**Государственное бюджетное общеобразовательное учреждение города Москвы "Школа № 1564 имени Героя Советского Союза А. П. Белобородова"**

**Отчет по лабораторной работе №**

по дисциплине «Программирование»

Выполнил: ученик 11Б класса

Проверила:

Изотова Анастасия Андреевна

Москва

2022

**Задание**

Создать свой тамагочи.

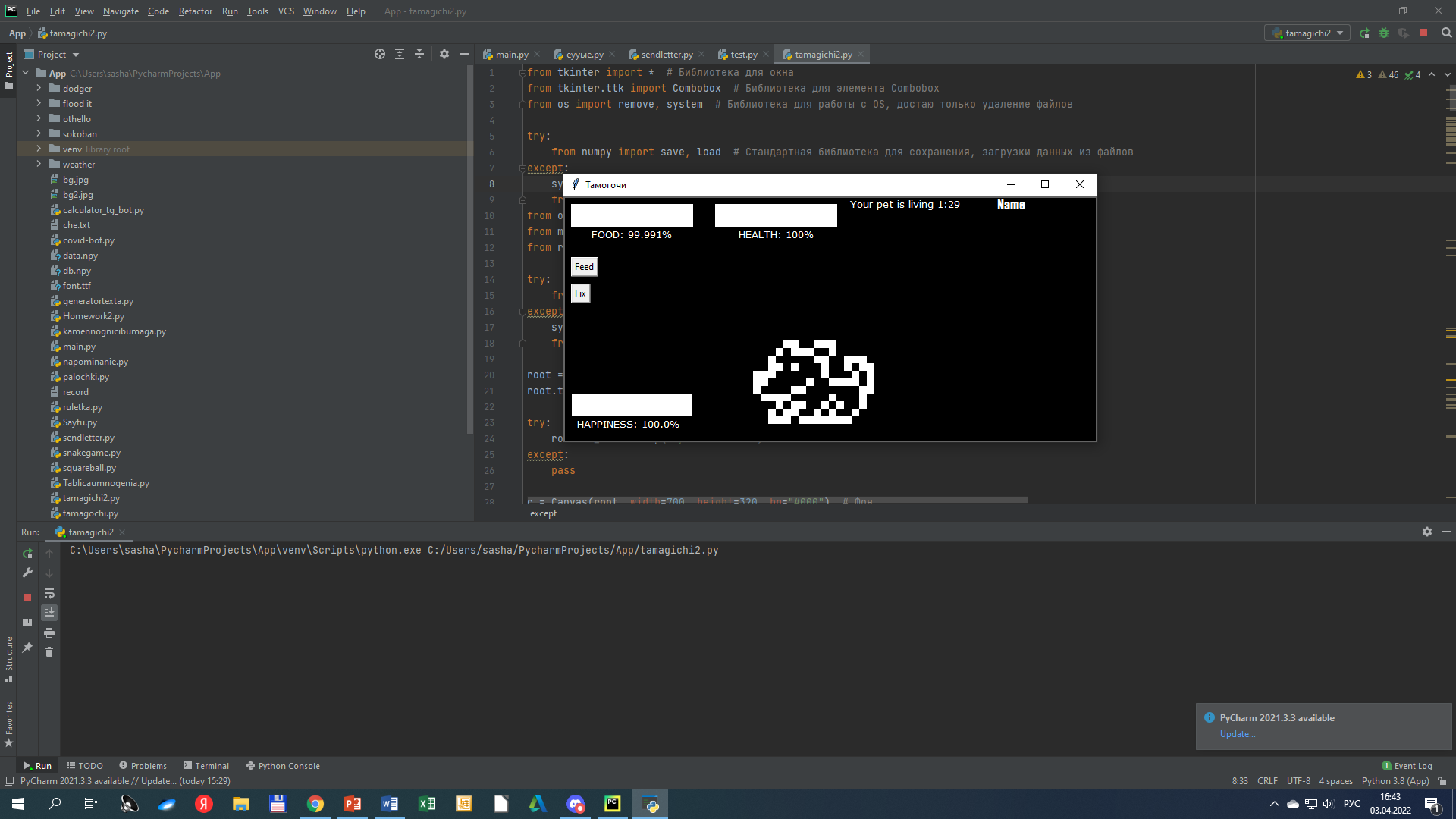
**Код программы**

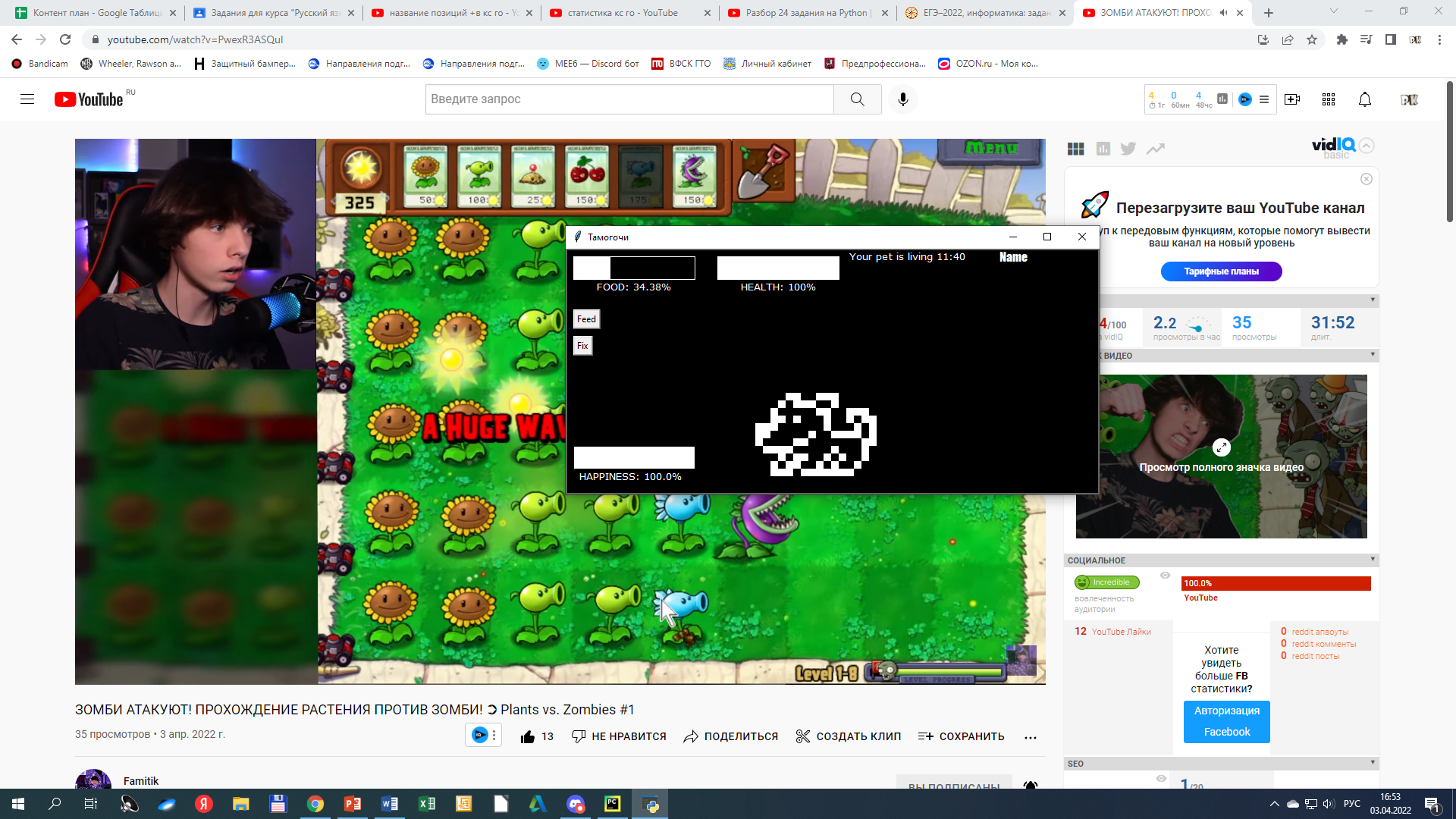
from tkinter import \* # Библиотека для окна  
from tkinter.ttk import Combobox # Библиотека для элемента Combobox  
from os import remove, system # Библиотека для работы с OS, достаю только удаление файлов  
  
try:  
 from numpy import save, load # Стандартная библиотека для сохранения, загрузки данных из файлов  
except:  
 system("pip3 install numpy")  
 from numpy import save, load  
from os.path import exists # Библиотека для работы с файловой системой  
from math import floor # Библиотека для работы с числами, достаю только округление  
from random import random # Библиоетка случайных чисел  
  
try:  
 from requests import post # Библиотека для работы с запросами (Для рекордов)  
except:  
 system("pip3 install requests")  
 from requests import post  
  
root = Tk()  
root.title("Тамогочи")  
  
  
c = Canvas(root, width=700, height=320, bg="#000") # Фон  
c.grid()  
c.focus\_set()  
  
# BITMAP'Ы ЖИВОТНЫХ  
animals = {  
 "cat": '0000000000000000000000000000000000000000000000001110000011100011100111110010001110100000101000110101000101001101010100010111101001000100010001000010101010000010000111111010001000010001001110100010011001101010001110111001110000000000000000000000000000000000',  
 # Кот  
 "horse": '0000000000000000000000000000000000000000000000000000111100000000000111011000000001100011100000001001001111001110100000011111111101110000000001110000100100000011000011010010011100010101111010110001010101011010000110110111011000000000000000000000000000000000',  
 # Лошадь  
 "pig": '0000000000000000000000000000000000000000000000000000110011000000000101110100000000100000001100100011001110001001011110010000010110000100000001011101010000000010100001000010001001111001010000100001011101100100000110011001100000000000000000000000000000000000',  
 # Свинья  
 "dog": "0000000000000000000000000000000000000000000000000000110011100000000101110010000000100000110011100011010001001011111000000100010111000001001111011000011000000011011110000010001000010110010100100010110010100100001110111111100000000000000000000000000000000000",  
 # Собака (Корги)  
 "hen": "0000000000000000000000000000000000000000000000000111100000000000011111000000000001000100000000001101010000011000111001011111010000010010001001000001000111001000000010000001000000000111111000000000001001000000000001101100000000000000000000000000000000000000",  
 # Курица  
 "turtle": "0000000000000000000000000000000000000000000000000000000000000000000000000000000000110001111100000100101010101000101011111111110010001010101010100110011111111110000101101010101000001111111111110000010010100100000001110001110000000000000000000000000000000000",  
 # Черепаха  
 "rabbit": "0000000000000000000000000000000000000000000000000000111111000000000100001000000000100111111000000101100001000000100000111001100010100111111001001000010000010100110000000000100001000010110010000100011100010000011110011110000000000000000000000000000000000000",  
 # Кролик  
 "koala": "0000000000000000000000000000000000000000000000001111000000111100010011111100100011010000010011000110100100110000001000000010011000101100001110010001110001000101000011111000001000010000001100100001001001010010000011011110110000000000000000000000000000000000",  
 # Коала  
 "elephant": "0000000000000000000000000000000000000000000000000001111000000000001000011000000001000010011100000100100001001000100000100100010010001010010001001001010110000100101010000000010010110101010101001001010101010100011001101011100000000000000000000000000000000000",  
 # Слон  
 "beagle": "0000000000000000000000000000000000000000000000000011100000000000010001100000011001010001000010101000010010010100110001001111100010001011100010000111000000001000001010011101010001011101010101000111011001101100000000000000000000000000000000000000000000000000",  
 # Бигль  
 "lion": "0000000000000000000000000000000000000000000000000010111000000000011111111000000011111111110001101110001110000110101101101100001111010101111110011100000111001101111010111000011001111111101101000011111101110100000111011101110000000000000000000000000000000000",  
 # Лев  
 "panda": "0000000000000000000000000000000000000000000000000110000011000000111111111110000011100000111000000101101101000110010110110100011001000100011111100010000011100001011111111110011101100001110011111111111111111111111001110011110000000000000000000000000000000000",  
 # Панда  
 "penguin": "0000000000000000000000000000000000000000000000000000111110000000000110001100000000010010110000000011100011100000011110011110000000010000111110000011000001111100011100000011111000011000001100000000111111110000001111001110000000000000000000000000000000000000",  
 # Пенгвин  
 "whale": "0000000000000000000000000000000000000000000000000011011000110010010010010011111100001000000111110011111000001110011111111000111011101111110111101111111111111110111111111111110011111101111111000111111000111000001111111111000000000000000000000000000000000000",  
 # Кит  
 "sheep": "000000000000000000000000000000000000000000000000000011011100000000110010001101100111100000001001111101000000010111011100000001101111000000000100111000000000010000101001010010000011111111111000001101101111000000000000000000000000000000000000"  
 # Баран  
}  
#  
  
poop = "0100100110010010010010011000001000011000001111000111111011111111" # Рисунок говна (8x8)  
  
  
def Bool(bool):  
 if bool == "True":  
 return True  
 elif bool == "False":  
 return False  
 else:  
 return -1  
  
  
class Animal(object):  
 def \_\_init\_\_(self, name, type, food, health, time, happiness, timeToShit, isShit):  
 # ПЕРЕМЕННЫЕ  
 self.name = name  
 self.food = float(food)  
 self.type = type  
 self.health = float(health)  
 self.time = int(time)  
 self.happiness = float(happiness)  
 self.shit = Bool(isShit)  
 self.shitE = []  
 self.timeToShit = float(timeToShit)  
 #  
  
 # ОТРИСОВКА ЖИВОТНОГО  
 x = 0  
 y = 0  
 for i in range(len(animals[self.type])):  
 sym = animals[self.type][i]  
 if sym == "1":  
 c.create\_rectangle(250 + x, 160 + y, 250 + x + 9, 160 + y + 9, fill='#fff', outline='#fff')  
 x += 10  
 if x == 160:  
 x = 0  
 y += 10  
 #  
  
 # ТЕКСТЫ  
 c.create\_text(590, 10, text=self.name, justify=CENTER, font='Impact', fill='#fff')  
 self.timeText = c.create\_text(450, 10, text="Your pet is living " + str(floor(self.time / 60)) + ":" + str(  
 self.time % 60), justify=CENTER, font='Verdana 10', fill='#fff')  
 #  
  
 # ЕДА  
 c.create\_rectangle(10, 10, 170, 40, outline='#fff')  
 try:  
 self.foodBar = c.create\_rectangle(10, 10, int(170 / (100 / self.food)), 40, fill='#fff')  
 except ZeroDivisionError:  
 self.foodBar = c.create\_rectangle(10, 10, 10, 40, fill='#fff')  
 self.foodText = c.create\_text(50, 50, text="FOOD: " + str(round(self.food, 3)) + "%", justify=CENTER,  
 font="Verdana 10", fill='#fff')  
 #  
  
 # ЗДОРОВЬЕ  
 c.create\_rectangle(200, 10, 360, 40, outline='#fff')  
 try:  
 self.healthBar = c.create\_rectangle(200, 10, int(360 / (100 / self.health)), 40, fill='#fff',  
 outline='#fff')  
 except ZeroDivisionError:  
 self.healthBar = c.create\_rectangle(200, 10, 200, 40, fill='#fff', outline='#fff')  
 self.healthText = c.create\_text(250, 50, text="HEALTH: " + str(round(self.health, 3)) + "%", justify=CENTER,  
 font='Verdana 10', fill='#fff')  
 #  
  
 # КНОПКИ  
 self.ButtonEating = Button(root, text='Feed', command=lambda: self.eating())  
 self.ButtonEating.place(x=10, y=80)  
 self.ButtonHeling = Button(root, text='Fix', command=lambda: self.healing())  
 self.ButtonHeling.place(x=10, y=115)  
 #  
  
 # СЧАСТЬЕ  
 c.create\_rectangle(10, 260, 170, 290, outline='#fff')  
 try:  
 self.happinessBar = c.create\_rectangle(10, 260, 170 / (100 / self.happiness), 290, fill='#fff')  
 except ZeroDivisionError:  
 self.happinessBar = c.create\_rectangle(10, 260, 10, 290, fill='#fff')  
 self.happinessText = c.create\_text(85, 300, text='HAPPINESS: ' + str(round(self.happiness, 3)) + "%",  
 justify=CENTER, font='Verdana 10', fill='#fff')  
 #  
  
 self.save() # Сохранение при создании  
  
 def eating(self): # ПОКОРМИТЬ  
 self.food += 20  
 if self.food > 100:  
 self.food = 100  
  
  
 def healing(self): # ПОЧИНИТЬ  
 if self.food >= 5:  
 self.food -= 5  
 self.health += 1  
 if self.health > 100:  
 self.health = 100  
  
 def save(self): # СОХРАНИТЬ  
 datas = [self.name, self.type, self.food, self.health, self.time, self.happiness, self.timeToShit,  
 self.shit] # Данные о животном  
 try:  
 save("./data.npy", datas) # Сохраняю  
 except PermissionError:  
 pass  
  
 def UPDATE(self): # ОБНОВИТЬ  
 # ПОТРЕБНОСТИ  
 if self.food >= 0:  
 self.food -= 0.001  
 if self.food < 0:  
 self.food = 0  
 self.health -= 0.01  
 if self.health <= 0:  
 self.death()  
  
 if self.shit:  
 self.happiness -= 0.005  
 else:  
 if self.happiness <= 100 - 0.0025:  
 self.happiness += 0.0025  
 if self.happiness < 0:  
 self.happiness = 0  
  
 self.timeToShit -= 1  
 if self.timeToShit <= 0:  
 self.take\_to\_shit()  
 self.timeToShit = round(random() \* 28800) # Обновляю таймер  
 #  
  
 # ВРЕМЯ  
 self.time += 1  
 c.delete(self.timeText)  
 sec = str(self.time % 60)  
 if floor(int(sec) / 10) % 10 == 0:  
 sec = "0" + sec  
 else:  
 sec = int(sec)  
 self.timeText = c.create\_text(450, 10, text="Your pet is living " + str(floor(self.time / 60)) + ":" + str(sec),  
 justify=CENTER, font='Verdana 10', fill='#fff')  
 #  
  
 # ОБНОВЛЕНИЕ ЕДЫ  
 try:  
 c.delete(self.foodBar)  
 self.foodBar = c.create\_rectangle(10, 10, int(170 / (100 / self.food)), 40, fill='#fff', outline='#fff')  
 except ZeroDivisionError:  
 c.delete(self.foodBar)  
 self.foodBar = c.create\_rectangle(10, 10, 10, 40, fill='#fff', outline='#fff')  
 c.delete(self.foodText)  
 self.foodText = c.create\_text(90, 50, text="FOOD: " + str(round(self.food, 3)) + "%", justify=CENTER,  
 font="Verdana 10", fill='#fff')  
 #  
  
 # ОБНОВЛЕНИЕ ЗДОРОВЬЯ  
 try:  
 c.delete(self.healthBar)  
 self.healthBar = c.create\_rectangle(200, 10, int(360 / (100 / self.health)), 40, fill='#fff',  
 outline='#fff')  
 except ZeroDivisionError:  
 c.delete(self.healthBar)  
 self.healthBar = c.create\_rectangle(200, 10, 200, 40, fill='#fff', outline='#fff')  
 c.delete(self.healthText)  
 self.healthText = c.create\_text(280, 50, text="HEALTH: " + str(round(self.health, 3)) + "%", justify=CENTER,  
 font='Verdana 10', fill='#fff')  
 #  
  
 # ОБНОВЛЕНИЕ СЧАСТЬЯ  
 try:  
 c.delete(self.happinessBar)  
 self.happinessBar = c.create\_rectangle(10, 260, 170 / (100 / self.happiness), 290, fill='#fff')  
 except ZeroDivisionError:  
 c.delete(self.happinessBar)  
 self.happinessBar = c.create\_rectangle(10, 260, 10, 290, fill='#fff')  
 c.delete(self.happinessText)  
 self.happinessText = c.create\_text(85, 300, text='HAPPINESS: ' + str(round(self.happiness, 3)) + "%",  
 justify=CENTER, font='Verdana 10', fill='#fff')  
 #  
  
 def death(self): # СМЕРТЬ  
 global EmailPermission, Email  
 if EmailPermission:  
 self.send(Email)  
 c.delete()  
 remove("./data.npy")  
  
 def kill(self): # УБИЙСТВО  
 global EmailPermission, Email  
 if EmailPermission:  
 self.send(Email)  
 remove("./data.npy")  
 exit(0)  
  
 def take\_to\_shit(self):  
 if self.shit:  
 return  
 global poop  
 self.shit = True  
 x = 0  
 y = 0  
 for i in range(64):  
 sym = poop[i]  
 if sym == "1":  
 self.shitE.append(  
 c.create\_rectangle(430 + x, 250 + y, 430 + x + 5, 250 + y + 5, fill='#fff', outline='#fff'))  
 x += 6  
 if x == 48:  
 x = 0  
 y += 6  
  
 def clear\_shit(self, e):  
 if e == -1:  
 for i in self.shitE:  
 c.delete(i)  
 self.shit = False  
 return  
 if e.x > 430 and e.x < 478 and e.y > 250 and e.y < 298:  
 for i in self.shitE:  
 c.delete(i)  
 self.shit = False  
  
 def hotkeys(self, e): # Обработка горячих клавиш  
 a = e.keycode  
 if a == 81: # Q or Й  
 exit(0)  
 elif a == 69: # E or У  
 self.eating()  
 elif a == 72: # H or Р  
 self.healing()  
 elif a == 75: # K or Л  
 self.kill()  
 elif a == 67: # C or C  
 self.clear\_shit(-1)  
 elif a == 83: # S or Ы  
 self.save()  
 elif a == 80:  
 self.take\_to\_shit()  
  
  
def FPS():  
 global animal  
 animal.UPDATE() # ОБНОВЛЕНИЕ  
 animal.save() # АВТО СОХРАНЕНИЕ  
 root.after(1000, FPS) # ВЫЗЫВАТЬ КАЖДУЮ СЕКУНДУ  
  
  
if exists("./data.npy") == False:  
  
 # ДОБАВЛЕНИЕ COMBOBOX ДЛЯ ВЫБОРА ЖИВОТНОГО  
 ChooseAnimal = Combobox(root)  
 ChooseAnimal["values"] = (  
 "cat", "horse", "pig", "dog", "hen", "turtle", "rabbit", "koala", "elephant", "beagle", "lion", "panda", "penguin",  
 "whale", "sheep")  
 ChooseAnimal.current(3)  
 ChooseAnimal.place(x=87, y=5)  
  
  
 #  
  
 # ФУНКЦИЯ ДОБАВЛЕНИЕ ЖИВОТНОГО  
 def \_main():  
 global animal, \_\_a\_obj  
 animal = Animal(Name.get(), ChooseAnimal.get(), 100, 100, 0, 100, round(random() \* 28800),  
 "False") # Создаём животное по умолчанию  
 c.bind("<1>", animal.clear\_shit)  
 root.bind("<KeyPress>", animal.hotkeys)  
  
 # УДАЛЕНИЕ ЭЛЕМЕНТОВ  
 ChooseAnimal.destroy()  
 Click.destroy()  
 Name.destroy()  
 c.delete(\_\_a\_obj)  
 #  
  
 FPS()  
  
  
 #  
  
 # ДОБАВЛЕНИЕ ТЕКСТОВОГО ПОЛЯ ДЛЯ ИМЯ ЖИВОТНОГО  
 Name = Entry(root, width=12)  
 Name.place(x=10, y=5)  
 Name.insert(0, "Name")  
 #  
  
 \_\_a\_obj = c.create\_text(10, 50, text='', justify=CENTER, font='Arial 10', fill='#fff')  
  
  
 #  
 def \_\_a(e):  
 global \_\_a\_obj  
 description = {  
 "cat": ["Хорошое животное, у него острые когти", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Большая", "Маленькие кучи, 3 раза в сутки"],  
 "horse": ["Огромное животное, не подходит для детей", "Очень большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Небольшая", "ОГРОМЕННЫЕ КУЧИ ДЕРМИЩА"],  
 "pig": ["Господи, какое описания, это же мини пиг", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Нет", "Средние кучи, 3 раза в сутки"],  
 "dog": ["Какой челевоек не хотел бы завести себе корги?)", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Большая", "Небольшие кучи, 3 раза в сутки"],  
 "hen": ["Курочка =)", "Не большая, она может не есть ~24 часа", "Если голодает, то проживёт ~2 часа",  
 "Большая", "Очень маленькие кучи, 3 раза в сутки"],  
 "turtle": ["Как у Макса Максимова", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Небольшая", "Очень маленькие кучи, 3 раза в сутки"],  
 "rabbit": ["ЭТО КРОЛИК КАРЛ! БЕРИ ЕГО!", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Большая", "Небольшие кучи, 3 раза в сутки"],  
 "koala": ["Милая коала", "Не большая, он может не есть ~24 часа", "Если голодает, то проживёт ~2 часа",  
 "Небольшая", "Небольшие кучи, 3 раза в сутки"],  
 "elephant": ["Да это же слон! P.S. Вы можете продать его бивни =)", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Нет", "МЕГА ОГРОМНЫЕ ГИГАНСКИЕ КУЧИ ДЕРМИЩА"],  
 "beagle": ["Офигеная собака, только вот игривая...", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Очень большая", "Жидкие кучи, 999 раза в сутки"],  
 "lion": ["Он громко рычит", "Не большая, он может не есть ~24 часа", "Если голодает, то проживёт ~2 часа",  
 "Небольшая", "Большие, 3 раза в сутки"],  
 "panda": ["У неё такие же как и у меня мешки под глазами", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Нет", "Средние, 3 раза в сутки"],  
 "penguin": ["TRIGGERED", "Не большая, он может не есть ~24 часа", "Если голодает, то проживёт ~2 часа",  
 "Небольшая", "Жидкие кучи, 3 раза в сутки"],  
 "whale": ["Самое большое животное", "Не большая, он может не есть ~24 часа",  
 "Если голодает, то проживёт ~2 часа", "Нет", "ОГРОМЕННЫЕ, 3 раза в сутки"],  
 "sheep": ["Это же бараш", "Не большая, он может не есть ~24 часа", "Если голодает, то проживёт ~2 часа",  
 "Нет", "Средние, 3 раза в сутки"]  
 }  
 c.delete(\_\_a\_obj)  
 \_\_a\_obj = c.create\_text(270, 100, text="""  
 Описание: {}  
 Потребность в еде: {}  
 Здоровье: {}  
 Потребность к играм: {}  
 """.format(description[ChooseAnimal.get()][0], description[ChooseAnimal.get()][1],  
 description[ChooseAnimal.get()][2], description[ChooseAnimal.get()][3],  
 description[ChooseAnimal.get()][4]), font='Arial 10 italic bold', fill='#fff')  
  
  
 \_\_a("")  
 ChooseAnimal.bind("<<ComboboxSelected>>", \_\_a)  
 #  
  
 # ДОБАВЛЕНИЕ КНОПКИ  
 Click = Button(root, text='Ask parents', command=lambda: \_main())  
 Click.place(x=230, y=2)  
 #  
else:  
 datas = load("./data.npy", allow\_pickle=True) # ЗАГРУЗКА ДАННЫХ  
 animal = Animal(datas[0], datas[1], datas[2], datas[3], datas[4], datas[5], datas[6],  
 datas[7]) # СОЗДАНИЕ ЖИВОТНОГО  
 c.bind("<1>", animal.clear\_shit) # Отслеживание ЛКМ  
 root.bind("<KeyPress>", animal.hotkeys)  
 FPS()  
  
root.mainloop()

**Реализация**

Рассмотрим код программы сервера. Сначала я импортировал разные библиотеки для работы с файлами и созданием своего настраиваемого окна. Создаю окно с помощью команды ***root.title("Тамогочи").*** Далее задаю размеры окна ***c = Canvas(root, width=700, height=320, bg="#000") # Фон  
c.grid() c.focus\_set().*** Далее задаю изображения животных. После всего этого создаю ***class Animal(object),*** в котором прописываю основные потребности моего питомца, его здоровье, счастье и кнопки для взаимодействия. Прописываю функции на разные потребности: eating (еда), healing (починка), save (сохранение), UPDATE (обновление данных), death (смерть), kill (убийство), take\_to\_shit, clear\_shit (туалет). Обобщая, методы взаимодействия. В функции ***FPS*** создаю основные тексты и техническую часть (сохранение, удаление предметов).

**Результат работы программы**





**Вывод**

Во время написания программы я ближе познакомился с библиотеками, связанными с созданием своего приложения. Узнал из них новые функции, методы, как можно создать свою игру.