TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Äriinfotehnoloogia instituut

Programmerimise põhikursus

PROEKT

Mini RPG Game

Üliõpilane:

Aleksei Fjodorov

Tallinn 2014

**Содержание:**

*Описание задачи*.

* 1. Цель работы.
  2. Общение с пользователем
  3. Исходные данные и получаемые результаты

*Описание программы*

* 1. Обаща схема действий(UML Activity Diagramm) для всей программы и для каждой функции.
  2. Текст программы с комментариями.

*Список используемой литературы.*

*Описание задачи.*

Цель проекта – создание мини РПГ игры на языке программирования Python. Научится работать с функциями, списками, циклами, лог.функциями, работа с переменными.

Цель игры – суметь развить за короткое время своего персонажа так, чтоб в коненчом итоге можно было бы победить босса. Бой с боссом можно выбрать в любое время, вся сложность состоит в том, в случае поражения Вашему персонажу придется игру начинать заного, под этим подрузумевается и то, что все накопленные «статы» за все время игры будут утрачены, что приведет к начальному уровню игры.

Всего существует 5 персонажей:

Player

Monster1(stat\*0.8)

Monster2(stat\*1)

Monster3(stat\*1.2)

Boss(stat random [30,50])

В начале игры каждому персонажу дается пять очков(stat). Статы распределяются между:

HP(кол-во жизней) statHP \* 10

ATC(атака) Распределяется между минимумом statATC\*3 и максимумом statATC\*4.5

ACC(точность) увеличивает шанс попадания

Dex(уворотливость)

Общение с пользователем:

Персонаж(Player) распределяет каждый stat на каждую из данных изначально характеристик последовательно.

После распределения характеристик, игроку предоставляется возможность выбора противников:

1. Monster1(клавиша нм. 1)
2. Monster2(клавиша нм. 2)
3. Monster3(клавиша нм. 3)
4. Boss (клавиша нм. 4)
5. Схема(UML Activity Diagram) функции def Player(stat):



На первом этапе происходит непосредственное распределение характеристик Вашего пресонажа. Происходит это следующим образом:

Если количество ваших статов больше чем 0, то на дисплей функция выводит Ваше общее количество очков. Далее следует их распределение. Игрок выбирает самостоятельно куда будет рспределять свои очки. Характеристики персонажа разделены на четыре группы, в каждую из которых игрок может определить лишь одно очко. После чего создается отдельный текстовый документ, где сохраняются характеристики игрока.

1. Функциия def fight():

def fight():

Существует главный цикл который проверяет, выбрал ли персонаж главного босса, а также в ней же лежит, функция чтобы игрок распределял статы, создание монстров, функция генерации. Затем внутри цикла существует, цикл который проверяет закончился бой или идет до сих пор.

Затем идет сам бой, где ходы между игроком и монстром идут по порядку.

1. Функция def generation():

Генерация открывается текст файл монстра и игрока с помошью его определяет его по статам. Так же в ней вычисляется шанс попадания игрока и монстра.

1. Схема(UML Activity Diagram) функции def enemy(statm): 

while statm1>0:

x=int(random.uniform(1,5))

if x==1:

monster1[0]=monster1[0]+1

statm1=statm1-1

elif x==2:

monster1[1]=monster1[1]+1

statm1=statm1-1

elif x==3:

monster1[2]=monster1[2]+1

statm1=statm1-1

elif x==4:

monster1[3]=monster1[3]+1

statm1=statm1-1

Изначально всем монтсрам и боссу дается по одному стату на каждую характеристику, затем в зависимомсти от стата персонажа он умножается на коэфициент для того, что бы определить его силу. Затем игрок выбирает с каким монстор он будет сражаться. После этого создается текстовый фаил монтсра с его данными характеристик.

*Список используемой литературы:*

1. <https://www.ibm.com/developerworks/ru/library/l-python_part_3/>
2. <http://stackoverflow.com/questions/9755538/how-do-i-create-a-list-of-unique-random-numbers>
3. <https://programmeerimine.cs.ut.ee/>
4. <https://docs.python.org/3/tutorial/>