**ARAKNOID**

A PROJECT REPORT

Submitted by

c saminathan 180701210

m saravanan 180701219

**In partial fulfillment of the award of the degree**

**of**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**RAJALAKSHMI ENGINEERING COLLEGE , CHENNAI**

  ****

**MAY -2019**

**ANNA UNIVERSITY**

**CHENNAI**

**BONAFIDE CERTIFICATE**

Certified that this project **“ARAKNOID”**is the bonafide work of **“M SARAVANAN and C SAMINATHAN”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr.P.KUMAR M.E Ph.D.,** | **Mr.M.CHITHAMBARATHANU** |
| **HEAD OF THE DEPARTMENT** | **ASSISTANT PROFESSOR(SS)** |
| Dept. of Computer Science and Engg, | Dept. of Computer Science and Engg, |
| Rajalakshmi Engineering College,Chennai | Rajalakshmi Engineering College,Chennai |

Submitted for the **ANNA UNIVERSITY** practical examination Mini-Project work viva voce held on \_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**ABSTRACT**

Arkanoid is one of the oldest block buster games which made the previous generation go crazy. Here we have recreated the modern version of it. In this, the ball initially is in touch with the bar. After the ball is released, it hits the series of rectangular blocks at the top of it and the ball returns. If the ball hits a block, that particular block gets vanished and a part of the block falls down,if the residue of the block hits the bar,the life gets reduced with the exception to the power blocks. Each time when the ball returns to the bar, the player has to move the block (only left and right) and project it to the blocks. If the player fails to move the bar where ball is coming, and leaves the ball without touching, he loses a life. For each player, initially 3 lives are given. After all the blocks are disappeared, the player completes that level. We have 5 levels and at each level, the difficulty of the game gets increased and also the player gets extra one life.There are also powers like Big ball,Long Bar,Slow ball remain hidden in random blocks . They will fall only after hitting those particular blocks.If the player catch those block while falling he will get the power for the upcoming ten seconds.The player has to complete all the Five levels to win the game.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson

**DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance

No Words Of Gratitude Will Suffice For The Unquestioning Support Extended To Us By Our Head Of The Department **Dr.P.KUMAR M.E Ph.D.,** For Being Ever Supporting Force During Our Project Work

We also extend our sincere and hearty thanks to our internal guide **Mr.M.CHITHAMBARATHANU B.E.,M.Tech.,**for his valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of information technology.

SARAVANAN M(180701219)

SAMINATHAN C(180701210)

**TABLE OF CONTENTS**

CHAPTER TITLE PAGE

**NO**

**ABSTRACT**

**1 INTRODUCTION 1**

1.1 introduction 1

1.2 SCOPE OF THE WORK 1

1.3 PROBLEM STATEMENT 2

1.4 AIM AND OBJECTIVES OF THE PROJECT 2

**2 SYSTEM SPECIFICATIONS 3**

2.1 HARDWARE SPECIFICATIONS 3

2.2 SOFTWARE SPECIFICATIONS 3

**3 MODULE DESCRIPTION 4**

**4 SYSTEM DESIGN 6**

4.1 ARCHITECTURE DIAGRAM 6

4.2 FLOW CHART 7

**5 CODING 8**

**6 SCREEN SHOTS 31**

**7 CONCLUSION AND FUTURE ENHANCEMENT 41**

**REFERENCES 42**

**CHAPTER 1**

**INTRODUCTION**

* 1. INTRODUCTION

Araknoid is a arcade game which was developed in 19th century. It was a great entertainer to a large mob of all ages. For nearly 40 years it has been a unforgettable game in every ones lifetime.The main theme of the game is to hit all the blocks in the screen while balancing it with a bar without leaving it below. Will the player break all the blocks and win the game?

* 1. **SCOPE OF THE WORK**

Araknoid is purely an entertainment game created only for the enjoyment of the people. We have recreated a new version of the classic game which has been one of the most loved game of our childhood.It is a mind relaxing game made just for fun. Without giving a second to take a look out of the screen, the game keeps the player fully into the game. The main change from the previous versions is the breaking blocks function. Each time when we hit a block, a piece of the block falls down and the player should be cautious not to touch them. Few even has powers in them. By Catching those blocks the player get some advantages like slow ball,long bar,big ball which will make his gameplay easy

* 1. **PROBLEM STATEMENT**

We have to control the ball and direct it towards the appropriate place considering boundaries and collision conditions. We have to manage the attributes of the blocks, bar and the ball, be userfriendly and allow him to operate the game as easy as possible.

* 1. **AIM AND OBJECTIVES OF THE PROJECT**

The aim of the project is to recreate the araknoid game with added functions and modified gameplay. We have created this game with more different levels and new functions and with a little increased difficulty level.The distinct thing in this game is its hard gameplay. One would even say that it is impossible to complete even one level. Difficulty keeps accreting at each second.

**CHAPTER 2**

**SYSTEM SPECIFICATIONS**

**2.1 HARDWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
| Processor | **:** | Pentium IV Or Higher |
| Memory Size | **:** | 256 GB (Minimum) |
| HDD | **:** | 40 GB (Minimum) |

**2.2 SOFTWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
| Operating System | **:** | WINDOWS XP or Higher |
| Front – End | **:** | python |
| Back - End | **:** | Microsoft access |
| Language | **:** | python |

**CHAPTER 3**

**MODULE DESCRIPTION**

1. Pygame:

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.Pygame is primariry used to create games using python platform. Using pygame only the whole game is created . Pygame is not a built in module. It is necessary to install it before running this game.Pygame adds functionality on top of the excellent SDL 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.

1. **Stack:**

Stack is a user defined module created to use attributes of ball and bar such as speed length and radius. The stack module consists of a stack which contains the above listed attributes. We retrieve and use these attributes in the specified levels.At each level we pop up the attributes and use them to create the images. The stack module consist of attributes of all the objects used in the game for all the levels

1. **Time:**

Time is a inbuilt module in python. This module provides a number of functions to deal with dates and the time. The sleep function from the time module is used in the game for a small pause at appropriate places. Python time module provides the ability to read, represent, and reset the time information in many ways.

**Some of the useful time functions in Python:**

* **time.time()**
* **time.clock()**
* **time.ctime()**
* **time.sleep()**
* **time.struct\_time class**
* **time.strftime()**

1. **Random:**

Random is a inbuilt module in python. It is used to generate random numbers.Random module implements a pseudo-random number generator, and contains methods that let us directly solve many different programming issues where randomness comes into play.

Here it used for randomly allocating colors for the blocks and hiding random powers in random blocks .

**CHAPTER 4**

**SYSTEM DESIGN**

* 1. ARCHITECTURE DIAGRAM

INTRO SCREEN

Exit

START GAME

HIGH SCORE

**4.2 FLOW CHART**

Highscore.txt

stack.py

INTRO SCREEN

QUIT

START

HIGH SCORE

LEVEL=1,RUN=True,

WIN=false

WHILE LEVEL<=5 and RUN

PRINT “YOU LOST”

PRINT “YOU WON”

LEVEL+=1

TRUE

PLAY

if Run == True

**CHAPTER 5**

**SAMPLE CODING**

import pygame

from pygame.locals import \*

from stack import \*

import time

import random

#COLORS

BLACK=[0,0,0]

WHITE=[255,255,255]

GREY=[128,128,128]

RED=[255,0,0]

SILVER=[192,192,192]

PINK=(255,20,147)

YELLOW=(255,255,0)

BLUE=(0,255,255)

GREEN=(0,255,0)

VIOLET=( 108, 52, 131 )

ORANGE=(255, 117, 26)

GREEN2= [ 0,255,127]

PINK2=[144, 12, 63]

def reset():

#stack

global s,emptystr,highscore,show\_score,launched,Start,win,speedvar,change\_speed,changespeedinterval,bar,lostlife

global total\_score,Level\_Score,block\_list,all\_sprites\_list,player\_sprite,sprited\_block\_rect,iron\_block\_list,hs,level,run

global counterval0,counterval1,counterval2,counter0,counter1,counter2

counter0=False

counter1=False

counter2=False

counterval0=counterval1=counterval2=0

s=Stack()

emptystr=str()

highscore=int()

show\_score=bool()

launched=bool()

Start=bool()

win=bool()

speedvar=int() #speedchange for each changespeedinterval

change\_speed=bool() # whether the speed change at the instant

changespeedinterval=int() # for no.of.Level\_Score increase, the speed will increase

bar=0

lostlife=0

total\_score=0 # total Level\_Score

Level\_Score=0 # Level\_Score of each level

#Group

block\_list=pygame.sprite.Group()

all\_sprites\_list=pygame.sprite.Group()

player\_sprite=pygame.sprite.Group()

#List

sprited\_block\_rect=list()

iron\_block\_list=list()

power\_block\_list=list()

hs=0

with open("HighScore.txt",'r') as f:

hs=f.read()

clock=pygame.time.Clock()

level=1

run=True

# self.life=3

class Block(pygame.sprite.Sprite):

def \_\_init\_\_(self,color,w,h):

super().\_\_init\_\_()

self.color=color

self.image=pygame.Surface([w,h])

self.image.fill(self.color)

self.rect=self.image.get\_rect()

self.health= 200 if color == [128,128,128] else 100

def update(self):

self.rect.y+=10

class Bar(pygame.sprite.Sprite):

def \_\_init\_\_(self,color,w,h):

super().\_\_init\_\_()

self.length=w

self.height=h

self.color=color

self.image=pygame.Surface([self.length,self.height])

self.image.fill(self.color)

self.rect=self.image.get\_rect()

self.rect.x= screen\_width//2

self.rect.y= screen\_height - (self.height-1) # (h is the bar's length)

class Ball(pygame.sprite.Sprite):

def \_\_init\_\_(self,color,radius,speed,speedvar):

global lostlife

super().\_\_init\_\_()

self.color=color

self.radius=radius

self.x\_speed=self.y\_speed=speed

self.speedvar=speedvar

self.life=3-lostlife

self.image=pygame.Surface([self.radius,self.radius],pygame.SRCALPHA) #SCRALPHA for TRANSOARENCY

self.image.fill(BLUE)

self.image.set\_colorkey(WHITE)

self.rect=self.image.get\_rect()

pygame.draw.ellipse(self.image,self.color,self.rect)

def update(self):

global Start

global changespeedinterval

global changespeed

global win

global launched

global bar

global lostlife

global fall\_list

global counterval0,counterval1,counterval2

global counter0,counter1,counter2

global ball

global specialcondition

index=0

if self.rect.x+self.radius > screen\_width or self.rect.x < 0 :

self.x\_speed\*=-1

#index=self.rect.collidelistall(all\_block\_rect)

hit=True if len(sprited\_block\_rect) >0 else False

# hit become True if the ball collides with any blocks

if hit :

self.rect.y+=10

self.y\_speed\*=-1

if self.rect.y< 0 or self.rect.colliderect(bar.rect) :

self.y\_speed\*=-1

if bar.rect.x<= self.rect.x <= bar.rect.x + bar.length//3 :

if self.x\_speed > 0 :

self.x\_speed\*=-1

elif bar.rect.x+bar.length//2 <= self.rect.x <= bar.rect.x + bar.length :

if self.x\_speed < 0 :

self.x\_speed\*=-1

if self.rect.y> screen\_height and self.life >0.5 and launched :

self.life-=1

lostlife+=1

time.sleep(0.5)

self.rect.x=bar.rect.x+30

self.rect.y=bar.rect.y-30

launched=False

if self.rect.y+self.radius > screen\_height and self.life <=1:

Start=False

win=False

for block in fall\_list:

if block.rect.y<screen\_height:

block.update()

if block.rect.y>screen\_height:

fall\_list.remove(block)

if block.rect.colliderect(bar.rect):

for block in fall\_list:

block.rect.y+=screen\_height

fall\_list.clear()

if block.color==PINK2:

counter1=True

colorb=bar.color

heightb=bar.height

lenghtb=bar.length

xpot=bar.rect.x

ypot=bar.rect.y

all\_sprites\_list.remove(bar)

bar=Bar(PINK2,lenghtb+18,20)

all\_sprites\_list.add(bar)

bar.rect.x=xpot

bar.rect.y=ypot

specialcondition=True

break

elif block.color==BLACK: #slow ball

counter2=True

self.x\_speed\*=0.5

self.y\_speed\*=0.5

break

elif block.color==WHITE:

counter0=True

rad=self.radius

speed=self.x\_speed

speedv=self.speedvar

xpo=self.rect.x

ypo=self.rect.y

all\_sprites\_list.remove(ball)

player\_sprite.remove(ball)

ball=Ball(GREEN,rad+30,speed,speedv)

ball.rect.x=xpo

ball.rect.y=ypo

all\_sprites\_list.add(ball)

player\_sprite.add(ball)

self.update()

self.radius+=30

break

if self.life > 0.5:

self.life-=0.25

lostlife+=0.25

self.rect.x=bar.rect.x+30

self.rect.y=bar.rect.y-30

launched=False

else:

Start=False

win=False

break

if counter0 or counter1 or counter2:

if counterval0>=1000:#reverse condition

counterval0=0

counter0=False

rad=self.radius

speed=self.x\_speed

speedv=self.speedvar

xpo=self.rect.x

ypo=self.rect.y

all\_sprites\_list.remove(ball)

player\_sprite.remove(ball)

ball=Ball(GREEN,rad-30,speed,speedv)

ball.rect.x=xpo

ball.rect.y=ypo

all\_sprites\_list.add(ball)

player\_sprite.add(ball)

self.radius-=30

self.update()

elif counterval1>=1000:

counter1=False

counterval1=0

xpot=bar.rect.x

ypot=bar.rect.y

all\_sprites\_list.remove(bar)

bar=Bar(SILVER,bar\_length,20)

all\_sprites\_list.add(bar)

bar.rect.x=xpot

bar.rect.y=ypot

specialcondition=False

elif counterval2>=1000:

counterval2=0

counter2=False

self.x\_speed\*=2

self.y\_speed\*=2

if len(block\_list) <= 0 : #when all the blocks are demolished, finish the level

Start=False

win=True

if Level\_Score == changespeedinterval : #for every changespeedinterval value increment the speed of the ball

changespeed=True

if Level\_Score%10== 0 and Level\_Score>=10 and changespeed: #Level\_Score>=10 is not to increment the speed for values less than zero since mod 10 of every value below 11 is 0

self.x\_speed\*=self.speedvar

self.y\_speed\*=self.speedvar

changespeed=False

changespeedinterval+=10

# finally increment x and y speed

self.rect.x+=self.x\_speed

self.rect.y-=self.y\_speed

hs=0

with open("HighScore.txt",'r') as f:

hs=f.read()

def introscreen():

global hs

global Start

global emptystr

global show\_score

global level

titlefont=pygame.font.SysFont("Comic sans MS",48)

font=pygame.font.SysFont(None,48)

title=titlefont.render("Araknoid",True,BLACK,GREEN)

title\_rect=title.get\_rect()

play\_text=font.render("Play",True,BLACK,GREEN)

play\_rect=play\_text.get\_rect()

quit\_text=font.render("Quit",True,BLACK,GREEN)

quit\_rect=quit\_text.get\_rect()

high\_score=font.render("High Score",True,BLACK,GREEN)

high\_score\_rect=high\_score.get\_rect()

show\_hscore=font.render(emptystr,True,BLACK)

show\_hscore\_rect=show\_hscore.get\_rect()

title\_rect.center=screen.get\_rect().center

title\_rect.centery=screen.get\_rect().centery - 130

play\_rect.center=screen.get\_rect().center

quit\_rect.centerx=screen.get\_rect().centerx

quit\_rect.centery=screen.get\_rect().centery +60

high\_score\_rect.centerx=screen.get\_rect().centerx

high\_score\_rect.centery=screen.get\_rect().centery +120

show\_hscore\_rect.centerx=screen.get\_rect().centerx

show\_hscore\_rect.centery=screen.get\_rect().centery +200

screen.blit(title,title\_rect)

screen.blit(play\_text,play\_rect)

screen.blit(quit\_text,quit\_rect)

screen.blit(high\_score,high\_score\_rect)

screen.blit(show\_hscore,show\_hscore\_rect

for event in pygame.event.get():

mpos=pygame.mouse.get\_pos()

if event.type == pygame.QUIT:

pygame.quit()

quit()

elif event.type==pygame.MOUSEBUTTONDOWN and event.button == 1:

if play\_rect.collidepoint(mpos) :

Start=True

breeak

elif quit\_rect.collidepoint(mpos):

pygame.quit()

quit()

# setting highscore button as ON AND OFF switch

elif high\_score\_rect.collidepoint(mpos) and not show\_score :

emptystr=str(hs)

show\_score=True

elif high\_score\_rect.collidepoint(mpos) and show\_score :

emptystr=str()

show\_score=False

def Start\_Game(iron):

# iron arguments says number of iron block rows

pygame.mouse.set\_visible(False)

posy=0

colourlist=[RED,SILVER,PINK,YELLOW,VIOLET,GREEN2]

power\_block\_list=[]

powlist=['bigball','longbar','slowball']

for i in range(5):

posx=0

for j in range(10):

color= GREY if i==0 else colourlist[random.randint(0,5)]

block=Block(color,47,25)

block.rect.x=posx

block.rect.y=posy

#screen.blit(block.image,block.rect)

block\_list.add(block)

all\_sprites\_list.add(block)

if i ==0:

iron\_block\_list.append(block)

#all\_block\_rect.append(block.rect)

rand= random.randint(0,7)

if (rand <3) and i!=0 :

power\_block\_list.append(block)

if rand==0:

block.color=WHITE #for larger ball

elif rand==1:

block.color=PINK2 #for longer bar

elif rand==2:

block.color=BLACK #for slower ball

posx+=50

posy+=28

all\_sprites\_list.add(bar)

def GameOver(message,Escape):

global highscore

global total\_score

if Escape:

score=total\_score

else:

score=highscore

basicfont=pygame.font.SysFont('Comic Sans MS',46)

scorefont=pygame.font.SysFont(None,50)

text=basicfont.render(message,True,WHITE,BLUE) # render(text,anti-aliasing,color,background)

textrect=text.get\_rect()

textrect.centerx=screen.get\_rect().centerx

textrect.centery=screen.get\_rect().centery-50

show\_score=scorefont.render(str(score),True,RED,GREEN)

score\_rect=show\_score.get\_rect()

score\_rect.centerx=screen.get\_rect().centerx

score\_rect.centery=screen.get\_rect().centery

intro=scorefont.render("Press Enter to Play Again",True,BLACK,BLUE)

introrect=intro.get\_rect()

introrect.centerx=screen.get\_rect().centerx

introrect.centery=screen.get\_rect().centery+60

screen.blit(text,textrect)

screen.blit(show\_score,score\_rect)

screen.blit(intro,introrect)

for event in pygame.event.get():

if event.type == pygame.KEYDOWN and event.key == pygame.K\_RETURN :

return True

def showscore(ball):

font=pygame.font.SysFont(None,40)

levelfont=font.render("LEVEL "+str(level),True,BLACK,BLUE)

lifefont=font.render("LIFE:"+str(ball.life),True,BLACK,BLUE)

scorefont=font.render("SCORE:"+str(total\_score),True,BLACK,BLUE)

level\_rect=levelfont.get\_rect()

life\_rect=lifefont.get\_rect()

score\_rect=scorefont.get\_rect()

level\_rect.centerx=screen.get\_rect().centerx

level\_rect.centery=screen.get\_rect().centery

score\_rect.centerx=screen.get\_rect().centerx+70

score\_rect.centery=screen.get\_rect().centery+40

life\_rect.centerx=screen.get\_rect().centerx-60

life\_rect.centery=screen.get\_rect().centery+40

screen.blit(levelfont,level\_rect)

screen.blit(scorefont,score\_rect)

screen.blit(lifefont,life\_rect)

pygame.init()

screen\_width=500

screen\_height=500

screen=pygame.display.set\_mode([screen\_width,screen\_height])

pygame.display.set\_caption("ARAKNOID")

clock=pygame.time.Clock()

level=1

run=True

fall\_list=[]

#is\_falling=bool()

#fall\_img=pygame.Surface([30,30])

#fall\_img.fill(RED)

#fall\_rect=fall\_img.get\_rect()

def NextStage(speedVar,bar\_len,ball\_rad,ball\_speed):

global emptystr

global hs # high score to be stored in file

global highscore

global show\_score

global Start

global win

global change\_speed

global changespeedinterval

global speedvar

global bar

global Level\_Score

global total\_score

global level

global run

global launched

global fall\_list

global counterval0,counter0

global counterval1,counter1

global counterval2,counter2

global ball

global specialcondition

bar\_length=bar\_len

global bar\_length

specialcondition=False

# global fall\_rect

speedvar=speedVar

#bar

bar=Bar(SILVER,bar\_len,20)

emptystr=str()

show\_score=False

Start=False if level==1 else True

win=False

#execute at the beginning level only

while not Start and level==1:

for event in pygame.event.get():

if event.type == pygame.QUIT :

pygame.quit()

exit()

break

screen.fill(BLUE)

introscreen()

pygame.display.flip()

block\_hit\_list=[]

ball=Ball(GREEN,ball\_rad,ball\_speed,speedvar)

ball.rect.x=screen\_width//2 + 20

ball.rect.y=screen\_height-45

all\_sprites\_list.add(ball)

player\_sprite.add(ball)

#creating blocks

Start\_Game(level):

launched=False

while Start:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

if event.type==pygame.KEYDOWN and event.key==pygame.K\_SPACE and not launched :

player\_sprite.update()

launched=True

#for quitting in middle of the running game

if event.type == pygame.KEYDOWN and event.key == pygame.K\_ESCAPE:

run=False

changeHighscore(total\_score,hs)

while True:

screen.fill(BLUE)

pygame.mouse.set\_visible(True)

if GameOver("You Escaped , Coward !!",True):

return

pygame.display.update()

if counter0:

if counterval0<=1000: counterval0+=1

elif counter1:

if counterval1<=1000: counterval1+=1

elif counter2:

if counterval2<=1000: counterval2+=1

keys=pygame.key.get\_pressed()

if keys[pygame.K\_LEFT] and bar.rect.x>0 :

if not launched :

ball.rect.x-=6 # to move the ball with respect to bar when it is not launched yet

bar.rect.x-=7

if keys[pygame.K\_RIGHT] and bar.rect.x<screen\_width-bar.length:

if not launched :

ball.rect.x+=6

bar.rect.x+=7

screen.fill(BLUE)

remove=bool()

iron=[block.rect for block in iron\_block\_list]

if ball.rect.collidelist(iron)+1 : # the rect1.collidelist(rect2list) will return the index of the total rectangles of rect2list that collide with rect1

for block in iron\_block\_list:

if ball.rect.colliderect(block.rect):

remove=False

block.image.fill(RED)

iron\_block\_list.remove(block)

block.color=RED

break

'''for block in power\_block\_list:

if ball.rect.colliderect(block.rect):

remove=False

block.image.fill(RED)

power\_block\_list.remove(block)

block.color=PINK2'''

else:

remove=True

block\_hit\_list=pygame.sprite.spritecollide(ball,block\_list,remove)

for block in block\_hit\_list:

if block.color !=GREY and len(fall\_list)<3:

block.rect.width-=35

block.rect.height-=5

fall\_list.append(block)

if specialcondition==True: block.color=PINK2

sprited\_block\_rect.append(block.rect)

total\_score+=1

Level\_Score+=1

if launched:

player\_sprite.update()

sprited\_block\_rect.clear()

all\_sprites\_list.draw(screen)

player\_sprite.draw(screen)

for block in fall\_list:

pygame.draw.rect(screen,block.color,block.rect)

showscore(ball)

clock.tick(60)

pygame.display.flip()

while not Start :

ball.image.fill(BLUE)

bar.image.fill(BLUE)

player\_sprite.remove(ball) # del ball

pygame.mouse.set\_visible(True)

screen.fill(BLUE)

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

highscore=total\_score

if win:

if gotonextlevel():

break

else:

run=False

changeHighscore(highscore,hs)

if GameOver('GAME OVER',False):

return

pygame.display.flip()

Level\_Score=0

#NextStage(ch\_s,ch\_in,bar\_len,ball\_rad,ball\_speed)

def gotonextlevel():

global level

if level != 5:

font=pygame.font.SysFont(None,28)

nextlev=font.render("Press ENTER to continue",True,BLACK,GREEN)

nextlev\_rect=nextlev.get\_rect()

nextlev\_rect.center=screen.get\_rect().center

for event in pygame.event.get():

if event.type == pygame.KEYDOWN and event.key == pygame.K\_RETURN :

level+=1

return True

screen.blit(nextlev,nextlev\_rect)

pygame.display.update()

else:

font=pygame.font.SysFont(None,28)

nextlev=font.render("You Won\n Score :"+str(total\_score),True,BLACK,GREEN)

nextlev\_rect=nextlev.get\_rect()

nextlev\_rect.center=screen.get\_rect().center

screen.blit(nextlev,nextlev\_rect)

return

def changeHighscore(highscore,hs):

if highscore>int(hs):

with open("HighScore.txt",'w') as f:

f.write(str(highscore))

reset()

def Next():

global run,level,lostlife,s

while run and not s.isEmpty() :

arg=s.pop()

if level>1:

lostlife-=1

NextStage(arg[0],arg[1],arg[2],arg[3])

while True:

if run :

Next()

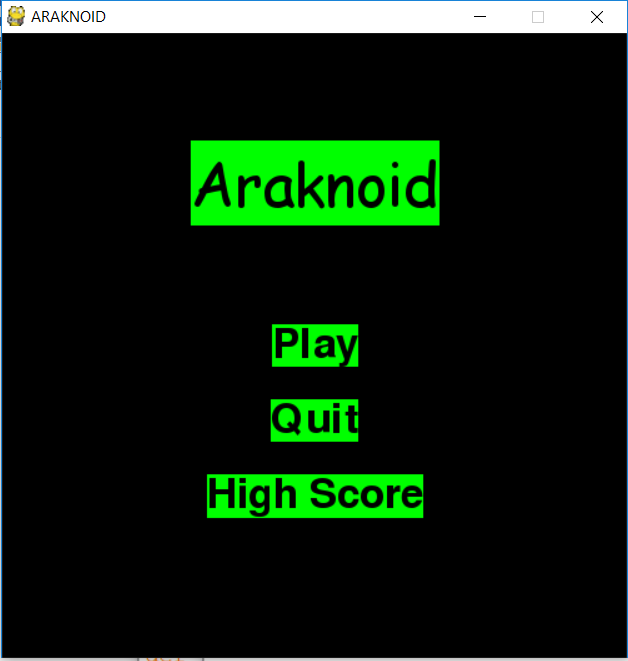
else:

reset()

**CHAPTER 6**

**SCREEN SHOTS**

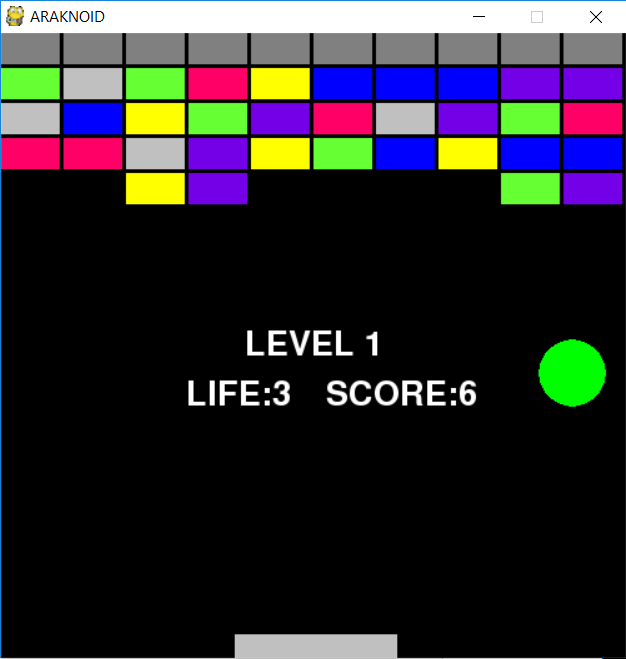
**INTRO SCREEEN**





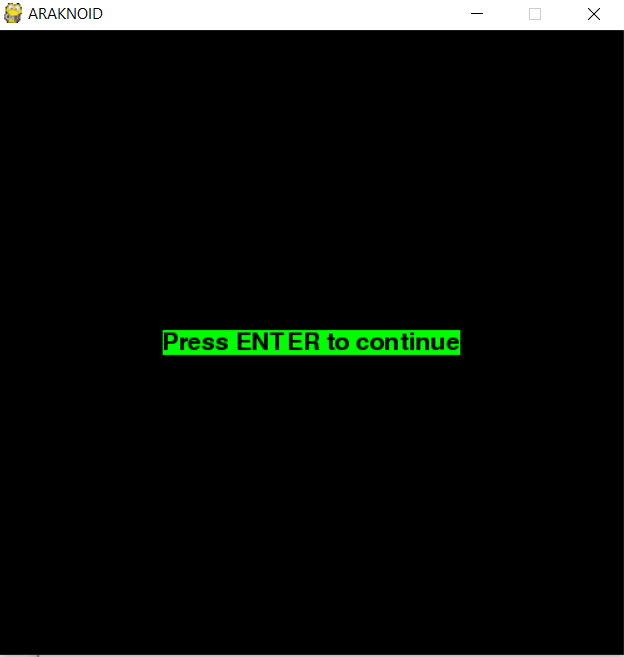
Game Starts

Big Ball Power



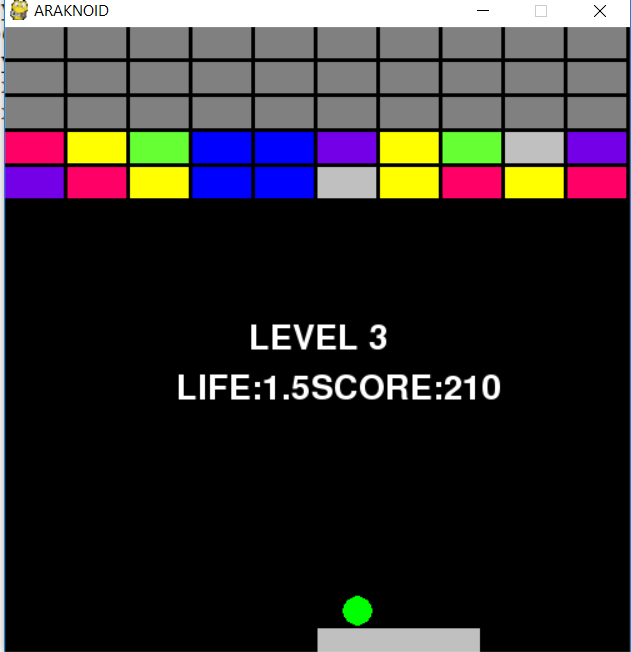


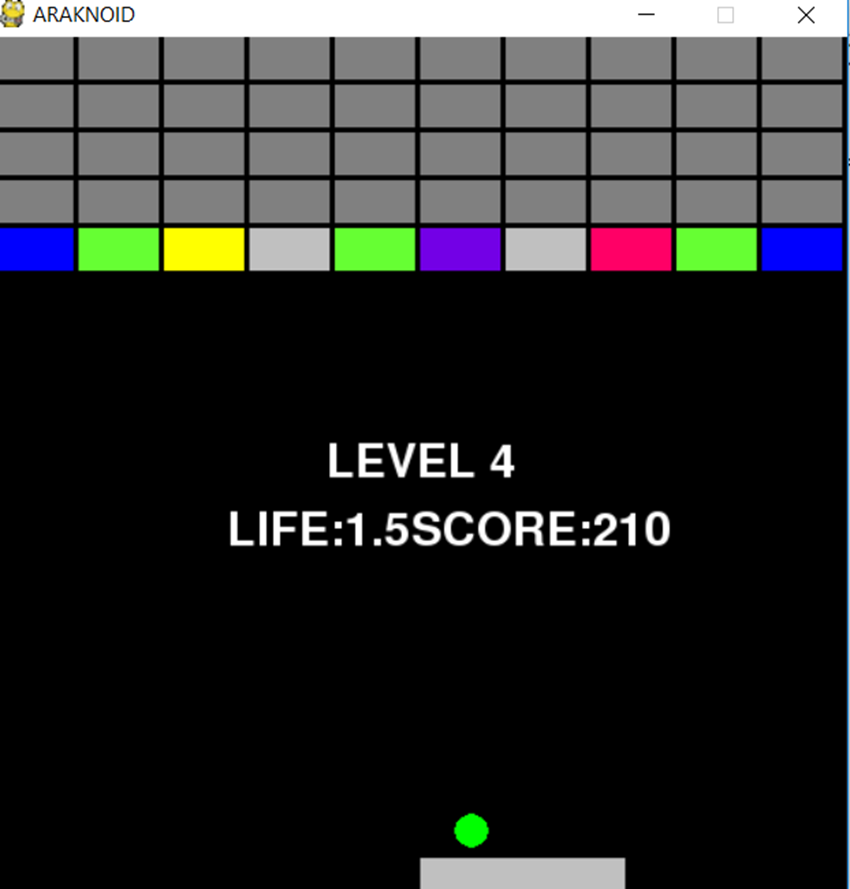
Long Bar Power

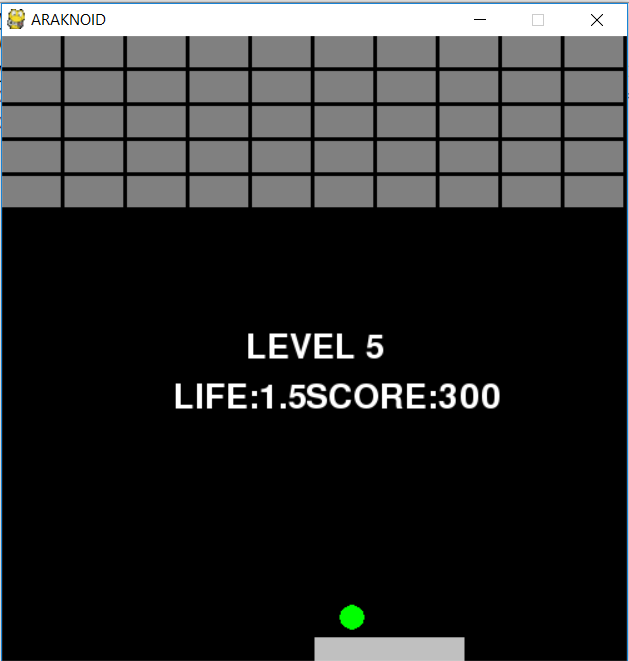


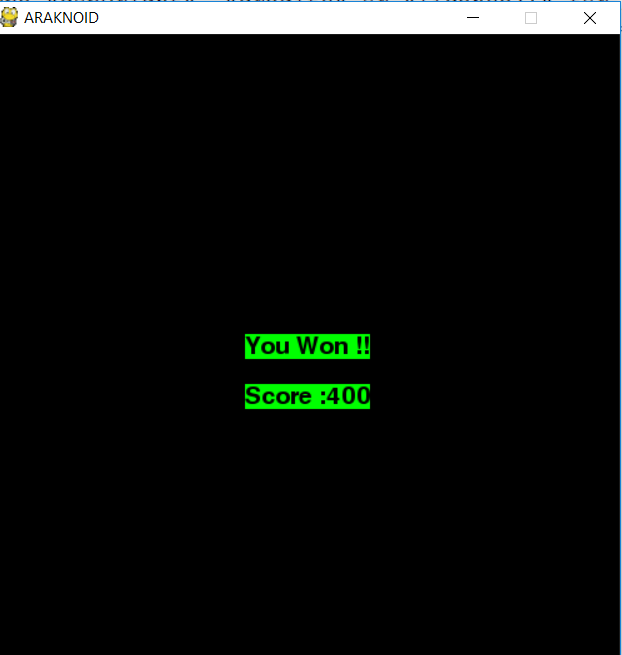
NEXT LEVEL





****





**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

We made this game with only limited functions and levels. In future we can upgrade the gameplay in a more user friendly manner. We will also make this game in 3 dimensional format as here we have did this game only with 2 dimensional manner. In future we are willing to make more powers and different arrangement of blocks in more colourfull way. We are also willing to add more user customizing attributes like selecting

the colour of the ball and bar.

**REFERENCe**

1. https://stackoverflow.com
2. https://programarcadegames.com
3. [http://pygametutorials.wikidot.com](http://pygametutorials.wikidot.com/)
4. <https://pygame.org>
5. <https://www.edureka.co/blog/pygame-tutorial>
6. <https://nerdparadise.com/programming/pygame/>