

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеський національний технологічний університет
Університет Інформатики і прикладних знань, м.Лодзь, Польща
Національний технічний університет України «Київський
політехнічний інститут»
Навчально-науковий інститут комп'ютерних систем і технологій
«Індустрія 4.0» ім. П.М. Платонова

XXIII Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів

«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»

Матеріали конференції



Одеса

20-21 квітня 2023 р.

Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXIII Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 20-21 квітня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – 449 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

Збірник буде корисним як для фахівців і працівників фірм, зайнятих в області ІТ, так і для викладачів, магістрів і студентів вищих навчальних закладів, які навчаються за напрямками і спеціальностями програмного забезпечення, обчислювальної техніки і автоматизованих систем, прикладної математики та обробки інформації, буде корисним професіоналам з комп'ютерного моделювання та розробки комп'ютерних ігор.

Результати досліджень у збірнику представляють собою своєрідний зріз сучасного стану справ в перерахованих галузях знань, який може допомогти як фахівцям, так і студентам університетів скласти загальну картину розвитку інформаційних технологій та пов'язаних з ними питань.

Наукові праці згруповані за напрямками роботи конференції та наведені в алфавітному порядку прізвищ авторів.

Матеріали (тези доповідей) друкуються в авторській редакції. Відповідальність за якість та зміст публікацій несе автор.

Матеріали подано українською та англійською мовами.

Редактор збірника Котлик С.В.

університет)	
30. Концепція інтелектуальної інформаційної системи аналізу та обробки антропометричних даних. Бондаренко А.С. (Національний університет «Одеська політехніка»)	373
Розділ 7: Комп'ютерні ігри і WEB-дизайн	375
1. The role of game projects in teaching programming. Pan A., Kim Ye.R. (University "Turan", Kazakhstan)	375
2. Development of the computer game "fade out" on the unity platform in the mode of 2d modeling. Tarasov V.A., Lagutkin B.D., Bibik I.V., Kasimova A.K., Mamyrova A.K. (College "turan", University "Turan", Kazakhstan)	377
3. Особливості дизайну для людей з особливими потребами. Альпашкін М.І., Романюк О. Н., Романюк О.В., Котлик С.В. (Вінницький національний технічний університет, Одеський національний технологічний університет)	380
4. Вплив трасування променів на рендер game-ready моделей. Бойцова М.П., Ломовцев П.Б. (Одеський національний технологічний університет)	382
5. Аналіз безкоштовних графічних редакторів для веб-дизайну. Бондаренко Н.О., Романюк О.Н. (Вінницький національний технічний університет)	383
6. Розробка ігрового додатку в середовищі UNITY. Булах В.О., Сахарова С.В., Іванова Л.В. (Одеський національний технологічний університет, Одеський технічний фаховий коледж ОНТУ)	385
7. Дослідження засобів бібліотеки react для розробки текстових блогів. Гандзюк К. Р. (Волинський національний університет імені Лесі Українки).	388
8. Штучний інтелект у комп'ютерних іграх. Геселева Н.В., Коваль А. К. (Державний торговельно-економічний університет)	389
9. Розробка бізнес-гри для навчання керівничого персоналу. Горбатюк М.В., Стельмашенко А.В. (Український державний хіміко-технологічний університет)	391
10. Кастомізабельний Тетріс на основі рушія HGE. Данченков В.О., Данченков Я.В. (Національний університет водного господарства та природокористування)	393
11. Розробка інтелектуальної гри-вікторини за допомогою UNITY 3D. Деркач Т.М., Вітер В.В. (Національний університет "Полтавська політехніка імені Юрія Кондратюка")	395
12. Дослідження та розробка анімаційної візуалізації інтер'єру у середовищі Unreal Engine. Єрощенко М.В., Ломовцев П.Б. (Одеський національний технологічний університет)	397
13. Комп'ютерні ігри і WEB-дизайн. Жилін М. (Національний університет "Одеська політехніка")	398
14. Аналіз ігор жанру «Симулятор колонії». Зелененький А.О., Ненов О.Л. (Одеський національний технологічний університет)	399
15. Розробка гри в жанрі ENDLESS RUNNER. Карякін Д. (Національний університет кораблебудування ім. адмірала Макарова)	402
16. Середовище розробки ігор Gamedeaker: огляд основних функцій та можливостей. Костюк В. В., Мельник А. В. (Житомирський державний університет ім. Івана Франка)	403
17. Комп'ютерні ігри та WEB-дизайн. Кравцова А. О. (Національний університет «Одеська політехніка».)	406
18. Розробка відеогри для стимуляції покращення зорових функцій. Курашин Є.О. (Український державний хіміко-технологічний університет)	407
19. Переваги RPG жанру відеоігор. Кухаренко В.С., Гайдаєнко О.В. (Національний університет кораблебудування імені адмірала Макарова)	410
20. Комп'ютерна RPG-гра «FATEFUL STORY». Лисогурський М.Л., Швець Н.В. (ВСП "Фаховий коледж промислової автоматики та інформаційних технологій ОНТУ)	412

РОЗРОБКА БІЗНЕС-ГРИ ДЛЯ НАВЧАННЯ КЕРІВНИЧОГО ПЕРСОНАЛУ

ГОРБАТЮК М.В. (neironogamiiko@gmail.com)

СТЕЛЬМАШЕНКО А.В. (astelmashenko@gmail.com)

Український державний хіміко-технологічний університет

Представлена робота присвячується проектуванню та реалізації функціональних та структурних особливостей програмного продукту під назвою «Dudes in Space», бізнес-гри для навчання керівничого персоналу.

Розробка будь-яких програмних продуктів пов'язана з роботою над структурними та функціональними особливостями проектів. Тому, окремим питанням в індустрії розробки комп'ютерних ігор стоїть вирішення, або запобігання проблем реалізації цих особливостей [1].

Спочатку, треба пояснити, що структурні особливості – це погляд команди програмістів на те, якою буде структура програми. Тобто, які фреймворки, методи, та принципи написання коду будуть використані.

До структурних особливостей можна віднести:

- використання MonoBehaviour, як базового класу, від якого успадковуються усі скрипти, окрім інтрефейсів (при використанні C# необхідно явно успадкуватися від MonoBehaviour);

- використання методу Invoke() для виклику іншого заданого методу, після закінчення деякого заданого часу;

- використання FixedUpdate(), методу, який викликається з фіксованою частотою кадрів;

- використання делегатів та медіаторів для створення жорсткого, або слабкого зв'язку між ігровими об'єктами.

- використання серіалізації для перетворення структури даних в послідовність бітів, для безпечного зв'язку між різними ігровими об'єктами.

Функціональні особливості – це погляд користувача на програмний продукт, а саме на те, який функціонал та інтерфейс йому надається.

До функціональних особливостей можна віднести:

- взаємодія з ігровими завданнями та подіями;

- ізометрична камера;

- особливості взаємодії з ігровим інтерфейсом;

- кросплатформність.

На основі концепції програмного продукту, з врахуванням його структурних особливостей, було створено діаграму класів (рис. 1).

З них, інтерфейси: IAstronaut, IShipComponent, ITask, IShipEvent, які вміщують логіку керування аватарами, модулями корабля, завдань та івентів. Класи-контролери: AvatarMoving, DoorActivator, TaskGenerator, ShipEventGenerator, CameraMover, які відповідають за взаємодію між ігровими об'єктами. Класи-сутності: Astronaut, AstronautVitals, ShipComponent, MarkerCollection, AstronautInfoFolder, SpaceShip, для реалізації інтерфейсів та забезпечення зв'язку гравця з ігровим світом, user interface (UI).

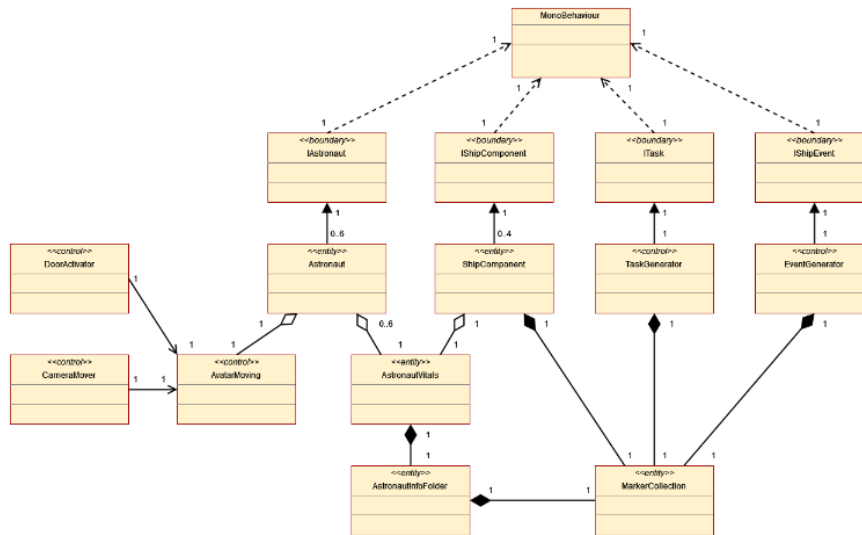


Рис.1 – Діаграма класів (без полів, методів, властивостей на подій)

Приклад структури одного з класів представлено на рис. 2.

```

using System.Collections.Generic;
using Star_Ship;
using UnityEngine;

namespace User_Interface.Tasks
{
    /// <summary>
    /// Controller that is responsible for enabling/disabling visibility of the Tasks
    List Frame.
    /// </summary>
    /// <remarks> Testing remarks comment. </remarks>
    public class TasksListUIController : MonoBehaviour
    {
        [SerializeField] private TasksListItem _tasksListItem;

        public void UpdateTasks(IEnumerable<TaskToSolveInfo> taskToSolveInfos)
        {
            RemoveAllItems();
            foreach (var info in taskToSolveInfos) AddNewItem(info);
        }

        private void AddNewItem(TaskToSolveInfo info)
        {
            var instance = Instantiate(_tasksListItem, transform);
            instance.SetImage(info.Icon);
            instance.SetText(
                $"{{info.TasksInQueue}}{{info.ComponentName}} is broken : must be fixed with
                {{info.RoleToSolve}}! ({{info.Time}} seconds)");
        }

        private void RemoveAllItems()
        {
            var childrenCount = transform.childCount;
            for (var i = 0; i < childrenCount; i++)
                Destroy(transform.GetChild(i).gameObject);
        }
    }
}

```

Рис.2 – Діаграма класів (без полів, методів, властивостей на подій)

З врахуванням функціональних особливостей, було реалізовано інтерфейс, який представлено на рис.3.



Рис. 3 – Загальний user interface

Висновок: щоб уникнути найпоширеніших проблем, пов'язаних з розробкою програмних продуктів, необхідно використовувати чітко сформульовану систему класів та створити інтуїтивно зрозумілий user interface.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1.Schell J. The Art of Game Design (second edition) / Jesse Schell. – Pittsburgh, Pennsylvania, USA: Crnegie Mellon University and Schell Games, 2014. – (2).URL: [https://iums.ac.ir/uploads/%5BJesse_Schell%5D_The_Art_of_Game_Design_A_book_of_1\(BookFi\).pdf](https://iums.ac.ir/uploads/%5BJesse_Schell%5D_The_Art_of_Game_Design_A_book_of_1(BookFi).pdf).

УДК 004.738.5

КАСТОМІЗАБЕЛЬНИЙ ТЕТРІС НА ОСНОВІ РУШІЯ HGE

ДАНЧЕНКОВ В.О. (danchenkov_ak20@nuwm.edu.ua),

ДАНЧЕНКОВ Я.В. (ja.v.danchenkov@nuwm.edu.ua).

Національний університет водного господарства та природокористування

2D гра у стилі Тетріс виконана на рушії Haaf's Game Engine, що працює з мовою C++. На початковій стадії розробки було створено оптимальну структуру наслідування, що допомагає логічно розподілити об'єкти за їх призначенням.

Custom (англ.) – користувацький, ability (англ.) – здатність, тобто слово «кастомізабельний» характеризує спроможність гри пристосовуватись до потреб користувача в аспекті дизайну, параметрів поля, та складності.

Для відображення ігрового поля, фігури не розміщуються прямо в матриці, а зберігаються у вигляді масиву фігур, оскільки це позбавляє програміста багатьох проблем при роботі з переміщенням. Впровадження механізму наслідування значно скоротило дублювання коду – застосування механізму є необхідним, адже наявність семи стандартних фігур передбачає визначення полів та методів не лише для кожної з видів фігур [1]. Також, без наслідування неможливим було б централізоване збереження різнотипних фігур, адже C++ є статично типізованою мовою. Для представлення самої фігури було описано 4 програмні структури, що агрегують одна одну (Рис. 1).

Для перевірки зіткнень використовується систему прев'ю-чекінга. Вона є кращою за звичайну перевірку за ітерацією, адже вона може використовуватися ще й при обертальних