

Создание игры «Свеча» на языке программирования Python

Маринчук Александр Сергеевич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Создание игр в современном мире является достаточно популярной профессиональной сферой деятельности людей. Развитие информационных технологий позволило производить игровые продукты высоко качества и уже нельзя утверждать, что разработкой игр могут заниматься только люди с соответствующим образованием. Каждый человек, которому интересна данная тема, может установить один из множества доступных инструментов и создать собственную игру, начиная от простых крестиков-ноликов и заканчивая масштабными 3D проектами. В данной статье рассмотрена разработка игры кликера на языке программирования Python.

Ключевые слова: кликер, игра, python, свеча, tkinter.

Creating a game "Candle" in the Python programming language

Marinchuk Alexander Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

Game creation in the modern world is a fairly popular professional field of human activity. The development of information technology has made it possible to produce high quality gaming products and it can no longer be argued that only people with the appropriate education can be engaged in the development of games. Anyone interested in this topic can install one of the many available tools and create their own game, ranging from simple tic-tac-toe to large-scale 3D projects. This article discusses the development of a clicker game in the Python programming language.

Keywords: clicker, game, python, candle, tkinter.

1. Введение

1.1 Актуальность исследования

Создание игр в современном мире является достаточно популярной профессиональной сферой деятельности людей. Развитие информационных технологий позволило производить игровые продукты высоко качества и уже нельзя утверждать, что разработкой игр могут заниматься только люди с соответствующим образованием. Каждый человек, которому интересна

данная тема, может установить один из множества доступных инструментов и создать собственную игру, начиная от простых крестиков-ноликов и заканчивая масштабными 3D проектами. Одним из главных качеств, которым должен обладать программист, который берется за создание игр, является умение поставить себя на место игрока и выявить все недостатки, с которыми может столкнуться пользователь в процессе игры. Именно взгляд со стороны в качестве играющего поможет понять и представить конечный продукт в наиболее интуитивно понятном и приятном для игры виде.

1.2 Обзор исследований

М. М. Белолобский и М. А. Сорочинский в своей статье рассматривают функциональные особенности модулей (библиотеки) Pygame языка программирования Python, которые позволяют создавать 2D игры, мультимедийные приложения и выступают в роли «игрового движка», как программной среды для разработки игр [1]. В работе S. Watkiss описывается создание нескольких простых игр на Python для того, чтобы понять структуру разработки и проиллюстрировать различные техники программирования [2]. J. R. Parker в своей книге рассматривает основные идеи и некоторые дизайнерские приемы при создании игр на языке программирования Python. Также он подчеркивает программные особенности, которые могут стать полезными при разработке игровых проектов [3]. Создатели новых информационных технологий заинтересованы во всём большем потреблении новых продуктов, не задумываясь о результатах. Яркий пример - компьютерные игры, которые уже считаются неотъемлемой частью современной культуры, однако, являются ли они благом, способствуют ли они социализации и укреплению коммуникации между людьми? В своем исследовании Е. Ш. Ефимова предлагает еще один подход к ответам на эти вопросы и подчеркивает важность осознания данной проблемы специалистами в области информационных технологий [4]. За недолгое время своего существования гейм-дизайн произвел на свет множество бессмертных шедевров. За последние годы игровая индустрия поднялась на небывалые высоты, принося миллиарды долларов создателям игр. Но как проходит сам процесс создания игры от идеи до продаж? В статье А. Д. Бирюковой и М. В. Новиковой рассматриваются все этапы создания компьютерной игры [5]. Описал создание обучающей 3D-игры по химии, которая способствует познавательному интересу учащихся к урокам химии в своей работе А. З. Хатуев [6].

1.3 Цель исследования

Целью данного исследования является создание игры «Свеча» на языке программирования Python.

1.4 Суть игры

На данный момент довольно популярным жанром игры на просторах интернета являются игры-кликеры. Они подразумевают нажатия кнопки

мышь для набора очков или просто выживания, это так называемые «времяубивалки», которых существует огромное множество. Данная игра основана на принципе «Не дай сгореть свече» и подразумевает нажатие кнопки для увеличения продолжительности горения свечи. Если игрок будет нажимать кнопку слишком рано, то свеча горит быстрее, а если свеча полностью сгорит, то игра окончена.

2. Методы исследования

Для начала скачаем Python [7] с официального сайта и интегрированную среду разработки Pycharm [8] и создадим пустой проект. Далее подключим библиотеку tkinter для создания графических окон, которая нарисует окно, в котором будет располагаться сама игра. Также объявим необходимые переменные, зададим размеры окна, создадим метки и кнопку, загрузим в переменные нужные изображения и зададим нажатие клавиши Enter как начало игры (рис. 1).

```
from tkinter import *

okToPressReturn = True

candle = 100
day = 0

root = Tk()
root.title("Свеча")
root.geometry("600x800")

no_candle = PhotoImage(file="candle2.png")
candle1 = PhotoImage(file="candle.png")
endgame = PhotoImage(file="endgame.png")

startLabel = Label(root, text="Нажмите Enter, чтобы начать игру", font=("Times New Roman", 14))
candleLabel = Label(root, text="Свеча:" + str(candle), font=("Times New Roman", 14))
dayLabel = Label(root, text="День:" + str(day), font=("Times New Roman", 14))
candle_normal = Label(root, image=candle1)

button_nocandle = Button(root, text="Клики на меня", command=change)

startLabel.pack()
candleLabel.pack()
dayLabel.pack()
candle_normal.pack()
button_nocandle.pack()

root.bind('<Return>', start)
root.mainloop()
```

Рисунок 1 – Создание проекта

Далее объявим функцию start, которая будет работать если переменная okToPressReturn равна True (рис. 2).

```
def start(event):
    global okToPressReturn
    global candle
    global day
    if okToPressReturn == False:
        pass
    else:
        candle = 100
        day = 0
        startLabel.config(text="Не дайте свече сгореть!")
        updateCandle()
        updateDay()
        updateDisplay()
```

Рисунок 2 – Функция start

Следующим шагом будет описание трех функций, которые задействованы при старте программы: `updateCandle`, `updateDay` и `updateDisplay`, которые служат для убавления значения метки свечи, убавления значения метки дня и обновления дисплея в целом соответственно (рис. 3).

```
def updateCandle():
    global candle
    candle -= 1
    if isAlive():
        candleLabel.after(500, updateCandle)

def updateDay():
    global day
    day += 1
    if isAlive():
        dayLabel.after(5000, updateDay)

def updateDisplay():
    global candle
    global day
    if candle != 0:
        if candle <= 50:
            candle_normal.config(image=no_candle)
            candleLabel.config(text="Свеча:" + str(candle))
            dayLabel.config(text="День:" + str(day))
            candle_normal.after(100, updateDisplay)
        else:
            candle_normal.config(image=candle1)
            candleLabel.config(text="Свеча:" + str(candle))
            dayLabel.config(text="День:" + str(day))
            candle_normal.after(100, updateDisplay)
```

Рисунок 3 – Функции `updateCandle`, `updateDay` и `updateDisplay`

Последним шагом будет объявление функций `change` и `isAlive`, которые отвечают за горение свечи и состояние игры на текущий момент (рис. 4).

```
def change():
    global candle
    if isAlive():
        if candle <= 70:
            candle += 10
        else:
            candle -= 15

def isAlive():
    global candle
    global day
    global okToPressReturn
    if candle <= 0:
        startLabel.config(text="Игра окончена, нажмите Enter, чтобы сыграть снова")
        candle_normal.config(image=endgame)
        root.bind('<Return>', start)
        return False
    else:
        return True
```

Рисунок 4 – Функции change и isAlive

Запустим программу и проверим ее работоспособность (рис. 5).

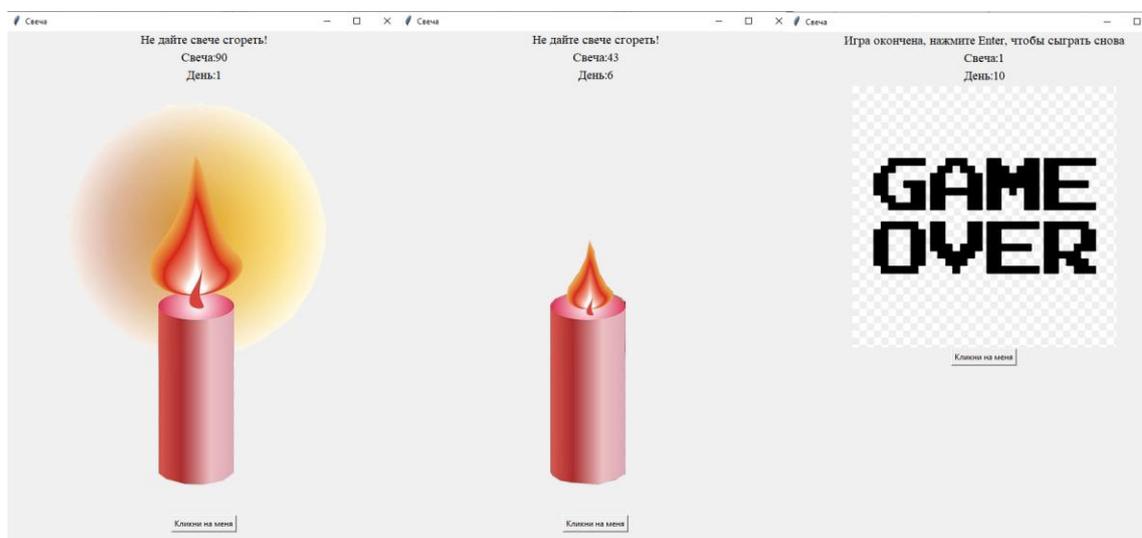


Рисунок 5 – Пример работы игры

3. Выводы

Создание игр достаточно интересная и познавательная область информационных технологий, которая в настоящее время стремительно набирает обороты, а специалисты в этой области становятся все востребованнее на рынке труда. С нынешними возможностями интернета почти каждый человек может заняться разработкой игр или приложений и набраться опыта для построения своей будущей профессиональной карьеры.

В данной статье была рассмотрена разработка игры «Свеча» на языке программирования Python.

Библиографический список

1. Белолобский М. М., Сорочинский М. А. Изучение модуля "Pygame" для создания игр на языке программирования Python // Современное образование: традиции и инновации. 2020. №2. С. 220-222.
2. Watkiss S. Creating Computer Games //Beginning Game Programming with Pygame Zero. – Apress, Berkeley, CA, 2020. С. 1-10.
3. Parker J. R. Game Development Using Python. Stylus Publishing, LLC, 2018.
4. Ефимова Е. Ш. Разработка компьютерных игр как этическая проблема // Научные коммуникации. Научная этика. Инженерная этика. Омск: Омский государственный технический университет, 2017. С. 17-21.
5. Бирюкова А. Д., Новикова М. В. Жизненный цикл разработки компьютерных игр // Вестник научных конференций. 2016. №12-4. С. 24-25.
6. Хатуев А. З. Разработка компьютерных игр. Помощь учителю // Весенний школьный марафон. Чебоксары: Общество с ограниченной ответственностью "Центр научного сотрудничества "Интерактив плюс", 2017. С. 17-22.
7. Язык программирования Python URL: <https://www.python.org/> (дата обращения: 03.09.2020).
8. Pycharm URL: <https://www.jetbrains.com/ru-ru/pycharm/> (дата обращения: 03.09.2020).