

Министерство науки и образования Украины
Харьковский национальный университет имени В.Н. Каразина
Факультет компьютерных наук

КУРСОВА РОБОТА
з дисципліни «Об'єктно-орієнтоване програмування»

Тема «Гра Google T-Rex»

Выполнили:
Кириченко О.С., студент гр. КУ-21

Перевірів:
доцент кафедри искусственного интеллекта и программного
обеспечения
Владимирова М.В.

Харьков 2020

Зміст

ВСТУП.....	3
РОЗДІЛ 1 КОНЦЕНЦІЯ.....	4
1.1 Загальна характеристика програмного продукту.....	4
1.2 Основні можливості та обмеження.....	4
РОЗДІЛ 2 АНАЛІЗ.....	5
2.1 Механіка руху.....	5
2.2 Механіка зіткнення.....	5
2.3 Механіка генерації перешкоди.....	6
2.4 Механіка перезавантаження гри.....	6
2.5 Механіка ігри.....	6
РОЗДІЛ 3 ПРОЕКТУВАННЯ.....	7
3.1 Загальний опис архітектури.....	7
РОЗДІЛ 4 ТЕСТУВАННЯ.....	9
РОЗДІЛ 5 ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАННЯ.....	10
ВИСНОВОК.....	12
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	13
ДОДАТКИ.....	14

ВСТУП

Тема даної курсової роботи гра “Google T-Rex” була вибрана мною для того щоб продемонструвати як за допомогою мови програмування Java та графічної бібліотеки JavaFX можна робити ігри. Я обрав для цього саме “Google T-Rex” тому що це напевне найпопулярніша гра в стилі примітивного ранера яку просто реалізувати. Всю програму я писав сам з нуля а спрайти взяв з офіційного “Google T-Rex”. Під час виконання курсової я користувався середовищем програмування для мови Java IntelliJ IDEA. Пояснювальна записка складається з концепції, аналізу, проектування, тестування . Більшість труднощів виникала саме в тому як представити ігрового персонажа та ворогів як перевіряти столкновения. В процесі розробки прийшлося використовувати стандартний прийом для таких ігор а саме що насправді персонаж стоїть на місці а середовище рухається на нього. Це вимушений метод прийшлося використовувати через те що по своїй суті ця гра не має кінця це означає що не можна заздалегідь побудувати рівень і двигати по цьому рівню персонажа. Саме через це і було вирішено використовувати таку методику.

РОЗДІЛ 1 КОНЦЕНЦІЯ

1.1 Загальна характеристика програмного продукту

Моя програмна реалізація гри “*Google T-Rex*” відрізняється від оригіналу можливістю обирати колір гри на вибір представляється один з 9 можливих кольорів. Також я додав приємну музику на фон щоб розслабити гравця.

1.2 Основні можливості та обмеження

Можливості:

1. Наповнення рівня відбувається випадково але так щоб гравець міг мій долати перешкоди.
2. Можливість обирати колір рівняння.
3. Мелодія на фоні.
4. Підрахунок кількості пройдених перешкод.
5. Можливість почати з початку після смерті.
6. Уменшений хітбокс для комфортнішої гри.
7. Збільшення швидкості перешкод за кількістю пройдених перешкод

Обмеження :

1. Не можна змінювати розмір вікна(вікно unresizable) це обумовлено тим що перешкоди генеруються по координатам відносно верхнього лівого кута тому зміна розміру ігрового поля може призвести до того що частина ігрового контенту буде прихована від гравця самим гравцем або буде видно те що розробник хотів приховати від гравця
2. Для запуску гри потрібно мати java не менше 8 версії так як використовується JavaFX

Всі файли для роботи програми ідуть в jar файлі наприклад: музичні треки в форматі wav зберігаються в середині jar як і спрайти в форматі png.

РОЗДІЛ 2 АНАЛІЗ

В цьому розділі будуть описані основні труднощі з якими я зіштовхнувся під час створення програмної реалізації гри “*Google T-Rex*” .

2.1 Механіка руху

Одна з самих важких на мою думку проблем це була зв'язана з тим як рухати персонажа. Було рівно 2 варіанти або закріпити на ньому камеру і рухати його, або рухати всі всі об'єкти а персонажа залишати не рухомим. По ходу написання роботи більш оптимальним був метод переміщення всього відносно персонажа. Хоч на перший погляд здається що переміщати тільки персонажа вигідніше тому що менше роботи знач гра буде легша для комп'ютера але чим довше іграв тим більше пам'яті з'їдала б гра. Використовуючи метод руху всього відносно відносно персонажа потребує чуть більше ресурсів але є стабільним по пам'яті.

2.2 Механіка зіткнення

Другою по важливості і по важкості було саме механіка зіткнення персонажа з ворогами. Вся сіль була в тому що я використовував для представлення персонажа і оточення нащадків класу *Rape* а він приймає форму найбільшого об'єкта в формі, а як ви розумієте що якщо гранями об'єкта буде квадрат розміром з персонажа а персонаж то не не квадратний тобто для комфортної гри потрібно щоб хітбокс персонажа був менший чим персонаж для цього я просто перевіряю чуть менший квадрат по кординатам чуть меншим і розмір квадрата чуть менший.

2.3 Механіка генерації перешкоди

Генерація перешкод досить зрозуміла як на мій погляд ми повинні згенерувати перешкоду так щоб її можливо було подолати. Так як з кількістю пройдених перешкод швидкість їх зростає, а висота стрибка персонажа залишається не змінною то зрозуміло що розстояніє між перешкодами це функція яка залежить від швидкості перешкод

2.4 Механіка перезапуску гри

Механіка перезапуску досить проста просто очищаємо екран та заново створюємо всі об'єкти заново.

2.5 Механіка гри

Вся гра запускається в таймері, тобто все що на ігровому екрані оновлюється 60 раз за секунду значення в 60 кадрів було взято для плавності проісходящого на ігровому екрані.

РОЗДІЛ 3 ПРОЕКТУВАННЯ

В цьому розділі я опишу як відбувалася проектування моєї реалізації “*Google T-Rex*”. Все починалось з простого : поставити якийсь прямокутник на ігрове поле та змусити його рухатися. Першою спробою було змусити саме персонажа рухатися по ігровому полю, а не щоб рухалось ігрове поле. По те як реалізувати це я пішов в ютуб наткнувся на відео де автор розбирав як за допомогою JavaFX робити 2D ігри на прикладі Flappy Bird[1] . Але його реалізація мені не підійшла через те що з часом потребувалося більше пам’яті. Тому використав метод переміщення сцени а не персонажа. Коли я добився того що по ігровому полі рухались квадрати різної величини приступив до того щоб в грі була можливість програти. Спочатку я просто виводив чи пересікся персонаж з перепонами і виводив це в консоль для перевірки чи пересікаються перешкода та персонаж я використовував стандартний метод з класа `Bouns intersects(Bouns)` і він підходив мені до того моменту поки я не вирішив костемізувати гру тобто замість квадратів поставити спрайтову анімацію[2]. Після цього мені вже не підходила стандартна реалізація і я був змушений реалізувати по своєму. Далі потрібно було зробити щоб можна було почати гру заново після смерті тут я не став якось ускладняти і просто пересоздавав ігрове поле. Під час додавання анімації я зрозумів що вся графіка в мене залежить від однієї картинки де зберігаються всі спрайти і вирішив додати можливість вибору кольору гри бо це було просто реалізувати і додавало шарму грі. Вже під кінець написання програми я подумав а чому б не додати якусь мелодію для фону. На цьому я закінчив реалізовувати програмну частину моєї курсової роботи.

3.1 Загальний опис архітектури.

Вся гра поділена на класи виконавці і класи частини гри. Тобто є класи які управляють ігровим процесом і класи які представляють об’єкти на ігровому полі.

Класи предметної області:

1. Динозаврик — має можливість стрибати та присідати має ворогів та анімацію якщо пересікається з перешкодою вмирає.
2. Вороги — мають можливість рухатися на персонажа, можуть мати анімацію або буди статичними, мать швидкість одну для всіх.
3. Декорації — можуть мати або не мати анімації, рухаються зі сталою швидкістю.

РОЗДІЛ 4 ТЕСТУВАННЯ

РОЗДІЛ 5 ІНСТРУКЦІЯ ДЛЯ КОРИСТУВАННЯ

Щоб почати грати в мою гру потрібно запустити jar файл для цього отрібно:

1. Перейти в деректорію з файлом див. Рис 5.1
2. Відкрити термінал з цієї папки див. Рис 5.2
3. Відкрити файз за допомогою команди `java -jar <file>` див. Рис 5.3
4. Після виконаних даних пунктів ви почуєте мелодію а після екран ігри див Рис 5.4
5. Натисьні одну з представлених кнопок що вибрати колір або любую кнопку на клавіатурі що автоматично вибере стандартний режим
6. Для того щоб пригати використовується `space` та стрілка вверх для того щоб присісти використовується стрілочка вниз.
7. Смерть настає тоді коли динозаврик зістовхнувся з перешкодою щоб почати заново натисніть любую з представлених на стартовому екрані кнопок кнопку.

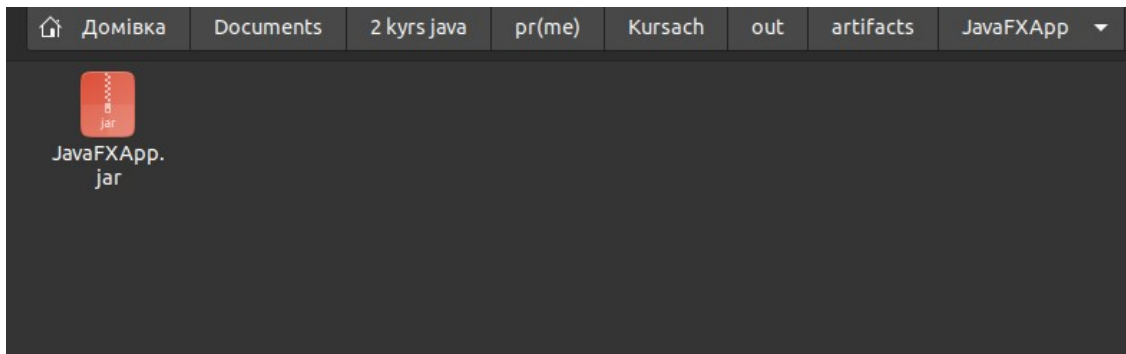


Рисунок 5.1 — Деректорія з файлом

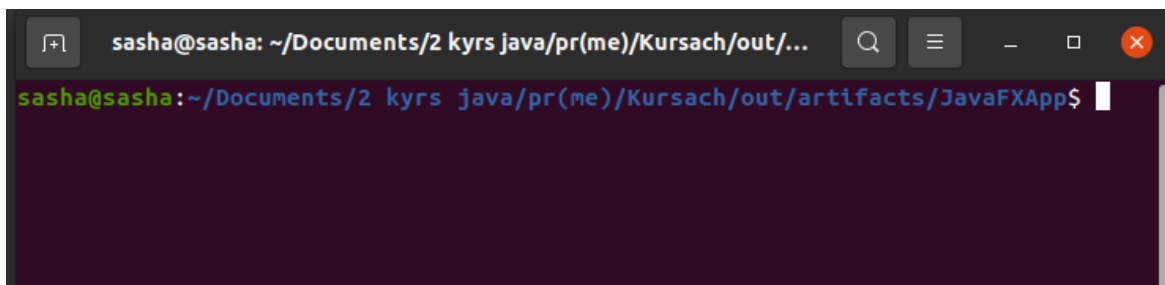


Рисунок 5.2 — Відкрили термінал з папки

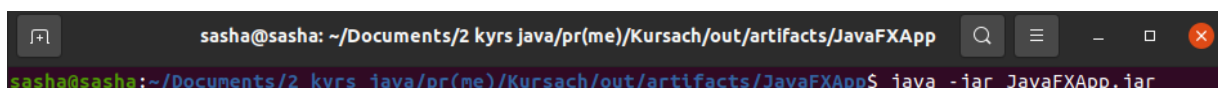


Рисунок 5.3 — Запуск jar файла

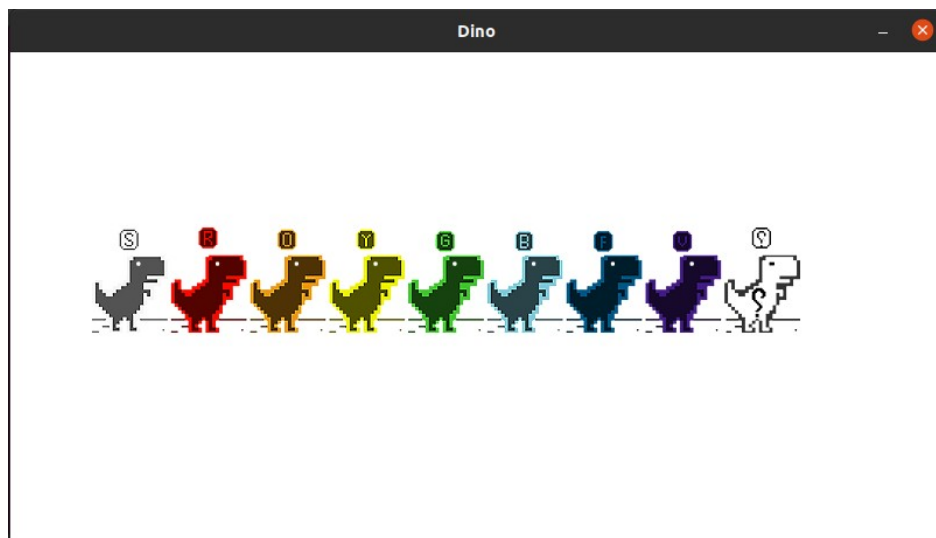


Рисунок 5.4 — Стартовый экран гри

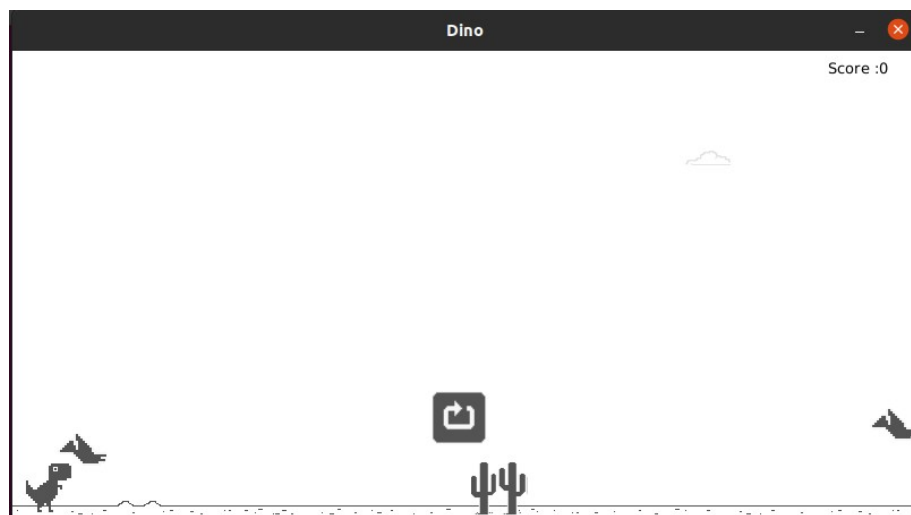


Рисунок 5.5 — Смерть

ВИСНОВОК

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. JavaFX. Tutorial 24. Flappy Bird Game (Урок 24. Пишем игру) rus.
//youtube.com URL https://www.youtube.com/watch?v=L5WwtGV3JvE&t=1137s&ab_channel=VladislavKavunenko
Дата звернення 28.11.20
2. JavaFx 2.0 - an approach to game sprite animation //stackoverflow.com
URL : <https://stackoverflow.com/ru/q/2836752>
Дата звернення 28.11.20

ДОДАТКИ