Державна служба України з надзвичайних ситуацій
Львівський державний університет безпеки життєдіяльності
Навчально-науковий інститут цивільного захисту
Кафедра управління проектами, інформаційних технологій та телекомунікацій

«Допущено до захисту»
Завідувач кафедри УПІТтаТ
доктор технічних наук
професор
_____ Євген МАРТИН
“____” _____20___ року

# ДИПЛОМНА РОБОТА МАГІСТРА

на тему **«Розробка моделі та алгоритму роботи дитячого навчального симулятора вивчення правил пожежної безпеки»**

Виконав:
здобувачка VI курсу, групи КН-61мз
спеціальності 122 «Комп’ютерні науки»
(шифр і назва спеціальності)

Маріанна БАРТКО
(прізвище та ініціали)
Керівник Назарій БУРАК
(прізвище та ініціали)
Рецензент Роман ДУНЕЦЬ
(прізвище та ініціали)

Львів – 2020 року

# АНОТАЦІЯ

**Маріанна БАРТКО** «Розробка моделі та алгоритму роботи дитячого навчального симулятора вивчення правил пожежної безпеки». Дипломна робота за спеціальністю 122 “Комп’ютерні науки” складається з текстової частини, що містить 4 розділи, 65 с., 10 рис., 3 табл., 26 джерел.

Об’єкт дослідження – навчальні комп’ютерні ігри і симулятори педагогічного спрямування..

Мета роботи – дослідження моделей, та алгоритмів для розробки навчального комп’ютерного симулятора, призначеного для вивчення правил пожежної безпеки дітьми молодшого шкільного віку.

Магістерська дипломна робота спрямована на виконанні стадії препродакшну розробки навчального комп’ютерного симулятора, призначеного для вивчення правил пожежної безпеки дітьми молодшого шкільного віку.

Здійснено аналіз літературних джерел про використання інформаційно-комунікаційних технологій, зокрема ігор і симуляторів, в навчальному процесі молодшої школи.

Виконано порівняльний аналіз наявних засобів створення комп’ютерних ігор, обрано оптимальні інструменти.

Виконано математичне моделювання існуючої настільної гри, ідею якої взято за основу при розробці моделі ігрового симулятора та виведено залежність тривалості гри від кількості та конфігурації «червоних» та «зелених» позицій на ігровому полі.

Оптимізовано вищенаведену залежність та встановлено модель та алгоритм навчального симулятора з оптимальними статистичними параметрами.

Розроблено ескізи прототипів персонажів гри;

КОМП’ЮТЕРНІ НАУКИ, ПОЖЕЖНА БЕЗПЕКА, НАВЧАЛЬНИЙ ІГРОВИЙ СИМУЛЯТОР, ПРОТОТИПУВАННЯ, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, ОПТИМІЗАЦІЙНА ПРОГРАМА

# ВИСНОВКИ

В Україні показник смертності дітей на пожежах за останні 5 років (2015-2019 рр.) становить, в середньому, 0,96 на 100000 населення, тоді як середнє значення для країн Євросоюзу – 0,28. Окрім того встановлено, що близько 60% пожеж, на яких гинуть або травмуються діти молодшого шкільного віку, ними ж самими і спричинені, що свідчить про неналежне проведення заходів із профілактики дитячого травматизму внаслідок пожеж.

Широке впровадження в усі сфери людського життя, в тому числі і освіту, комп'ютерних технологій потребує розробки і впровадження ефективних засобів навчання. Особлива роль у такому навчанні належить іграм. Співставивши ці дані, підсумуємо, що використання інформаційних технологій для навчання дітей правилам безпеки життєдіяльності загалом і пожежної безпеки зокрема є актуальним і корисним, хоча і малодослідженим завданням.

Роботу присвячено розробці моделі і алгоритму навчального симулятора для навчання дітей молодшого шкільного віку правилам пожежної безпеки.

Під час виконання роботи на підставі аналізу було встановлено, що для даної вікової категорії найоптимальнішим симуляторм буде гра на базі гри «Snakes and Ladder».

В процесі роботи було реалізовано наступні завдання:

– здійснено аналіз літературних джерел про використання інформаційно-комунікаційних технологій, зокрема ігор і симуляторів, в навчальному процесі молодшої школи;

– виконано порівняльний аналіз наявних засобів створення комп'ютерних ігор, обрано оптимальні інструменти;

– виконано математичне моделювання існуючої настільної гри, ідею якої взято за основу при розробці моделі ігрового симулятора та виведено залежність тривалості гри від кількості та конфігурації «червоних» та «зелених» позицій на ігровому полі;

– оптимізовано вищенаведену залежність та встановлено модель та алгоритм навчального симулятора з оптимальними статистичними параметрами;

– розроблено ескізи прототипів персонажів гри;

– наведено рекомендації щодо безпечної роботи дітей молодшого шкільного віку з комп’ютером.

Розроблена математична модель та встановлена залежність дає змогу обирати різноманітні конфігурації механіки для ігор такого типу і може бути використана при реалізації інших проектів, пов’язаних із розробкою навчальних та розвиваючих ігор.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Apperley T. H. Genre and game studies: Toward a critical ap-proach to video game genres / T. H. Apperley // Simulation & Gaming. – 2006. – Vol. 37. – No. 1. – P. 6–23.

2. Clearwater D. What Defines Videogame Genre? Thinking about Genre Study after the Great Divide / D. Clearwater // The Journal of the Canadian Game Studies Association. – 2011. – No. 5 (8). – P. 29–49.

3. Crawford C. The Art of Computer Game Design / C. Crawford. – Osborne: McGraw-Hill, 1997. – 81 p.

4. Crawford C. A. Taxonomy of Computer Games / C. A. Crawford // Crowford C. The Art of Computer Game Design.

5. Egenfeldt-Nielsen S. Understanding VideoGames. The Essential Introduction / S. Egenfeldt-Nielsen, J. H. Smith, S. P. Tosca. – NewYork : Routlege, 2008. – 304 p.

6. Perron B. TheVideo GameTheory / B. Perron, M. Wolf. – NewYork : Routledge, 2009. – 430 p.

7. Офіційний сайт Construct2. – Режим доступу: www.scirra.com.

8. Офіційний портал розробників XNA – Режим доступу https://msdn.microsoft.com/ruRU/games-development-msdn.

9. Офіційний сайт MONO game – Режим доступу http://www.monogame.net

10. Офіційний сайт Phaser – Режим доступу http://phaser.io/

11. Офіційний githab репозиторій Pixijs – Режим доступу https://github.com/pixijs/pixi.js

12. Офіційний сайт Blender – Режим доступу https://www.blender.org/features/

13. Офіційний сайт easeljs – Режим доступу http://createjs.com/easeljs –

14. Портал розробників Html5 ігор – Режим доступу https://html5gameengine.com/

15. Офіційний сайт Melonjs – Режим доступу http://melonjs.org/

16. Офіційний сайт PandaJs – Режим доступу http://www.pandajs.net/ .

17. Офіційний сайт GameMaker – Режим доступу http://www.yoyogames.com/gamemaker

18. Офіційний сайт Unity – Режим доступу http://unity3d.com/ru/

19. Портал ігрових новин – Режим доступу http://www.3dnews.ru/games/622071

20. Портал ігрових новин – Режим доступу http://www.skgaming.com/content/4811

21. Аналіз популярності веб сайтів – Режим доступу http://compare.easycounter.com/

22. Популярний портал розробників ігор – Режим доступу http://www.gamedev.net/

23. How Long Is a Game of Snakes and Ladders? S. C. Althoen, L. King and K. Schilling, The Mathematical Gazette, Vol. 77, No. 478 (Mar., 1993), pp. 71-76

24. Афанасьєва К. В. Вплив комп'ютерізації на соціалізацію дітей / К. В. Афанасьєва [Електронний ресурс]. – Режим доступу: http://bibliofond.ru/view.aspx?id=586206

25. Бондаровська В.М. Діти та нові інформаційні технології : позитивні та негативні наслідки нової культури людського життя / В.М. Бондаровська [Електронний ресурс]. – Режим доступу: http://vydavnytstvo.plastscouting.org/vor/arkhiv/146/7.html

26. Reflections of an Master Game Designer. – Berkeley, California, Osborne : McGraw-Hill, 1984. – P. 19–40.

## Додаток А.

## Код для проходження гри з розрахунку на 2 гравців

```
public class
SnakeNLadderTest
{
                        public static void main(String[] args) {
                              SnakeNLadder s = new SnakeNLadder();
                              s.startGame();
                        }
                  }
                  class SnakeNLadder
                  {

                        final static int WINPOINT = 100;


                        static Map<Integer,Integer> snake = new
                  HashMap<Integer,Integer>();
                        static Map<Integer,Integer> ladder = new
                  HashMap<Integer,Integer>();

                        {
                              snake.put(99,54);
                              snake.put(70,55);
                              snake.put(52,42);
                              snake.put(25,2);
                              snake.put(95,72);

                              ladder.put(6,25);
                              ladder.put(11,40);
                              ladder.put(60,85);
                              ladder.put(46,90);
                              ladder.put(17,69);
                        }



                        public int rollDice()
                        {
                              int n = 0;
                              Random r = new Random();
                              n=r.nextInt(7);
                              return (n==0?1:n);
                        }

                        public void startGame()
```

```
	{
		int player1 =0, player2=0;
		int currentPlayer=-1;
		Scanner s = new Scanner(System.in);
		String str;
		int diceValue =0;
		do
		{
			System.out.println(currentPlayer==-
1?"\n\nFIRST PLAYER TURN":"\n\nSECOND PLAYER TURN");
			System.out.println("Натисніть R, щоб
кинути кубик");
			str = s.next();
			diceValue = rollDice();


			if(currentPlayer == -1)
			{
				player1 =
calculatePlayerValue(player1,diceValue);
				System.out.println("Перший гравец
:: " + player1);
				System.out.println("Другий
гравець :: " + player2);
				System.out.println("-------------
-----");
				if(isWin(player1))
				{
					System.out.println("Перший
гравець переміг");
					return;
				}
			}
			else
			{
				player2 =
calculatePlayerValue(player2,diceValue);
				System.out.println("Перший
гравець :: " + player1);
				System.out.println("Другий
гравець :: " + player2);
				System.out.println("-------------
-----");
				if(isWin(player2))
				{
					System.out.println("Другий
гравець переміг");
					return;
```

```
                    }
                }

                currentPlayer= -currentPlayer;



            }while("r".equals(str));
        }


        public int calculatePlayerValue(int player, int
diceValue)
        {
            player = player + diceValue;

            if(player > WINPOINT)
            {
                player = player - diceValue;
                return player;
            }

            if(null!=snake.get(player))
            {
                System.out.println("потрапив на
червоне поле");
                player= snake.get(player);
            }

            if(null!=ladder.get(player))
            {
                System.out.println("потрапив на зелене
поле");
                player= ladder.get(player);
            }
            return player;
        }

        public boolean isWin(int player)
        {
            return WINPOINT == player;
        }

}
```