 PROJECT REPORT ON ANGRY BIRD

GROUP MEMBERS

Name -Nitin Sinha

Reg No.-11910096

Name-Piyush Mishra

Reg No.-11911715

Name-Rahul Pathak

Reg No.-11910125

Submitted To : Mr. Sagar Pande

INTRODUCTION OF PROJECT(ANGRYBIRD)

In the making of this project we have used python language in building the game and used

many libraries like pygame , pymunk etc . Our game has five levels and as the player

progresses the difficulty level increases thus creating interest in the users . Our main goal is

to provide a good gaming experience and the user should feel comfortable to use and play

but it’s not easy to cross the levels and the user has think how to clear leve’s . We have used

pygame library to build the game and this is the library used for creating games in python .

We have inserted many pictures in the game and also added sound . The two characters that

we have used in the game are red bird and pig . After passing each level the player gets the

stars but maximum number of stars that can be gained are three .

**ABOUT PYGAME**

Pygame

1. Pygame is a cross-platform set of Python modules which is used to create video games.

2. It consists of computer graphics and sound libraries designed to be used with the

Python programming language .

3. Pygame was officially written by Pete Shinners to replace PySDL .

4. Pygame is suitable to create client-side applications that can be potentially wrapped in

a standalone executable.

Python is the most popular programming language or nothing wrong to say that

it is the next-generation programming language. In every emerging field in

computer science, Python makes its presence actively.

[Pygame](https://www.pygame.org/) is a Free and Open Source cross-platform library for the development

of multimedia applications like video games on Python . It uses the Simple

DirectMedia Layer libray and several other popular libraries to abstract most

common functions and makes writing these program a more intuitive task.



**Pygame** is a set of [Python](http://www.python.org/) modules designed for writing video games. Pygame adds

functionality on top of the excellent [SDL](http://www.libsdl.org/) library. This allows you to create fully featured

games and multimedia programs in the python language.

Pygame is highly portable and runs on nearly every platform and operating system.

Pygame itself has been downloaded millions of times.

**Multi core CPUs can be used easily**. With dual core CPUs common, and 8 core CPUs

cheaply available on desktop systems, making use of multi core CPUs allows you to do more

in your game. Selected pygame functions release the dreaded python GIL, which is

something you can do from C code.

**Uses optimized C and Assembly code for core functions**. C code is often 10-20 times faster

than python code, and assembly code can easily be 100x or more times faster than python

code.

**Comes with many operating systems**. Just an apt-get, emerge, pkg\_add, or yast install away.

No need to mess with installing it outside of your operating system's package manager.

Comes with binary pos system installers (and uninstallers) for Windows or MacOSX.

Pygame does not require setup tools with even ctypes to install.

**Truly portable**. Supports Linux (pygame comes with most main stream linux distributions),

Windows (95, 98, ME, 2000, XP, Vista, 64-bit Windows, etc), Windows CE, BeOS, MacOS,

Mac OS X, FreeBSD, NetBSD, OpenBSD, BSD/OS, Solaris, IRIX, and QNX. The code

contains support for AmigaOS, Dreamcast, Atari, AIX, OSF/Tru64, RISC OS, SymbianOS

and OS/2, but these are not officially supported. You can use it on hand held devices, game

consoles and the One Laptop Per Child (OLPC) computer.

**It's Simple** and easy to use. Kids and adults make shooter games with pygame. Pygame is

used in the OLPC project and has been taught in essay courses to young kids and college

students. It's also used by people who first programmed in z80 assembler or c64 basic.

**Many games have been published**. Including Indie Game Festival finalists, Australian Game

festival finalists, popular shareware, multimedia projects and open source games. Over 660

projects have been published on the pygame websites such as: list needed. Many more games

have been released with SDL (which pygame is based on), so you can be sure much of it has

been tested well by millions of users.

**You control your main loop**. You call pygame functions, they don't call your functions. This

gives you greater control when using other libraries, and for different types of programs.

**Does not require a GUI to use all functions**. You can use pygame from a command line if you

want to use it just to process images, get joystick input, or play sounds.

**Fast response to reported bugs**. Some bugs are patched within an hour of being reported. Do a

search on our mailing list for BUG... you'll see for yourself. Sometimes we suck at bug fixes,

but mostly we're pretty good bug fixers. Bug reports are quite rare these days, since a lot of

them have been fixed already.

**Small amount of code**. It does not have hundreds of thousands of lines of code for things you

won't use anyway. The core is kept simple, and extra things like GUI libraries, and effects are

developed separately outside of pygame .

**Modular** . You can use pieces of pygame separately . You can use a different sound libraries

And it will be fine . Many of the core modules can be initialized and used separately .

PYGAME DOCUMENTATION

# Pygame

1. [pygame.\_sdl2.touch](https://devdocs.io/pygame/ref/touch#module-pygame._sdl2.touch) pygame module to work with touch input

2. [pygame.camera](https://devdocs.io/pygame/ref/camera#module-pygame.camera) pygame module for camera use

3. [pygame.cdrom](https://devdocs.io/pygame/ref/cdrom#module-pygame.cdrom) pygame module for audio cdrom control

4. [pygame.cursors](https://devdocs.io/pygame/ref/cursors#module-pygame.cursors) pygame module for cursor resources

5. [pygame.display](https://devdocs.io/pygame/ref/display#module-pygame.display) pygame module to control the display window and screen

6. [pygame.draw](https://devdocs.io/pygame/ref/draw#module-pygame.draw) pygame module for drawing shapes

7. [pygame.event](https://devdocs.io/pygame/ref/event#module-pygame.event) pygame module for interacting with events and queues

8. [pygame.examples](https://devdocs.io/pygame/ref/examples#module-pygame.examples) module of example programs

9. [pygame.font](https://devdocs.io/pygame/ref/font#module-pygame.font) pygame module for loading and rendering fonts

10. [pygame.freetype](https://devdocs.io/pygame/ref/freetype#module-pygame.freetype) Enhanced pygame module for loading and rendering computer fonts

11. [pygame.gfxdraw](https://devdocs.io/pygame/ref/gfxdraw#module-pygame.gfxdraw) pygame module for drawing shapes

12. [pygame.image](https://devdocs.io/pygame/ref/image#module-pygame.image) pygame module for image transfer

13. [pygame.joystick](https://devdocs.io/pygame/ref/joystick#module-pygame.joystick) Pygame module for interacting with joysticks, gamepads, and trackballs.

14. [pygame.key](https://devdocs.io/pygame/ref/key#module-pygame.key) pygame module to work with the keyboard

15. [pygame.locals](https://devdocs.io/pygame/ref/locals#module-pygame.locals) pygame constants

16. [pygame.mask](https://devdocs.io/pygame/ref/mask#module-pygame.mask) pygame module for image masks.

17. [pygame.math](https://devdocs.io/pygame/ref/math#module-pygame.math) pygame module for vector classes

18. [pygame.midi](https://devdocs.io/pygame/ref/midi#module-pygame.midi) pygame module for interacting with midi input and output.

19. [pygame.mixer](https://devdocs.io/pygame/ref/mixer#module-pygame.mixer) pygame module for loading and playing sounds

20. [pygame.mixer.music](https://devdocs.io/pygame/ref/music#module-pygame.mixer.music) pygame module for controlling streamed audio

21. [pygame.mouse](https://devdocs.io/pygame/ref/mouse#module-pygame.mouse) pygame module to work with the mouse

22. [pygame.pixelcopy](https://devdocs.io/pygame/ref/pixelcopy#module-pygame.pixelcopy) pygame module for general pixel array copying

23. [pygame.scrap](https://devdocs.io/pygame/ref/scrap#module-pygame.scrap) pygame module for clipboard support.

24. [pygame.sndarray](https://devdocs.io/pygame/ref/sndarray#module-pygame.sndarray) pygame module for accessing sound sample data

25. [pygame.sprite](https://devdocs.io/pygame/ref/sprite#module-pygame.sprite) pygame module with basic game object classes

26. [pygame.surfarray](https://devdocs.io/pygame/ref/surfarray#module-pygame.surfarray) pygame module for accessing surface pixel data using array interfaces

27. [pygame.tests](https://devdocs.io/pygame/ref/tests#module-pygame.tests) Pygame unit test suite package

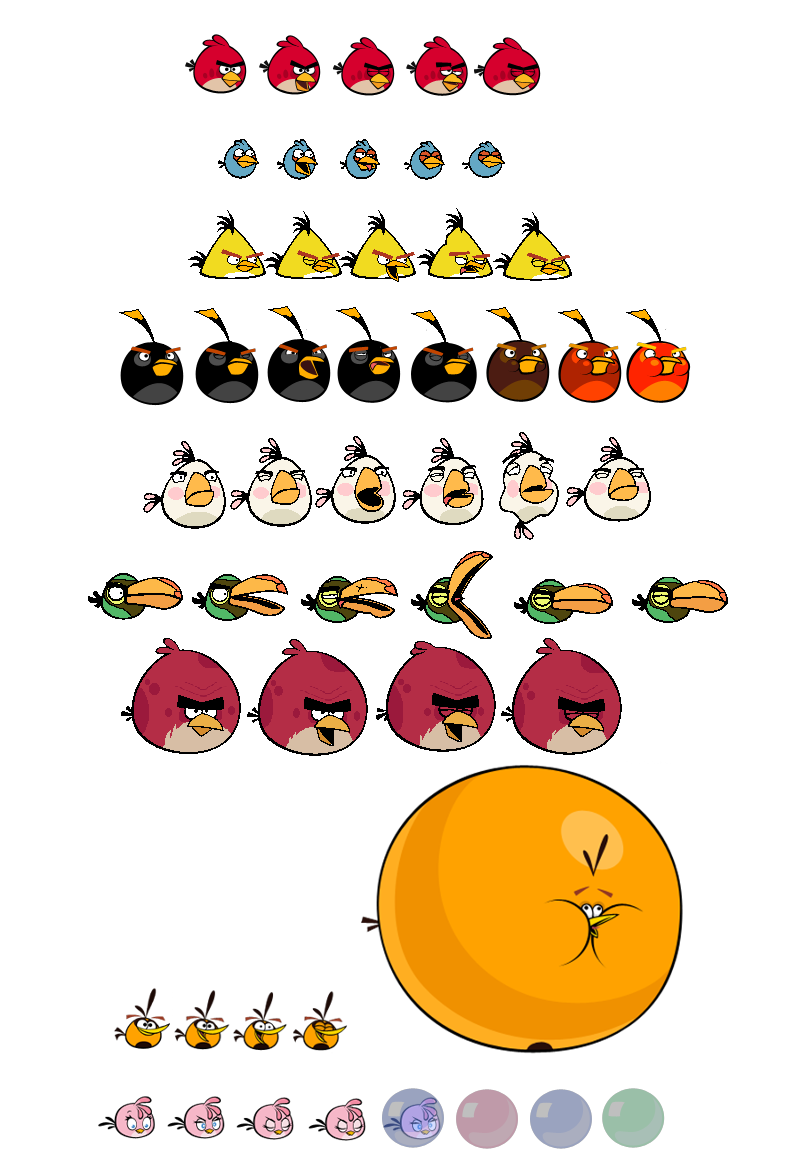
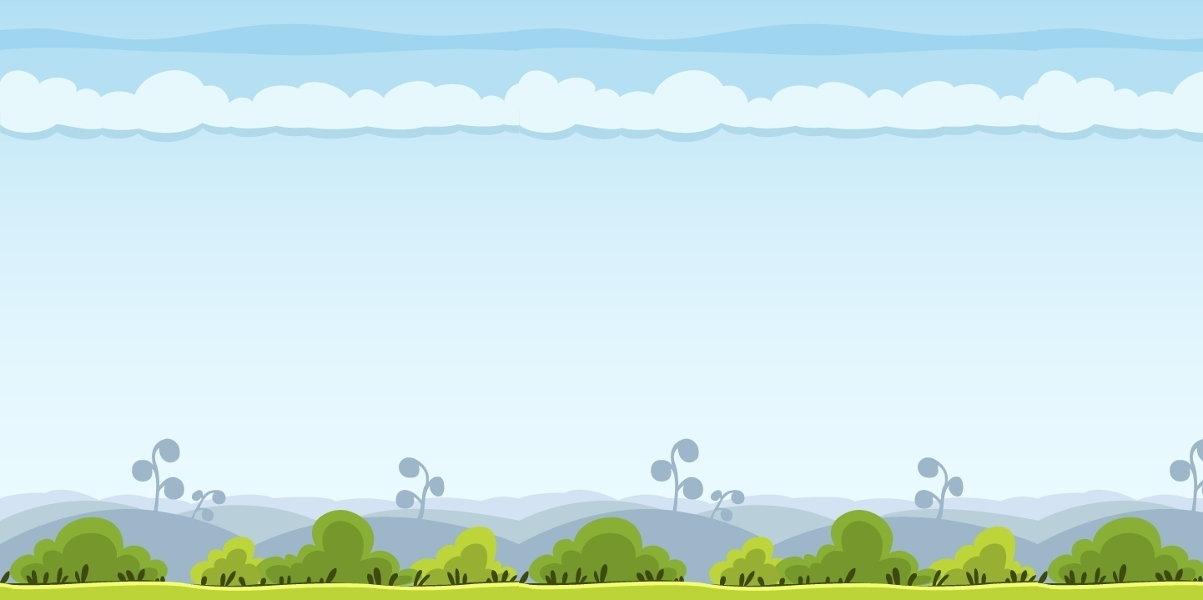
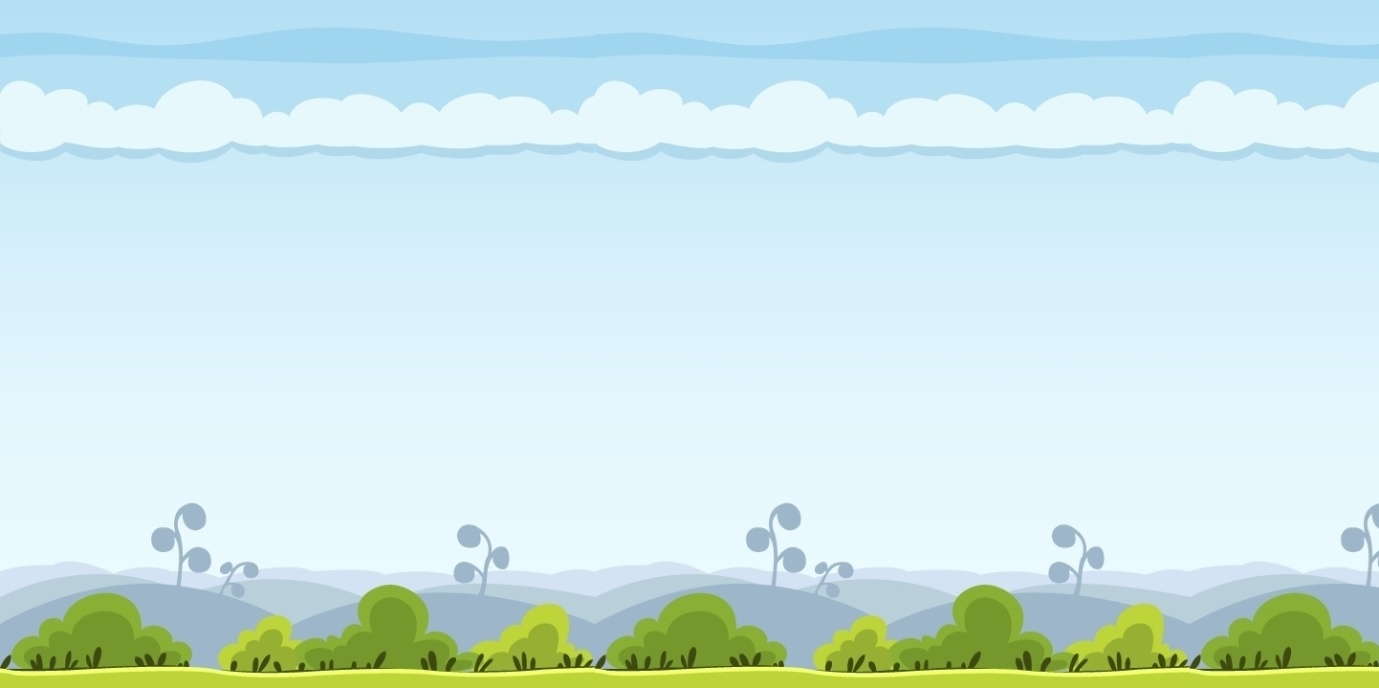
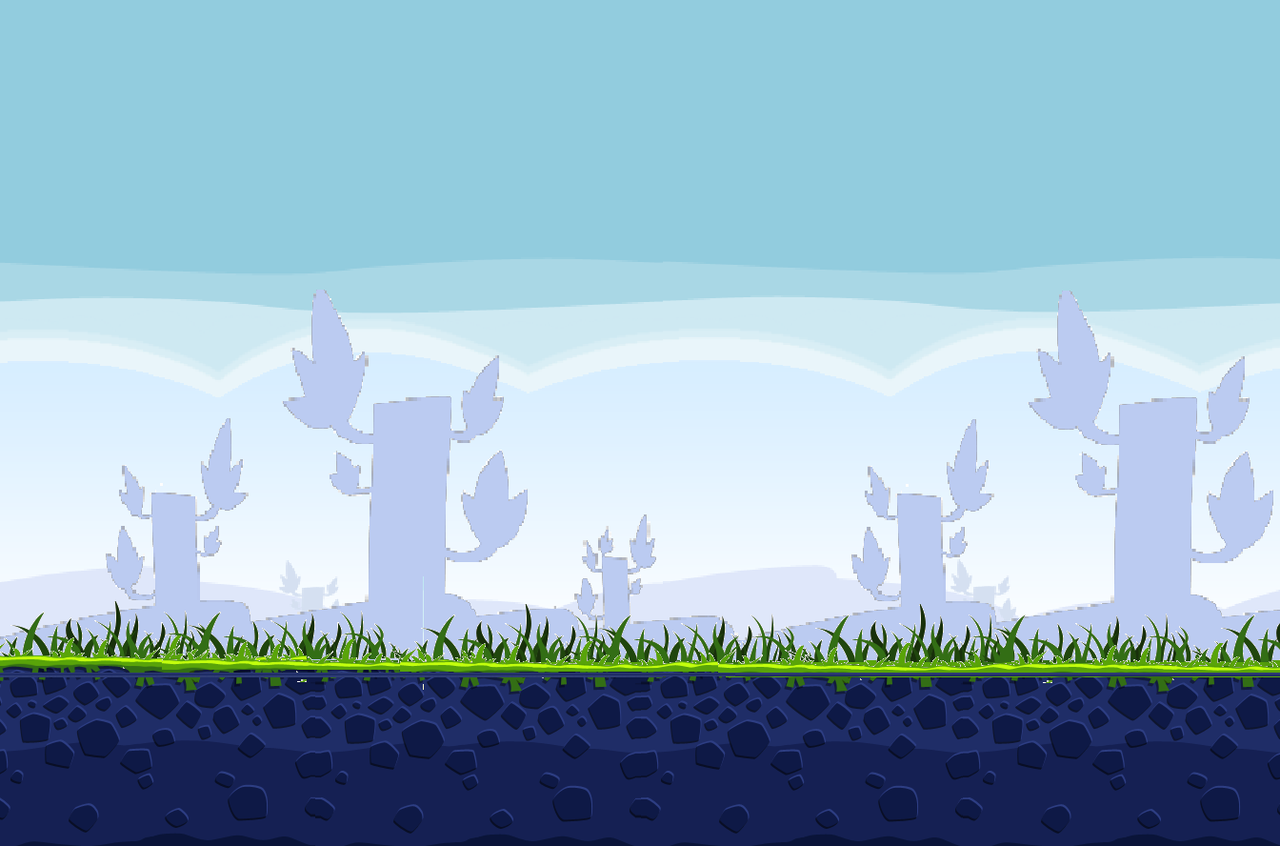
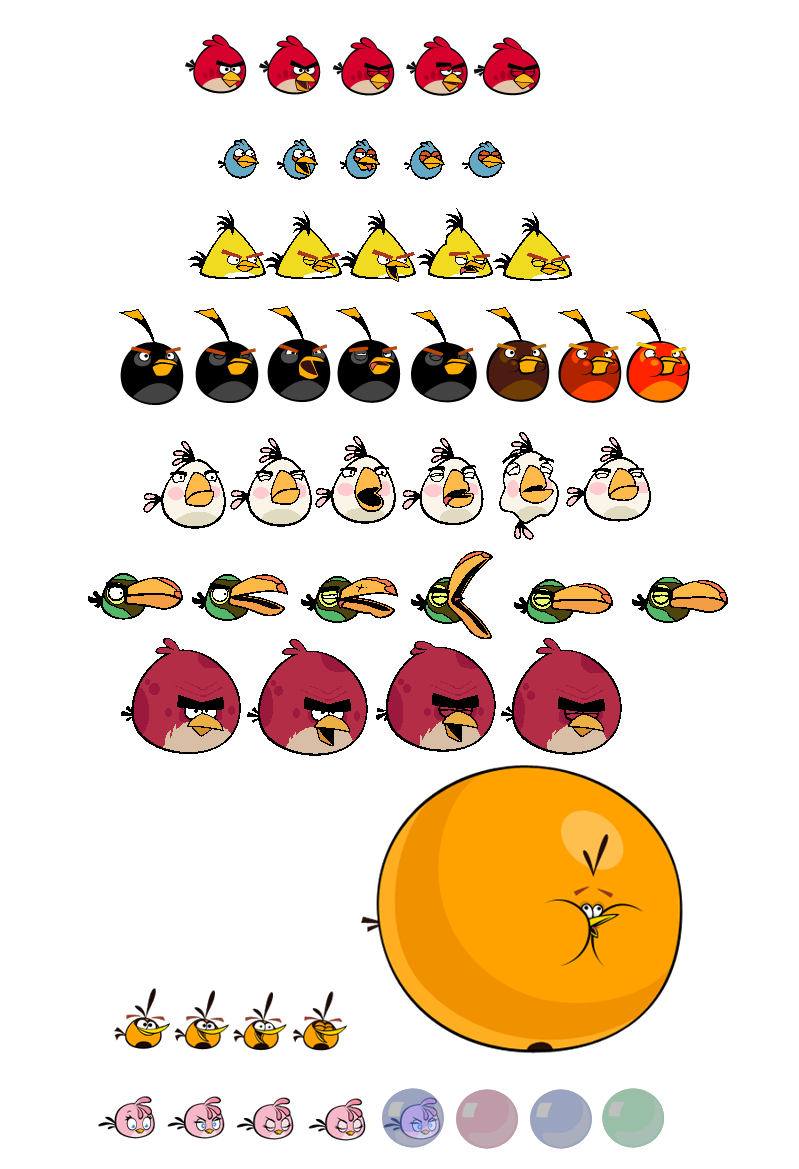
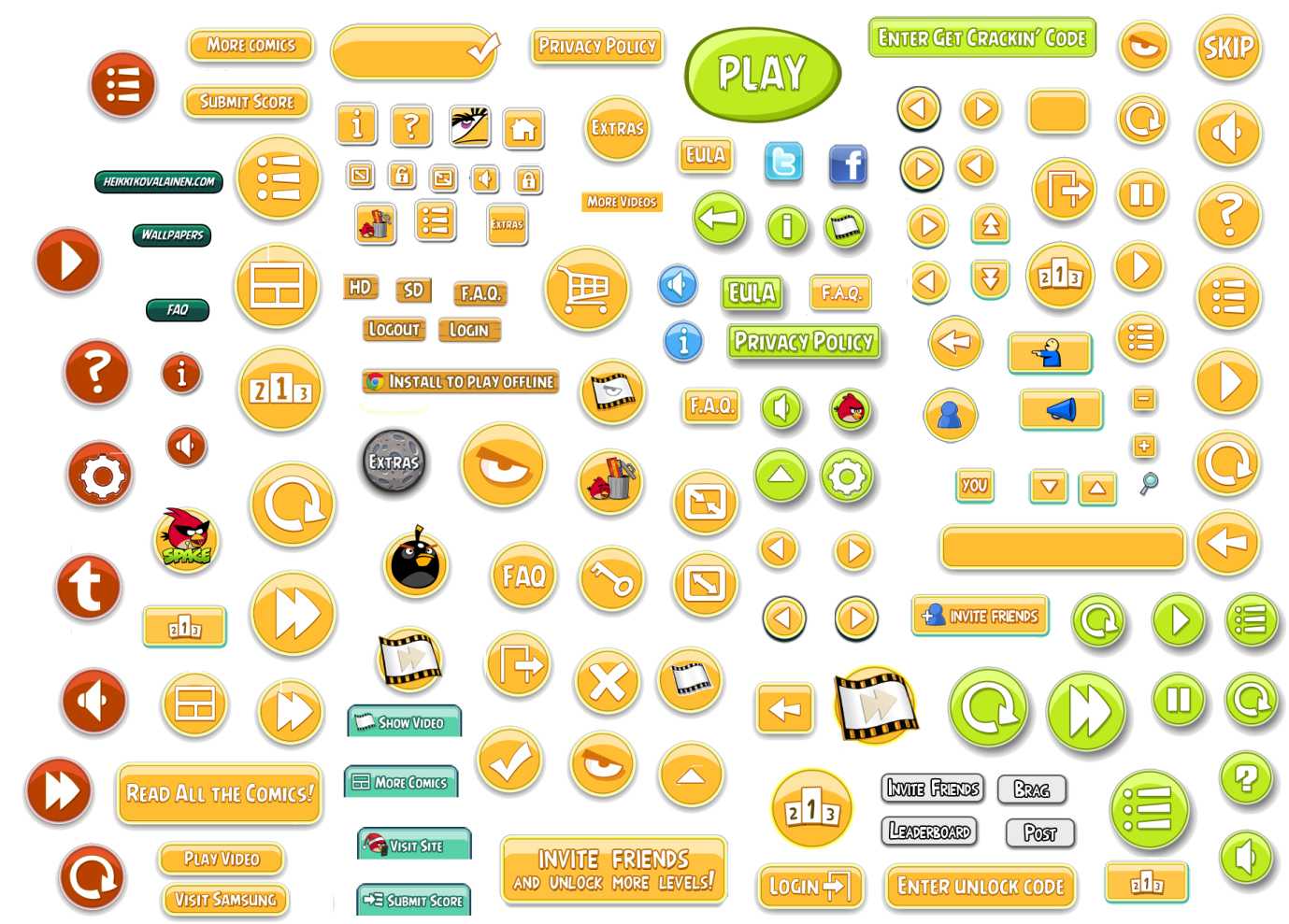
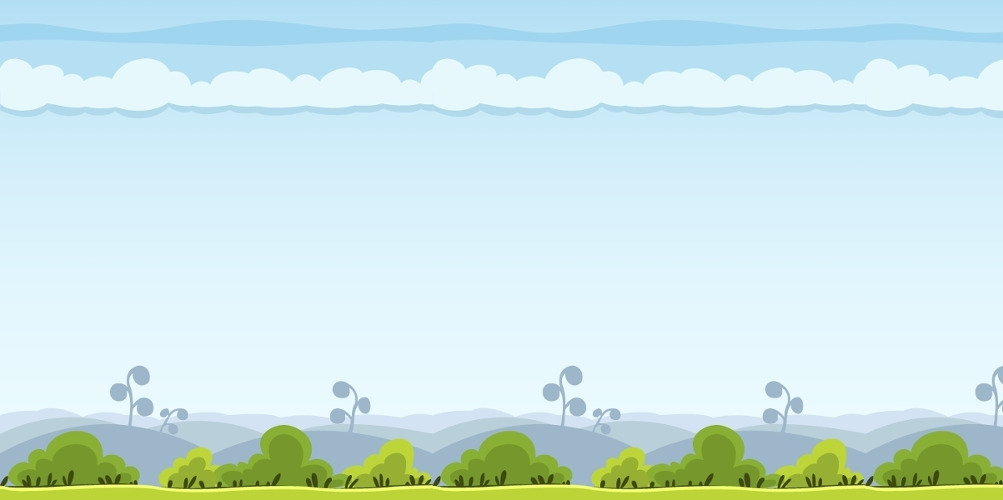
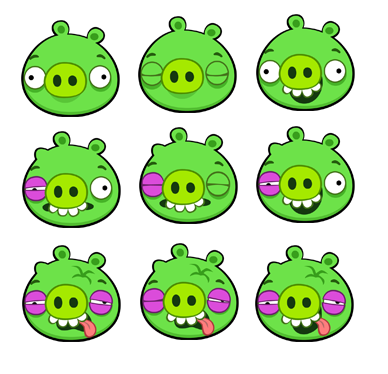
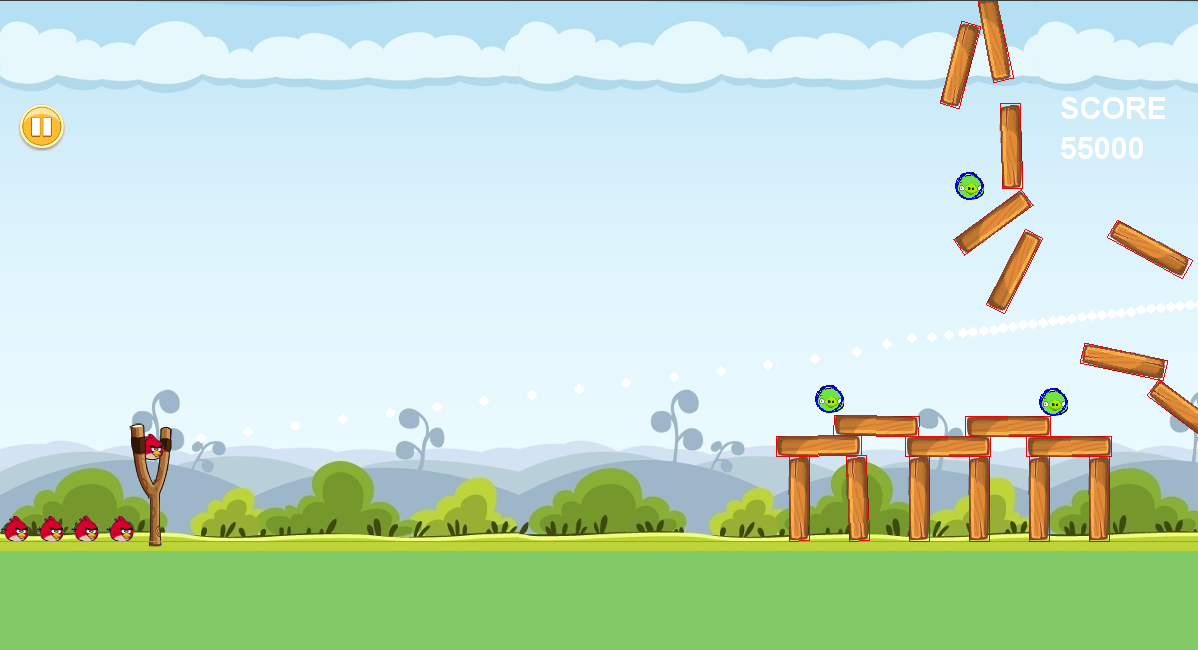
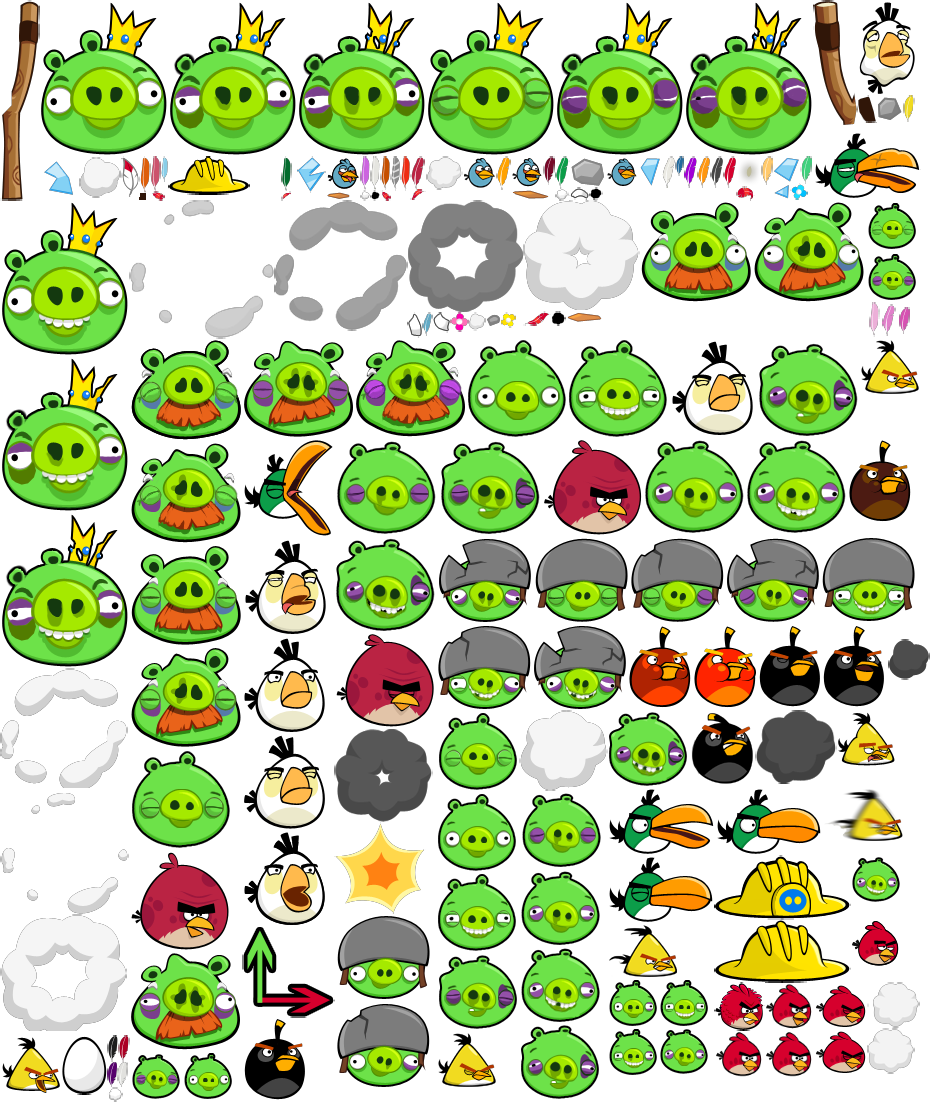
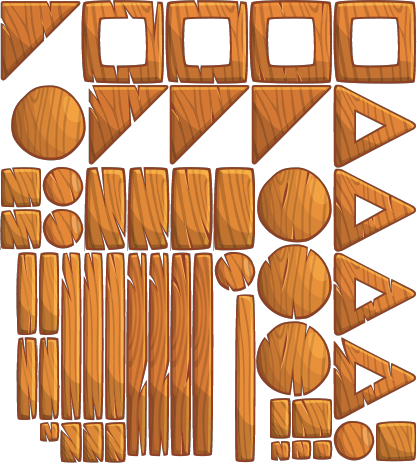
28. [pygame.time](https://devdocs.io/pygame/ref/time#module-pygame.time) pygame module for monitoring time

29. [pygame.transform](https://devdocs.io/pygame/ref/transform#module-pygame.transform) pygame module to transform surfaces

30. [pygame.version](https://devdocs.io/pygame/ref/pygame#module-pygame.version) small module containing version information

31. [pygame](https://devdocs.io/pygame/ref/pygame#module-pygame) the top level pygame package

IMAGES IN GAME USED

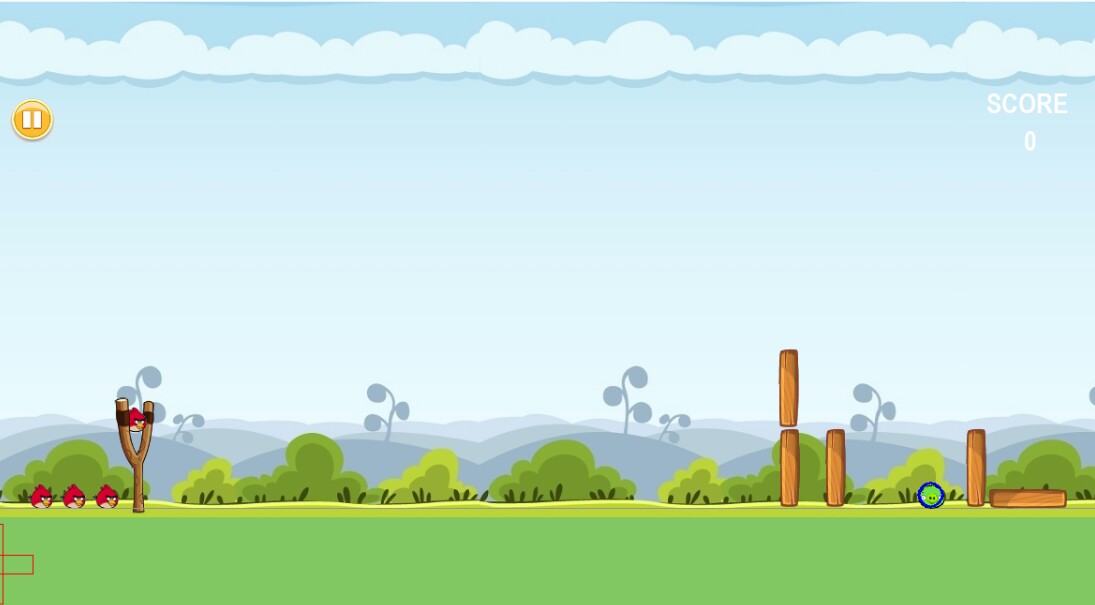
 reeeecolumn.pngred-bird2.pngred-bird3.png

ANGRY BIRD DIFFERENT LEVELS IMAGES

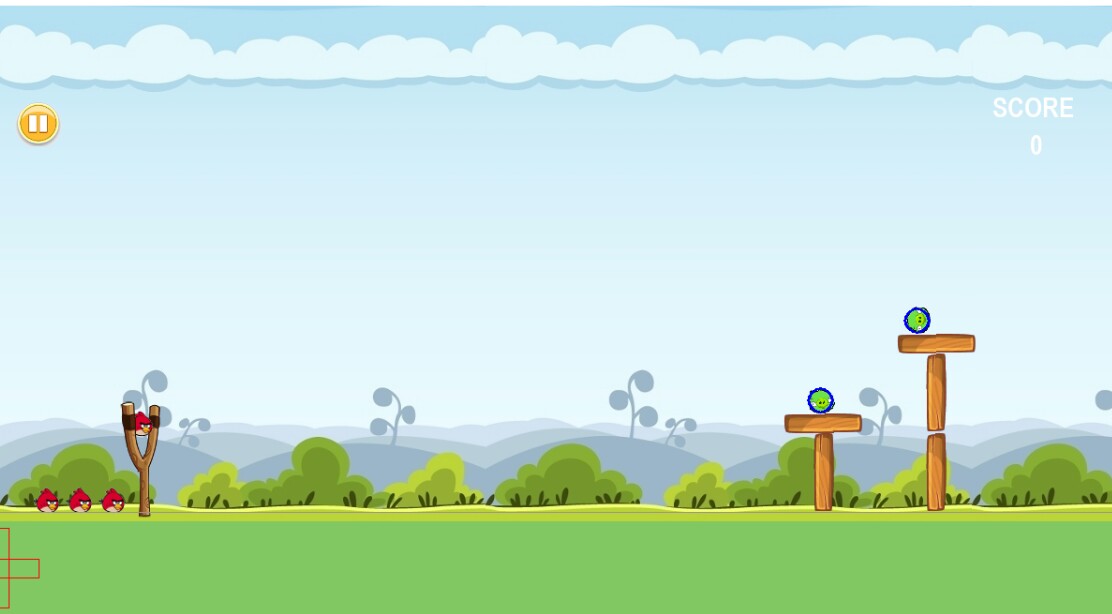
LEVEL 1 :



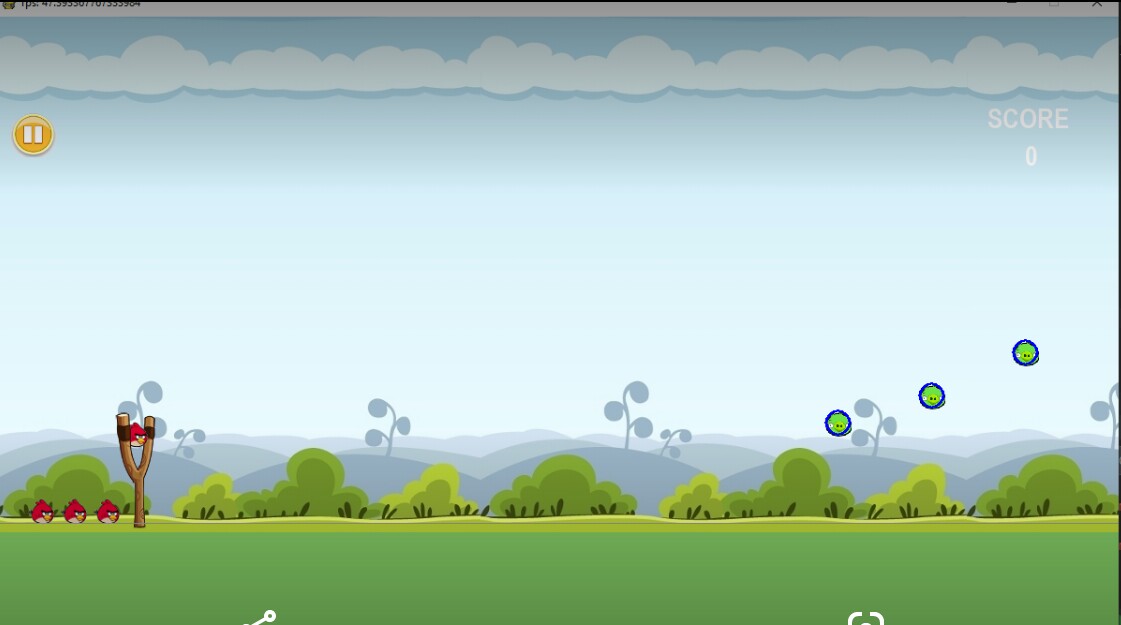
LEVEL 2 :



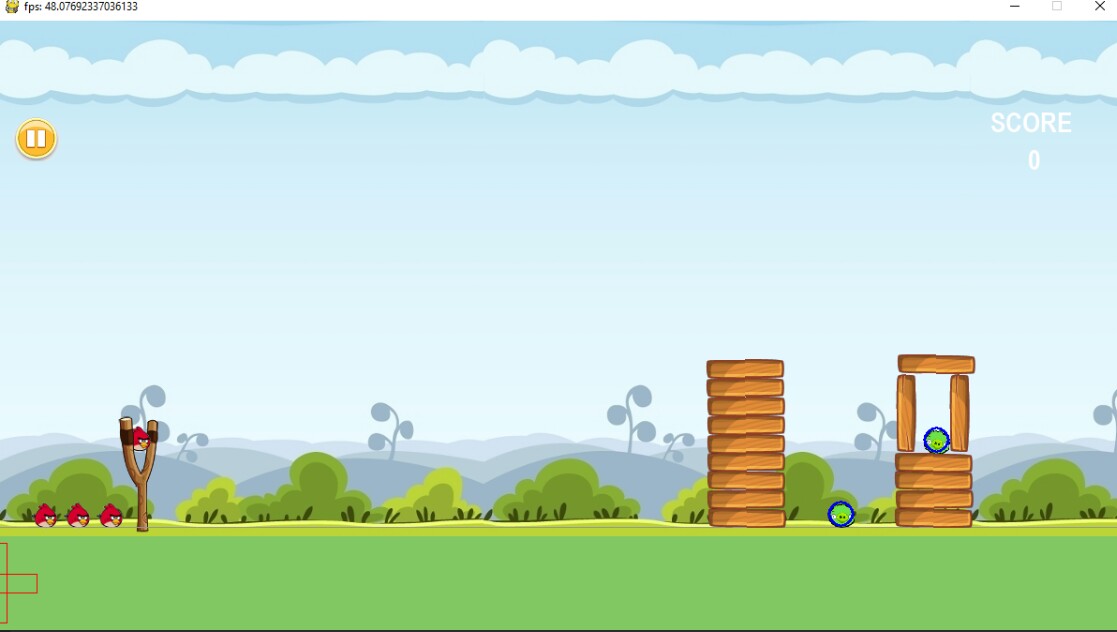
LEVEL 3:



LEVEL 4:



LEVEL 5:



**REFERENCES :**

1. [www.wikipedia.org](http://www.wikipedia.org)

2. [www.pygame.org](http://www.pygame.org)

3. [www.python.org](http://www.python.org)

4. [www.w3schools.org](http://www.w3schools.org)

5. [www.tutorialspoint.com](http://www.tutorialspoint.com)