

ИГРОВО БАЗИРАНО ОБУЧЕНИЕ ПО ПРОГРАМИРАНЕ ЗА НАЧИНАЕЩИ В РЕЖИМ НА PYGAME ZERO – ПРИМЕРНИ ЗАДАЧИ С TURTLE ГРАФИКА НА PYTHON

Росица Георгиева

Югозападен университет „Неофит Рилски“ – Благоевград

Резюме. В последните години се наблюдава все по-голяма дигитализация във всички сфери на живота в световен мащаб. Търсенето на специалисти в областта на ИКТ налага подготовката им да започне още в училище с изучаването на програмиране и ИТ. Във всички степени на средното образование се наблюдават добри практики и засилване на интереса към игрово базираното обучение. Използват се различни среди, езици за програмиране и подходи за обучение по учебните дисциплини „Компютърно моделиране“ и „Информатика“. Във фокуса на статията са примерни задачи и методическа реализация за непрофилирано обучение по програмиране на ученици в гимназиален етап чрез създаване на игри. Представена е терминологията, свързана с игрово базираното обучение и ролята на задачите за усвояване на учебното съдържание. Основният фокус е върху създаването на забавни игри, реализирани с език за програмиране Python с използване на turtle графика в среда 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 обучение по програмиране в профилирани и непрофилирани гимназии.

Ключови думи: образователни компютърни игри; игровизация; игрово базирано обучение; програмиране; език за програмиране Python; turtle графика; среда за програмиране Mu

1. Въведение

В последните години се наблюдава все по-голяма дигитализация във всички сфери на живота в световен мащаб. Търсенето на специалисти в областта на ИКТ сектора е огромно. Подготовката на такива кадри трябва да започне още в училище с изучаването на програмиране и информационни технологии. Във всички степени на средното образование се наблюдават добри практики и засилване на интереса към игрово базираното обучение. Използват се

различни среди, езици за програмиране и подходи за обучение по учебните дисциплини „Компютърно моделиране“ и „Информатика“.

Във фокуса на статията са примерни задачи и методическа реализация за непрофилирано обучение по програмиране на ученици в гимназиален етап чрез създаване на игри. В Раздел 2. е представена терминологията, свързана с игрово базираното обучение. В Раздел 3. е представена ролята на задачите за усвояване на учебното съдържание, а в Раздел 4. фокусът е върху създаването на забавни игри, реализирани с език за програмиране Python с използване на turtle графика в среда 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 обучение по програмиране в профилирани и непрофилирани гимназии.

2. Роля на игрово базираното обучение

Игровизацията (Gamification) може да се определи като „използване на базирана на играта механика, естетика и игрово мислене, с цел да ангажира хората, да мотивира действията, да насърчава ученето и решаването на проблеми“ (Turarova et al. 2018). Понятието може да бъде описано и като концепция за въвеждане на игрови техники в различни процеси и области, като бизнес, маркетинг и образование (Trybus 2015).

В сферата на образованието игровизацията се проявява чрез концепцията за игрово базирано обучение (ИБО). ИБО е средство за придобиване на знания и умения чрез игра, където игровите дейности изискват от играча да решава проблеми и предизвикателства, предоставени за постигане на резултати (Kirriemuir & McFarlane 2004). Други автори дефинират ИБО като „форма на обучение, насочено към ученика, което използва дигитални игри за образователни цели“ (Aleksić et al. 2016).

В научната литература се срещат различни термини и формулировки на понятия, свързани с приложението на компютърни игри в образователния процес – образователни видеоигри, дидактически компютърни игри, образователни компютърни игри, сериозни игри. „Образователните игри са онези сериозни игри, чиято цел е предаването на специфични знания и умения в някаква област и предоставяне на нови знания чрез игрови подход“ (Terzieva et al. 2007). В (Dureva & Kaseva 2011) образователните компютърни игри са определени като „положителен фактор за развитието на мисленето, паметта и вниманието на детето и формиране на умения за обобщение и класификация“.

Използването на образователни компютърни игри е широко практикувано в началното училище и поради това, че „включването на образователните компютърни игри в учебно-възпитателен процес го прави по-интересен, занимателен, наситен с положителни емоции, като разбира се, неминуемо води

до повишаване плътността на урока и по-успешно усвояване на знанията и тяхната трайност“ (Dureva & Kaseva 2011).

Проучването от (Spirova 2018) съпоставя традиционното и интерактивното обучение, разглежда реализацията на някои от основните дидактически принципи чрез игрово базираното обучение и прави извода, че „чрез използването на образователни игри принципите на обучение могат да бъдат реализирани успешно и да доведат до по-висока мотивация и по-добри резултати“.

По отношение на обучението по програмиране ИБО може да се разглежда в три направления: използване на съществуващи образователни компютърни игри за придобиване на знания и умения и развитие на алгоритмично мислене; проектиране и разработване на компютърни игри, включително математически модел на играта, дизайн на графичен потребителски интерфейс (GUI) с реализация на C# в MS Visual Studio, кодиране, тестване и проверка; среда, базирана на игри (Tuparova et al. 2020).

При игрово базираното обучение от съществено значение са педагогическите аспекти на игрите, които се използват в обучението. В (Tuparova et al. 2018) са посочени основните педагогически изисквания към игрите, свързани с: учебното съдържание, което се покрива от дадена игра и предметната област; целите на обучение, измерени с дадената игра; дидактическото предназначение на играта – „формиране и усвояване на нови знания и умения, за затвърдяване на знанията и уменията, за проверка (самопроверка) и оценка на знанията и уменията, или има комплексно предназначение“.

3. Задачите в обучението по програмиране

Обучението по програмиране е базирано основно на решаване на различни по тип задачи. Редица автори разглеждат мястото и ролята на задачите по информатика (Asenova 1990), методическите проблеми при решаването на задачи в училищните курсове по информатика и ИТ (Dureva 2003). Съставят се системи от опорни задачи за подготовката на ученици за участие в олимпиади и състезания по информатика и информационни технологии (Grozdev & Garov 2008), (Garov & Todorova 2006), (Garov 2000).

Функциите на задачите, които се прилагат при обучението по програмиране, се обуславят от горепосочените основни акценти. Освен основните функции: методическа, дидактическа, организираща и управляваща (Dureva 2003), (Garov 2010), могат да се посочат и някои специфични: формиране на алгоритмичен стил на мислене, усвояване на основите на моделирането, усвояване формализацията при разкриване на взаимовръзката между различните форми на едно и също понятие, явление или процес, приложение на задачите в практиката.

В (Dernovaya 2008) и (Dureva 2003) е представен модел на етапите при решаване на проблеми от различни предметни области със средствата на ин-

форматиката, който успешно може да бъде приложен и при игрово базираното обучение по програмиране.

Посочените по-горе функции на задачите по информатика и ИТ и етапите за тяхното решаване могат да се приложат при задачи по програмиране, свързани с разработване на игри. Така например в (Nikolova & Tuparova 2018) е представена система от задачи, базирани на създаване на игри, с посочени цели на обучение и очертани етапи за решаването им. Тази система от задачи е насочена към курса по информатика в профили „Софтуерни и хардуерни науки“ и „Математически“ със задължителни профилиращи учебни предмети информатика и информационни технологии в VIII клас по Наредба № 7 от 11.08.2016 г. за профилираната подготовка¹.

Основните изисквания за изграждане на система от задачи в курсове по информатика и ИТ са: ясно определяне на целите на обучение, които се постигат или измерват с дадената задача; декомпозиране на задачата на отделни подзадачи, които са относително независими (Dureva 2003). В следващия раздел на статията са предложени задачи и методика за реализацията им, подходящи и за непрофилирано обучение на начинаещи.

4. Избор на среда за програмиране

За реализация на задачите е избран езикът за програмиране Python с използване на turtle графика в режим Pygame Zero в среда за програмиране Mu.

Python е съвременен, интерпретативен, интерактивен, обектно ориентиран език за програмиране от високо ниво, създаден от Гуидо ван Росум в началото на 90-те години². В сравнение с други съвременни езици за програмиране, като Java или C/C#, Python има много предимства, като много по-кратък код, което е свързано и с по-малко грешки и спестяване на ресурс. Освен това Python има изчерпателна стандартна библиотека и хиляди допълнителни библиотеки в Python Package Index. Библиотеките на много други програмни езици могат да се използват и в Python. Езикът е „един от най-доминиращите скриптови езици през последните години. По последни данни според редица тренд платформи най-голямото търсене на специалисти в Европа и САЩ е било насочено именно към програмисти, които владеят този език“ (Harizanov 2023). Така например според различни сайтове^{3: 4} към месец март 2023 г. езикът Python е на първо място по популярност и търсене.

В учебните програми по „Компютърно моделиране и информационни технологии“ от 2020 г.⁵ е заложено преминаване от език с блоково програмиране към скриптов текстов език по избор на преподавателя, като са посочени конкретно Python или JavaScript. В по-голямата част от учебниците, одобрени от МОН, разделът „Компютърно моделиране“ е реализиран с Python.

Средите за програмиране на Python предоставят множество практически възможности за програмиране, включително на игри (Pygame Zero), крипто-

графия и изкуствен интелект, обработка и визуализация на данни и др. Това го прави подходящ както за ученици от прогимназиален етап, така и от гимназиален етап, които не са изучавали до този момент програмен език или имат само базови знания по програмиране. Езикът подпомага и бързото изграждане на алгоритмично мислене.

Предложените задачи за обучение по програмиране са използвани в извънкласна форма на обучение с ученици от VIII и IX клас, които не изучават предмета информатика. Използван е игровият подход от гледна точка на обучение по програмиране чрез създаване на игри. Осъществява се със средствата на средата за разработка Mu на Python в режим Pygame Zero.

Първоначално Mu е създаден от Python Software Foundation за проекта micro:bit на BBC. Превръща се в среда за програмиране за начинаещи благодарение на подкрепата на Raspberry Pi Foundation. Mu е подходящ за ученици, които правят първите си стъпки в програмирането с Python, работи под Windows, OSX, Linux и Raspberry Pi. Лесен е за използване, предлага се на всички основни платформи и е добре документиран, дори и за начинаещи⁶. Mu е безплатен и може да бъде лесно свален и инсталиран от официалната страница на Mu⁷.

В примерните задачи се използва библиотека turtle. Графиката на костенурките е популярен начин за запознаване на начинаещи с програмирането и се базира на част от оригиналния език за програмиране LOGO. Библиотеката turtle предоставя графични примитиви за костенурки както по обектно ориентиран, така и по процедурно ориентиран начин. Чрез комбиниране на различни команди лесно могат да се нарисуват сложни форми и картини⁸. Основните функции за работа могат да бъдат намерени в учебниците за VI и VII клас, където има предложени и множество различни задачи. Тук няма да се спираме на функционалностите на библиотека turtle.

5. Примерни задачи за обучение по програмиране за начинаещи

В началото на обучението учениците се запознават със среда за програмиране Mu на Python и се предлагат задачи за изчертаване на геометрични фигури с помощта на библиотека turtle с цел да се усвоят основни оператори и програмни конструкции на езика Python.

5.1. Запознаване с редактор Mu на Python. Изчертаване на фигури

Основни цели, които трябва да се постигнат на този етап, са: ученикът да познава менюто на редактора; да може да отваря, затваря и запазва файл; да описва прост линеен алгоритъм с езика Python; да използва вградени функции за примитиви от библиотека turtle. Към тази тема на учениците могат да се предложат задачи за изчертаване на геометрични фигури – квадрат, равностранен триъгълник, правилни n -ъгълници и др.

5.2. Цветове и запълване на фигури. Рисуване с turtle

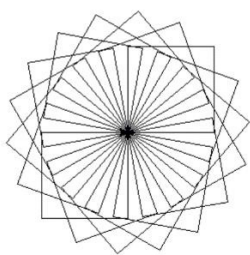
Постепенно, след изчертаване на елементарни фигури, се преминава към оператор за цикъл `for` и изчертаване на по-сложни фигури. Ето някои примерни задачи.

Задача 1. Начертайте фигурата (Фигура 1): А) оцветете страните в червено;

Б) запълнете фигурата с жълт цвят; В) променете ъгъла на завъртане.

Задача 2. Начертайте фигурата (Фигура 2). Използвайте оператор за цикъл `for`. Оцветете фигурата с цветове по свое желание.

Задача 3. „Баскетболна топка“⁴⁹. Начертайте баскетболната топка (Фигура 3).



Фигура 1



Фигура 2



Фигура 3

Основни цели, които трябва да се постигнат на този етап, са: ученикът създава променлива – обект на `turtle`; използва функциите за промяна на цвета и запълване на фигури от библиотека `turtle` – `fillcolor()`, `begin_fill()`, `end_fill()`; замества повтарящи се действия, като използва цикъл `for`; определя позиция и движи писалката – `pensize(arg)`, `forward(arg)`, `goto(arg1,arg2)`, `up()`, `down()`, `left()`, `right()`.

Методически бележки: със Задача 3 се разяснява действието на вградените функции и координатите на работното пространство – припомнят се координатна система и координати на точка (математика) и се уточнява, че координати (0,0) са в горния ляв ъгъл на работното пространство при графика с `turtle`. Като самостоятелна задача може да се предложи на учениците да нарисуват картина по свое желание, евентуално на милиметрова хартия за по-лесно ориентиране в координатната система.



Фигура 4

5.3. Изчертаване на по-сложни фигури. Потребителски функции

Основни цели на този етап са: прилагане на усвоените знания, умения и навици за инициализиране на променливи и използване на оператор за цикъл `for` за

описване на повтарящи се действия; използване на функцията `randint(arg1, arg2)` от библиотеката `Random`; използване на списък; използване на предварително създадени потребителски функции със задаване на параметри на функциите.

На учениците се представя предварително изходът от решението на следната задача.

Задача 4. Напишете програма, която изчертава на екрана правилни многоъгълници с различен цвят и брой на страните, както е показано на (Фигура 4). Използвайте списък за цветовете и потребителски функции за изчертаване на многоъгълниците и позицията им.

Реализация на задачата (Фигура 5):

```

1 import turtle
2 import random
3
4 turtle.Screen().bgcolor('light steel blue')
5 t = turtle.Turtle()
6 t.pensize(3)
7
8 def random_square(lenth, sides):      # Създава многоъгълника
9     for i in range(sides):
10         t.fd(lenth)
11         t.lt(360/sides)
12
13 def goto(x, y):                      # Позиция на фигурата
14     t.up()
15     t.goto(x, y)
16     t.down()
17
18 for i in range(50):
19     color = ['magenta', 'yellow', 'green', 'blue', 'orange', 'indigo', 'red', 'purple', 'cyan', 'deep pink']
20     t.pencolor(color[i % 10])
21     random_square(random.randint(10, 50), random.randint(3, 12))
22     goto(random.randint(-250, 250), random.randint(-250, 250))
23
24 if t.pos()[0] >= 250 or t.pos()[0] <= -250 :
25     goto(t.pos()[0]-100, t.pos()[1]-100)
26     continue
27 elif t.pos()[1] >= 250 or t.pos()[1] <= -250:
28     goto(t.pos()[0]-100, t.pos()[1]- 100)
29     continue

```

Фигура 5. Реализация на задача „Многоъгълници“

Методически бележки: в тази задача е използван типът „Списък“ за съхраняване на цветовете на многоъгълниците. Списъците се използват за съхраняване на множество елементи в една променлива и са един от четирите типа контейнери, съхраняващи много елементи в една променлива. Елементите на списъка могат да бъдат от различен тип.

Синтаксисът за деклариране на списък е:

<име на променлива> = [стойности, разделени със запетайи]

Пример за такава декларация е дефинирането на списъка `color` на Фигура 5.

Семантика: елементите на списъка са подредени и са индексирани – първият елемент има индекс 0, вторият елемент е с индекс 1 и т.н. Достъпът

до елемент на списъка се осъществява с името му и индекса на елемента. Например `color[2]='green'`. Добавянето на нов елемент в списъка става в края му. Учителят може да даде наготово конструкцията и да обясни синтаксиса и семантиката в практическата задача.

В тази задача се демонстрира и използването на библиотека `Random` и вградената функция `randint(arg1,arg2)`. Синтаксисът на извикването е:

`random.randint(начална стойност, крайна стойност)`

Например, `random.randint(10,50)`.

Семантика: параметрите на функцията са цели числа, а функцията връща като резултат „случайно“ генерирано цяло число между началната и крайната стойност на интервала. Всяко ново извикване на функцията ще връща различна стойност от диапазона. В примера резултатът от функцията ще бъде цяло число между 10 и 50.

Задача 5. Класическа игра „Лабиринт“. На учениците се предлага част от кода на игра тип „Лабиринт“ (Maze)¹⁰.

Основни цели: прилагане на усвоените знания, умения и навици за инициализиране на променливи, използване на оператори за цикъл, използване на условен оператор; използване на готови потребителски функции и предаване параметри на функция; използване на функции от `turtle` – `screen.clear()`, `screen.blit(tile,(x,y))`, `player.draw()` и от `Pygame` – `animate(player,duration=0.1,pos=(x,y))`, `enemy.colliderect(player)`; работа със списъци от данни и индексирание на елементите в списък; изграждане на умения за работа с клавиши за движение; изграждане на умения за работа със собствени и готови изображения; създаване на собствени изображения.

Етапи на създаване на играта

Първи етап (Фигура 6). Повечето от средите за програмиране работят с `gif` формат на изображенията, но `Му` и `Pygame` позволяват използване и на формата `png`. В тази задача всички файлове с изображения са с формат `png`.


```

1 import turtle
2
3 tile_size = 64
4 width = tile_size * 8
5 height = tile_size * 8
6
7 tiles = ['empty', 'wall', 'goal', 'door', 'key']
8 unlock = 0
9
10 maze = [
11     [1, 1, 1, 1, 1, 1, 1, 1],
12     [1, 0, 0, 0, 0, 0, 0, 1],
13     [1, 0, 1, 0, 1, 1, 4, 1],
14     [1, 0, 0, 0, 0, 0, 0, 1],
15     [1, 2, 1, 0, 1, 3, 0, 1],
16     [1, 0, 1, 0, 1, 1, 0, 1],
17     [1, 0, 0, 0, 0, 0, 0, 1],
18     [1, 1, 1, 1, 1, 1, 1, 1]
19 ]
20

```

Фигура 6. Първи етап на играта „Лабиринт“

Създава се лабиринт от отделни изображения (плочки), които се рисуват много пъти – `empty.png` (бяла плочка), `wall.png` (червена плочка) и `goal.png` (синя плочка). Създават се и още две изображения – за врата в лабиринта (`door.png`) и ключ за нея (`key.png`). Всяко от тях трябва да е с размер 64×64 пиксела. Записват се в `tu_code/images`. Имената на файловете на изображенията за плочките се съхраняват в списък `tiles`, а лабиринтът се съхранява като списък `maze` с 8 реда и 8 колони. В него стойност 0 отговаря на плочка `empty`, 1 отговаря на плочка `wall`, 2 – на плочка `goal`, 3 – на плочка `door`, а 4 – на плочка `key`, т.е. на индексите на елементите от списък `tiles`.

Втори етап (Фигура 7). За да се нарисува лабиринт, се използват вложени цикли `for`. Външният цикъл повтаря редовете, а вътрешният – колоните, т.е. елементите на всеки ред. Записват се във функция `draw()`.

```
24
25 def draw():
26     screen.clear()
27     for row in range(len(maze)):
28         for column in range(len(maze[row])):
29             x = column * tile_size
30             y = row * tile_size
31             tile = tiles[maze[row][column]]
32             screen.blit(tile, (x, y))
33     player.draw()
34     enemy.draw()
```

Фигура 7. Втори етап на играта „Лабиринт“

На учениците се предлага сами да определят къде в програмата трябва да добавят кода за създаване на играча и „врага“ (Фигура 8).

```
player = Actor("player", anchor=(0, 0), pos=(1 * TILE_SIZE, 1 * TILE_SIZE))
enemy = Actor("enemy", anchor=(0, 0), pos=(3 * TILE_SIZE, 6 * TILE_SIZE))
enemy.yv = -1
```

Фигура 8. Създаване на играча и „врага“

Трети етап (Фигура 9). Движение на играча и „врага“. Създава се функция `on_key_down(key)`.

```
36 def on_key_down(key):
37     # движение на играча
38     row = int(player.y / tile_size)
39     column = int(player.x / tile_size)
40     if key == keys.UP:
41         row = row - 1
42     if key == keys.DOWN:
43         row = row + 1
44     if key == keys.LEFT:
45         column = column - 1
46     if key == keys.RIGHT:
47         column = column + 1
48     tile = tiles[maze[row][column]]
49     if tile == 'empty':
50         x = column * tile_size
51         y = row * tile_size
52         animate(player, duration=0.1, pos=(x, y))
53     global unlock
54     if tile == 'goal':
55         print('Well done')
56         exit()
57     elif tile == 'key':
58         unlock = unlock + 1
59         maze[row][column] = 0 # 0 празна плочка
60     elif tile == 'door' and unlock > 0:
61         unlock = unlock - 1
62         maze[row][column] = 0 # 0 празна плочка
63
```

Фигура 9. Трети етап – функция `on_key_down(key)`

До този момент учениците не са използвали условен оператор. Чрез тази функция се обясняват синтаксисът и използването на оператора, както и използването на клавишите за движение. С конструкцията (Фигура 10): се описва движението на играча чрез стрелките за движение от клавиатурата: UP – нагоре, DOWN – надолу, LEFT – наляво, и RIGHT – надясно. Проверява се кой клавиш е натиснат, и се променя позицията на играча съответно по ред и стълб.

```

if key == keys.UP:
    row = row - 1
if key == keys.DOWN:
    row = row + 1
if key == keys.LEFT:
    column = column - 1
if key == keys.RIGHT:
    column = column + 1

```

Фигура 10. Движение на играча

Изяснява се и използването на вградената функция за автоматично движение `animate(player, duration=0.1, pos=(x, y))`, с помощта на която преместването на обектите става по-плавно. Първият параметър на функцията е обектът (героят), когото движим, вторият е продължителността на едно движение, а третият – текущата позиция на героя.

Към функцията `on_key_down(key)` се добавя и движението на „врага“ (Фигура 11).

```

64     # движение на врага
65     row = int(enemy.y / tile_size)
66     column = int(enemy.x / tile_size)
67     row = row + enemy.yv
68     tile = tiles[maze[row][column]]
69     if not tile == 'wall':
70         x = column * tile_size
71         y = row * tile_size
72         animate(enemy, duration=0.1, pos=(x, y))
73     else:
74         enemy.yv = enemy.yv * -1
75     if enemy.colliderect(player):
76         print('You died')
77         exit()

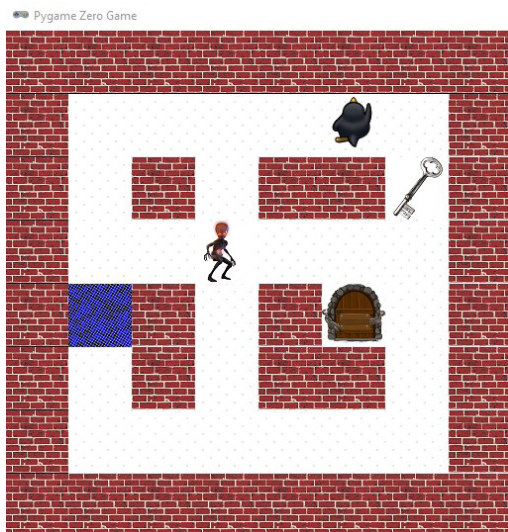
```

Фигура 11. Добавяне на движението на „врага“ към функция `on_key_down(key)`

В края на функцията се използва и вградената функция `enemu.colliderect(player)`. Функцията връща като резултат `true`, ако има сблъсък (удар) между играча и „врага“ и `false` – в противен случай.

Четвърти етап. Проверка и тестване на програмата. Функциите в задачата са предоставени наготово на учениците. Те сами коригират синтактичните грешки в кода и проиграват играта. Един възможен изход¹¹ е показан на Фигура 12.

Методически бележки: на учениците може да се предложи като самостоятелна работа да заменят плочките с готови изображения, да добавят още видове плочки към лабиринта: вода, скала, дърво; да добавят монети, които играчът събира, за да увеличи резултата; да добавят сандък със съкровища (вместо вратата), който се отключва с ключа и увеличава резултата, вместо да прекратят играта; да дадат на играча 3 живота; да покажат резултата на играча.



Фигура 12. Примерен изход от играта „Лабиринт“

На вниманието на учениците може да се предложат някои онлайн галерии със свободен достъп и/или без авторски права:

<https://opengameart.org/>, <https://krita.org/en/>

<https://itch.io/game-assets/free/tag-2d>

<https://www.vecteezy.com/free-vector/animation-sprite>

<https://www.freepik.com/free-photos-vectors/sprite-animation>

6. Заключение

Предвид използването на езика Python в VI и VII клас в модул „Компютърно моделиране“ по учебния предмет „Компютърно моделиране и информационни технологии“ по новите учебни програми, авторът е убеден, че предложените задачи, реализирани с език Python, режим Pygame Zero и чрез Mu Editor, могат да бъдат включени като надграждащи елементи в обучението по програмиране в гимназиален етап. Нещо повече, обучението може да продължи с използването на библиотека Pygame, която позволява да се създават напълно функционални игри и мултимедийни програми на езика Python. Задачите са подходящи за обучение както в непрофилираните гимназии, така и при начално обучение в профилирани училища.

БЕЛЕЖКИ

1. <https://cioo.mon.bg/>. https://cioo.mon.bg/wp-content/uploads/2014/07/naredba_7_11.08.2016_profilirana_podgotovka.pdf. https://cioo.mon.bg/wp-content/uploads/2014/07/naredba_7_11.08.2016_profilirana_podgotovka.pdf(02.03.2023)
2. Python a programming language. <https://brochure.getpython.info/media/releases/psf-python-brochure-vol.-i-final-download.pdf/view> (18.02.2023)
3. <https://www.tiobe.com/tiobe-index/> (05.03.2023)
4. <https://pypl.github.io/PYPL.html> (05.03.2023)
5. Национална стратегия за въвеждане на ИКТ в българските училища. <https://www.strategy.bg/StrategicDocuments/View.aspx?lang=bg-BG&Id=398> (02.03.2023)
6. <https://mu.readthedocs.io/en/latest/> (19.02.2023)
7. <https://codewith.mu/en/download> (19.02.2023)
8. Turtle. <https://docs.python.org/3/library/turtle.html> (19.02.2023)
9. Извлечено от <https://www.chegg.com/homework-help/questions-and-answers/please-draw-following-image-python-code-using-turtle-add-image-mean-draw-2d-example-drew-b-q46180382> (27.03.2022)
10. Извлечено от: <https://electronstudio.github.io/pygame-zero-book/chapters/maze.html> (15.03.2022)
11. Изображенията в задачата са със свободен достъп и са извлечени от <https://www.vecteezy.com/free-png/png>(15.03.2022)

ЛИТЕРАТУРА

АСЕНОВА, П., 1990. *Построение и использование системы задач в курсе алгоритмизации*. Автореферат диссертации, Russian Academy of Science, Москва.

- ГРОЗДЕВ, С., ГЪРОВ, К., 2008. За системите от опорни задачи при подготовката за участие в олимпиади по информатика. Комбинаторни обекти и алгоритми. *37 Пролетна конференция на СМБ, София*, 304 – 311, http://www.math.bas.bg/smb/2008_PK/2008/pdf/304-311.pdf (02.04.2022)
- ГЪРОВ, К., ТОДОРОВА, Е., 2006. Примерна система от опорни задачи по темата „Алгоритми и задачи от теория на числата“ за подготовка на талантиливи ученици по информатика. *35. пролетна конференция на СМБ, Боровец*, 374 – 380, http://www.math.bas.bg/~smb/2004_2007_PK/2006/pdf/374-380.pdf (02.04.2022)
- ГЪРОВ, К., 2000. Обучението по информатика и информационни технологии в средното училище – състояние и перспективи. *Юбилейна научна сесия – 30 години ФМИ, ПУ „Паисий Хилендарски“*, Пловдив, 28 – 37.
- ГЪРОВ, К., 2010. Задачите в обучението по информатика и информационни технологии. *Национална конференция „Образованието в информационното общество“*. Пловдив, <https://core.ac.uk/download/pdf/62659791.pdf> (04.04.2022)
- ДЕРНОВАЯ, С., 2008. Этапы конструирования системы задач по информатике. *Альманах современной науки и образования*. https://www.gramota.net/articles/issn_1993-5552_2008_7_21.pdf (29.03.2022)
- ДУРЕВА, Д., 2003. *Проблеми на методиката на обучение по информатика и информационни технологии*. Университетско издателство ЮЗУ „Неофит Рилски“, Благоевград.
- ДУРЕВА, Д., Касева, М. 2011. Компютърните образователни игри в обучението в началното училище. *Информационно-комуникационни технологии, медии и образование*, Благоевград.
- НИКОЛОВА Е., ТУПАРОВА, Д., 2018. Създаване на игри в часовете по информатика чрез използване на генератор на случайни числа, *Математика и информатика*, **61**(3), ISSN 1314-8532 (online) 1310-2230 (print), 232 – 258.
- СПИРОВА, М. 2018. Място на игрово базираното обучение в училище. *XI национална конференция „Образованието и изследванията в информационното общество“*, Пловдив, 41 – 46. <http://webcache.googleusercontent.com/search?q=cache:MDvQKGmK58MJ:sci-gems.math.bas.bg/jspui/bitstream/10525/2945/1/ERIS2018-book-p04.pdf+&cd=1&hl=bg&ct=clnk&gl=bg&client=firefox-b-d> (15.10.2021)
- ТЕРЗИЕВА, Т., ГОЛЕВ, А., СТАВРЕВ, С., 2017. Сериозните игри – иновативно средство за обучение. *Иновативни софтуерни инструменти*

- и технологии с приложения в научни изследвания по математика, информатика и педагогика на обучението, Пампорово, 107 – 114.
- ТУПАРОВА, Д., КАСЕВА, М., СТОЯНОВА, М.^b 2018. Педагогически аспекти при разработването на образователна компютърна игра за началното училище. *XI Национална конференция „Образованието и изследванията в информационното общество”*.
- ALEKSIĆ, V., IVANOVIC, M., POPESCO, E. & Budimac, Z., 2016. Commercial Off-the-Shelf Games as Learning Media. *17th International Conference on Computer Systems and Technologies 2016 – CompSysTech '16, Palermo, Italy*, 355 – 360.
- HARIZANOV, K., 2023. Several opportunities for implementing the training in „Computer modeling and information technologies” in 7th class. *Mathematics and Informatics*, **66**(1), 67 – 72, ISSN 1314-8532 (online) 1310-2230 (print).
- KIRRIEMUIR, J., McFarlane, A., 2004. Literature Review in Games and Learning., *NESTA Futurelab series*: report 8, 1 – 38. Available from: https://www.researchgate.net/publication/32231341_Literature_Review_in_Games_and_Learning (12.09.2021)
- TRYBUS, J., 2015. Game-Based Learning: What It Is, Why It Works, and Where It's Going., New Media Institute, Available from: <http://www.newmedia.org/game-based-learning--what-it-is-why-it-works-and-where-its-going.html> (29.09.2021)
- TUPAROVA, D., NIKOLOVA E. & TUPAROVA, E., 2020. Integrated Game-based Learning in an Informatics Secondary course: Is There a Difference between Girls' and Boys' Achievements?, *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU2020)*, 702 – 709, DOI: 10.5220/0009818407020709, ISBN: 978-989-758-417-6, <https://www.scitepress.org/Papers/2020/98184/98184.pdf> (09.03.2023)
- TUPAROVA, D., TUPAROV, G., VELEVA, V. & NIKOLOVA, E., 2018. Educational computer games and gamification in informatics and information technology education – Teachers' points of view. *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 842 – 847, ISSN 1847-3938, Извлечено от: <http://www.mipro.hr/LinkClick.aspx?fileticket=j9GUcUpxsNY%3D&tabid=196&language=hr-HR&fbclid=IwAR0cOh0XrwBgX1cx0cVxomYC Bmj-7zVABWh4w1MtvfjJrr7FvQzGnNFeiQQ> (09.03.2023)

REFERENCES

- ASENOVA, P., 1990. *Postroenie i ispolzovanie sistemi zadachi v kurs algoritimizacii*, Avtoreferat disertacii, Russian Academy of Science, Moskva. [in Russian]
- GROZDEV, S., Garov, K., 2008. Za sistemite ot oporni zadachi pri podgotovkata za uchastie v olimpiadi po informatika. Kombinatorni obekti i algoritmi, Sofia, 37 *Proletna konferentsia na SMB*. 304 – 311, Available from: http://www.math.bas.bg/smb/2008_PK/2008/pdf/304-311.pdf [in Bulgarian] (02.04.2022)
- GAROV, K., TODOROVA E., 2006. Primerna sistema ot oporni zadachi po temata “Algoritmi i zadachi ot teoria na chislata” za podgotovka na talantlivi uchenitsi po informatika, Borovets, 35. *proletna konferentsia na SMB*, 374-380. Available from: http://www.math.bas.bg/~smb/2004_2007_PK/2006/pdf/374 – 380.pdf [in Bulgarian] (02.04.2022)
- GAROV, K., 2000. Obuchenieto po informatika i informatsionni tehnologii v srednoto uchilishte – sastoyanie i perspektivi. *Yubileyna nauchna sesia – 30 godini FMI*, PU “Paisiy Hilendarski”, Plovdiv. [in Bulgarian]
- GAROV, K., 2010. Zadachite v obuchenieto po informatika i informatsionni tehnologii, Plovdiv, *Natsionalna konferentsia “Obrazovaniето v informacionnoto obshtestvo”*. Available from: <https://core.ac.uk/download/pdf/62659791.pdf> [in Bulgarian] (04.04.2022)
- DERNOVAYA, S., 2008. Etapi konstruirovania sistemi zadach po informatike. Almanah sovremennoy nauki i obrazovaniya. Available from: https://www.gramota.net/articles/issn_1993-5552_2008_7_21.pdf. [in Russian] (29.03.2022)
- DUREVA, D., 2003. *Problemi na metodikata na obuchenie po informatika i informatsionni tehnologii*. Neofit Rilski, Blagoevgrad. [in Bulgarian]
- DUREVA, D., & KASEVA, M., 2011. Kompyutarnite obrazovatelni igri v obuchenieto v nachalnoto uchilishte. *Informatsionno-komunikacionni tehnologii, medii i obrazovanie*, Blagoevgrad. [in Bulgarian]
- NIKOLOVA E., TUPAROVA, D., 2018. Creating of Games in the Informatics Classes by Using a Generator of Random Numbers, *Mathematics and Informatics*, 61(3), ISSN 1314-8532 (online) 1310 – 2230 (print), 232 – 258 [in Bulgarian]
- SPIROVA, M., 2018. Myasto na igrovo-bazironoto obuchenie v uchilishte. Plovdiv, *XI natsionalna konferentsia “Obrazovaniето i izsledvaniyata v informatsionnoto obshtestvo”*, 41 – 46. Available from: <http://webcache.googleusercontent.com/search?q=cache:MDvQKGmK58MJ:sci-gems.math.bas.bg/jspui/bitstream/10525/2945/1/ERIS2018-book-p04.pdf>

- +&cd=1&hl=bg&ct=clnk&gl=bg&client=firefox-b-d [in Bulgarian] (15.10.2021)
- TERZIEVA, T., GOLEV, A., STAVREV, S., 2017. Serioznite igri – inovativno sredstvo za obuchenie. Pamporovo, *Inovativni softuerni instrumenti i tehnologii s prilozhenia v nauchni izsledvania po matematika, informatika i pedagogika na obuchenieto*. 107 – 114. [in Bulgarian]
- TUPAROVA, D., KASEVA, M., STOYANOVA, M.^b, 2018. Pedagogicheski aspekti pri razrabotvaneto na obrazovatelna kompyutarna igra za nachalното uchilishte. *XI Nacionalna konferentsia “Obrazovaniето i izsledvaniyata v informatsionното obshtestvo”*. [in Bulgarian]
- ALEKSIĆ, V., IVANOVIC, M., POPESCO, E. & Budimac, Z., 2016. Commercial Off-the-Shelf Games as Learning Media. *17th International Conference on Computer Systems and Technologies 2016 – CompSysTech '16, Palermo, Italy*, 355 – 360.
- HARIZANOV, K., 2023. Several opportunities for implementing the training in „Computer modeling and information technologies” in 7th class. *Mathematics and Informatics*, **66**(1), 67 – 72, ISSN 1314-8532 (online) 1310-2230 (print).
- KIRRIEMUIR, J., McFarlane, A., 2004. Literature Review in Games and Learning., *NESTA Futurelab series: report 8*, 1 – 38. Available from: https://www.researchgate.net/publication/32231341_Literature_Review_in_Games_and_Learning (12.09.2021)
- TRYBUS, J., 2015. Game-Based Learning: What It Is, Why It Works, and Where It’s Going., New Media Institute, Available from: <http://www.newmedia.org/game-based-learning--what-it-is-why-it-works-and-where-its-going.html> (29.09.2021)
- TUPAROVA, D., NIKOLOVA E. & TUPAROVA, E., 2020. Integrated Game-based Learning in an Informatics Secondary course: Is There a Difference between Girls’ and Boys’ Achievements?, *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU 2020)*, 702 – 709, DOI: 10.5220/0009818407020709, ISBN: 978-989-758-417-6, Available from: <https://www.scitepress.org/Papers/2020/98184/98184.pdf> (09.03.2023)
- TUPAROVA, D., TUPAROV, G., VELEVA, V. & NIKOLOVA, E.^a, 2018. Educational computer games and gamification in informatics and information technology education - Teachers’ points of view. *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 842 – 847, ISSN 1847-3938, Available from: <http://www.mipro.hr/LinkClick.aspx?fileticket=j9GUcUpxsNY%3D&tabid=>

196&language=hr-HR&fbclid=IwAR0cOh0XrwBgX1cx0cVxomYCBmj-
7zVABWh4w1MtvfjJrr7FvQzGnNFeiQQ (09.03.2023)

GAME-BASED PROGRAMMING TRAINING FOR BEGINNERS IN THE PYGAME ZERO ENVIRONMENT - SAMPLE ASSIGNMENTS WITH PYTHON TURTLE GRAPHICS

Abstract: In recent years, an increasing digitalization has been observed in all spheres of life on a global scale. The demand for specialists in the field of ICT requires their preparation to start at school with the study of programming and IT. Good practices and increasing interest in game-based learning are observed at all levels of secondary education. Different environments, programming languages and learning approaches is used in the Computer Modeling and Informatics disciplines. The focus of the article are sample tasks and methodical implementation for non-profiled training in programming for high school students by creating games. The terminology related to game-based learning and the role of tasks in mastering learning content are presented. The main focus is on creating fun games implemented in Python programming language using turtle graphics in 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 training in programming in profiled and non-profiled high schools are discussed.

Keywords: educational computer games; gamification; game-based learning; programming; Python programming language; turtle graphics; 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