

# Міні-проект Donut Dream

## 1) Опис гри

Події гри Donut Dream відбуваються частково в реальному світі та у сні головного героя. Розпочинається гра проханням ввести гравця його нікнейм, після чого розпочинається перший акт(Всього їх три)

### Перший акт

Гра розпочинається і головний герой розуміє, що у нього закінчилось усе солодке, а особливо пончики, які він обожає. У гравця з'являється вибір: або піти сумним та розбитим спати, або піти в магазин та купити щось солоденьке. Якщо гравець іде купляти щось солодке то він може або вдягнути куртку, або ні і втратити 10/100 здоров'я. Пізніше, гравець отримує вибір що купити в магазині: пончик або шоколадку. Залежно від цього вибору він отримує більше або менше додаткових очків. Гравець повертається додому, і з'являється перша міні-гра, де йому потрібно перевести цифри з бінарних в звичайні щоб потрапити до себе в кімнату за три спроби, після чого код змінюється.

### Другий акт

Гравець засинає і бачить перед собою розбиту та наполовину з'їджену вафлю в уявному світі здорової їжі. У гравця є 4 вибори: або поговорити з нею, або не реагувати на неї, або вбити її. Усі варіанти вибору приводять героя до замку та до фінального боса, проте якщо гравець обирає вбити вафлю то він отримує або додатковий урон або збільшує собі максимальну кількість здоров'я, втративши при цьому якусь його частину під час битви. Якщо гравець не вибирає вбити вафлю, то головний герой отримує колу, яка регенерує 25 здоров'я гравця, зразу опісля гра рекомендує гравцеві її використати, адже гравець падає з обриву і втрачає 50 очків здоров'я. Гравець може не використовувати її, тоді в кінці гри, якщо гравець виживе, йому будуть нараховані додаткові 200 очків

### Третій акт(бос)

Усі варіанти вибору гравця приводять його до замку, де він буде змагатись з першим босом. У нього є 100 здоров'я та його урон – 30.

Гравець може частково відбивати атаку, якщо він відгадає, яке число було помножене на дане число щоб вийшло показане число за 3 спроби.

Гравець може нанести у два рази більше урону якщо відгадає випадкове число від 0 до 150 за 10 спроб. Після кожної спроби гра буде підказувати гравцю чи менше чи більше загадане число від введеного

Після кожних двох атак та захистів, у гравця є можливість відновити частково своє здоров'я, вгадавши відповідь до легеньких загадок. Гравець відновлює або 25 або 50 здоров'я, залежно від того, що йому випаде.

Уся битва з босом знаходиться в циклі, який переривається або коли гравець помер, або коли гравець вбив боса.

## 2) Опис того, як працює гра, приклади

```
===      WELCOME      ===  
To the RPG game Donut Dream!  
You will now experience the sugar overdose and chill vibes!  
===      Press enter      ===  
  
Please, enter your name:  
>>> █
```

Гравця зустрічає даний інтерфейс та прохання ввести нікнейм

```
Your depression strikes because you can't eat anything sweet  
Press Enter to see your actions:  
  
1. Go to sleep (start the game)  
2. Go outside  
  
Enter the number of an action:  
>>>  
  
Please, enter your choise again:  
>>> █
```

Ось, наприклад, вибір гравцем дії. Якщо гравець не ввів нічого або вводить не варіанти відповіді, а текст, то вивидиться поле, яке просить ввести варіант ще раз

```
The door generates a random password. It consists of four numbers
Guess the password if it is printed in binary
```

```
0100
```

```
1000
```

```
1000
```

```
1001
```

```
Enter the password where each number is seperated with space:
```

```
█
```

Приклад гри з відгадуванням паролю до дверей, переводячи бінарні числа в натуральні.

```
You are sleeping, and in your dream there is something strange. You feel everything
The result of the first act is:
```

```
Your points: 100
```

```
Your health: 100/100
```

```
Your damage: 10
```

```
You are moving on█
```

Після кожного з трьох актів гравцю виводиться його статистика

```
He attacks you and you need to pull off your shield
```

```
Guess what number was multiplied to activate your shield!
```

```
13 * ? = 104
```

```
Enter your guess:
```

```
>>>
```

```
You did not enter a number. Please try again!
```

```
>>> █
```

Міні-гра під час битви з босом на вгадування числа на яке було помножене число

Now, YOU attack!

Quick, guess the right number from 0 to 150!

Enter your guess:

>>> 75

You have 9 attempts left

You didn't enter the right number. The number is Smaller

>>> 35

You have 8 attempts left

You didn't enter the right number. The number is Smaller

>>> 25

You have 7 attempts left

You didn't enter the right number. The number is Smaller

>>> 10

You have 6 attempts left

You didn't enter the right number. The number is Bigger

>>> 15

You have 5 attempts left

You didn't enter the right number. The number is Smaller

>>> 14

Приклад міні-гри з відгадуванням числа від 0 до 150

Now, you can get a health boost!

Answer the riddle to get either a coke or a pepsi!

Enter "True" if you think that the answer is True and "False" if you think that the answer is False

There are just a few of galaxies in our Universe

Enter your guess:

>>> False

The guess was right!

Приклад загадки для відновлення здоров'я

The boss has: 0 health remaining

=== IMPOSSIBLE! YOU COULDN'T HAVE WON ===

=== YOU ARE JUST A DONUT! ===

=== IT IS NOT THE END! ===

===

===

===

=== TO BE CONTINUED! ===

=== Thanks for playing! ===

Your result is: 550 points

## Консоль після виграшу гравця

### 3) Процес розробки

Розпочалось все з ідеї – зробити щось незвичне. Коли я придумував дану ідею, я дуже сильно хотів чогось солоденького, і мені чомусь на думку спали пончики. І далі я продовжив ідею, а події відбуваються в основному уві сні, тому що я в той час дуже хотів спати.

Потім я почав малювати маркерами на дошці у себе в кімнаті структуру та пізніше переносив все це в нотатки, насичуючи репліками, а опісля переносив на python.

І в кінці-кінців було тестування гри, під час якого було виправлено багато помилок та багів.

### 4) Опис структури гри

У грі наявні 10 функцій, 3 з яких є вкладеними в іншій. Розпочинається код ф-цією `start_game()`. У ній гравець вводить свій нікнейм, а система впевнюється, що ім'я було введено.

Наступна функція – `rand_num_game()`. Функція генерує випадковий пароль, де кожна цифра – двійковий бінарний код. Гравцеві треба відгадати пароль, написавши кожен цифру через пробіл.

Бінарний код – код, де кожна наступна цифра(0 чи 1) починаючи справа показує, чи добавляти до числа два в степені починаючи з нуля і добавляючи один до степеня з кожним кроком вправо. 1 – добавляти, 0 – ні. У гравця є три спроби, після чого пароль змінюється.

Функція `first_act(health, points, nickname)`. Ця функція представляє гравцю сюжет першої глави гри, приймає нік, здоров'я та очки гравця, повертає після закінчення своєї роботи здоров'я та очки гравця.

Функція `second_act(max_health, health, damage, points, coke)`. Функція представляє гравцю продовження сюжету гри, приймає значення статистик гравця(урон, максимальна к-сть здоров'я та саме здоров'я, очки), а також таку змінну як coke. В один момент гри гравець отримує колу, яка може добавити здоров'я гравцю, або він може зберегти її, за що в кінці гри він отримає додаткові бали. Функція змінює статистику гравця залежно від того, який вибір він робив та повертає статистику.

Функція `the_boss(max_health, health, damage, points)`. Функція приймає статистику гравця та представляє битву з босом гравцю. Якщо гравець перемагає, то він отримує виграшний екран, привітання з перемогою та кількість зароблених очків.

Якщо ж гравець помирає, то він отримує кількість очків та гра зупиняється.

У середині функції `the_boss()` є три функції, які забезпечують якісний та безперервний босфайт.

Функція `your_damage(is_more_damage)` дає гравцеві можливість відгадати загадане число від 0 до 150 за 10 спроб і при успішному відгадуванні числа ф-ція повертає значення `True` в `is_more_damage`, що дозволяє гравцю нанести більше урону. Якщо не відгадав, то `False`

Функція `reflection(is_active_sheild)` дає гравцю можливість відгадати, на яке число було помножене дане, щоб отримати певний результат за три спроби. У разі успіху, головний герой отримує менше урону, тобто функція повертає значення `True`. Якщо не відгадав, то `False`

Функція `health_boost(health)` дозволяє гравцю під час битви відновити певну кількість здоров'я, якщо гравець відгадає з першої спроби відповідь на загадку (`True or False`). Вона випадковим чином після успішного відгадування загадки повертає або `soke(+ 25 здоров'я)` або `pepsi(+ 50 здоров'я)`. Якщо гравець неправильно відповів на загадку – додаткового здоров'я не отримує. Функція повертає значення здоров'я (`health`)

Функція `results(points, max_player_health, player_health, player_damage)` функція приймає значення статистик гравця та виводить їх в кінці кожного акту

Функція `main()` запускає всі функції у правильному порядку

У грі використовується велика кількість циклів та `if-else` функцій. В основному використовується цикл `while` при кожній можливості гравця вибрати варіант продовження гри, для отримання від гравця правильного вибору, а не рандомного набраного тексту, також `while` використовується під час фінальної битви з босом. Цикли `for` використовуються, коли потрібно порахувати, чи використав користувач усі спроби під час міні-ігор, якщо їх у нього кінчна кількість