from kivy.app import App #Делает приложение (запуск)  
from kivy.uix.label import Label #Вывод текста на экран  
from kivy.uix.button import Button #Кликабельные кнопочки  
from kivy.config import Config #Настройки  
from pygments.lexers import HtmlLexer #Настройки кнопок  
from kivy.uix.boxlayout import BoxLayout #Ориентация текста/кнопочек (горизонтально/вертикально)  
from kivy.uix.gridlayout import GridLayout #Ориентация текста/кнопочек (2х2/3х3)  
from kivy.uix.stacklayout import StackLayout #Ориентация текста/кнопочек (стек/как мешки в машинку)  
from kivy.uix.anchorlayout import AnchorLayout #Ориентация текста/кнопочек (более гибкое расположение на экране)  
from kivy.uix.floatlayout import FloatLayout #Ориентация текста/кнопочек (любое расположение на экране)  
from kivy.core.window import Window #Для масштабирования окна  
import time #Остановка выполнения программы (если понадобится пауза)  
from kivy.uix.screenmanager import ScreenManager, Screen #Для смены экрана  
from kivy.config import ConfigParser #Для настройки перехода и пр  
from kivy.uix.recycleview import RecycleView #Для прокрутки  
import random #Для рандомизации  
from kivy.metrics import dp  
import os  
import ast  
  
Window.size = (590, 680)  
Window.clearcolor = (.75, .60, .35, 1)  
Window.title = "Андердарк"  
  
XP = 0 # ХП персонажа (начальное рассчитывается по формуле: XP = 100 + n  
MAX\_XP = 0 # Максимальное XP персонажа  
Experience = 0 # Опыт  
Trophies = 0 # Трофеи  
LVL = 1 # Уровень (Увеличивается по условию: if(Experience >= 50 + 20\*LVL): LVL += 1  
  
Class = " " # Класс персонажа, определяющий дальнейшие начальные характеристики:  
n = 0 # Число, необходимое для высчитывания начального и максимального XP персонажа  
Skill = 0 # Ловкость (зависит от класса)  
Luck = 0 # Удача (зависит от класса)  
Strenght = 0 # Сила (зависит от класса)  
Endurance = 1 # Выносливость (зависит от класса)  
  
Difficulty = " " # Сложность игры, определяющая дальнейшие характеристики:  
k = 2 # Коэфициент удара моба по персонажу (зависит от сложности игры, изменяемой в настройках)  
  
XP\_normal\_monster = (random.randint(60, 90)) # XP обычного монстра  
XP\_average\_monster = (random.randint(70, 100)) # XP среднего монстра  
XP\_elite\_monster = (random.randint(100, 130)) # XP элитного монсра  
  
damage\_from\_normal\_monster = (random.randint(4, 6) \* k) // Endurance # Урон от обычного монстра  
damage\_from\_average\_monster = (random.randint(8, 10) \* k) // Endurance # Урон от среднего монстра  
damage\_from\_elite\_monster = (random.randint(14, 16) \* k) // Endurance # Урон от элитного монстра  
  
damage\_to\_normal\_monster = (random.randint(8, 9) \* Strenght) # Урон по обычному монстру  
damage\_to\_average\_monster = (random.randint(7, 8) \* Strenght) # Урон по среднему монстру  
damage\_to\_elite\_monster = (random.randint(6, 7) \* Strenght) # Урон по элитному монстру  
  
# Инвентарь персонажа (сюда будут добавлены все возможные предметы игры, начальное количество которых будет равно 0)  
  
Usual\_healing\_potion = 0 # Обычное зелье лечения  
Powerful\_healing\_potion = 0 # Сильное зелье лечения  
Epic\_healing\_potion = 0 # Эпическое зелье лечения  
  
# Сохранение  
  
try:  
 with open (u"Proverka.txt", 'r') as Save:  
 Save = (Save.read())  
except FileNotFoundError:  
 with open ("Proverka.txt", "w") as Save:  
 Save.write("Я есть!")  
 with open("XP.txt", "w") as Save:  
 Save.write(XP=0)  
 with open("Experience.txt", "w") as Save:  
 Save.write(Experience = 0)  
 with open("Trophies.txt", "w") as Save:  
 Save.write(Trophies = 0)  
 with open("LVL.txt", "w") as Save:  
 Save.write(LVL = 1)  
 with open("Class.txt", "w") as Save: # Класс персонажа и всё, что от него зависит  
 Save.write(Class = " ")  
 with open("n.txt", "w") as Save:  
 Save.write(n = 0)  
 with open("Skill.txt", "w") as Save:  
 Save.write(Skill = 0)  
 with open("Luck.txt", "w") as Save:  
 Save.write(Luck = 0)  
 with open("Strenght.txt", "w") as Save:  
 Save.write(Strenght = 0)  
 with open("Endurance.txt", "w") as Save:  
 Save.write(Endurance = 0)  
 with open("Difficulty.txt", "w") as Save: # Сложность игры и всё, что от неё зависит  
 Save.write(Difficulty = " ")  
 with open("k.txt", "w") as Save:  
 Save.write(k = 2)  
 with open("Usual\_healing\_potion.txt", "w") as Save: # Инвентарь и его содержимое  
 Save.write(Usual\_healing\_potion = 0)  
 with open("Powerful\_healing\_potion.txt", "w") as Save:  
 Save.write(Powerful\_healing\_potion = 0)  
 with open("Epic\_healing\_potion.txt", "w") as Save:  
 Save.write(Epic\_healing\_potion = 0)  
  
with open("XP.txt", "w") as Save:  
 Save.write(str(XP))  
with open("Experience.txt", "w") as Save:  
 Save.write(str(Experience))  
with open("Trophies.txt", "w") as Save:  
 Save.write(str(Trophies))  
with open("LVL.txt", "w") as Save:  
 Save.write(str(LVL))  
with open("Class.txt", "w") as Save: # Класс персонажа и всё, что от него зависит  
 Save.write(str(Class))  
with open("n.txt", "w") as Save:  
 Save.write(str(n))  
with open("Skill.txt", "w") as Save:  
 Save.write(str(Skill))  
with open("Luck.txt", "w") as Save:  
 Save.write(str(Luck))  
with open("Strenght.txt", "w") as Save:  
 Save.write(str(Strenght))  
with open("Endurance.txt", "w") as Save:  
 Save.write(str(Endurance))  
with open("Difficulty.txt", "w") as Save: # Сложность игры и всё, что от неё зависит  
 Save.write(str(Difficulty))  
with open("k.txt", "w") as Save:  
 Save.write(str(k))  
with open("Usual\_healing\_potion.txt", "w") as Save: # Инвентарь и его содержимое  
 Save.write(str(Usual\_healing\_potion))  
with open("Powerful\_healing\_potion.txt", "w") as Save:  
 Save.write(str(Powerful\_healing\_potion))  
with open("Epic\_healing\_potion.txt", "w") as Save:  
 Save.write(str(Epic\_healing\_potion))  
  
  
with open("XP.txt", "r") as Save:  
 XP = int(Save.read())  
with open("Experience.txt", "r") as Save:  
 Experience = int(Save.read())  
with open("Trophies.txt", "r") as Save:  
 Trophies = int(Save.read())  
with open("LVL.txt", "r") as Save:  
 LVL = int(Save.read())  
with open("Class.txt", "r") as Save: # Класс персонажа и всё, что от него зависит  
 Class = (Save.read())  
with open("n.txt", "r") as Save:  
 n = int(Save.read())  
with open("Skill.txt", "r") as Save:  
 Skill = int(Save.read())  
with open("Luck.txt", "r") as Save:  
 Luck = int(Save.read())  
with open("Strenght.txt", "r") as Save:  
 Strenght = int(Save.read())  
with open("Endurance.txt", "r") as Save:  
 Endurance = int(Save.read())  
with open("Difficulty.txt", "r") as Save: # Сложность игры и всё, что от неё зависит  
 Difficulty = (Save.read())  
with open("k.txt", "r") as Save:  
 k = int(Save.read())  
with open("Usual\_healing\_potion.txt", "r") as Save: # Инвентарь и его содержимое  
 Usual\_healing\_potion = int(Save.read())  
with open("Powerful\_healing\_potion.txt", "r") as Save:  
 Powerful\_healing\_potion = int(Save.read())  
with open("Epic\_healing\_potion.txt", "r") as Save:  
 Epic\_healing\_potion = int(Save.read())  
  
  
class Saving:  
 with open("XP.txt", "w") as Save:  
 Save.write(str(XP))  
 with open("Experience.txt", "w") as Save:  
 Save.write(str(Experience))  
 with open("Trophies.txt", "w") as Save:  
 Save.write(str(Trophies))  
 with open("LVL.txt", "w") as Save:  
 Save.write(str(LVL))  
 with open("Class.txt", "w") as Save: # Класс персонажа и всё, что от него зависит  
 Save.write(str(Class))  
 with open("n.txt", "w") as Save:  
 Save.write(str(n))  
 with open("Skill.txt", "w") as Save:  
 Save.write(str(Skill))  
 with open("Luck.txt", "w") as Save:  
 Save.write(str(Luck))  
 with open("Strenght.txt", "w") as Save:  
 Save.write(str(Strenght))  
 with open("Endurance.txt", "w") as Save:  
 Save.write(str(Endurance))  
 with open("Difficulty.txt", "w") as Save: # Сложность игры и всё, что от неё зависит  
 Save.write(str(Difficulty))  
 with open("k.txt", "w") as Save:  
 Save.write(str(k))  
 with open("Usual\_healing\_potion.txt", "w") as Save: # Инвентарь и его содержимое  
 Save.write(str(Usual\_healing\_potion))  
 with open("Powerful\_healing\_potion.txt", "w") as Save:  
 Save.write(str(Powerful\_healing\_potion))  
 with open("Epic\_healing\_potion.txt", "w") as Save:  
 Save.write(str(Epic\_healing\_potion))  
  
class Return:  
 with open("XP.txt", "r") as Save:  
 XP = int(Save.read())  
 with open("Experience.txt", "r") as Save:  
 Experience = int(Save.read())  
 with open("Trophies.txt", "r") as Save:  
 Trophies = int(Save.read())  
 with open("LVL.txt", "r") as Save:  
 LVL = int(Save.read())  
 with open("Class.txt", "r") as Save: # Класс персонажа и всё, что от него зависит  
 Class = (Save.read())  
 with open("n.txt", "r") as Save:  
 n = int(Save.read())  
 with open("Skill.txt", "r") as Save:  
 Skill = int(Save.read())  
 with open("Luck.txt", "r") as Save:  
 Luck = int(Save.read())  
 with open("Strenght.txt", "r") as Save:  
 Strenght = int(Save.read())  
 with open("Endurance.txt", "r") as Save:  
 Endurance = int(Save.read())  
 with open("Difficulty.txt", "r") as Save: # Сложность игры и всё, что от неё зависит  
 Difficulty = (Save.read())  
 with open("k.txt", "r") as Save:  
 k = int(Save.read())  
 with open("Usual\_healing\_potion.txt", "r") as Save: # Инвентарь и его содержимое  
 Usual\_healing\_potion = int(Save.read())  
 with open("Powerful\_healing\_potion.txt", "r") as Save:  
 Powerful\_healing\_potion = int(Save.read())  
 with open("Epic\_healing\_potion.txt", "r") as Save:  
 Epic\_healing\_potion = int(Save.read())  
  
class Main\_menu(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Main\_menu, self).\_\_init\_\_(\*\*kw)  
 menu = BoxLayout(orientation="vertical")  
 menu.add\_widget(Button(text="Новая игра",  
 font\_size=50,  
 background\_color=[.75, .60, .35, 1],  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_normal="", on\_press=self.Новая))  
 menu.add\_widget(Button(text="Продолжить",  
 font\_size=50,  
 background\_color=[.75, .60, .30, 1],  
 color=[62/255, 28/255, 0/255, 1],  
 background\_normal="", on\_press=self.Прода))  
 menu.add\_widget(Button(text="Мой персонаж",  
 font\_size=50,  
 background\_color=[.75, .60, .35, 1],  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_normal="", on\_press=self.MyPers))  
 menu.add\_widget(Button(text="Магазин",  
 font\_size=50,  
 background\_color=[.75, .60, .30, 1],  
 color=[62/255, 28/255, 0/255, 1],  
 background\_normal="", on\_press=lambda x: set\_screen('Магазин')))  
 menu.add\_widget(Button(text="Правила",  
 font\_size=50,  
 background\_color=[.75, .60, .35, 1],  
 color=[62/255, 28/255, 0/255, 1],  
 background\_normal="", on\_press=lambda x: set\_screen('Правила')))  
 menu.add\_widget(Button(text="История",  
 font\_size=50,  
 background\_color=[.75, .60, .30, 1],  
 color=[62/255, 28/255, 0/255, 1],  
 background\_normal="", on\_press=lambda x: set\_screen('История')))  
 menu.add\_widget(Button(text="Настройки",  
 font\_size=50,  
 background\_color=[.75, .60, .35, 1],  
 color=[62/255, 28/255, 0/255, 1],  
 background\_normal="", on\_press=lambda x: set\_screen('Настройки')))  
 menu.add\_widget(Button(text="Выход",  
 font\_size=50,  
 background\_color=[.75, .60, .30, 1],  
 color=[62/255, 28/255, 0/255, 1],  
 background\_normal="", on\_press=exit))  
 self.add\_widget(menu)  
 def Новая(self, event):  
 set\_screen("Выбор\_класса")  
 return Saving  
 def Прода(self, event):  
 if Class != " ":  
 i = random.randint(1, 7)  
 set\_screen(str(i))  
 else:  
 set\_screen("Нет\_класса")  
 return Return  
 def MyPers(self, event):  
 if Class != " ":  
 set\_screen("Мой\_персонаж")  
 else:  
 set\_screen("Нет\_класса")  
  
class My\_character(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(My\_character, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Сharacter = FloatLayout()  
 Сharacter.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.9, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Сharacter.add\_widget(Button(text=" \*тут будет\nэмблема класса\*",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=15,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.3, .2), pos=(80, 400), background\_normal=""))  
 Сharacter.add\_widget(Label(text="Мой персонаж\n\n\n\n\n\n\n\n\n\n",  
 font\_size=40,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Сharacter.add\_widget(Label(text="\n\n\n\n\n"  
 " Класс : "+str(Class)+"\n"  
 " Уровень: "+str(LVL)+"\n"  
 " Опыт: "+str(Experience)+"\n"  
 " Трофеи: "+str(Trophies)+"\n\n"  
 "Характеристики:\n\n"  
 "Ловкость: "+str(Skill)+"\n"  
 "Сила: "+str(Strenght)+"\n"  
 "Выносливость: "+str(Endurance)+"\n"  
 "Удача: "+str(Luck)+"\n\n"  
 "Инвентарь:\n\n"  
 "Простое зелье лечения: "+str(Usual\_healing\_potion)+"\n"  
 "Сильное зелье лечения: "+str(Powerful\_healing\_potion)+"\n"  
 "Эпическое зелье лечения: "+str(Epic\_healing\_potion)+"\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 self.add\_widget(Сharacter)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class N0\_class(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(N0\_class, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 No = FloatLayout()  
 No.add\_widget(Button(text="У Вас нет созданного персонажа!\n\n (нажмите для продолжения)",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1], size\_hint=(.7, .3), pos=(90, 275),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 self.add\_widget(No)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Privet(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Privet, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 priv = BoxLayout(orientation="vertical")  
 priv.add\_widget(Label(text="Пока что это всё \ ( ^ - ^ ) / ", font\_size=40,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 priv.add\_widget(Button(text="Назад",  
 background\_color=[.75, .60, .30, 1],  
 font\_size=40,  
 color=[62/255, 28/255, 0/255, 1],  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 self.add\_widget(priv)  
 print("Класс персонажа: ", Class, "\nБонус к ХП: ", n, "\nКоличество ХП: ", XP, "\nЛовкость: ", Skill, "\nУдача: ", Luck, "\nСила: ", Strenght, "\nВыносливость: ", Endurance)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class History(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(History, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 self.layout = GridLayout(cols=1, spacing=10, size\_hint\_y=2.3)  
 self.layout.bind(minimum\_height=self.layout.setter('height'))  
 Istoria = Button(text='<-- В главное меню',  
 background\_color=[176 / 255, 136 / 255, 68 / 255, 1],  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 font\_size=20,  
 background\_normal="",  
 on\_press=lambda x: set\_screen('Главное\_меню'),  
 size\_hint\_y=None, height=dp(40))  
 self.layout.add\_widget(Istoria)  
 root = RecycleView(size\_hint=(1, None), size=(Window.width, Window.height))  
 root.add\_widget(self.layout)  
 self.add\_widget(root)  
 btn = BoxLayout(orientation="vertical")  
 btn.add\_widget(Button(text="В Андердарке, расположенном под гордом\n"  
 "Фаэрун, всегда было неспокойно. Там\n"  
 "обитало множество рас, считающихся\n"  
 "«тёмными» Жили там тёмные эльфы-дроу,\n"  
 "иллитиды, наблюдатели,\nтемные дварфы - дуергары.\n\n"  
 "Андердарк – это огромный лабиринт\n"  
 "пещер и тоннелей. Здесь нет\n"  
 "практически никаких источников света,\n"  
 "за исключением светящихся камней или\n"  
 "грибов. Сам воздух может содержать\n"  
 "яды или взрывоопасные газы. Но\n"  
 "всё это не помешало разумным\n"  
 "существам жить в Андердарке.\n\n"  
 "Пусть это и подземный мир, однако,\n"  
 "растительный и животный мир отличается\n"  
 "удивительным, для своего климата,\n"  
 "разнообразием. Растительная жизнь не"  
 "\nмогла использовать солнечный свет в "  
 "\nкачестве источника энергии и, таким"  
 "\nобразом, приняла странные формы,"  
 "\nприспособленные к жизни под землёй."  
 "\nБудучи неспособными переваривать"  
 "\nнормальные питательные вещества,"  
 "\nбольшую часть времени они полагались"  
 "\nна поглощение магической энергии, или,"  
 "\nкак её тут называют, фаэрзресс. Животный\n"  
 "же мир не менее разнообразен. Здесь есть"  
 "\nпочти всё: от маленьких насекомых до"  
 "\nтеневых драконов, создавших свои"  
 "\nцарства глубоко внутри.\n\n"  
 "Вам с детства знакомы эти холодные\n"  
 "туннели, темнота и чувство опасности.\n"  
 "И пусть со временем у вас развилось\n"  
 "тёмное зрение, менее опасно здесь не\n"  
 "стало. Вы привыкли жить среди тьмы,\n"  
 "с редким вкраплением фосфоресцирующих\n"  
 "грибов, среди хищников и редко\n"  
 "встречающихся обитателей Андердарка.\n"  
 "\nНо ведь всегда было интересно, что там,\n"  
 "наверху, над многочисленными проходами\n"  
 "и тоннелями пещер, над подземными\n"  
 "государствами, над всем этим \n"  
 "подземным городом.\n\n",  
 font\_size = 25, color = [62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[.75, .60, .35, 1],  
 background\_normal=""))  
 self.layout.add\_widget(btn)  
 def on\_leave(self):  
 self.layout.clear\_widgets()  
  
class Rules(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Rules, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 self.layout = GridLayout(cols=1, spacing=10, size\_hint\_y=2.9)  
 self.layout.bind(minimum\_height=self.layout.setter('height'))  
 Pravila = Button(text='<-- В главное меню',  
 background\_color=[176 / 255, 136 / 255, 68 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_normal="",  
 on\_press=lambda x: set\_screen('Главное\_меню'),  
 size\_hint\_y=None, height=dp(40))  
 self.layout.add\_widget(Pravila)  
 root = RecycleView(size\_hint=(1, None), size=(Window.width, Window.height))  
 root.add\_widget(self.layout)  
 self.add\_widget(root)  
 Pravila2 = BoxLayout(orientation="vertical")  
 Pravila2.add\_widget(Button(text="На начальном этапе вам предстоит выбрать\n"  
 "класс персонажа. От этого будут зависеть\n"  
 "Ваши начальные характеристики, количество\n"  
 "хп, а также урон, получаемый от\n"  
 "ловушек.\n\n"  
 "Ваши характеристики:\n"  
 "◊ Сила – отвечает за силу атаки\n"  
 "◊ Ловкость (ловк) – отвечает за уклонение\n"  
 "◊ Выносливость (вын) – отвечает за урон\n"  
 "получаемый от монстров\n"  
 "◊ Удача – отвечает за ценность и\n"  
 "количество найденных трофеев, а также за\n"  
 "частоту встречи ловушек\n\n"  
 "Классов всего 4:\n\n"  
 "1. Маг 2. Убийца\nКоличество хп – 100 Количество хп – 130\n"  
 "Ловкость – 2 Ловкость – 5\nУдача – 3 Удача – 2\n"  
 "Сила – 6 Сила – 5\nВыносливость – 2 Выносливость – 3\n"  
 "\n3. Воин 4. Страж\n"  
 "Количество хп – 170 Количество хп – 210\nЛовкость – 3 Ловкость – 2\n"  
 "Удача – 2 Удача – 1\nСила – 4 Сила – 3\n"  
 "Выносливость – 5 Выносливость – 6\n\n"  
 "После выбора класса начинается\nпрохождение подземелья\n\n Опыт и трофеи\n\n"  
 "Опыт персонажа накапливается при\nпрохождении каждой локации\n(кроме локаций магазина и отдыха)\n\n"  
 "Опыт необходим для перехода в более\nсложные участки подземелья, этажи\n(всего их 10)\n\n"  
 "Трофеи персонажа служат валютой, за\nкоторую он может приобретать вещи в\nмагазине.\n\n"  
 "Трофеи можно получить на локациях\n(«викторина», монстр, сундук)\n\n "  
 " Бой и механика\n\n"  
 "Урон, уклонение, количество и качество\nтрофеев не фиксированы, они зависят от\n"  
 "рандома (с учётом характеристик персонажа,\nкак начальных, так и приобретённых за счёт\n"  
 "предметов)\nУрон и уклонение (в битве с монстром)\nтакже зависят от рандома, но с учётом не\n"  
 "только характеристик персонажа, но и\nхарактеристик самого монстра.\n",  
 font\_size = 25, color = [62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[.75, .60, .35, 1],  
 background\_normal=""))  
 self.layout.add\_widget(Pravila2)  
 def on\_leave(self):  
 self.layout.clear\_widgets()  
  
class Election\_class(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Election\_class, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 ViborClass = FloatLayout()  
 ViborClass.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62/255, 28/255, 0/255, 1],  
 size\_hint=(.9, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 ViborClass.add\_widget(Label(text="\n\n\n\n Выбор класса\n\n"  
 "Вы решили попробовать выбраться из\nПодземного Города, но для этого Вам\n"  
 "предстоит выбрать свой дальнейший путь…\n"  
 "Быть может, Вы мечтали стать магом? или\n"  
 "воином? В любом случае, выбирайте с умом,\n"  
 "ведь у каждого класса есть как плюсы,\nтак и минусы.\n\n\n\n\n\n\n\n\n\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 ViborClass.add\_widget(Button(text=" Маг\n\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n XP - 100\n Ловк - 2\n Удача - 2\n Сила - 6\n"  
 " Выносл - 2\n",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.2, .4), pos=(25, 30),  
 on\_press=self.Маг))  
 ViborClass.add\_widget(Button(text=" Убийца\n\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n XP - 130\n Ловк - 5\n Удача - 2\n Сила - 5\n"  
 " Выносл - 3\n",  
 background\_color=[176 / 255, 136 / 255, 68 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 on\_press=self.Убийца,  
 size\_hint=(.2, .4), pos=(164, 30)))  
 ViborClass.add\_widget(Button(text=" Воин\n\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n XP - 170\n Ловк - 3\n Удача - 2\n Сила - 4\n"  
 " Выносл - 5\n",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 on\_press=self.Воин,  
 size\_hint=(.2, .4), pos=(301, 30)))  
 ViborClass.add\_widget(Button(text=" Страж\n\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n XP - 210\n Ловк - 2\n Удача - 1\n Сила - 3\n"  
 " Выносл - 6\n",  
 background\_color=[176 / 255, 136 / 255, 68 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 on\_press=self.Страж,  
 size\_hint=(.2, .4), pos=(440, 30)))  
 self.add\_widget(ViborClass)  
 def Маг(self, event):  
 global Class  
 global n  
 global Skill  
 global Luck  
 global Strenght  
 global Endurance  
 global XP  
 global MAX\_XP  
 Class = "Маг"  
 n = 0  
 XP = 100 + n  
 MAX\_XP = XP  
 Skill = 2  
 Luck = 3  
 Strenght = 6  
 Endurance = 2  
 set\_screen('Начало\_пути\_маг')  
 print("Класс персонажа: ", Class, "\nБонус к ХП: ", n, "\nКоличество ХП: ", XP, "\nЛовкость: ", Skill, "\nУдача: ", Luck, "\nСила: ", Strenght, "\nВыносливость: ", Endurance)  
 def Убийца(self, event):  
 global Class  
 global n  
 global Skill  
 global Luck  
 global Strenght  
 global Endurance  
 global XP  
 global MAX\_XP  
 Class = "Убийца"  
 n = 30  
 XP = 100 + n  
 MAX\_XP = XP  
 Skill = 5  
 Luck = 2  
 Strenght = 5  
 Endurance = 3  
 set\_screen('Начало\_пути\_убийца')  
 print("Класс персонажа: ", Class, "\nБонус к ХП: ", n, "\nКоличество ХП: ", XP, "\nЛовкость: ", Skill, "\nУдача: ", Luck, "\nСила: ", Strenght, "\nВыносливость: ", Endurance)  
 def Воин(self, event):  
 global Class  
 global n  
 global Skill  
 global Luck  
 global Strenght  
 global Endurance  
 global XP  
 global MAX\_XP  
 Class = "Воин"  
 n = 70  
 XP = 100 + n  
 MAX\_XP = XP  
 Skill = 3  
 Luck = 2  
 Strenght = 4  
 Endurance = 5  
 set\_screen('Начало\_пути\_воин')  
 print("Класс персонажа: ", Class, "\nБонус к ХП: ", n, "\nКоличество ХП: ", XP, "\nЛовкость: ", Skill, "\nУдача: ", Luck, "\nСила: ", Strenght, "\nВыносливость: ", Endurance)  
 def Страж(self, event):  
 global Class  
 global n  
 global Skill  
 global Luck  
 global Strenght  
 global Endurance  
 global XP  
 global MAX\_XP  
 Class = "Страж"  
 n = 110  
 XP = 100 + n  
 MAX\_XP = XP  
 Skill = 2  
 Luck = 1  
 Strenght = 3  
 Endurance = 6  
 set\_screen('Начало\_пути\_страж')  
 print("Класс персонажа: ", Class, "\nБонус к ХП: ", n, "\nКоличество ХП: ", XP, "\nЛовкость: ", Skill, "\nУдача: ", Luck, "\nСила: ", Strenght, "\nВыносливость: ", Endurance)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Beginning\_Wizard(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Beginning\_Wizard, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Class = FloatLayout()  
 Class.add\_widget(Button(text="Выбрать другой класс",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62/255, 28/255, 0/255, 1],  
 size\_hint=(.9, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Выбор\_класса')))  
 Class.add\_widget(Label(text="\n\nПоздравляем!\nВы выбрали Класс Мага и Ваше путешествие\n"  
 "по Андердарку начинается с этого момента.\n\n"  
 "Ваши характеристики:\nКоличество xp - 100\nловкость – 2\nудача – 3\n"  
 "сила – 6\nвыносливость – 2\n\n"  
 "На Вашем пути будет множество трудностей\n"  
 "и испытаний, но награда того стоит.\n\n"  
 "Пусть удача сопутствует Вам!\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Class.add\_widget(Button(text="Продолжить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62/255, 28/255, 0/255, 1],  
 size\_hint=(.9, .1), pos=(27,1),  
 background\_normal="", on\_press=lambda x: set\_screen('1')))  
 self.add\_widget(Class)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Beginning\_Assassin(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Beginning\_Assassin, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Class = FloatLayout()  
 Class.add\_widget(Button(text="Выбрать другой класс",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.9, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Выбор\_класса')))  
 Class.add\_widget(Label(text="\n\nПоздравляем!\nВы выбрали Класс Убийцы и Ваше путешествие\n"  
 "по Андердарку начинается с этого момента.\n\n"  
 "Ваши характеристики:\nКоличество xp - 130\nловкость – 5\nудача – 2\n"  
 "сила – 5\nвыносливость – 3\n\n"  
 "На Вашем пути будет множество трудностей\n"  
 "и испытаний, но награда того стоит.\n\n"  
 "Пусть удача сопутствует Вам!\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Class.add\_widget(Button(text="Продолжить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.9, .1), pos=(27, 1),  
 background\_normal="", on\_press=lambda x: set\_screen('1')))  
 self.add\_widget(Class)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Beginning\_Warrior(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Beginning\_Warrior, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Class = FloatLayout()  
 Class.add\_widget(Button(text="Выбрать другой класс",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.9, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Выбор\_класса')))  
 Class.add\_widget(Label(text="\n\nПоздравляем!\nВы выбрали Класс Воина и Ваше путешествие\n"  
 "по Андердарку начинается с этого момента.\n\n"  
 "Ваши характеристики:\nКоличество xp - 170\nловкость – 3\nудача – 2\n"  
 "сила – 4\nвыносливость – 5\n\n"  
 "На Вашем пути будет множество трудностей\n"  
 "и испытаний, но награда того стоит.\n\n"  
 "Пусть удача сопутствует Вам!\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Class.add\_widget(Button(text="Продолжить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.9, .1), pos=(27, 1),  
 background\_normal="", on\_press=lambda x: set\_screen('1')))  
 self.add\_widget(Class)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Beginning\_Guard(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Beginning\_Guard, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Class = FloatLayout()  
 Class.add\_widget(Button(text="Выбрать другой класс",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.9, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Выбор\_класса')))  
 Class.add\_widget(Label(text="\n\nПоздравляем!\nВы выбрали Класс Стража и Ваше путешествие\n"  
 "по Андердарку начинается с этого момента.\n\n"  
 "Ваши характеристики:\nКоличество xp - 210\nловкость – 2\nудача – 1\n"  
 "сила – 3\nвыносливость – 6\n\n"  
 "На Вашем пути будет множество трудностей\n"  
 "и испытаний, но награда того стоит.\n\n"  
 "Пусть удача сопутствует Вам!\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Class.add\_widget(Button(text="Продолжить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.9, .1), pos=(27, 1),  
 background\_normal="", on\_press=lambda x: set\_screen('1')))  
 self.add\_widget(Class)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Settings(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Settings, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Class = FloatLayout()  
 Class.add\_widget(Button(text="\n\n\n\n\nСложность игры",  
 font\_size=40,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[.75, .60, .35, 1],  
 background\_normal=""))  
 Class.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.9, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Class.add\_widget(Button(text="Лёгкая",  
 background\_color=[176 / 255, 136 / 255, 68 / 255, 1],  
 font\_size=30,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.3, .2), pos=(27, 40),  
 background\_normal="", on\_press=self.easy))  
 Class.add\_widget(Button(text="Средняя",  
 background\_color=[176 / 255, 136 / 255, 68 / 255, 1],  
 font\_size=30,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.3, .2), pos=(205, 40),  
 background\_normal="", on\_press=self.middle))  
 Class.add\_widget(Button(text="Сложная",  
 background\_color=[176 / 255, 136 / 255, 68 / 255, 1],  
 font\_size=30,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.3, .2), pos=(383, 40),  
 background\_normal="", on\_press=self.hard))  
 self.add\_widget(Class)  
 def easy(self, event):  
 global k  
 global Difficulty  
 k = 1  
 Difficulty = "Лёгкий уровень сложности: "  
 print(Difficulty, " ", k)  
 set\_screen('Сложность\_установлена')  
 def middle(self, event):  
 global k  
 global Difficulty  
 k = 2  
 Difficulty = "Средний уровень сложности: "  
 print(Difficulty, " ", k)  
 set\_screen('Сложность\_установлена')  
 def hard(self, event):  
 global k  
 global Difficulty  
 k = 3  
 Difficulty = "Максимальный уровень сложности: "  
 print(Difficulty, " ", k)  
 set\_screen('Сложность\_установлена')  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Shop(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Shop, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 self.layout = GridLayout(cols=1, spacing=10, size\_hint\_y=1)  
 self.layout.bind(minimum\_height=self.layout.setter('height'))  
 Shop=(Button(text='<-- В главное меню',  
 background\_color=[176 / 255, 136 / 255, 68 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_normal="",  
 on\_press=lambda x: set\_screen('Главное\_меню'),  
 size\_hint\_y=None, height=dp(40)))  
 self.layout.add\_widget(Shop)  
 root = RecycleView(size\_hint=(1, None), size=(Window.width, Window.height))  
 root.add\_widget(self.layout)  
 self.add\_widget(root)  
 Shop2 = FloatLayout()  
 Shop2.add\_widget(Label(text=" Предметы Купить \n\n"  
 "Обычное зелье лечения\n\nСильное зелье лечения\n\nЭпическое зелье лечения\n\n\n\n\n\n\n\n\n\n\n\n",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Shop2.add\_widget(Button(text="5 трофеев",  
 font\_size = 25, color = [62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1], size\_hint=(.3, .1), pos=(400, 515),  
 background\_normal="", on\_press=self.обычное\_зелье))  
 Shop2.add\_widget(Button(text="10 трофеев",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1], size\_hint=(.3, .1), pos=(400, 455),  
 background\_normal="", on\_press=self.сильное\_зелье))  
 Shop2.add\_widget(Button(text="15 трофеев",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1], size\_hint=(.3, .1), pos=(400, 395),  
 background\_normal="", on\_press=self.эпическое\_зелье))  
 self.add\_widget(Shop2)  
 def обычное\_зелье(self, event):  
 global Usual\_healing\_potion  
 global Trophies  
 if (Trophies - 5) > 0:  
 Usual\_healing\_potion += 1  
 Trophies -= 5  
 else:  
 set\_screen('Нет\_средств')  
 def сильное\_зелье(self, event):  
 global Powerful\_healing\_potion  
 global Trophies  
 if (Trophies - 10) > 0:  
 Powerful\_healing\_potion += 1  
 Trophies -= 10  
 else:  
 set\_screen('Нет\_средств')  
 def эпическое\_зелье(self, event):  
 global Epic\_healing\_potion  
 global Trophies  
 if (Trophies - 15) > 0:  
 Epic\_healing\_potion += 1  
 Trophies -= 15  
 else:  
 set\_screen('Нет\_средств')  
 def on\_leave(self):  
 self.layout.clear\_widgets()  
  
class N0\_money(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(N0\_money, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 No = FloatLayout()  
 No.add\_widget(Button(text="У Вас недостаточно трофеев!\n\n(нажмите для продолжения)",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1], size\_hint=(.7, .3), pos=(90, 275),  
 background\_normal="", on\_press=lambda x: set\_screen('Магазин')))  
 self.add\_widget(No)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Difficulty\_is\_set(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Difficulty\_is\_set, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 No = FloatLayout()  
 No.add\_widget(Button(text="Сложность игры установлена!\n\n(нажмите для продолжения)",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1], size\_hint=(.7, .3), pos=(90, 275),  
 background\_normal="", on\_press=lambda x: set\_screen('Настройки')))  
 self.add\_widget(No)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class LovyshkaOFF(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(LovyshkaOFF, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Lovyshka = FloatLayout()  
 Lovyshka.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Lovyshka.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Lovyshka.add\_widget(Label(text="Проходя по очередному коридору,\nвы в последний момент смогли\n"  
 "уклониться от пролетающей над\nвашей головой стрелы.\n\n"  
 "Сегодня вам повезло, но в\nследующий раз стоит внимательнее \n"  
 "смотреть под ноги!\n\nОпыт + 10\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Lovyshka.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=self.proda))  
 Lovyshka.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Lovyshka)  
 def piy(self, event):  
 return Saving  
 def proda(self, event):  
 global Experience  
 Experience += 10  
 set\_screen('2')  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Viktorina1(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Viktorina1, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Viktorina = FloatLayout()  
 Viktorina.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Viktorina.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Viktorina.add\_widget(Label(text="Вы сосредоточенно передвигаетесь по\nкоридорам, перешагивая опасные места,\n"  
 "но тут до вашего слуха доносится\nчьё-то мяуканье.\n"  
 "Добравшись до источника звука вы\nобнаруживаете маленького сфинкса,\n"  
 "сидящего на большом камне.\n"  
 "'Кто из 4-х героев ялвяется\nсамым сильным из всех?'\n"  
 "- спрашивает у вас сфинкс...\n\n\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Viktorina.add\_widget(Button(text="Маг",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .1), pos=(40, 165),  
 on\_press=self.ПравильныйВыбор))  
 Viktorina.add\_widget(Button(text="Убийца",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .1), pos=(310, 55),  
 on\_press=self.НеправильныйВыбор))  
 Viktorina.add\_widget(Button(text="Воин",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .1), pos=(40, 55),  
 on\_press=self.НеправильныйВыбор))  
 Viktorina.add\_widget(Button(text="Страж",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .1), pos=(310, 165),  
 on\_press=self.НеправильныйВыбор))  
 Viktorina.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Viktorina)  
 def piy(self, event):  
 return Saving  
 def ПравильныйВыбор(self, event):  
 global Experience  
 global Trophies  
 Experience += 10  
 Trophies += 4  
 set\_screen('2.1')  
 def НеправильныйВыбор(self, event):  
 global XP  
 global Experience  
 XP -= 10  
 Experience += 5  
 set\_screen('2.2')  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Viktorina1T(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Viktorina1T, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Viktorina = FloatLayout()  
 Viktorina.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Viktorina.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Viktorina.add\_widget(Label(text="Вы правильно ответили на вопрос!\n\n"  
 "Свинкс благосклонно кивнул вас\nи быстро скрылся за ближайшим проходом.\n"  
 "На его месте остался неприметный\nдеревянный сундучок!\n\n"  
 "Вы подходите и обнаруживаете там:\n\n"  
 " Трофеи - 4 штуки\n\n"  
 "Опыт + 10\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Viktorina.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=lambda x: set\_screen('3')))  
 Viktorina.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Viktorina)  
 def piy(self, event):  
 return Saving  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Viktorina1F(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Viktorina1F, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Viktorina = FloatLayout()  
 Viktorina.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Viktorina.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Viktorina.add\_widget(Label(text="Кажется, что-то пошло не так...\n"  
 "Сфинкс зашипил и два огромных\nпрыжка приблизился к вам!\n\n"  
 "Быстро полоснув острыми когтями\nпо лицу, он скрылся в проходе\n"  
 "Вам стоит быть аккуратнее, прежде\nчем отвечать на подобные вопросы.\n"  
 "\nXP - 10\nОпыт + 5\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Viktorina.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=lambda x: set\_screen('3')))  
 Viktorina.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Viktorina)  
 def piy(self, event):  
 return Saving  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Syndyk1(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Syndyk1, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Syndyk = FloatLayout()  
 Syndyk.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Syndyk.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Syndyk.add\_widget(Label(text="Переходя из одного коридора в другой, \nвы замечаете засыпанную пылью и\n"  
 "камнями нишу в полу. \n\n"  
 "Подумав, что там может быть что-то\nинтересное, вы принимаетесь\n"  
 "раскапывать яму.\n\n\n\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Syndyk.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=self.proda))  
 Syndyk.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Syndyk)  
 def piy(self, event):  
 return Saving  
 def proda(self, event):  
 global Trophies  
 global Usual\_healing\_potion  
 global Experience  
 Trophies += 6  
 Usual\_healing\_potion += 1  
 Experience += 10  
 set\_screen("3.2")  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Syndyk2(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Syndyk2, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Syndyk = FloatLayout()  
 Syndyk.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Syndyk.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Syndyk.add\_widget(Label(text="Заметив железный блеск отделки\nсундука, вы начали копать активнее\n"  
 "и уже через минуту вы вытаскиваете \nна поверхность дубовый сундук!\n\n"  
 "Открыв его, вы обнаруживаете:\n\n"  
 " Трофеи - 6 штук\n Простое зелье лечения - 1 штука\n\n"  
 "Опыт + 10\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Syndyk.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=lambda x: set\_screen('4')))  
 Syndyk.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Syndyk)  
 def piy(self, event):  
 return Saving  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class LovyshkaON(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(LovyshkaON, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Lovyshka = FloatLayout()  
 Lovyshka.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Lovyshka.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Lovyshka.add\_widget(Label(text="Блуждая по коридорам тёмной пещеры, \nвы не заметили, как наступили на\n"  
 "исписанный рунами булыжник,\nактивировав ловушку!\n\n"  
 "Шипы, резко выдвинувшиеся из стены,\nсильно задели вас!\n\n"  
 "Опыт + 5\nXP - 20\n\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Lovyshka.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=self.proda))  
 Lovyshka.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Lovyshka)  
 def piy(self, event):  
 return Saving  
 def proda(self, event):  
 global Experience  
 global XP  
 Experience += 5  
 XP -= 20  
 set\_screen('5')  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Magazin1(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Magazin1, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Magazin = FloatLayout()  
 Magazin.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Magazin.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Magazin.add\_widget(Label(text="Выйдя на освещённую площадку пещеры,\nвы замечаете старика в неприметных\n"  
 "одеждах. На его переносной торговочной\nлавке расставленно несколько\n"  
 "знакомых вам вещей.\n\n"  
 "Может быть, вам удасться что-то\nкупить у него?\n\n\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Magazin.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=lambda x: set\_screen('5.2')))  
 Magazin.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Magazin)  
 def piy(self, event):  
 return Saving  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Magazin2(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Magazin2, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Magazin = FloatLayout()  
 Magazin.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Magazin.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Magazin.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=lambda x: set\_screen('6')))  
 Magazin.add\_widget(Label(text=" Предметы Купить \n\n"  
 "Обычное зелье лечения\n\nСильное зелье лечения\n\nЭпическое зелье лечения\n\n\n\n\n\n\n\n",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Magazin.add\_widget(Button(text="5 трофеев",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1], size\_hint=(.3, .1),  
 pos=(400, 455),  
 background\_normal="", on\_press=self.обычное\_зелье))  
 Magazin.add\_widget(Button(text="10 трофеев",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1], size\_hint=(.3, .1),  
 pos=(400, 395),  
 background\_normal="", on\_press=self.сильное\_зелье))  
 Magazin.add\_widget(Button(text="15 трофеев",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1], size\_hint=(.3, .1),  
 pos=(400, 335),  
 background\_normal="", on\_press=self.эпическое\_зелье))  
 self.add\_widget(Magazin)  
 def обычное\_зелье(self, event):  
 global Usual\_healing\_potion  
 global Trophies  
 if (Trophies - 5) > 0:  
 Usual\_healing\_potion += 1  
 Trophies -= 5  
 else:  
 set\_screen('Нет\_средств2')  
 def сильное\_зелье(self, event):  
 global Powerful\_healing\_potion  
 global Trophies  
 if (Trophies - 10) > 0:  
 Powerful\_healing\_potion += 1  
 Trophies -= 10  
 else:  
 set\_screen('Нет\_средств2')  
 def эпическое\_зелье(self, event):  
 global Epic\_healing\_potion  
 global Trophies  
 if (Trophies - 15) > 0:  
 Epic\_healing\_potion += 1  
 Trophies -= 15  
 else:  
 set\_screen('Нет\_средств2')  
 def piy(self, event):  
 return Saving  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class N0\_money2(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(N0\_money2, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 No = FloatLayout()  
 No.add\_widget(Button(text="У Вас недостаточно трофеев!\n\n(нажмите для продолжения)",  
 font\_size=25, color=[62 / 255, 28 / 255, 0 / 255, 1],  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1], size\_hint=(.7, .3), pos=(90, 275),  
 background\_normal="", on\_press=lambda x: set\_screen('5.2')))  
 self.add\_widget(No)  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Ozero(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Ozero, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Ozero = FloatLayout()  
 Ozero.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Ozero.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Ozero.add\_widget(Label(text="Вы вышли на большую ровную площадку.\nПод потолком пещеры кружат маленькие\n"  
 "летучие мышки, вроде бы безобидные...\n\n"  
 "Вы замечаете небольшой костёр,\nоставленный кем-то в этом месте.\n\n"  
 "Что ж, может и вам стоит передохнуть?\n\n\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Ozero.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=self.proda))  
 Ozero.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Ozero)  
 def piy(self, event):  
 return Saving  
 def proda(self, event):  
 global XP  
 XP = MAX\_XP  
 set\_screen('6.2')  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
class Ozero2(Screen):  
 def \_\_init\_\_(self, \*\*kw):  
 super(Ozero2, self).\_\_init\_\_(\*\*kw)  
 def on\_enter(self):  
 Ozero = FloatLayout()  
 Ozero.add\_widget(Button(text="<-- В главное меню",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(27, 610),  
 background\_normal="", on\_press=lambda x: set\_screen('Главное\_меню')))  
 Ozero.add\_widget(Button(text="Сохранить",  
 background\_color=[185 / 255, 145 / 255, 75 / 255, 1],  
 font\_size=20,  
 color=[62 / 255, 28 / 255, 0 / 255, 1],  
 size\_hint=(.4, .1), pos=(320, 610),  
 background\_normal="", on\_press=self.piy))  
 Ozero.add\_widget(Label(text="Вам удалось отдохнуть без приключений.\n\n"  
 "Поев и восстановив силы, вы готовы\nпродолжить путь.\n\n"  
 "Жаль, но вам всё же придётся\nпокинуть это красивое и тихое место.\n\n"  
 "Ваше XP полностью восстановлено\n\n\n\n",  
 font\_size=25,  
 color=[62 / 255, 28 / 255, 0 / 255, 1]))  
 Ozero.add\_widget(Button(text="Продолжить",  
 background\_color=[180 / 255, 140 / 255, 75 / 255, 1],  
 font\_size=20, color=[62 / 255, 28 / 255, 0 / 255, 1], background\_normal="",  
 size\_hint=(.4, .2), pos=(320, 30),  
 on\_press=lambda x: set\_screen('1')))  
 Ozero.add\_widget(Button(text="XP = "+str(XP),  
 background\_color=[150 / 255, 81 / 255, 69 / 255, 1],  
 font\_size=20, color=[64 / 255, 21 / 255, 21 / 255, 1], background\_normal="",  
 size\_hint=(.3, .10/11), pos=(27, 580)))  
 self.add\_widget(Ozero)  
 def piy(self, event):  
 return Saving  
 def in\_leave(self):  
 self.add\_widget.clear\_widgets()  
  
def set\_screen(name\_screen):  
 sm.current = name\_screen  
  
sm = ScreenManager()  
sm.add\_widget(Main\_menu(name='Главное\_меню'))  
sm.add\_widget(Privet(name='Привет'))  
sm.add\_widget(History(name="История"))  
sm.add\_widget(Rules(name="Правила"))  
sm.add\_widget(My\_character(name="Мой\_персонаж"))  
sm.add\_widget(N0\_class(name="Нет\_класса"))  
sm.add\_widget(Election\_class(name="Выбор\_класса"))  
sm.add\_widget(Beginning\_Wizard(name="Начало\_пути\_маг"))  
sm.add\_widget(Beginning\_Assassin(name="Начало\_пути\_убийца"))  
sm.add\_widget(Beginning\_Warrior(name="Начало\_пути\_воин"))  
sm.add\_widget(Beginning\_Guard(name="Начало\_пути\_страж"))  
sm.add\_widget(Settings(name="Настройки"))  
sm.add\_widget(Shop(name="Магазин"))  
sm.add\_widget(N0\_money(name="Нет\_средств"))  
sm.add\_widget(Difficulty\_is\_set(name="Сложность\_установлена"))  
sm.add\_widget(LovyshkaOFF(name="1"))  
sm.add\_widget(Viktorina1(name="2"))  
sm.add\_widget(Viktorina1T(name="2.1"))  
sm.add\_widget(Viktorina1F(name="2.2"))  
sm.add\_widget(Syndyk1(name="3"))  
sm.add\_widget(Syndyk2(name="3.2"))  
sm.add\_widget(LovyshkaON(name="4"))  
sm.add\_widget(Magazin1(name="5"))  
sm.add\_widget(Magazin2(name="5.2"))  
sm.add\_widget(N0\_money2(name="Нет\_средств2"))  
sm.add\_widget(Ozero(name="6"))  
sm.add\_widget(Ozero2(name="6.2"))  
  
class Андердарк(App):  
 def \_\_init\_\_(self, \*\*kwargs):  
 super(Андердарк, self).\_\_init\_\_(\*\*kwargs)  
 self.config = ConfigParser()  
 def build\_config(self, config):  
 config.adddefaultsection('General')  
 config.setdefault('General', 'user\_data', '{}')  
 def set\_value\_from\_config(self):  
 self.config.read(os.path.join(self.directory, '%(appname)s.ini'))  
 self.user\_data = ast.literal\_eval(self.config.get(  
 'General', 'user\_data'))  
 def get\_application\_config(self):  
 self.config.read(os.path.join(self.directory, '%(appname)s.ini'))  
 self.user\_data = ast.literal\_eval(self.config.get('General', 'user\_data'))  
 def build(self):  
 return sm  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 Андердарк().run()