Project

CHARMANDER

C - Cricket

H - Head Ball

A - And

R - Ricochet,

M - Multiprogramming

A - And Analysing

N - New,

D - Diversified

E - Entertaining,

R - Realistic Games

By :

Vijai Aravindh.R

V.S. Arnav Ajay Krishna

K.V. Badhrinarayanan

Contents

* Acknowledgement
* Synopsis
* Salient Features
* Project Analysis
* Project Listing
* Output

Acknowledgement

We sincerely thank our Computer Science teachers Ms. Harifa Begum and Ms. Nirmala Devi in helping and guiding us to make this program. Their expert inputs and contributions have helped us in developing our game immensely.

Synopsis

Project CHARMANDER contains 3 games:

* An interactive game based on cricket : “Chennai’28”
* A challenging physics-based game :

“Ricochet”

* A game that offers a twist on football :

“Head Ball”

Chennai’28 is a single player game based on cricket, whose core aspect focuses on sheer timing of the ball, where the player is the batsman. The aim of the player is to chase down the target in a given number of balls. It has 3 difficulty levels, and the target relatively scales up with only 3 wickets per inning.

Ricochet is a game that tests your physics intuition. The aim of the game is to use a cannon to shoot a ball into a box, while taking care to ‘ricochet’ your ball perfectly to land in the box. You have 3 tries and a time limit, so make sure the ball lands on time!

Head Ball is a 1v1 game based on football but with new mechanics. The program is all about correctly timing and placing the ball into the opponent’s court, with witty player movements. The first player to score 5 goals wins the game.

Salient Features

Chennai’28: It is a single player game based on regular kinetics of cricket, where the user gets to play as a batsman. It primarily focuses on perfecting the timing of the ball. Being an interactive game, there are a number of shots which can be played, with respect to the timing. Accordingly, the user gets rewarded with a four, six, or two, and a poorly timed shot may result in a wicket. As the difficulty level rises, the target exponentially increases, with only a few extra overs and the same number of wickets in hand.

Ricochet: It is a physics-based game where the goal is to shoot a ball into a box by bouncing it off the walls within the time limit. The special feature of this game is that different coloured walls will bounce the ball differently – red will bounce more, yellow will bounce less. The ball can be shot 3 times and each successive ball gets faster to allow you to win within the time limit.

Head Ball: This is a game programmed such that 2 people can face each other and the first to score 5 goals wins. The game screen consists of 2 players controlled by WASD and arrow keys and a ball. The main objective of the game is to score goals using the enhanced mechanics .The game allows a player to hold the ball if the player is moving at the time he collides with the ball and moving away from the ball releases it.

Project Analysis

Project CHARMANDER has been programmed using the Python language primarily with the use of pygame module. Some Python concepts that have been used in Ball are:

* Functions
* Loops
* Arrays/Lists/Tuples
* Classes
* Reference variables
* Default parameters
* Libraries/Modules

In addition to these, many built-in functions, methods and graphics commands have been used in the program to help

The following functions have been used in the Cricket’28 program :

1. bowler() – draws the image of the bowler on the display screen.
2. batsman() - draws the image of the batsman on the display screen.
3. bowling() – sets a variable to the ball when it is in motion after it is bowled by the bowler, and draws the ball progressively as it moves to the batsman.
4. fieldercollision1() – checks for collision between the ball and a fielder, and returns True if the collision is true. This accounts for a wicket in the game.
5. fieldercollision2() – checks for collision between the ball and a fielder, and returns True if the collision is true. This accounts for a wicket in the game.

This function has been used in the Head Ball program:

1. checkcollision() – a user defined function which takes the position of the player and the ball and returns a measure of the distance between them in order to detect collision. It is used to check the collision between a player and a ball as well as to include sound effects.

The following functions have been used in the Ricochet program :

1. screen.get\_at((x,y)) – inputs x & y as a tuple and checks and returns the colour of the pixel at the co-ordinates (x,y).

Project Listing

Chennai’28

import pygame

import math

import random

import sys

from pygame.locals import \*

restart = True

while restart:

pygame.init()

#window of the game

screen=pygame.display.set\_mode((800,600))

gamedisplay=pygame.display.set\_mode((800,600))

#Title

pygame.display.set\_caption("Chennai'28")

#bowler

bowlerimg=pygame.image.load('bowlercs2.png')

bowlerX=375

bowlerY=480

#batsman

batsmanimg=pygame.image.load('batsmanimg.png')

batsmanimg1=pygame.image.load('batsmanimg1.png')

batsmanimg2=pygame.image.load('batsmanimg2.png')

batsmanimg3=pygame.image.load('batsmanimg3.png')

batsmanimg4=pygame.image.load('batsmanimg4.png')

batsmanimg5=pygame.image.load('batsmanimg5.png')

batsmanimg6=pygame.image.load('batsmanimg6.png')

batsmanX=360

batsmanY=100

#ball1

ballimg=pygame.image.load('ballfinal-2.png')

ballX=370

ballY=480

ballstate='ready'

#ballYchange=10

#fonts

font=pygame.font.Font('freesansbold.ttf',32)

font1=pygame.font.Font('Cooper Black Regular.ttf',32)

font2=pygame.font.Font('Impact.ttf',32)

font3=pygame.font.Font('Minecrafter.Reg.ttf',30)

font4=pygame.font.Font('Minecrafter.Reg.ttf',22)

font5=pygame.font.Font('Impact.ttf',38)

font6=pygame.font.Font('freesansbold.ttf',24)

font7=pygame.font.Font('Minecrafter.Reg.ttf',30)

#count

score=0

wicket=0

ballcount=0

#stumps

stumpimg=pygame.image.load('stumps1.png')

#fieldersimages

fielder1img=pygame.image.load('fielder1.png')

fielder2img=pygame.image.load('fielder2.png')

fielder3img=pygame.image.load('fielder3.png')

#fielders

fielder1X=692

fielder1Y=480

fielder2X=54

fielder2Y=480

fielder3X=160

fielder3Y=260

#background

bg2=pygame.image.load('cricketbg.png')

bg1=pygame.image.load('cricketbg1.png')

gameoverlost=pygame.image.load('gameoverlost.png')

gameoverwon=pygame.image.load('gameoverwon.png')

#menu

menu=True

clock=pygame.time.Clock()

#boundarymessage

fourleft1=font2.render("The ball races",True,(255,255,255))

fourleft2=font2.render("to the boundary!",True,(255,255,255))

fourright1=font2.render("That's a lovely stroke",

True,(255,255,255))

fourright2=font2.render("off the middle of the

bat!",True,(255,255,255))

sixleft1=font2.render("Oh he's put it away!",True,(255,255,255))

sixleft2=font2.render("Yes its into the

crowd!",True,(255,255,255))

sixright1=font2.render("That is a biggie,",True,(255,255,255))

sixright2=font2.render("its out of here!",True,(255,255,255))

twoleft1=font2.render("Lovely flick of the

wrist.",True,(255,255,255))

twoleft2=font2.render("Gets two runs.",True,(255,255,255))

tworight1=font2.render("Guides it into the

gap.",True,(255,255,255))

tworight2=font2.render("Two runs.",True,(255,255,255))

#wicketmessage

bowled1=font2.render('What a ball! Chopped',True,(255,255,255))

bowled2=font2.render('it off the legstump.',True,(255,255,255))

caughtleft1=font2.render('The fielder at long

on',True,(255,255,255))

caughtleft2=font2.render('takes a blinder!',True,(255,255,255))

caughtright1=font2.render('What a catch!

Great',True,(255,255,255))

caughtright2=font2.render('effort from the

fielder.',True,(255,255,255))

#gameovermessage

gameovermsg=font3.render('Press Escape to exit the

Game',True,(255,255,255))

gamerestartmsg=font7.render('Press Enter to play the game

again',True,(255,255,255))

#music

pygame.mixer.init()

pygame.mixer.music.load("cannon rock.wav")

pygame.mixer.music.set\_volume(0.7)

pygame.mixer.music.play(-1)

def bowler(x,y):

screen.blit(bowlerimg,(x,y))

def batsman(x,y):

screen.blit(batsmanimg,(x,y))

def bowling(x,y):

global ballstate

ballstate='fire'

screen.blit(ballimg,(x+10,y+16))

def fieldercollision1(fielder1X,fielder1Y,ballX,ballY):

distance= math.sqrt((math.pow(fielder1X-

ballX,2))+(math.pow(fielder1Y-ballY,2)))

if distance<=20:

return True

else:

return False

def fieldercollision2(fielder2X,fielder2Y,ballX,ballY):

distance= math.sqrt((math.pow(fielder2X-

ballX,2))+(math.pow(fielder2Y-ballY,2)))

if distance<=20:

return True

else:

return False

#mainloop

run=True

while run==True:

while menu==True:

keys=pygame.key.get\_pressed()

screen.blit(bg1,(0,0))

pygame.display.update()

for event in pygame.event.get():

if keys[pygame.K\_e]:

mode='easy'

wicketcount=4

i=random.randint(18,24)

balls=12

screen.blit(bg2,(0,0))

target=font.render("You need to score:"+str(i)+" "+ "runs in" +" "+str(balls)+" "+ "balls",True,(255,255,255))

screen.blit(target,(120,250))

displaymessage=font.render("Press SPACE to bowl

the first ball" + " ",True,(0,255,0))

screen.blit(displaymessage,(140,520))

wicketsinhand=font.render("Wickets in

hand:"+"4"+" ",True,(255,255,255))

screen.blit(wicketsinhand,(120,280))

pygame.display.update()

menu=False

elif keys[pygame.K\_m]:

mode='medium'

wicketcount=4

i=random.randint(40,60)

balls=18

screen.blit(bg2,(0,0))

target=font.render("You need to score:"+str(i)+" "+ "runs in" +" "+str(balls)+" "+ "balls",True,(255,255,255))

screen.blit(target,(120,250))

displaymessage=font.render("Press SPACE to bowl

the first ball" + " ",True,(0,255,0))

screen.blit(displaymessage,(140,520))

wicketsinhand=font.render("Wickets in

hand:"+"4"+" ",True,(255,255,255))

screen.blit(wicketsinhand,(120,280))

pygame.display.update()

menu=False

elif keys[pygame.K\_h]:

wicketcount=4

mode='hard'

i=random.randint(80,100)

balls=24

screen.blit(bg2,(0,0))

target=font.render("You need to score:"+str(i)+" "+ "runs in" +" "+str(balls)+" "+ "balls",True,(255,255,255))

screen.blit(target,(120,250))

displaymessage=font.render("Press SPACE to bowl

the first ball" + " ",True,(0,255,0))

screen.blit(displaymessage,(140,520))

wicketsinhand=font.render("Wickets in

hand:"+"4"+" ",True,(255,255,255))

screen.blit(wicketsinhand,(120,280))

pygame.display.update()

menu=False

elif event.type==QUIT:

menu=False

run=False

pygame.quit()

exit()

screen.fill((34,139,34))

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

pygame.draw.line(screen,(0,0,0,),(320,450),(480,450),1)

screen.blit(stumpimg,(380,55))

screen.blit(stumpimg,(380,445))

screen.blit(fielder1img,(fielder1X,fielder1Y))

screen.blit(fielder2img,(fielder2X,fielder2Y))

screen.blit(fielder3img,(fielder3X,fielder3Y))

for event in pygame.event.get():

if event.type==QUIT:

run=False

pygame.quit()

exit()

keys=pygame.key.get\_pressed()

if keys[pygame.K\_SPACE]:

if ballstate is 'ready':

ballYchange=random.randint(6,13)

ballX=bowlerX

bowling(ballX,ballY)

bowler(bowlerX,bowlerY)

batsman(batsmanX,batsmanY)

if ballY <= batsmanY-60:

wicket=wicket+1

ballcount=ballcount+1

SCORE=font.render("Score :"+str(score)+"/"

+str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

screen.blit(bowled1,(20,100))

screen.blit(bowled2,(500,100))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:"

+str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

screen.blit(ballimg,(385,58))

pygame.display.update()

ballY = 480

ballstate = "ready"

if ballstate is 'fire':

bowling(ballX,ballY)

ballY=ballY-ballYchange

if ballY>batsmanY+10 and ballY<batsmanY+25:

if keys[pygame.K\_LEFT]:

while ballX!=0 and ballY!=600:

ballX=ballX-5

ballY=ballY+2

screen.blit(ballimg,(ballX+12,ballY+2))

if ballX>3:

pygame.draw.circle(screen,(34,139,34),(ballX+6,ballY+18),25)

screen.blit(ballimg,(ballX,ballY-3))

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

pygame.draw.circle(screen,(34,139,34),(10,268),12)

pygame.draw.circle(screen,(34,139,34),(388,168),36)

pygame.draw.circle(screen,(34,139,34),(batsmanX+25,batsmanY),32)

screen.blit(stumpimg,(380,55))

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

screen.blit(batsmanimg4,(batsmanX,batsmanY))

clock.tick(60)

pygame.display.update()

elif keys[pygame.K\_RIGHT]:

while ballX<775 and ballY!=600:

ballX=ballX+5

ballY=ballY+2

screen.blit(ballimg,(ballX+12,ballY+2))

pygame.draw.circle(screen,(34,139,34),(ballX+2,ballY),13)

pygame.draw.circle(screen,(34,139,34),(778,294),9)

pygame.draw.circle(screen,(34,139,34),(388,168),36)

pygame.draw.circle(screen,(34,139,34),(batsmanX+25,batsmanY),32)

screen.blit(stumpimg,(380,55))

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

screen.blit(batsmanimg3,(batsmanX,batsmanY))

clock.tick(60)

pygame.display.update()

if ballX==0 or ballY==600:

screen.blit(fourleft1,(100,100))

screen.blit(fourleft2,(500,100))

score=score+4

ballcount=ballcount+1

SCORE=font.render("Score :"+str(score)+"/"

+str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:"

+str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

ballstate='ready'

ballY=480

ballX=370

elif ballX>770 or ballY==600:

screen.blit(fourright1,(40,100))

screen.blit(fourright2,(480,100))

score=score+4

ballcount=ballcount+1

SCORE=font.render("Score :"+str(score)+"/"

+str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render ("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:" +str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

ballstate='ready'

ballY=480

ballX=370

elif ballY>batsmanY+25 and ballY<batsmanY+40:

if keys[pygame.K\_LEFT]:

while ballX>=0 and ballY!=600:

ballX=ballX-6

ballY=ballY+2

screen.blit(ballimg,(ballX+4,ballY+3))

if ballX>=3:

pygame.draw.circle(screen,(34,139,34),(ballX,ballY+5),23)

screen.blit(ballimg,(ballX,ballY))

pygame.draw.circle(screen,(34,139,34),(9,ballY+1),14)

pygame.draw.circle(screen,(34,139,34),(388,168),36)

pygame.draw.circle(screen,(34,139,34),(batsmanX+25,batsmanY),32)

screen.blit(stumpimg,(380,55))

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

screen.blit(batsmanimg6,(batsmanX,batsmanY))

clock.tick(60)

pygame.display.update()

elif keys[pygame.K\_RIGHT]:

while ballX<770 and ballY!=600:

ballX=ballX+6

ballY=ballY+2

screen.blit(ballimg,(ballX+12,ballY+2))

pygame.draw.circle(screen,(34,139,34),(ballX+1,ballY+4),15)

pygame.draw.circle(screen,(34,139,34),(774,263),5)

pygame.draw.rect(screen,(34,139,34),(775,269,10,10))

pygame.draw.circle(screen,(34,139,34),(388,168),36)

pygame.draw.circle(screen,(34,139,34),(batsmanX+25,batsmanY),32)

screen.blit(stumpimg,(380,55))

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

screen.blit(batsmanimg5,(batsmanX,batsmanY))

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

clock.tick(60)

pygame.display.update()

if ballY==600 or ballX<=0 :

screen.blit(sixleft1,(50,100))

screen.blit(sixleft2,(500,100))

score=score+6

ballcount=ballcount+1

SCORE=font.render("Score :"+str(score)+"/"

+str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:"

+str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

ballstate='ready'

ballY=480

ballX=370

elif ballY==600 or ballX>765:

screen.blit(sixright1,(100,100))

screen.blit(sixright2,(500,100))

score=score+6

ballcount=ballcount+1

SCORE=font.render("Score :"+str(score)+"/"

+str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:"

+str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

ballstate='ready'

ballY=480

ballX=370

elif ballY>batsmanY+40 and ballY<batsmanY+55:

if keys[pygame.K\_LEFT]:

while ballX!=0 and ballY!=600:

ballX=ballX-3

ballY=ballY+1

screen.blit(ballimg,(ballX+4,ballY+3))

pygame.draw.circle(screen,(34,139,34),(ballX-3,ballY+10),23)

screen.blit(ballimg,(ballX,ballY))

pygame.draw.circle(screen,(34,139,34),(388,168),34)

pygame.draw.circle(screen,(34,139,34),(batsmanX+25,batsmanY),32)

screen.blit(stumpimg,(380,55))

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

screen.blit(batsmanimg2,(batsmanX,batsmanY))

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

clock.tick(60)

pygame.display.update()

elif keys[pygame.K\_RIGHT]:

while ballX<775 and ballY!=600:

ballX=ballX+2.5

ballY=ballY+1

screen.blit(ballimg,(ballX+12,ballY+2))

pygame.draw.circle(screen,(34,139,34),(ballX-1,ballY+4),30)

screen.blit(ballimg,(385,180))

pygame.draw.circle(screen,(34,139,34),(388,168),37)

screen.blit(ballimg,(ballX,ballY))

pygame.draw.circle(screen,(34,139,34),(batsmanX+25,batsmanY),32)

screen.blit(stumpimg,(380,55))

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

screen.blit(batsmanimg3,(batsmanX,batsmanY))

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

clock.tick(60)

pygame.display.update()

if ballX>770 or ballY==600:

screen.blit(tworight1,(30,100))

screen.blit(tworight2,(500,100))

score=score+2

ballcount=ballcount+1

SCORE=font.render("Score :"+str(score)+"/"

+str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render ("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:"

+str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

ballstate='ready'

ballY=480

ballX=370

elif ballX==0 or ballY==600:

screen.blit(twoleft1,(10,100))

screen.blit(twoleft2,(500,100))

score=score+2

ballcount=ballcount+1

SCORE=font.render("Score :"+str(score)+"/"

+str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:"

+str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

ballstate='ready'

ballY=480

ballX=370

elif ballY>batsmanY+55 and ballY<batsmanY+70:

if keys[pygame.K\_LEFT]:

flag=True

while fieldercollision2(fielder2X,fielder2Y,ballX,ballY) is False and flag==True:

ballX=ballX-2.5

ballY=ballY+2.5

screen.blit(ballimg,(ballX+4,ballY+3))

pygame.draw.circle(screen,(34,139,34),(ballX+19,ballY+5),16.5)

pygame.draw.circle(screen,(34,139,34),(388,168),39)

pygame.draw.circle(screen,(34,139,34),(batsmanX+25,batsmanY),36)

screen.blit(stumpimg,(380,55))

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

screen.blit(batsmanimg1,(batsmanX,batsmanY))

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

screen.blit(ballimg,(ballX,ballY))

clock.tick(60)

pygame.display.update()

if fieldercollision2(fielder2X,fielder2Y,ballX,ballY) is True:

wicket=wicket+1

ballcount=ballcount+1

screen.blit(caughtleft1,(35,100))

screen.blit(caughtleft2,(500,100))

SCORE=font.render("Score :"+str(score)+"/" +str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:"

+str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

ballstate='ready'

ballY=480

ballX=370

flag=False

elif keys[pygame.K\_RIGHT]:

opt=True

while fieldercollision1(fielder1X,fielder1Y,ballX,ballY) is False and opt==True:

ballX=ballX+2.5

ballY=ballY+2.5

screen.blit(ballimg,(ballX+12,ballY+2))

pygame.draw.circle(screen,(34,139,34),(ballX+7,ballY-3),29)

pygame.draw.circle(screen,(34,139,34),(388,168),37)

screen.blit(ballimg,(ballX,ballY))

pygame.draw.circle(screen,(34,139,34),(batsmanX+25,batsmanY),32)

screen.blit(stumpimg,(380,55))

pygame.draw.line(screen,(0,0,0),(320,100),(480,100),1)

screen.blit(batsmanimg1,(batsmanX+15,batsmanY))

pygame.draw.rect(screen,(0,0,0),[320,70,160,414],1)

clock.tick(60)

pygame.display.update()

if fieldercollision1(fielder1X,fielder1Y,ballX,ballY) is True:

wicket=wicket+1

ballcount=ballcount+1

screen.blit(caughtright1,(40,100))

screen.blit(caughtright2,(500,100))

SCORE=font.render("Score :"+str(score)+"/" +str(wicket)+" ",True,(255,255,255))

screen.blit(SCORE,(505,50))

required=i-score

ballsleft=balls-ballcount

REQUIRED=font6.render("Required:"+str(required),True,(255,255,255))

BALLSLEFT=font6.render("Balls left:"

+str(ballsleft),True,(255,255,255))

screen.blit(REQUIRED,(530,220))

screen.blit(BALLSLEFT,(530,260))

ballstate='ready'

ballY=480

ballX=370

opt=False

if mode is 'easy':

if score>i:

run=False

j=False

elif ballcount==balls:

run=False

j=False

elif wicket==wicketcount:

run=False

j=False

clock.tick(60)

pygame.display.update()

elif mode is 'medium':

if score>i:

run=False

j=False

elif ballcount==balls:

run=False

j=False

elif wicket==wicketcount:

run=False

j=False

clock.tick(60)

pygame.display.update()

elif mode is 'hard':

if score>i:

run=False

j=False

elif ballcount==balls:

run=False

j=False

elif wicket==wicketcount:

run=False

j=False

clock.tick(60)

pygame.display.update()

gamedisplay.fill((0,0,0))

postrun=True

while postrun==True:

if score>i:

gamedisplay.blit(gameoverwon,(0,0))

gamedisplay.blit(gameovermsg,(70,520))

gamedisplay.blit(gamerestartmsg,(70,450))

gamewon=font5.render("Congrats!You Won the

game!",True,(255,255,255))

gamedisplay.blit(gamewon,(80,50))

pygame.display.update()

elif wicket==wicketcount or ballcount==balls:

gamedisplay.blit(gameoverlost,(0,0))

gamedisplay.blit(gameovermsg,(70,520))

gamedisplay.blit(gamerestartmsg,(70,450))

gamelost=font4.render("You lost the game.Better luck

next time!",True,(255,255,255))

fullstop=font2.render(".",True,(255,255,255))

exclamation=font2.render("!",True,(255,255,255))

gamedisplay.blit(gamelost,(80,50))

gamedisplay.blit(fullstop,(340,30))

gamedisplay.blit(exclamation,(660,34))

pygame.display.update()

keys=pygame.key.get\_pressed()

for event in pygame.event.get():

if keys[pygame.K\_RETURN]:

restart = True

postrun=False

elif keys[pygame.K\_ESCAPE]:

restart = False

postrun = False

pygame.quit()

exit()

elif event.type==QUIT:

restart=False

postrun=False

pygame.quit()

exit()

Ricochet

# 1 - Import library

import pygame

from pygame.locals import \*

import math

import random

import time

# 2 - initialize the game

pygame.init()

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

width, height = 1280, 960

clock=pygame.time.Clock()

screen=pygame.display.set\_mode((width, height))

keys = [False, False]

playerpos=[333,855]

cannonpos=[300,834]

bullets=[]

p=0;end=0

gravity=0.01

font = pygame.font.Font('freesansbold.ttf', 30)

white=(255, 255, 255, 255)

red=(255, 0, 0, 255)

yellow=(230, 214, 37, 255)

green=(0, 255, 60, 255)

black=(7, 7, 8, 255)

class Ball(pygame.sprite.Sprite):

def \_\_init\_\_(self,x,y,velx,vely,screen):

pygame.sprite.Sprite.\_\_init\_\_(self)

self.image=pygame.image.load("resource/images/Ball.png")

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

self.vel\_x=velx

self.vel\_y=vely

self.x=x

self.y=y

self.maxsp=20

def update(self):

self.x+=velx

self.y+=vely

def draw(self):

screen.blit(Ballimg,(self.x,self.y))

pygame.display.update()

# 3 - Load images

player = pygame.image.load("resource/images/Cannon top.png")

bottom = pygame.image.load("resource/images/Cannon bottom.png")

Ballimg = pygame.image.load("resource/images/Ball.png")

gameover = pygame.image.load("resource/images/gameover.png")

youwin = pygame.image.load("resource/images/youwin.png")

mainmenu = pygame.image.load("resource/images/mainmenu.png")

# 4 - keep looping through

running=1;loss=1;menu=1;title=1

while running:

while title:

screen.blit(mainmenu,(0,0))

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key==K\_1:

bg = pygame.image.load("resource/images/Level1.png")

start\_time=pygame.time.get\_ticks()

title=0

if event.key==K\_2:

bg = pygame.image.load("resource/images/Level2.png")

start\_time=pygame.time.get\_ticks()

title=0

if event.key==K\_3:

bg = pygame.image.load("resource/images/Level3.png")

start\_time=pygame.time.get\_ticks()

title=0

if event.key==K\_q:

pygame.exit()

# 5 - clear the screen before drawing it again

screen.fill(0)

# 6 - draw the screen elements

screen.blit(bg,(0,0))

# 6.1 - Set player position and rotation

position = pygame.mouse.get\_pos()

angle = math.atan2(position[1]-(playerpos[1]+32),position[0]-

(playerpos[0]+26))

playerrot = pygame.transform.rotate(player, 360-angle\*57.29)

playerpos1 = (playerpos[0]-playerrot.get\_rect().width/2,

playerpos[1]-playerrot.get\_rect().height/2)

screen.blit(playerrot, playerpos1)

screen.blit(bottom, cannonpos)

# 6.2 - Ball drawing code

for bullet in bullets:

ball.vel\_y=ball.vel\_y+gravity

ball.x+=ball.vel\_x

ball.y+=ball.vel\_y

if ball.x<1250 and ball.x>50 and ball.y<1000 and ball.y>50:

if screen.get\_at((int(ball.x), int(ball.y+6)))==red or screen.get\_at((int(ball.x), int(ball.y+7)))==red or screen.get\_at((int(ball.x), int(ball.y+8)))==red or screen.get\_at((int(ball.x), int(ball.y+9)))==red or screen.get\_at((int(ball.x), int(ball.y+10)))==red or screen.get\_at((int(ball.x), int(ball.y+11)))==red or screen.get\_at((int(ball.x), int(ball.y+12)))==red or screen.get\_at((int(ball.x+18), int(ball.y+6)))==red or screen.get\_at((int(ball.x+18), int(ball.y+7)))==red or screen.get\_at((int(ball.x+18), int(ball.y+8)))==red or screen.get\_at((int(ball.x+18), int(ball.y+9)))==red or screen.get\_at((int(ball.x+18), int(ball.y+10)))==red or screen.get\_at((int(ball.x+18), int(ball.y+11)))==red or screen.get\_at((int(ball.x+18), int(ball.y+12)))==red:

ball.vel\_x=-ball.vel\_x\*1.5

elif screen.get\_at((int(ball.x+6), int(ball.y)))==red or screen.get\_at((int(ball.x+7), int(ball.y)))==red or screen.get\_at((int(ball.x+8), int(ball.y)))==red or screen.get\_at((int(ball.x+9), int(ball.y)))==red or screen.get\_at((int(ball.x+10), int(ball.y)))==red or screen.get\_at((int(ball.x+11), int(ball.y)))==red or screen.get\_at((int(ball.x+12), int(ball.y)))==red or screen.get\_at((int(ball.x+6), int(ball.y+18)))==red or screen.get\_at((int(ball.x+7), int(ball.y+18)))==red or screen.get\_at((int(ball.x+8), int(ball.y+18)))==red or screen.get\_at((int(ball.x+9), int(ball.y+18)))==red or screen.get\_at((int(ball.x+10), int(ball.y+18)))==red or screen.get\_at((int(ball.x+11), int(ball.y+18)))==red or screen.get\_at((int(ball.x+12), int(ball.y+18)))==red:

ball.vel\_y=-ball.vel\_y\*1.5

elif screen.get\_at((int(ball.x+5), int(ball.y+1)))==red or screen.get\_at((int(ball.x+4), int(ball.y+2)))==red or screen.get\_at((int(ball.x+3), int(ball.y+3)))==red or screen.get\_at((int(ball.x+2), int(ball.y+4)))==red or screen.get\_at((int(ball.x+1), int(ball.y+5)))==red or screen.get\_at((int(ball.x+13), int(ball.y+1)))==red or screen.get\_at((int(ball.x+14), int(ball.y+2)))==red or screen.get\_at((int(ball.x+15), int(ball.y+3)))==red or screen.get\_at((int(ball.x+16), int(ball.y+4)))==red or screen.get\_at((int(ball.x+17), int(ball.y+5)))==red or screen.get\_at((int(ball.x+13), int(ball.y+17)))==red or screen.get\_at((int(ball.x+14), int(ball.y+16)))==red or screen.get\_at((int(ball.x+15), int(ball.y+15)))==red or screen.get\_at((int(ball.x+16), int(ball.y+14)))==red or screen.get\_at((int(ball.x+17), int(ball.y+13)))==red or screen.get\_at((int(ball.x+5), int(ball.y+17)))==red or screen.get\_at((int(ball.x+4), int(ball.y+16)))==red or screen.get\_at((int(ball.x+3), int(ball.y+15)))==red or screen.get\_at((int(ball.x+2), int(ball.y+14)))==red or screen.get\_at((int(ball.x+1), int(ball.y+13)))==red:

ball.vel\_x=-ball.vel\_x\*1.5

ball.vel\_y=-ball.vel\_y\*1.5

elif screen.get\_at((int(ball.x), int(ball.y+6)))==white or screen.get\_at((int(ball.x), int(ball.y+7)))==white or screen.get\_at((int(ball.x), int(ball.y+8)))==white or screen.get\_at((int(ball.x), int(ball.y+9)))==white or screen.get\_at((int(ball.x), int(ball.y+10)))==white or screen.get\_at((int(ball.x), int(ball.y+11)))==white or screen.get\_at((int(ball.x), int(ball.y+12)))==white or screen.get\_at((int(ball.x+18), int(ball.y+6)))==white or screen.get\_at((int(ball.x+18), int(ball.y+7)))==white or screen.get\_at((int(ball.x+18), int(ball.y+8)))==white or screen.get\_at((int(ball.x+18), int(ball.y+9)))==white or screen.get\_at((int(ball.x+18), int(ball.y+10)))==white or screen.get\_at((int(ball.x+18), int(ball.y+11)))==white or screen.get\_at((int(ball.x+18), int(ball.y+12)))==white:

ball.vel\_x=-ball.vel\_x

elif screen.get\_at((int(ball.x+6), int(ball.y)))==white or screen.get\_at((int(ball.x+7), int(ball.y)))==white or screen.get\_at((int(ball.x+8), int(ball.y)))==white or screen.get\_at((int(ball.x+9), int(ball.y)))==white or screen.get\_at((int(ball.x+10), int(ball.y)))==white or screen.get\_at((int(ball.x+11), int(ball.y)))==white or screen.get\_at((int(ball.x+12), int(ball.y)))==white or screen.get\_at((int(ball.x+6), int(ball.y+18)))==white or screen.get\_at((int(ball.x+7), int(ball.y+18)))==white or screen.get\_at((int(ball.x+8), int(ball.y+18)))==white or screen.get\_at((int(ball.x+9), int(ball.y+18)))==white or screen.get\_at((int(ball.x+10), int(ball.y+18)))==white or screen.get\_at((int(ball.x+11), int(ball.y+18)))==white or screen.get\_at((int(ball.x+12), int(ball.y+18)))==white:

ball.vel\_y=-ball.vel\_y

elif screen.get\_at((int(ball.x+5), int(ball.y+1)))==white or screen.get\_at((int(ball.x+4), int(ball.y+2)))==white or screen.get\_at((int(ball.x+3), int(ball.y+3)))==white or screen.get\_at((int(ball.x+2), int(ball.y+4)))==white or screen.get\_at((int(ball.x+1), int(ball.y+5)))==white or screen.get\_at((int(ball.x+13), int(ball.y+1)))==white or screen.get\_at((int(ball.x+14), int(ball.y+2)))==white or screen.get\_at((int(ball.x+15), int(ball.y+3)))==white or screen.get\_at((int(ball.x+16), int(ball.y+4)))==white or screen.get\_at((int(ball.x+17), int(ball.y+5)))==white or screen.get\_at((int(ball.x+13), int(ball.y+17)))==white or screen.get\_at((int(ball.x+14), int(ball.y+16)))==white or screen.get\_at((int(ball.x+15), int(ball.y+15)))==white or screen.get\_at((int(ball.x+16), int(ball.y+14)))==white or screen.get\_at((int(ball.x+17), int(ball.y+13)))==white or screen.get\_at((int(ball.x+5), int(ball.y+17)))==white or screen.get\_at((int(ball.x+4), int(ball.y+16)))==white or screen.get\_at((int(ball.x+3), int(ball.y+15)))==white or screen.get\_at((int(ball.x+2), int(ball.y+14)))==white or screen.get\_at((int(ball.x+1), int(ball.y+13)))==white:

ball.vel\_x=-ball.vel\_x

ball.vel\_y=-ball.vel\_y

elif screen.get\_at((int(ball.x), int(ball.y+6)))==yellow or screen.get\_at((int(ball.x), int(ball.y+7)))==yellow or screen.get\_at((int(ball.x), int(ball.y+8)))==yellow or screen.get\_at((int(ball.x), int(ball.y+9)))==yellow or screen.get\_at((int(ball.x), int(ball.y+10)))==yellow or screen.get\_at((int(ball.x), int(ball.y+11)))==yellow or screen.get\_at((int(ball.x), int(ball.y+12)))==yellow or screen.get\_at((int(ball.x+18), int(ball.y+6)))==yellow or screen.get\_at((int(ball.x+18), int(ball.y+7)))==yellow or screen.get\_at((int(ball.x+18), int(ball.y+8)))==yellow or screen.get\_at((int(ball.x+18), int(ball.y+9)))==yellow or screen.get\_at((int(ball.x+18), int(ball.y+10)))==yellow or screen.get\_at((int(ball.x+18), int(ball.y+11)))==yellow or screen.get\_at((int(ball.x+18), int(ball.y+12)))==yellow:

ball.vel\_x=(-ball.vel\_x)//2

elif screen.get\_at((int(ball.x+6), int(ball.y)))==yellow or screen.get\_at((int(ball.x+7), int(ball.y)))==yellow or screen.get\_at((int(ball.x+8), int(ball.y)))==yellow or screen.get\_at((int(ball.x+9), int(ball.y)))==yellow or screen.get\_at((int(ball.x+10), int(ball.y)))==yellow or screen.get\_at((int(ball.x+11), int(ball.y)))==yellow or screen.get\_at((int(ball.x+12), int(ball.y)))==yellow or screen.get\_at((int(ball.x+6), int(ball.y+18)))==yellow or screen.get\_at((int(ball.x+7), int(ball.y+18)))==yellow or screen.get\_at((int(ball.x+8), int(ball.y+18)))==yellow or screen.get\_at((int(ball.x+9), int(ball.y+18)))==yellow or screen.get\_at((int(ball.x+10), int(ball.y+18)))==yellow or screen.get\_at((int(ball.x+11), int(ball.y+18)))==yellow or screen.get\_at((int(ball.x+12), int(ball.y+18)))==yellow:

ball.vel\_y=(-ball.vel\_y)//2

elif screen.get\_at((int(ball.x+5), int(ball.y+1)))==yellow or screen.get\_at((int(ball.x+4), int(ball.y+2)))==yellow or screen.get\_at((int(ball.x+3), int(ball.y+3)))==yellow or screen.get\_at((int(ball.x+2), int(ball.y+4)))==yellow or screen.get\_at((int(ball.x+1), int(ball.y+5)))==yellow or screen.get\_at((int(ball.x+13), int(ball.y+1)))==yellow or screen.get\_at((int(ball.x+14), int(ball.y+2)))==yellow or screen.get\_at((int(ball.x+15), int(ball.y+3)))==yellow or screen.get\_at((int(ball.x+16), int(ball.y+4)))==yellow or screen.get\_at((int(ball.x+17), int(ball.y+5)))==yellow or screen.get\_at((int(ball.x+13), int(ball.y+17)))==yellow or screen.get\_at((int(ball.x+14), int(ball.y+16)))==yellow or screen.get\_at((int(ball.x+15), int(ball.y+15)))==yellow or screen.get\_at((int(ball.x+16), int(ball.y+14)))==yellow or screen.get\_at((int(ball.x+17), int(ball.y+13)))==yellow or screen.get\_at((int(ball.x+5), int(ball.y+17)))==yellow or screen.get\_at((int(ball.x+4), int(ball.y+16)))==yellow or screen.get\_at((int(ball.x+3), int(ball.y+15)))==yellow or screen.get\_at((int(ball.x+2), int(ball.y+14)))==yellow or screen.get\_at((int(ball.x+1), int(ball.y+13)))==yellow:

ball.vel\_x=(-ball.vel\_x)//2

ball.vel\_y=(-ball.vel\_y)//2

if ball.vel\_x>5:

ball.vel\_x=5

if ball.vel\_x<-5:

ball.vel\_x=-5

if ball.vel\_y>5:

ball.vel\_y=5

if ball.vel\_y<-5:

ball.vel\_y=-5

if end!=1:

if ball.vel\_x==0:

ball.vel\_x=0.5

if ball.vel\_y==0:

ball.vel\_y=0.5

if screen.get\_at((int(ball.x+29), int(ball.y+14)))==black or screen.get\_at((int(ball.x-1), int(ball.y+14)))==black or screen.get\_at((int(ball.x+14), int(ball.y+29)))==black:

screen.blit(gameover, (0,0))

ball.vel\_x=0;ball.vel\_y=0

gravity=0;end=1

pygame.display.flip()

pygame.time.delay(5000)

running=0;loss=1

elif screen.get\_at((int(ball.x+29), int(ball.y+14)))==green or screen.get\_at((int(ball.x-1), int(ball.y+14)))==green or screen.get\_at((int(ball.x+14), int(ball.y+29)))==green:

screen.blit(youwin, (0,0))

ball.vel\_x=0;ball.vel\_y=0

gravity=0;end=1

pygame.display.flip()

pygame.time.delay(5000)

running=0;loss=1

elif (pygame.time.get\_ticks() - start\_time)>=30000:

screen.blit(gameover, (0,0))

ball.vel\_x=0;ball.vel\_y=0

gravity=0;end=1

text=font.render('You ran out of

time!',True,(255,255,255))

screen.blit(text,(520,500))

pygame.display.flip()

pygame.time.delay(5000)

running=0;loss=1

#Time remaining

timeleft=font.render('Time left : '+str(int((90000-(pygame.time.get\_ticks()-start\_time))/1000%60)).zfill(2),

True,(0,0,0))

textRect = timeleft.get\_rect()

textRect.topright=[1200,5]

screen.blit(timeleft, textRect)

for bullet in bullets:

screen.blit(ball.image,(ball.x,ball.y))

# 7 - update the screen

pygame.display.flip()

# 8 - loop through the events

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key==K\_a:

keys[0]=True

elif event.key==K\_d:

keys[1]=True

if event.type==pygame.MOUSEBUTTONDOWN:

if p<3: #max 3 balls

position=pygame.mouse.get\_pos()

bullet=[math.atan2(position[1]-(playerpos1[1]+32),position[0]-(playerpos1[0]+26)),playerpos1[0]+16,playerpos1[1]+32]

velx=math.cos(bullet[0])\*5

vely=math.sin(bullet[0])\*5

ball=Ball(bullet[1],bullet[2],velx,vely,screen)

bullets.append(ball)

p+=1

if event.type == pygame.KEYUP:

if event.key==pygame.K\_a:

keys[0]=False

elif event.key==pygame.K\_d:

keys[1]=False

# check if the event is the X button

if event.type==pygame.QUIT:

# if it is quit the game

pygame.quit()

exit(0)

# 9 - Move player

if keys[0]:

playerpos[0]-=5

cannonpos[0]-=5

elif keys[1]:

playerpos[0]+=5

cannonpos[0]+=5

Head Ball

import pygame,random

from pygame.locals import \*

pygame.init()

window=pygame.display.set\_mode((1200,800))

screen\_width=1200

pygame.display.set\_caption("Head football")

bg1=pygame.image.load('C:/Users/sarathy/OneDrive/Documents/Cs programs/Head football/bg.jpg')

#window.fill([255,69,50])

bg2=pygame.image.load("badhri\_bg.jpg")

bg3=pygame.image.load('C:/Users/sarathy/OneDrive/Documents/Cs programs/Head football/bg2.jpg')

#Values of Striker A

x\_A=65

y\_A=400

radius=45

vel=5

Co\_A=(x\_A,y\_A)

hitbox\_A=(x\_A,y\_A,45)

score\_A=4

PaddleA=pygame.image.load("Player1.png")

#Values of Striker B

x\_B=1135

y\_B=400

Co\_B=(x\_B,y\_B)

hitbox\_B=(x\_B,y\_B,45)

score\_B=2

PaddleB=pygame.image.load("Player2.png")

#Values of Ball

x\_b=600

y\_b=400

radius\_b=20

vel\_x=4

vel\_y=4

Ball=pygame.image.load("Ball.png")

#Others

L=[1,-1]

Goal=False

clock=pygame.time.Clock()

#Music

pygame.mixer.init()

pygame.mixer.music.load("Game-bg.mp3")

pygame.mixer.music.set\_volume(0.7)

Start=True

Goal=False

def checkcircle(x1,y1,x2,y2):

global vel\_x,vel\_y

a=((abs(x1-x2))\*\*2) +((abs(y1-y2))\*\*2) - ((radius+radius\_b)\*\*2)

if a==100:

vel\_x=-(vel\_x+2)

vel\_y=-(vel\_y+2)

return a

#Fonts

font1=pygame.font.Font('freesansbold.ttf',37)

font2=pygame.font.Font('freesansbold.ttf',45)

goal=font2.render("GOAL",True,(0,0,0))

#For menu

bool1=True

menu=1

j=0

timer=0

#Main program

bool=True

while bool==True:

pygame.mixer.init()

pygame.mixer.music.load("Game-bg.mp3")

pygame.mixer.music.set\_volume(0.7)

if j==0:

pygame.mixer.music.play()

while menu:

keys=pygame.key.get\_pressed()

window.blit(bg1,(0,0))

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key==K\_SPACE:

menu=0

j=1

pygame.mixer.music.stop()

keys=pygame.key.get\_pressed()

pygame.mixer.init()

pygame.mixer.music.load("FOOTBALLKICK.mp3")

pygame.mixer.music.set\_volume(0.7)

for event in pygame.event.get():

if type(event)==pygame.QUIT:

bool=False

#Controls

values=[x\_A,y\_A,x\_B,y\_B,x\_b,y\_b]

#for Striker A

if keys[pygame.K\_s]:

y\_A+=vel

if keys[pygame.K\_w]:

y\_A-=vel

if keys[pygame.K\_a]:

x\_A-=vel

if keys[pygame.K\_d]:

x\_A+=vel

#for Striker B

if keys[pygame.K\_DOWN]:

y\_B+=vel

if keys[pygame.K\_UP]:

y\_B-=vel

if keys[pygame.K\_LEFT]:

x\_B-=vel

if keys[pygame.K\_RIGHT]:

x\_B+=vel

#for the Ball

while Start==True:

a=random.choice(L)

b=random.choice(L)

vel\_x=vel\_x\*a

vel\_y=vel\_y\*b

Start=False

f=x\_A-x\_b

g=y\_A-y\_b

e=f\*\*2 + g\*\*2

i=-0.5

while i<0:

x\_b-=vel\_x

y\_b-=vel\_y

i=i+1

if checkcircle(x\_b,y\_b,x\_A,y\_A)<=8000 or checkcircle(x\_b,y\_b,x\_B,y\_B)<=8000:

pygame.mixer.music.play()

if checkcircle(x\_b,y\_b,x\_A,y\_A)<=100 or checkcircle(x\_b,y\_b,x\_B,y\_B)<=100:

vel\_x=-vel\_x

vel\_y=-vel\_y

#Goal checking

if x\_b==20 and y\_b>=400-radius\*3 and y\_b<=400+radius\*3:

print("Goal")

score\_B+=1

Goal=True

x\_b=600

y\_b=400

x\_B=1135

y\_B=400

x\_A=65

y\_A=400

continue

if x\_b==1180 and y\_b>=400 - radius\*3 and y\_b<=400 + radius\*3:

print("Goal")

Goal=True

score\_A+=1

x\_b=600

y\_b=400

x\_B=1135

y\_B=400

x\_A=65

y\_A=400

continue

#Constraints

if x\_A<=45:

x\_A=45

if x\_B<=45:

x\_B=45

if x\_A>=555:

x\_A=555

if x\_B<=645:

x\_B=645

if y\_A<=45:

y\_A=45

if y\_B<=45:

y\_B=45

if x\_B>=1155:

x\_B=1155

if x\_A>=1155:

x\_A=1155

if y\_A>=755:

y\_A=755

if y\_B>=755:

y\_B=755

if y\_b>=780 or y\_b<=20:

vel\_y=-vel\_y

if x\_b>=1180 or x\_b<=20:

vel\_x=-vel\_x

#Printing

window.blit(bg2,(0,0))

pygame.draw.circle(window,(27,3,163),(0,400),round(radius\*3),1)

pygame.draw.circle(window,(27,3,163),(1200,400),round(radius\*3),1)

pygame.draw.circle(window,(27,3,163),(600,400),round(radius\*3),1)

pygame.draw.rect(window,(27,3,163),(596,0,6,400-135))

pygame.draw.rect(window,(27,3,163),(596,400+135,6,800))

pygame.draw.rect(window,(27,3,163),(0,0,5,800))

pygame.draw.rect(window,(27,3,163),(1195,0,5,800))

pygame.draw.rect(window,(27,3,163),(0,1200,5,0))

"""pygame.draw.circle(window,(0,0,255),(x\_A,y\_A),radius)

pygame.draw.circle(window,(0,255,0),(x\_B,y\_B),radius)"""

window.blit(PaddleA,(x\_A-radius,y\_A-radius))

window.blit(PaddleB,(x\_B-radius,y\_B-radius))

window.blit(Ball,(x\_b-radius\_b,y\_b-radius\_b))

time=timer

while Goal==True and timer<=time+50000:

window.blit(goal,(550,150))

timer=timer+1

timer=time

timer=timer+1

Goal=False

"""pygame.draw.circle(window,(255,0,0),(x\_b,y\_b),radius\_b)"""

"""pygame.draw.circle(window,(255,0,0),(x\_A,y\_A),radius+1,1)

pygame.draw.circle(window,(255,0,0),(x\_B,y\_B),radius+1,1)"""

score=font1.render("Score : "+str(score\_A)+" - " + str(score\_B),True,(0,0,0))

window.blit(score,(495,50))

clock.tick(60)

i=0

pygame.display.update()

pygame.mixer.init()

pygame.mixer.music.load("Game-bg.mp3")

pygame.mixer.music.set\_volume(0.7)

#print(score\_A//2,score\_B//2)

if score\_A ==5:

k=0

if k==0:

pygame.mixer.music.play()

b=score\_B

window.blit(bg3,(0,0))

while score\_A==5:

keys=pygame.key.get\_pressed()

pygame.display.update()

end1=font2.render("Player A wins",True,(0,0,0))

window.blit(end1,(450,400))

end2=font1.render("Score :"+str(score\_A)+" - " + str(score\_B),True,(0,0,0))

window.blit(end2,(500,525))

end1=font1.render("Press SPACE to exit",True,(0,0,0))

window.blit(end1,(400,650))

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key==K\_SPACE:

bool=False

score\_A=0

k=1

if score\_B ==5:

b=score\_A

window.blit(bg3,(0,0))

k=0

if k==0:

pygame.mixer.music.play()

while score\_B==5:

keys=pygame.key.get\_pressed()

pygame.display.update()

end1=font2.render("Player B wins",True,(0,0,0))

window.blit(end1,(450,400))

end2=font1.render("Score :"+str(score\_A)+" - " + str(score\_B),True,(0,0,0))

window.blit(end2,(500,550))

end1=font1.render("Press SPACE to exit",True,(0,0,0))

window.blit(end1,(400,650))

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key==K\_SPACE:

bool=False

score\_B=0

k=1

pygame.quit()

quit()

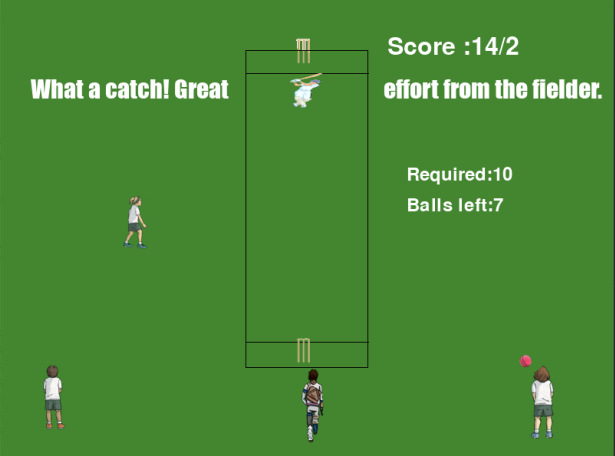
Output

Chennai’28:

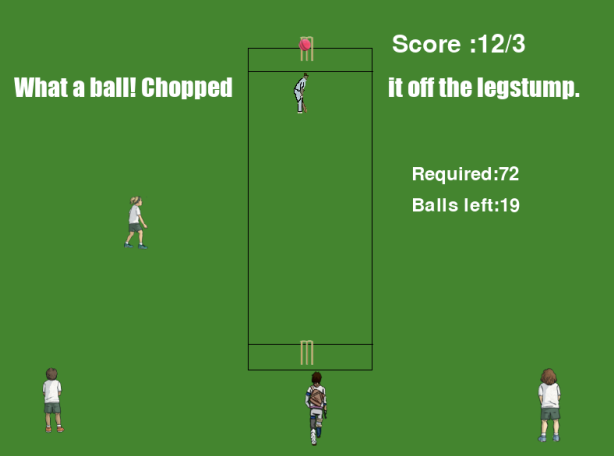






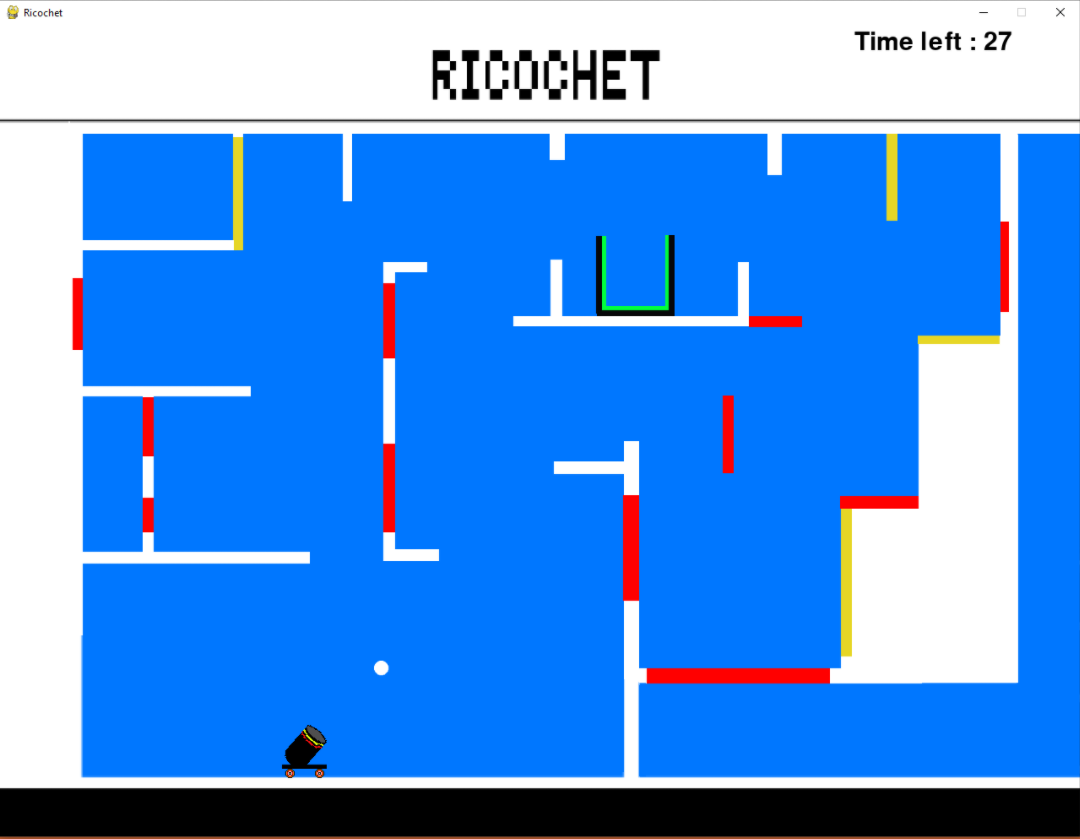






Ricochet:









Head Ball:



