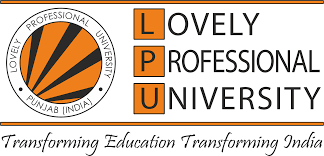
**GAMESON**

Section: K18GE

|  |  |  |
| --- | --- | --- |
| **Registration No.** | **Name of Students** | **Roll No.** |
| 11811502 | **Gourav Tanwar** | RK18GEA21 |
| 11811391 | **Devanshu Sharma** | RK18GEA22 |
| 11811407 | **Sanskar Mandothia** | RK18GEA23 |
| 11811441 | **Rishabh Verma** | RK18GEA24 |



**School of Computer Science Engineering**

**Lovely Professional University, Jalandhar**

**Index**

1.Abstract ...……………………………………………….. 3

2.Introduction ……………………………………………... 3

3.Proposed project ………………………………………… 3

4.Requirement …………………………………………….. 4

1. Software requirement
2. Hardware requirement

5.Module representation …………………………………... 4

6.Flow chart ……………………………………………….. 6

7.Implementation ………………………………………….. 7

**Abstract**

In this we are going to design project “Gameson” where the project consist of two games which are very simple , interesting, attractive and fun loving game. Our purpose to design the project “Gameson” is to built an interest again in the users who have stop playing this types of games which were most popular games in their time. So have design two games “Caterpillar” and “Egg Catcher” in this project and may later add more games like this which can have interest of users and which can interact users towards it. For the game “caterpillar” user will get 10 points each for eating and this will go on increasing until or unless the game has ended or user can directly end the game or game is ended if a user touches any side of the wall and after the game has been ended, it will display your total scores. For the game “Egg Catcher” a user will get total of 3 lives and there is also 10 points given to the user if he/she catches the egg and there is decrease in lives by one if he/she misses the egg but there game will go on until all the 3 lives has been used. But after using all the 3 lives the game will end and it will show your total scores or points.

**Introduction**

The project “Gameson” is a python based projects which is of two different types of games namely “Caterpillar” and “Egg Catcher”. Here to design these games we have used various python libraries and some are turtle, itertools, random and more. The project is not only for the gamers but also for the user with no gaming experience can use it or if a user want to play the game for the first time then it would be a great option. The project aim is to provide a customer or player a better and simple experience so that a user of any age can play this game easily with different option of games.

**Proposed Project**

Gameson is a game project based on the programming language “python”. In this project user will have two different games as mentioned above. The project objective is to provide user a easy, smooth and better experience. It is design considering all age groups and even for the user who is going to play a game for the first time.

As “Caterpillar” game is very famous, easy and interesting game almost of all time. But due to no much modification to this game has decreased the player’s interest as well as players. So we have tried to built this game in a better and more interesting way so that it can interact more and more users towards it, so that new generations can also have chance to play these fun loving games. In this game for every eating there will be increase in score with 10 points and with every increase of points there will be increase speed and if the object touches any side of the box the game will end.

On the other hand “Egg Catcher” is also very interesting and fun loving game that have been designed in various ways with different designs so we have tried to built this game in different and attracting design so that users who had loose their interest in these type of game can built their interest again but now with more better and smooth experience. In this player has total 3 lives. For 1 egg the user will 10 points and this will go on increasing till the user has lives. But if user miss an egg then there will decrease in lives by 1 and when there is no lives left then the game will end.

**Requirement**

Software Requirement :

Python IDLE 2.7 or above.

Operating system : Windows 7 or above, macOS, Linux

Hardware Requirement :

**Recommended system requirements :**

* Processor: Intel i5 processor 4300M at 2.6GHz or 2.59GHz ( 1 socket, 2 cores, 2 threads per core ) 8GB of DRAM Intel ; Xeon processor E5-2698 v3 at 2.30 GHz ( 2 socket, 16 cores, 1 threads per core), 64 GB of DRAM Intel; Xeon Phi processor 7210 at 1.30 GHz ( 1 socket, 64 cores, 4 threads per core) 32 GB of DRAM, 16 GB of MCDRAM ( flat mode enabled).
* Disk space of 2GB to 3GB.

**Minimum requirements :**

Processor : Intel i3 or Intel Atom

Disk space : 1 GB

**Module Representation**

Turtle :

“Turtle” is a Python feature like a drawing board, which lets us command a turtle to draw all over it! We can use functions like turtle.forward(…) and turtle.right(…) which can move the turtle around.

Itertools :

Iterator in python is any python type that can be used with a ‘for in loop’. Python lists, tuples, dicts and sets are all examples of inbuilt iterators. These types are iterators because they implement following methods. In fact, any object that wants to be an iterator must implement following methods.

1. **\_\_iter\_\_** method that is called on initialization of an iterator. This should return an object that has a next or \_\_next\_\_ (in Python 3) method.
2. **next ( \_\_next\_\_ in Python 3)** The iterator next method should return the next value for the iterable. When an iterator is used with a ‘for in’ loop, the for loop implicitly calls next() on the iterator object. This method should raise a StopIteration to signal the end of the iteration.

Random :

**Random** — Generate pseudo-**random** numbers. ... Almost all module functions depend on the basic function **random**() , which generates a **random** float uniformly in the semi-open range [0.0, 1.0).

Tkinter :

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.  
**To create a tkinter:**

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

**Flow Chart**

Start game

Egg Catcher

Lives 3

**Catch egg Miss egg**

1Egg=10 score

Lives 2

**Catch Egg Miss egg**

n E=n\*10score

Lives 1

End game

Start Game

Caterpillar

**Eat**

1Cater=10score

**Touches any side**

End Game

**Implementation**

