

**Міністерство освіти і науки України
Одеський національний технологічний університет
Вінницький національний технічний університет
Інститут комп'ютерних систем і технологій
"Індустрія 4.0" ім.П.Н.Платонова**

**II Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів**

**«КОМП'ЮТЕРНІ ІГРИ ТА МУЛЬТИМЕДІА ЯК
ІННОВАЦІЙНИЙ ПІДХІД ДО КОМУНІКАЦІЇ»**

Матеріали конференції



Одеса

29-30 вересня 2022 р.

Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації / Матеріали II Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 29-30 вересня 2022 р. - Одеса, Видавництво ОНТУ, 2022 р. – 178 с.

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

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова - Богдан Єгоров, президент ОНТУ

Заступники голови:

Наталя Поварова, проректор з наукової роботи, ОНТУ,

Сергій Котлик, директор навчально-наукового інституту Комп'ютерних систем і технологій «Індустрія 4.0» ім. П.Н. Платонова, ОНТУ,

Сергій Шестопалов, декан факультету Комп'ютерної інженерії, програмування і кіберзахисту, ОНТУ

Члени комітету:

Олексій Ізвалов, регіональний координатор Global Game Jam в Східній Європі, ETI ім.Ельворті,

Сергій Артеменко, зав.каф. Комп'ютерної інженерії, ОНТУ,

Михайло Кисленко, Unity Developer, DAL'S Games,

Олександр Романюк, зав.каф. Програмного забезпечення, ВНТУ,

Ольга Чолишкіна, директор Інституту комп'ютерно-інформаційних технологій і дизайну, МАУП,

Олександр Терьошин, Unity 3d developer, BlueGoji,

Валерій Плотніков, зав.каф. Інформаційних технологій і кібербезпеки, ОНТУ,

Павло Івасюк, Senior Snapchat JS Developer, BeVisioned,

Петро Горват, зав.каф. Комп'ютерних систем і мереж, ДВНЗ "Ужгородський національний університет".

Матеріали подано українською та англійською мовами.

Редактор збірника Котлик С.В.

СПИСОК
організацій, представники яких взяли участь у роботі конференції

Turan University, Almaty, Republic of Kazakhstan
University of food technologies, Plovdiv, Bulgaria
V.N. Karazin Kharkiv National University
Відокремлений структурний підрозділ "Фаховий коледж промислової автоматизації та інформаційних технологій ОНТУ"
Відокремлений структурний підрозділ «Одеський технічний фаховий коледж ОНТУ»
Вінницький національний технічний університет
Волинський національний університет імені Лесі Українки
ДВНЗ «Приазовський державний технічний університет»
Державний торговельно-економічний університет
Донецький національний медичний університет
Донецький національний університет імені Василя Стуса
Економіко-технологічний інститут імені Роберта Ельворті
Запорізький національний університет
Київський національний економічний університет імені Вадима Гетьмана
Київський національний університет технологій та дизайну
Книжкова палата України ім. Івана Федорова
Мелітопольський державний педагогічний університет ім. Богдана Хмельницького
Науково-дослідний інститут інтелектуальної власності Національної академії правових наук України
Національна академія сухопутних військ імені гетьмана П. Сагайдачного
Національний авіаційний університет
Національний лісотехнічний університет України
Національний технічний університет «Харківський політехнічний інститут»
Національний університет «Львівська політехніка»
Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Національний університет харчових технологій
Одеська національна морська академія
Одеський національний технологічний університет
Одеський національний університет імені І. І. Мечникова
Первомайська гімназія №2 Первомайської міської ради Миколаївської обл.
Українська академія друкарства
Хмельницький національний університет
Центральноукраїнський інститут розвитку людини Відкритого міжнародного університету розвитку людини «Україна»

Котлик С. В. Аналіз гексогональних ігор. (Вінницький національний технічний університет, Одеський національний технологічний університет)	
Романюк О.В., Романюк О.Н. Тестування ігор: типові помилки відеоігор. (Вінницький національний технічний університет)	140
Сечін Ю.Д., Сіренко О.І. Основні сфери застосування NFT. (Одеський національний технологічний університет)	143
Станіславенко Є. Г., Романюк О.Н., Денисюк А.В., Рейда О.М., Котлик С.В. Етапи розробки персонажів у відеоіграх. (Вінницький національний технічний університет, Одеська національний технологічний університет)	145
Тимошенко О.В., Шестопапов С.В. Технологія доповненої реальності. (Одеський національний технологічний університет)	149
Шабатура Ю.В., Поповченко О.М. Застосування методів штучного інтелекту у вирішенні задач оцінки технічного стану складних систем. (Національна академія сухопутних військ імені гетьмана Петра Сагайдачного)	152
Шабатура Ю.В., Рибак В.Р. Технології virtual reality у підготовці медичних фахівців. (Національна академія сухопутних військ імені гетьмана П. Сагайдачного, Національний лісотехнічний університет України)	155
Шпак О.І. Використання алгоритмів штучного інтелекту. (Національний університет «Львівська політехніка»)	157
Розділ 5. Дизайн (геймдизайн, дизайн рівнів, саунддизайн, арт)	160
Belov A.M., Kim Ye.R. Features of creating arcade games in Python. (Turan University, Almaty, Republic of Kazakhstan)	160
Fedorov V., Konovalov K., Kim Ye.R. Object-oriented approach when developing computer games on the example of the game "Chess". (Turan University, Almaty, Republic of Kazakhstan)	163
Ivanka Kr. Krasteva, Vladimira Kr. Ganchovska. Multimedia presentation of an experiment in food industry. (University of Food Technologies, Plovdiv, Bulgaria)	166
Драченко А.В., Жуковецька С.Л. Особливості відмальовки орнаментів з використанням комп'ютерних технологій. (Одеський національний технологічний університет)	170
Марін М.С., Ненов О.Л. Створення анімації 3D-персонажу за допомогою технології motion capture. (Одеський національний технологічний університет)	171
Овдій А.А. Дизайн. Інноваційні елементи відеоігор (підсумок 2021). (Одеська національний технологічний університет)	173
Суліма Ю.Є., Подольський В.І., Савельєв В.В. Основи геймдизайну. Правила і принципи проектування ігор на прикладі розробки мобільної гри «Bee Arena» (Відокремлений структурний підрозділ «Одеський технічний фаховий коледж ОНТУ»)	175

Розділ 5.

Дизайн (геймдизайн, дизайн рівнів, саунддизайн, арт)

UDC 004.928

FEATURES OF CREATING ARCADE GAMES IN PYTHON

BELOV A.M., KIM YE.R. (e.kim@turan-edu.kz)

Turan University, Almaty, Republic of Kazakhstan

The paper considers the creation of arcade games as one of the directions of a more in-depth study of the programming language.

Today, the gaming industry is interested in a large number of people, but not many people know how they are made and what kind of work they put into creating even the simplest game. This article will consider a simple arcade game in the Python programming language using the Pygame library [1].

In order to start working, you need to connect the library, adjust the width and height of the screen in pixels, set a limit on the frame refresh rate:

```
#window size and FPS
WIDTH = 800
HEIGHT = 700
FPS = 60
```

These basic variables will be used to create the game window. Next, initialize the window using the following commands:

```
pygame.init()#window launch
screen = pygame.display.set_mode((WIDTH, HEIGHT)) #setting the height and width
of pygame.display.set_caption("War zone") #window name
clock = pygame.time.Clock() #new variable for more precise FPS control
```

pygame.init() is the command that launches pygame;
screen is the program window that is created when we set its size in the settings.

Next, you need to create a clock to make sure that the game is running at the specified frame rate.

After that, you can start the game cycle:

```
running = True
while running:
    clock.tick(FPS) #FPS control
```

Character creation, enemies and obstacles can be found on the internet.

One of the difficulties in developing the game was the difficulty in making the game register hits not on one enemy, but on all separately.

To solve this issue, the following solutions were proposed: a sheet was created and each enemy added to the field was entered into it (Figure 1):

```
 mobs = pygame.sprite.Group()#enemy sprites
 mob_list=[]#enemy list
 for i in range(0,max_enemy,1):
     enemy_vel = random.randrange(0, 2)
     mob = Enemy.Mob(random.randrange(100, WIDTH - 50),
         random.randrange(100, HEIGHT - 50),enemy_vel)
     mobs.add(mob)           #trigger for checking collisions
     mob_list.append(mob)    #list of live mobs
     all_sprites.add(mob)
     life = True
```

After adding all the enemies to the list, they can be accessed in the standard way: `mob_list[n]`.

In order for the game to understand when damage should be done, the `spritecollide` function built into Pygame was used. But the problem was that the input data should be of type `sprite`, `groupsprit`. Since the "mob" variable was always assigned the last added enemy at the end of the enemy addition cycle, the game dealt damage only to him, regardless of who the player hit. To correct this error, a loop was created that addressed each enemy in order and checked for collisions.:

```
 for i in range(0,len(mob_list),1):
     hits = pygame.sprite.spritecollide(mob_list[i], bullets, False)
     if hits:
         pygame.sprite.groupcollide(mobs, bullets, False, True)
         mob_list[i].kill()
```

In the second line, the function accesses a specific enemy through the sheet and checks for a collision with an object. The third argument of this function is a variable of type `bool` (logical) and specifies what to do with the first argument. `True` means that the first object will be destroyed, `False` says that the object does not need to be destroyed.

The `groupcollide` function works almost like the first one, but it already checks for collisions of a group of sprites and here it is needed to simply destroy the bullet in a collision. If a collision occurs, the bullet is destroyed and line 5 kills the enemy it hit.

Thus, the game always registers hitting the desired mob (enemy) and will only remove it from the screen.

To implement resistance from the enemies, and not one, but all, the following cycle was used:

```
 for i in range(0, len(mob_list), 1):
     if mob_list[i].rect.x <= player.rect.x + 100 and mob_list[i].rect.x >= player.rect.x - 100 and mob_list[i].rect.y < player.rect.y:
         mob_bullet = Enemy_bullet.Enemy_Bullet((mob_list[i].rect.x+10),( mob_list[i].rect.y+5), 0, 10,1)
         enemy_bullets.add(mob_bullet)
     elif mob_list[i].rect.x <= player.rect.x + 100 and mob_list[i].rect.x >= player.rect.x - 100 and mob_list[i].rect.y > player.rect.y:
         mob_bullet = Enemy_bullet.Enemy_Bullet(mob_list[i].rect.x+21, mob_list[i].rect.y, 0, -10,0)
         enemy_bullets.add(mob_bullet)
     elif mob_list[i].rect.y <= player.rect.y + 100 and mob_list[i].rect.y >= player.rect.y - 100 and mob_list[i].rect.x > player.rect.x:
         mob_bullet = Enemy_bullet.Enemy_Bullet(mob_list[i].rect.x, mob_list[i].rect.y+10, -10, 0,2)
         enemy_bullets.add(mob_bullet)
     elif mob_list[i].rect.y <= player.rect.y + 100 and mob_list[i].rect.y >= player.rect.y - 100 and mob_list[i].rect.x < player.rect.x:
         mob_bullet = Enemy_bullet.Enemy_Bullet(mob_list[i].rect.x+5, mob_list[i].rect.y+22, 10, 0,3)
         enemy_bullets.add(mob_bullet)
```


All the verification conditions in this function are needed so that the enemy shoots towards the player, and not in one direction (figure 1). As we can see, the enemy shoots only when the player is in a certain area (figure 2).

As can be seen from Figure 2, enemies do not shoot until the player enters a certain radius. But then you can notice another problem, the enemies are shooting non-stop, it looks beautiful, but there is no way to defeat such rivals.

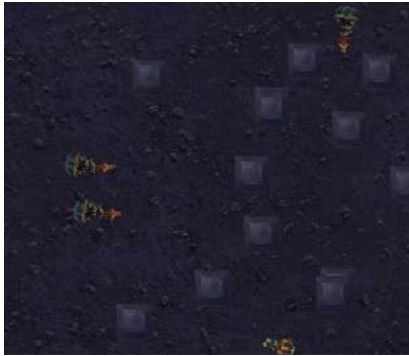


Figure 1

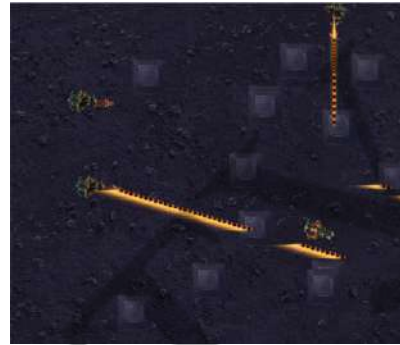


Figure 2

Therefore, to do this, it is necessary to add a delay before the shots, for which a new variable and a list were created, in which the time variable will be placed, starting from zero and increasing with each iteration of the cycle.

```
Time=50
running = True
while running:
    # Keeping the cycle at the right speed
    clock.tick(FPS)
    # Process Input (events)
    for i in range (0,len(shootTime),1):
        if shootTime[i]!=Time:
            shootTime[i]+=1
```

For the correct operation of the function, a check was added for the ability of the enemy to shoot at the moment:

```
for i in range(0, len(mob_list), 1):
    if shootTime[i] == Time:
```

Thus, the enemies start shooting intermittently, which makes it possible to play normally.

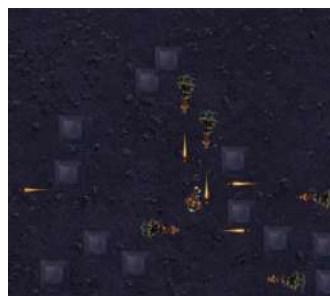


Figure 3

Figure 3 shows that there is no more "fan" of shots, therefore, they coped with the task.

In conclusion, I would like to note that game development when learning any programming languages contributes to a more in-depth study of programming and the formation of skills and competencies necessary for a modern IT specialist.

Reference:

[1] Mathis E. Learning Python: game programming, data visualization, web applications. 3rd ed." - Piter, 2020. - 512 p.

UDC 004.9

OBJECT-ORIENTED APPROACH WHEN DEVELOPING COMPUTER GAMES
ON THE EXAMPLE OF THE GAME "CHESS"

FEDOROV V., KONOVALOV K., KIM YE.R. (e.kim@turan-edu.kz)

Turan University, Almaty, Republic of Kazakhstan

The article discusses the creation of a game on the Unity game engine. An object-oriented approach to the development of a computer game is considered and the main methods of the main class of the computer game "Chess" are described.

In the modern world, video game development is one of the largest segments of the entire entertainment industry. The scale of the gaming industry is comparable, for example, with the film industry. And in terms of the growth rate over the past five years, the video game industry has been significantly ahead of it [1].

Currently, computer technologies have become actively used in various areas of our lives. The creation of various simple and complex computer programs for various fields of knowledge is constantly evolving. A computer program can act as an opponent in the game, be a storyteller, tutor, examiner. There are already computer programs aimed at developing various human functions, such as visual and auditory perception, attention, memory, verbal and logical reasoning, etc.

With the gradual development of technology, prerequisites have appeared for the emergence of universal games that could be run on various platforms, such as: personal computers, game consoles, mobile devices, universal web platforms. It was the idea of creating a board game on the Windows platform that was studied in detail and discussed in this article [2, 3].

One of the main classes of the CHESS game is the "ChessRules" class, which describes the main functions of the game, such as: Start(), Update(), LightOn(), LightOff(), ShowFigures().

The Start() function is designed to load the chessboard and place the pieces. This function is called once when loading the scene.

The code of this function is described below:

```
public void Start()
{
    string s = PlayerPrefs.GetString("SavedFen");
    chess = new ProjectChess(s);
    ShowFigures();
}
```

The Update method is called every time a frame is updated. Inside it there are algorithms designed to record the history and summarize the game, functions for obtaining the coordinates of

**II Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів**

**«КОМП'ЮТЕРНІ ІГРИ ТА МУЛЬТИМЕДІА ЯК
ІННОВАЦІЙНИЙ ПІДХІД ДО КОМУНІКАЦІЇ»**

Одеса

29-30 вересня 2022 р.

Збірник включає доповіді учасників конференції. Тези доповідей публікуються у вигляді, в якому вони були подані авторами.

Відповідальність за зміст і форму подачі матеріалу несуть автори статей.

Редакційна колегія: Котлик С.В., Шестопапов С.В.,
Корнієнко Ю.К.

Комп'ютерний набір і верстка: Соколова О.П.

Відповідальний за випуск: Котлик С.В.