Передумови: <ul style="list-style-type: none">Наявність ігрових сесій (мінімум 1)Наявність створеного акаунту (логін, пароль)
Вхідні дані: <ul style="list-style-type: none">Дані акаунту
Основний успішний сценарій (основний процес): <ul style="list-style-type: none">Гра надає можливість переглянути кількість зіграних сесій, найкращий результат, час у грі, загальний рахунок, рівень гравця
Частота виконання: <ul style="list-style-type: none">для кожного гравця
Постумови (результати): <ul style="list-style-type: none">Отримання загальної статистики гравця
Вихідні дані: <ul style="list-style-type: none">Загальна статистика гравця

Опис прецеденту «Дані поточної гри»

Основний виконавець – Гра Зацікавлені особи – Гравець
Передумови: <ul style="list-style-type: none">Запуск поточної гри
Вхідні дані: <ul style="list-style-type: none">Дані поточної гри
Основний успішний сценарій (основний процес): <ul style="list-style-type: none">Гра надає можливість переглянути поточний рахунок у грі, рівень складності, кількість складених ліній, наступну фігуру (залежно від налаштувань)
Частота виконання: <ul style="list-style-type: none">для кожного гравця
Постумови (результати): <ul style="list-style-type: none">Отримання поточної статистики гравця
Вихідні дані: <ul style="list-style-type: none">Поточна статистика гравця

Опис прецеденту «Геймплей»

Основний виконавець – Гра Зацікавлені особи – Гравець
Передумови: <ul style="list-style-type: none">Запуск поточної гри
Вхідні дані: <ul style="list-style-type: none">Дані поточної гриДії гравця
Основний успішний сценарій (основний процес): <ul style="list-style-type: none">Гравець має можливість виконувати певні дії для отримання результатів у грі (рухати та повертати фігури, складати лінії)
Частота виконання: <ul style="list-style-type: none">для кожного гравця
Постумови (результати): <ul style="list-style-type: none">Гравець завершує гру самостійно або у результаті виконання умов гри (кінець гри) та отримує додаткову статистику у профілі
Вихідні дані: <ul style="list-style-type: none">Результуюча статистика гри (час, рівень, рахунок)

3) Виконаємо розрахунок витрат на створення програмного продукту:

Для визначення договірної ціни розраховуються поточні витрати на виконання програмного продукту. Ці витрати розраховуються як сума прямих та накладних витрат за весь період створення програми.

Поточні витрати на створення програмного продукту, пов'язані з використанням ПК для виконання даної роботи, розраховуються виходячи з загального часу $T_{\text{заг}}$, собівартості людино-години $C_{\text{люд}}$ та години експлуатації ПК $C_{\text{ПК}}$, а також вартості ПЗ, необхідного для розробки:

$$V_{\text{ПП}} = (C_{\text{люд}} + C_{\text{ПК}}) * T_{\text{заг}} + C_{\text{ПЗ}}$$

Почасова ставка $C_{\text{люд}} = 100$ грн/год.

Вартість години роботи ПК = 10 грн/год.

Вартість використаного ПЗ = 0 грн (Безкоштовна VS Community Edition 2019).

Розрахуємо загальний час розробки $T_{\text{заг}}$:

$T_{\text{заг}} = T_{\text{підг}} + T_{\text{код}} + T_{\text{зв}}$, де $T_{\text{підг}}$ – час аналізу замовлення, створення ТЗ, пошуку необхідної інформації та ресурсів, $T_{\text{код}}$ – час написання та відладки коду, $T_{\text{зв}}$ – час створення звітів з розробки ПП.

$$T_{\text{заг}} = 10 + 50 + 5 = 65 \text{ годин.}$$

Розрахуємо поточні витрати на розробку програмного продукту $V_{\text{ПП}}$:

$$V_{\text{ПП}} = (100 + 10) * 65 + 0 = 7150 \text{ грн.}$$

4) Сформулюємо технічне завдання:

Повна назва програмного продукту – «Tetris OOP»

Операційна система (платформа): Windows 7,8,10 x32, x64.

Виконуваний файл програми – TetrisOOP.exe.

Необхідне вільне місце на ПК – 20 МБ.

Оперативна пам'ять для запуску програми – 10 МБ.

Інтернет: тільки для оновлення топу гравців, достатньо швидкості у 5 Кб/с.

Дані користувачів будуть зберігатися локально на ПК у xml файлі, а також частково на Github, для відображення глобального топу гравців.

Підстави для розробки:

1.1 Документ, на підставі якого ведеться розробка

Робота ведеться на підставі завдання з дисципліни «Технологія створення програмних продуктів».

1.2 Організація, що затвердила цей документ, і дата його затвердження

Завдання затверджено викладачем кафедри КІТ Добряком Сергієм Костянтиновичем.

1.3 Найменування теми розробки

Найменування теми розробки - «Тетріс у ООП виконанні».

Призначення розробки:

Програмний продукт призначений перш за все для розваги, отримання позитивних емоцій від процесу гри, а також для розвитку когнітивних навичок у будь-якому віці. Також у грі присутня система користувачів та топів, що додає змагальну складову до гри, тобто користувач буде зацікавлений у досягненні певного рівня гри.

Вимоги до функціональних характеристик програмного продукту:

Програма повинна виконувати наступні функції:

- Реєструвати або авторизовувати гравця (Функція виконується у межах 1 с.)
- Надавати детальну статистику гравця (Функція виконується у межах 1 с.)
- Надавати глобальний топ гравців (Функція виконується у межах 1 с.)
- Запускати або встановлювати гру на паузу (Функція виконується у межах 1 с.)
- Розраховувати рахунок та визначати рівень у грі (Функція виконується у межах 1 с.)
- Змінювати налаштування згідно до заданих користувачем (Функція виконується у межах 1 с.)

Вимоги до надійності:

- Гра повинна надійно функціонувати, виконувати усі вказані команди та не призводити до проблем іншого програмного забезпечення;
- Програмний продукт повинен забезпечувати обробку невірних дій користувача та програмних конфліктів з видачею відповідних повідомлень.

Вимоги до інформаційної та програмної сумісності:

Продукт створюється за допомогою інтегрованого середовища розробки Visual Studio на мові C# 8.0 з використанням інтерфейсу Windows Forms на платформі Microsoft .NET 4.8.

Вимоги до програмної документації

Програмна документація повинна містити:

- керівництво користувача;
- робочий проект програми мовою UML;

Порядок контролю та приймання

Контроль коректності функціонування та придатності гри до експлуатації виконується розробником на стадіях тестування та пост-аналізу, а також замовником ПП на підставі приймальних тестів, що ним надаються.

Рішення про приймання в експлуатацію приймається замовником, судячи з відповідності його умовам та результатам тестування продукту.

Висновки: я навчився розраховувати витрати на створення програмного продукту, виконувати планування робіт та складати технічне завдання на створення програмного продукту.

Я вивчив теоретичні відомості, а також виконав аналіз і формалізацію вимог замовника на розробку гри Tetris у ООП виконанні.

Розробив діаграму прецедентів використання й виконав опис прецедентів додатку. Розрахував витрати на створення програмного продукту, у результаті отримав суму у 7150 грн, яку вважаю справедливою для цього проекту. Також написав детальне технічне завдання на створення програми, завдяки якому можливо у найменші строки розробити максимальну кількість функціоналу гри.