Звіт з практики. Гра «Diamond Rush»

Структура даних

Скорочення: екземпляри класу – е. к.

У програмі для різних цілей було використано масиви та колекції. Об’єкти, що створюються безпосередньо під час виконання програми, зберігаються в колекціях, а саме в ArrayList, перевагою якого є швидкодія під час довільного доступу до елементів, зручність видалення та додавання нових об’єктів до колекції (у порівнянні з масивами). Об’єкти, кількість яких відома заздалегідь і залишається незмінною в процесі виконання програми, розташовуються в масивах.

*Масиви:*

Кожен рівень гри – це двовимірний масив клітинок, е. к. Cell. Цей масив міститься в е.к. Level, який заповнюється в класі – карті, Maps, і використовується в багатьох класах програми. Level також має масив е. к. Checkpoint. Е. к. Checkpoint містить двовимірний масив е. к. Cell для відновлення певної ділянки карти та масив цілих чисел additionalSegment для виділення додаткової ділянки. Для створення інших об’єктів у лабіринті також використовуються масиви. Ламка стіна, е. к. BreakableWall, використовує масив точок Point для пошуку на карті інших таких самих стін. Скриня, е. к. Chest, містить масив картинок, за допомогою ітерації по якому відбувається анімація відкривання скрині. Також у скрині є масив об’єктів, які наслідуються від абстрактного класу ThingInChest і лежать в ній. Подвійні двері, е. к. DoubleDoor, містять масив змій, е. к. Snake, яких треба вбити, щоб подвійні двері відчинилися. Пастки, об’єкти, що імплементують інтерфейс Trap, мають масиви картинок, які використовуються залежно від направлення і стану об’єкту. Вогняна пастка, е. к. FireTrap, має також boolean масив станів вогню. У героя Boy є 16 масивів картинок, за допомогою ітерації по яким відбувається анімація руху хлопчика. Не тільки об’єкти в лабіринті, а й карта рівнів, е. к. MapPanel, містить масив кнопок-переходів на рівні, е. к. LevelUI, та масиви x і y рисочок між цими кнопками. У класі для збереження прогресу, ProgressStorage, використовуються масиви стрічок для зручної роботи з контентом файлу.

*Колекції ArrayList:*

Ламка стіна, е. к. BreakableWall, використовує колекції клітинок, е. к. Cell, для швидкого додавання і зберігання інших таких самих стін поблизу на карті. У е. к. Checkpoint є колекції предметів, що знаходяться на підконтрольній чекпоїнту території: колекція е. к. Chest та колекція пасток. При малюванні поточного рівня в е. к. PlayPanel створюється колекція секретних стін, е. к. SecretWall, для того, щоб малювати їх після промальовування інших об’єктів.

Опис методів та класів

* PlayPanel: клас наслідується від JPanel та імплементує KeyListener. Це ігрова панель, на якій малюються та взаємодіють всі об’єкти лабіринту. В класі реалізовані такі можливості: малювання об’єктів; рух хлопчика при натисканні на кнопки клавіатури одночасно по екрану, по карті та в масиві; переміщення видимої ділянки карти; рестарт, пауза та відновлення гри; оновлення зібраних хлопчиком речей в змінних та на верхній панелі, StatusBarPanel; віднімання у хлопчика енергії; застосування чекпоїнтів; закінчення рівня. Написані boolean методи перевірок типу itIsObject() для всіх об’єктів, що можуть знаходитись в клітинці, та деякі інші перевірки простору для каменів та хлопчика.
* ProgressStorage: клас для збереження прогресу в файл та його відновлення. Має приватний статичний файл Progress.txt, куди записуються дані по кожному рівню. Є статичні методи для отримання та оновлення контенту файлу; зміни прогресу та отримання даних по конкретному рівню; метод, щоб скинути прогрес.
* Boy: клас дозволяє створювати хлопчика – героя, який має поля xInArray та yInArray, проте в масив клітинок Cell не додається. В класі є Images хлопчика та їхні масиви, за допомогою ітерації по яким відбувається анімація його руху. У хлопчика є методи: ходити та бити молотком вверх, вниз, вліво, вправо; штовхати камінь та відкривати двері срібним або золотим ключем вліво, вправо; тримати камінь над головою, відкривати скриню.
* Stone: абстрактний клас імплементує інтерфейс Resettable для подальшої реалізації брили (каменя – пастки) та діаманта (нешкідливого каменя). Камінь має методи для падіння в різні сторони, поле змії типу Snake, якщо стоїть у неї на дорозі, метод checkSnake() для вбивання змії та власний таймер. В конструкторі Stone запускається таймер та додається слухач подій, який переміщує камінь як тільки для нього з’являється вільне місце. Прописано багато перевірок на колізії з іншими каменями при одночасному їх падінні.
* Rock: клас імплементує інтерфейс Trap та наслідується від абстрактного класу Stone, дозволяє створювати брилу. Має методи beShovenLeft() та beShovenRight() для штовхання брили хлопчиком.
* Harmless: інтерфейс для подальшої реалізації нешкідливих для героя об’єктів на карті.
* Diamond: клас імплементує інтерфейс Harmless та наслідується від абстрактного класу Stone. Дозволяє створювати фіолетові діаманти, які може збирати герой.
* Tumbleweed: клас імплементує інтерфейс Harmless і дозволяє створювати рослину перекотиполе, яка зникає, коли на неї наступити.
* Chest: клас імплементує інтерфейс Harmless і дозволяє створювати скрині. Метод openChest() викликається при натисканні на цифру 1 і запускає анімацію відкривання скрині і діставання з неї предметів.
* ThingInChest: абстрактний клас для подальшої реалізації об’єктів зі скрині. Має метод paintObject() і абстрактний метод disappear()для визначення дій при зникненні об’єкту з екрану.
* PurpleDiamond: клас наслідується від абстрактного класу ThingInChest і дозволяє створювати фіолетовий діамант, який можна знайти в скрині. Метод disappear() збільшує кількість фіолетових діамантів у героя.
* RedDiamond: клас наслідується від абстрактного класу ThingInChest і дозволяє створювати червоний діамант, який можна знайти в скрині. Метод disappear() збільшує кількість червоних діамантів у героя.
* SilverKey: клас наслідується від абстрактного класу ThingInChest і дозволяє створювати срібний ключ, який можна знайти в скрині. Метод disappear() збільшує кількість срібних ключів у героя.
* GoldKey: клас наслідується від абстрактного класу ThingInChest і дозволяє створювати золотий ключ, який можна знайти в скрині. Метод disappear() збільшує кількість золотих ключів у героя.
* Energy: клас наслідується від абстрактного класу ThingInChest і дозволяє створювати енергію, яку можна знайти в скрині. Метод disappear() збільшує кількість енергії у героя.

**Розподіл ролей**  
*Джосан Володимир:*

* Створення ієрархії об’єктів на карті.
* Створення пустої карти (без пасток та інших об’єктів).
* Створення таких пасток, як “Скорпіон”, “Змія”, ”Вогняна пастка”, їхній рух.
* Поведінка об’єкту “Ламка стіна” .
* 4 рівень гри.

*Сітьков Ілля:*

* Створення інтерфейсу гри: початкового меню, карти рівнів, панелі гри, меню паузи.
* Пересування персонажа по карті.
* Створення таких об’єктів для карти, як «Двері з ключем», «Двері, що відчиняються натисканням панелі», «Автоматичні двері».
* 2, 5 рівні гри.

*Матвієнко Ірина:*

* Створення персонажа, його рух.
* Створення об’єктів «Камінь» та «Діамант», їхній рух та колізії.
* Розміщення рослин «Перекотиполе» на карті.
* Реалізація об’єкту «Скриня» та 5 речей всередині.
* Boolean методи перевірок типу itIsObject() для всіх об’єктів на карті.
* 3, 5 рівні гри.
* Збереження прогресу в файл.

Спільні задачі:

* Пошук та виправлення помилок.
* Пошук картинок для об’єктів на карті.
* Обробка картинок у програмі Photoshop.
* Створення 1 рівня гри.
* Написання звіту.

**Опис гри**

**Diamond Rush**. Наша версія гри відрізняється від оригінальної новою локацією із назвою “Єгипетська піраміда”, а також новими об’єктами та пастками, з якими персонаж повинен або може взаємодіяти. Більше того, на відміну від оригінального Diamond Rush`у, у нашій грі можна зібрати артефакт, за умови ідеального проходження усіх рівнів (детальніше про артефакт та рівні читайте далі).

П’ять рівнів складності гри – це п’ять карт, які користувач може пройти. Кожна наступна карта складніша за попередню.

Щоб пройти рівень, користувач повинен знайти вихід із лабіринту. Проте це тільки мінімальна умова проходження рівня. Щоб отримати частинку артефакту, персонаж повинен зібрати усі діаманти, які розкидані по карті.

У чому ж складність цієї гри, спитаєте Ви? На кожному рівні є пастки, які заважатимуть пересуватися лабіринтом. Якщо гравець був неакуратний та потрапив в одну з них, його герой втрачає енергію, необхідну для продовження гри. Також під час проходження карти потрібно бути дуже обачним, щоб випадково себе не заблокувати. Адже тоді єдиним виходом буде скористатися checkpoint’ом, який коштує багато енергії. Отже, як можна було здогадатись, енергія досить цінний ресурс, саме тому потрібно завжди ретельно обмірковувати наступні кроки. Окрім цього, треба бути уважним, адже в лабіринті є замасковані переходи, що ведуть до таємних кімнат. Саме там найчастіше заховані найбільш цінні діаманти – червоні. Тепер усе здається не таким простим, чи не так?