

ИГРОВО БАЗИРАНО ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ ЗА НАЧИНАЕЩИ В РЕЖИМ НА PYGAME ZERO – ПРИМЕРНИ ЗАДАЧИ НА PYTHON. ЧАСТ II – БИБЛИОТЕКА PYGAME

Росица Георгиева

Югозападен университет „Неофит Рилски“ – Благоевград

Резюме. Настоящата статия представлява естествено продължение на статия, публикувана в предишен брой на списание „Математика и информатика“, в която е представена терминологията, свързана с игрово базираното обучение, ролята на задачите за усвояване на учебното съдържание, както и някои примерни задачи и тяхната методическа реализация, предназначени за целите на непрофилираното обучение по програмиране на ученици в гимназиален етап чрез създаване на игри с помощта на turtle графика на Python. Тя следва структурата на първата част и е фокусирана върху примерни задачи, реализирани с езика за програмиране Python, библиотека Pygame и средата за програмиране Mu. Примерните задачи са подкрепени с методически указания. Дадени са и насоки за използването им в обучението. В заключението се дискутират възможностите за игрово базирано обучение по програмиране в профилирани и непрофилирани гимназии с помощта на библиотека Pygame на Python.

Ключови думи: програмиране; език за програмиране Python; библиотека Pygame; среда за програмиране Mu

1. Въведение

Настоящата статия е естествено продължение на (Georgieva 2023), публикувана в предишен брой на списание „Математика и информатика“, в която е представена терминологията, свързана с игрово базираното обучение, ролята на задачите за усвояване на учебното съдържание, както и някои примерни задачи и методическа реализация за непрофилираното обучение по програмиране на ученици в гимназиален етап чрез създаване на игри с помощта на turtle графика със скриптовия език за програмиране Python. Настоящата статия следва структурата на първата част и фокусът в нея е поставен върху примерни задачи, реализирани с езика за програмиране Python, библиотека Pygame и средата за програмиране Mu. Примерните задачи са подкрепени с методически указания. Дадени са насоки за използването им в обучението. В заключението се дискутират възможностите за игрово базирано обучение

по програмиране в профилирани и непрофилирани гимназии с помощта на библиотеката Pygame на Python.

Примерните задачи и методическата им реализация, представени в статията, са подходящи както за начално обучение по програмиране, така и за продължаващо, след началното, с turtle графика на Python.

2. Задачите игри в учебниците по информатика за VIII клас

През 2016 г. влязоха в сила новият Закон за предучилищното и училищното образование¹ (ЗПУО) и новите Държавни образователни стандарти. С Наредба № 4 за учебния план² и Наредба № 5 за общообразователната подготовка³ се правят съществени промени в учебния план и учебното съдържание по информатика и информационни технологии. Предметът информатика в първи гимназиален етап се изучава само в VIII клас като общообразователен (2 учебни часа седмично), и то само в паралелките с профили „Математически“, „Софтуерни и хардуерни науки“, „Икономическо развитие“ и „Природни науки“. В IX и X клас не е заложено изучаване на информатика, а във втори гимназиален етап – XI и XII клас – е предвидена профилирана подготовка по предмета (4 – 8 учебни часа седмично) в четири модула⁴.

Съобразени с новата учебна програма по информатика в VIII клас, са издадени три учебника за общообразователна подготовка: „Информатика за VIII клас“ (Manev et al. 2017), (Momcheva et al. 2017) и (Boycheva et al. 2017). В разделите за програмиране в учебниците са представени системи от задачи за усвояване на знания, умения и навици по програмиране, като в (Manev et al. 2017) реализацията на задачите е с използване на езика за програмиране C#, в (Momcheva et al. 2017) примерните реализации използват езиците Java, C#, JavaScript и Python, а в (Boycheva et al. 2017) – езика Java. Във всеки от учебниците, на различни етапи от обучението, има предложени задачи игри основно за затвърждаване на знанията. Така например в учебника (Manev et al. 2017) в раздел VI „Създаване на софтуерен продукт“ са предложени като Проект 51 играта „Познай числото“ и Проект 54 тест от типа игри Question game за обобщаване на знанията, уменията и навиците в края на обучението по програмиране. Примерните реализации на задачите, предложени в учебника, са с графичен интерфейс.

В учебника (Momcheva et al. 2017) са представени условията на задачи игри „Камък – ножица – хартия“, Lights Puzzle, „Морски шах“ и „Разгъвка на куб“ – задача от типа Question game, както и проект, наречен funText за вмъкване на думи в зададен текст.

В третия учебник (Boycheva et al. 2017) играта GeographyQuiz е предложена за обобщение на темата „Разклонени алгоритми“, а играта „Бесеница“ е използвана като задача демонстрация на последователно търсене на елемент в масив.

3. Примерни задачи за обучение по програмиране за начинаещи

В този раздел ще представим няколко примерни задачи, подходящи за началното обучение по програмиране. В началото на обучението е добре учителят да представи на учениците официалния сайт на библиотеката Pygame⁵ на Python и да даде разяснения как да се ориентират в него: как да използват документацията, как да търсят по референтен индекс, кои са най-често използвани модули и функции, представени с техния синтаксис и семантика.

За начало на учениците се предлага да решат следната задача.

Задача 1. Напишете програма, която изчертава на екрана окръжност, оцветена в червено, с радиус 25. *Упътване.* Използвайте библиотека Pygame.

Основни цели: придобиване на нови знания и умения за инициализиране на променливи; използване на функциите от Pygame:

- `display.set_mode(size)`;
- `display.update()`;
- `quit()`;
- `draw.circle(screen,color,(x,y),radius),update()`.

Примерно решение на задачата е показано на фигура 1.

```

1 import pygame
2 import sys
3
4 pygame.init()
5 screen = pygame.display.set_mode((800,600))
6
7 while True:
8     for event in pygame.event.get():
9         if event.type == pygame.QUIT:
10             pygame.quit()
11             sys.exit()
12     pygame.draw.circle(screen,'red',(100,100),25)
13     pygame.display.update()
14 
```

Фигура 1. Реализация на задача 1

Методически бележки

Функциите, които се използват в задачата, са:

- `pygame.display.set_mode(width,height)` – инициализира прозорец с размери параметрите на функцията;

- `pygame.draw.circle(screen,color,(x,y),r)` – чертае окръжност, центърът на която е с координати (x,y) , а радиус – цялото число r , в прозореца `screen` със зададения като втори параметър цвят `color`;
- `pygame.quit()` – деактивира библиотеката на Pygame;
- `pygame.update()` – актуализира състоянието на прозореца (преначертава обектите на прозореца при всяко извикване).

```
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            pygame.quit()
            sys.exit()
```

Фигура 2. Излизане от прозореца на Pygame

Конструкцията от фигура 2 се предоставя на учениците наготово и без да се навлиза в подробности, се обяснява, че тази конструкция, при натискане на бутона Close, затваря прозореца на Pygame. Функцията `sys.exit()` се използва за излизане от интерпретатора и е от библиотека `sys`.

След като учениците са решили тази задача, може да се разискват приликите и разликите при използването на функциите от `turtle` и `Pygame`. Изяснява се синтаксисът, припомнят се понятията координатна система и координати на точка. Уточнява се, че координатите $(0,0)$ са в горния ляв ъгъл на работното пространство.

Задача 2. Променете кода на задача 1 така, че окръжността да се „движи“ чрез стрелките за навигация от клавиатурата в четирите посоки – наляво, надясно, нагоре, надолу.

Основни цели: изграждане на умения за работа с клавиши за движение, инициализиране на променливи.

На учениците се предлагат наготово конструкциите за инициализиране на променливите `circleX` и `circleY` с координатите (x, y) на центъра на окръжността, `radius` за радиуса, както и на променливата `circle_pos` от тип списък, която се инициализира с позицията на окръжността върху екрана (фигура 3). Наготово се предоставя и кодът за управление на движението на окръжността в четирите посоки (фигура 4).

```
9 circleX = 100
10 circleY = 100
11 radius = 10
12 circle_pos = [400, 550]
```

Фигура 3. Инициализиране на променливите

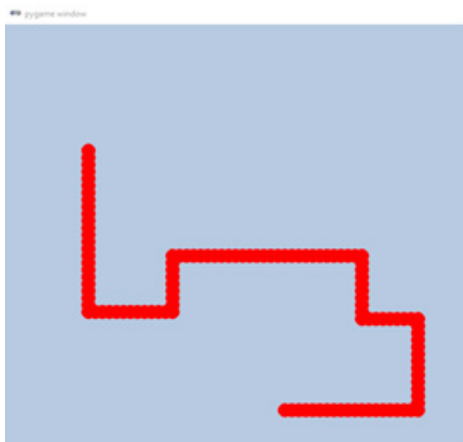
```

19
20     if event.type == pygame.KEYDOWN:
21         circleX = circle_pos[0]
22         circleY = circle_pos[1]
23         if event.key == pygame.K_UP:
24             circleY -= radius
25         elif event.key == pygame.K_DOWN:
26             circleY += radius
27         elif event.key == pygame.K_LEFT:
28             circleX -= radius
29         elif event.key == pygame.K_RIGHT:
30             circleX += radius
31         circle_pos = [circleX, circleY]
32
33 pygame.draw.circle(screen, 'red', (int(circle_pos[0]), int(circle_pos[1])), radius)
34 pygame.display.update()

```

Фигура 4. Движение на окръжността

Визуализация от изпълнение на решението на задача 2 е показана на фигура 5.



Фигура 5. Визуализация
от изпълнение
на решението на задача 1

Фигура 6. Визуализация от изпълнение на решението на задача 2

Задача 3. Съставете програма, която симулира удар на обект „квадрат“ в четирите края на работното пространство. Движението на обекта „квадрат“ да се извършва от програмата (без намесата на играч).

На учениците се предоставя наготово част от решението на задачата (фигура 7)

```
def update():
    global vx, vy
    square.x += vx
    square.y += vy
    if square.right > width or square.left < 0:
        vx = -vx
    if square.bottom > height or square.top < 0:
        vy = -vy
```

Фигура 7. Движение на обекта „квадрат“

Разяснява се правилото за област на действие на променливите и се обръща внимание на разликата в използването на локални и глобални променливи. За стойностите `square.right`, `square.left`, `square.bottom` и `square.top` може да се каже само, че са част от виртуалните атрибути на обекта `square` от клас `Rect`, които могат да се използват за проверка на местоположението на квадрата спрямо краищата на прозореца, без да се навлиза в подробности, свързани с обектно ориентираното програмиране.

Задача 4. Променете кода на предишната задача така, че да „удряте“ обекта „квадрат“ в обект „платформа“ (правоъгълник), позициониран в средата на долната част на прозореца. Движете платформата само наляво и надясно със стрелките за навигация от клавиатурата. Ако не „уцелите“ с платформата квадрата, играта приключва.

Основни цели: прилагане на усвоените знания, умения и навици за инициализиране на променливи; използване на оператори за цикъл; използване на условен оператор; използване на потребителски функции и параметри на функция; работа с клавишите за движение; обобщаване на знанията, уменията и навиците на учениците; използване на функции от Pygame:

- `display.set_mode(size);`
- `draw.rect(screen, color, rectangle);`
- `event.get();`
- `quit();`
- `update().`

Решенията на задача 3 и задача 4 могат да се намерят на адресите, дадени като бележки 6 и 7, а визуализация на момент от изпълнение на решението на задача 4 е показан на фигура 6.

Забележка. Решението на задача 3 може да бъде предоставено наготово на учениците, а те да направят в него необходимите за решението на задача 4 промени. Задача 3 и задача 4 са компоненти от следващата задача „Градушка“.

Задача 5. Игра „Градушка“. Напишете програма игра, при която от „небето пада градушка“. Играчът (платформата) няма къде да се скрие и трябва да се предпази от падащите ледени парчета с движения наляво/надясно. След падането на определен брой ледени парчета скоростта на падане да се увеличава. Играта приключва, когато първото ледено парче удари платформата. В края на играта програмата трябва да изведе на екрана броя на ледените парчета, които са били избегнати.

Основни цели: прилагане на усвоените знания, умения и навици за инициализиране на променливи; използване на локални и глобални променливи; използване на оператори за цикъл; използване на условен оператор; използване на потребителски функции, параметри на функция; работа с клавиши за движение; обобщаване на знанията, уменията и навиците на учениците; използване на функции от Pygame:

- `font.SysFont(name, size, [True/False, True/False]);`
- методите за работа със списъци – `append(item)` за добавяне на елемент в края на списъка и `pop(index)` за премахване от списъка на елемент на позиция `index`.

Етапи на създаване на играта

Първи етап: инициализиране на необходимите променливи, позициониране на играчите. В задачата се представя вградената функция за създаване на обект `Font` от системните шрифтове.

Синтаксис:

`pygame.font.SysFont (ширифт, размер на шрифт, [True/False, True/False])`

Семантика: функцията връща като резултат обект `Font`, който е зареден от системните шрифтове. Параметрите `[True/False, True/False]` не са задължителни. Ако са посочени, тогава при стойност `True` за третия параметър стилът на шрифта ще бъде `bold` (удебелен) и съответно `italic` (наклонен) при стойност `True` за четвъртия параметър.

Например `myFont = pygame.font.SysFont('monospace',35)` ще създаде обект от тип „Шрифт“ с име `'monospace'` и размер `35 pt`. Тъй като Pygame не предоставя директен начин за писане на текст върху екрана, за създаване на обекта текст, който след това да бъде изведен на екрана, трябва да се използва методът `render(text, True/False, color)`. Този метод може да визуализира само единични редове. Първият аргумент е текстът, който се изписва, вторият аргумент е булева стойност `True/False` или `1/0` и ако е `True`, знаците ще имат гладки ръбове, а третият аргумент е цветът на шрифта. Следният код:

- `text = ,Score:' + str(score)`
- `label = myFont.render(text, 1, ,red')`

ще се изпише Score и конкретна стойност на резултата с гладки ръбове и червен цвят.

Втори етап: създаване на потребителските функции:

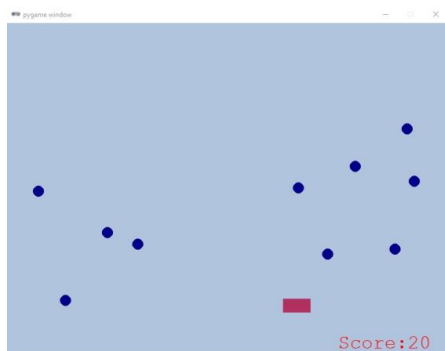
- `set_level(score, speed)` за увеличение на скоростта на „ледените парчета“;

- `drop_enemies(enemy_list)` за движението на „ледените парчета“;
- `draw_enemies(enemy_list)` за създаване на „ледените парчета“;
- `update_position(enemy_list, score)` за промяна на резултата;
- `check_collision(enemy_list, player_pos)` и `detect_collision(player_pos, enemy_pos)` за проверка на сблъсък между играча и „ледено парче“.

Трети етап: създаване на главната програма със следните елементи: описание на движението на играча с клавишите LEFT и RIGHT; извикване на потребителските функции; извеждане на резултата; прекратяване на играта.

Четвърти етап: проверка и тестване на програмата. Учениците сами коригират синтактичните грешки в кода и проиграват играта.

Решението на тази задача игра може да бъде намерено на адреса от бележка 8, а визуализацията от изпълнението е показана на фигура 8.



Фигура 8. Визуализация от изпълнение на играта „Градуска“



Фигура 9. Визуализация от изпълнение на играта „Змия“

Задача 6. Класическа игра „Змия“. Създайте програма, която реализира играта „Змия“. Змията трябва да се „нахрани“ с плодове. Целта на играта е змията да събере възможно най-много плодове. За изяден плод се счита онзи, при който змията „преминава“ през координатите на центъра на околността плод. Това да става, като играчът движи змията чрез стрелките за навигация в четирите посоки и внимава да не стигне края на прозореца на играта. При изяждането на всеки плод дължината на змията да се увеличава. По време на

играта извеждат броя на изядените плодове. Когато играта започне, змията не може да бъде спирана. Играта приключва, ако змията достигне края на прозореца на играта. Попитайте играча дали ще продължи играта.

Упътване: за създаване на змията използвайте обект „квадрат“ или „окръжност“. Можете да използвате като пример и кода на Задача 2. Плодът може да бъде например портокал – оцветена в оранжево окръжност.

Основни цели: прилагане на усвоените знания, умения и навици за инициализиране на променливи; използване на оператори за цикъл; използване на условен оператор; използване на потребителски функции за създаване на змията и плода, за извеждане на резултата и за край на играта; работа със списъци; работа с шрифтове; работа с клавиши за движение, използване на функции от Pygame:

- `time.Clock()`;
- `font.SysFont(name, color)`;
- `font.render(text, True, color)`.

Етапи на създаване на играта

Първи етап: инициализиране на необходимите променливи, позициониране на плода и змията.

Втори етап: създаване на потребителските функции:

- `snake (snake_block, snake_list)` – създава змията;
- `total_score (score)` – извеждане на резултата (колко плодове е „изяла“ змията);
- `message (msg, color)` – съобщение за край на играта;
- `draw_food (x, y, radius)` – създава плода/храната.

Трети етап: създаване на функция `main`. На учениците се предлага готово функцията. Основните стъпки в тази функция са: описание на движението на змията с клавишите `LEFT` и `RIGHT`; извикване на потребителските функции; извеждане на резултата и прекратяване на играта.

Четвърти етап: проверка и тестване на програмата. Учениците сами коригират синтактичните грешки в кода и проиграват играта.

Реализации на играта може да се намерят на адресите от бележка 9 и бележка 10, а визуализация от изпълнението ѝ е показана на фигура 9.

Методически бележки: учениците трудно възприемат да се говори за парадигмите на обектно ориентираното програмиране, но независимо от това могат да се споменават някои термини, без изрично да се дават дефиниции. В хода на уроците учителят може да въвежда новите понятия, без да навлиза в големи детайли за някои от тях и да отделя повече време на стила на писане на код, мнемониката в имената на променливите и функциите, практическата работа. Игрово базираното обучение по програмиране позволява да се избират и комбинират в уроците различните типове

данни, оператори, програмни конструкции. С оглед нивото на знания и мотивацията на учениците учителят може да предостави възможност и за използване на готови ресурси.

Заклучение

Повечето от задачите, предложени в двете статии, са използвани от автора при провеждането на експериментално игрово базирано обучение по програмиране в извънкласна форма. Анализът на резултатите дава основание да се счита, че предложените задачи, реализирани с езика Python, библиотека Pygame Zero и чрез Mu Editor, са подходящи за обучение по програмиране в гимназиален етап. Задачите са подходящи за обучение както в непрофилираните гимназии, така и при начално обучение в профилирани училища. Авторът има намерението на по-късен етап да публикува резултатите от проведенния експеримент.

БЕЛЕЖКИ

1. Закон за предучилищното и училищното образование, <https://www.navet.government.bg/bg/media/ZAKON-ZA-PREDUCHILISHTNOTO-I-UCHILISHTNOTO-OBRAZOVANIE.pdf>
2. НАРЕДБА № 4 от 30 ноември 2015 г. за учебния план, https://cioo.mon.bg/wp-content/uploads/2014/07/nrdb4_30.11.2015_ucheben_plan.pdf
3. НАРЕДБА № 5 от 30.11.2015 г. за общообразователната подготовка, https://cioo.mon.bg/wp-content/uploads/2014/07/nrdb5_30.11.2015_obshtoobr_podgotovka.pdf
4. НАРЕДБА № 7 от 11.08.2016 г. за профилираната подготовка, https://cioo.mon.bg/wp-content/uploads/2014/07/naredba_7_11.08.2016_profilirana_podgotovka.pdf
5. <https://www.pygame.org/docs/>
6. https://github.com/RosyGeorgieva/Codes/blob/main/ball_bat_bez.py
7. https://github.com/RosyGeorgieva/Codes/blob/main/ball_bat.py
8. https://github.com/RosyGeorgieva/Codes/blob/main/platform_game.py
9. https://github.com/RosyGeorgieva/Codes/blob/main/Snake_new1.py
10. Друга реализация на играта „Змия“ може да се намери на адрес <https://github.com/rajatdiptabiswas/snake-pygame/blob/master/Snake%20Game.py>

ЛИТЕРАТУРА

- БОЙЧЕВА, С., НИКОЛОВА, Н., СТЕФАНОВА, Е., ДЕНЕВ, А., 2017. *Информатика VIII клас общообразователна подготовка*. Просвета, София, ISBN 978-954-01-3444-4
- ГЕОРГИЕВА, Р., 2023. Игрово базирано обучение по програмиране за начинаещи в режим на Pygame zero – примерни задачи с turtle графика на Python, *Математика и информатика*, т. LXVI, бр. 2, с. 170 – 187.
- МАНЕВ, К., МАНЕВА, Н., ХРИСТОВА, В., 2017. *Информатика VIII клас. Общообразователна подготовка*. Изкуства, София, ISBN 978-619-7243-21-5, Извлечено от: https://izkustva.bg/pdf/TextBook_I8.pdf
- МОМЧЕВА, Г., ГЛУШКОВА, Т., МАРИНОВА, Р., 2017. *Информатика VIII клас Общообразователна подготовка*. Булвест 2000, София, ISBN 978-954-426-XXX-X

REFERENCES

- BOYCHEVA, S., NIKOLOVA, N., STEFANOVA, E., DENEV A., 2017. *Informatika VIII klas Obshtoobrazovatelna podgotovka*, izd. Prosveta, Sofia, ISBN 978-954-01-3444-4 [in Bulgarian]
- GEORGIEVA, R., 2023. Igrovo-bazirano obuchenie po programirane za nachinaeshiti v rezhim na Pygame zero – primerni zadachi s turtle grafika na Python, *Matematika I informatika*, v. LXVI, № 2, pp. 170 – 187. [in Bulgarian]
- MANEV, K., MANEVA, N., HRISTOVA, V., 2017. *Informatika 8. klas Obshtoobrazovatelna podgotovka*, izd. Izkustva, Sofia, ISBN 978-619-7243-21-5, Available from: https://izkustva.bg/pdf/TextBook_I8.pdf [in Bulgarian]
- MOMCHEVA, G., GLUSHKOVA, T., MARINOVA, R., 2017. *Informatika VIII klas Obshtoobrazovatelna podgotovka*, izd. Bulvest 2000, Sofia, ISBN 978-954-426-XXX-X [in Bulgarian]

GAME-BASED PROGRAMMING TEACHING FOR BEGINNERS IN PYGAME ZERO MODE – SAMPLE PYTHON TASKS. PART II – PYGAME LIBRARY

Abstract. This article is a natural continuation of its first part, published in the journal “Mathematics and Informatics”, where the terminology related to game-based learning and the role of tasks in learning the learning content, as well as example tasks and methodical implementation for non-profiled training in programming for high school students through the creation of games, are presented using Python’s turtle graph. This article follows the structure of its first part and focuses on example tasks implemented with the Python programming language, the Pygame library, and the Mu programming environment. The sample tasks are supported by methodical instructions. Guidelines are also given for their use in training. In the conclusion, the possibilities for game-based programming learning in profiled and non-profiled high schools are discussed using the Pygame Python library.

Keywords: programming; Python programming language; Pygame library; Mu programming environment

✉ **Ms. Rositsa Georgieva, PhD student**

Department of Informatics
South-West University “Neofit Rilski”
66, Ivan Mihailov St.
2700 Blagoevgrad, Bulgaria
E-mail: georgieva.rosy72@gmail.com