COMPUTER GRAPHICS

MINI PROJECT

SQUIRREL EAT SQUIRREL

**BY-**

**JAYASHREE DOMALA 29**

**MANMOHAN DOGRA 28**

**ABDUS SAMAD MUKRI 47**

**INTRODUCTION**

Pygame is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) set of [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) modules designed for writing [video games](https://en.wikipedia.org/wiki/Video_game). It includes [computer graphics](https://en.wikipedia.org/wiki/Computer_graphics) and sound [libraries](https://en.wikipedia.org/wiki/Library_(computing)) designed to be used with the Python [programming language](https://en.wikipedia.org/wiki/Programming_language).

Pygame uses the [Simple DirectMedia Layer](https://en.wikipedia.org/wiki/Simple_DirectMedia_Layer) (SDL) library, with the intention of allowing [real-time](https://en.wikipedia.org/wiki/Real-time_computer_graphics) [computer game](https://en.wikipedia.org/wiki/Computer_game) development without the [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language)mechanics of the [C programming language](https://en.wikipedia.org/wiki/C_(programming_language)) and its derivatives. This is based on the assumption that the most [expensive](https://en.wikipedia.org/wiki/Computationally_expensive) functions inside games, can be abstracted from the [game logic](https://en.wikipedia.org/wiki/Model-view-controller), making it possible to use a [high-level programming language](https://en.wikipedia.org/wiki/High-level_programming_language), such as Python, to structure the game.

Python has a module named time to handle time-related tasks. To use functions defined in the module, we need to import the module first.

**PROBLEM DEFINITION**

To create a squirrel game based on computer graphics using PyGame.

**IMPLEMENTATION**

import time

import random, sys, time, math, pygame

from pygame.locals import \*

FPS = 30 # frames per second to update the screen

WINWIDTH = 640 # width of the program's window, in pixels

WINHEIGHT = 480 # height in pixels

HALF\_WINWIDTH = int(WINWIDTH / 2)

HALF\_WINHEIGHT = int(WINHEIGHT / 2)

GRASSCOLOR = (24, 255, 0)

WHITE = (255, 255, 255)

RED = (255, 0, 0)

CAMERASLACK = 90 # how far from the center the squirrel moves before moving the camera

MOVERATE = 9 # how fast the player moves

BOUNCERATE = 6 # how fast the player bounces (large is slower)

BOUNCEHEIGHT = 30 # how high the player bounces

STARTSIZE = 25 # how big the player starts off

WINSIZE = 300 # how big the player needs to be to win

INVULNTIME = 2 # how long the player is invulnerable after being hit in seconds

GAMEOVERTIME = 4 # how long the "game over" text stays on the screen in seconds

MAXHEALTH = 3 # how much health the player starts with

NUMGRASS = 80 # number of grass objects in the active area

NUMSQUIRRELS = 30 # number of squirrels in the active area

SQUIRRELMINSPEED = 3 # slowest squirrel speed

SQUIRRELMAXSPEED = 7 # fastest squirrel speed

DIRCHANGEFREQ = 2 # % chance of direction change per frame

LEFT = 'left'

RIGHT = 'right'

# check if the player has collided with any squirrels

for i in range(len(squirrelObjs)-1, -1, -1):

sqObj = squirrelObjs[i]

if 'rect' in sqObj and playerObj['rect'].colliderect(sqObj['rect']):

# a player/squirrel collision has occurred

if sqObj['width'] \* sqObj['height'] <= playerObj['size']\*\*2:

# player is larger and eats the squirrel

playerObj['size'] += int( (sqObj['width'] \* sqObj['height'])\*\*0.2 ) + 1

del squirrelObjs[i]

if playerObj['facing'] == LEFT:

playerObj['surface'] = pygame.transform.scale(L\_SQUIR\_IMG, (playerObj['size'], playerObj['size']))

if playerObj['facing'] == RIGHT:

playerObj['surface'] = pygame.transform.scale(R\_SQUIR\_IMG, (playerObj['size'], playerObj['size']))

if playerObj['size'] > WINSIZE:

winMode = True # turn on "win mode"

elif not invulnerableMode:

# player is smaller and takes damage

invulnerableMode = True

invulnerableStartTime = time.time()

playerObj['health'] -= 1

if playerObj['health'] == 0:

gameOverMode = True # turn on "game over mode"

gameOverStartTime = time.time()

else:

# game is over, show "game over" text

DISPLAYSURF.blit(gameOverSurf, gameOverRect)

if time.time() - gameOverStartTime > GAMEOVERTIME:

return # end the current game

# check if the player has won.

if winMode:

DISPLAYSURF.blit(winSurf, winRect)

DISPLAYSURF.blit(winSurf2, winRect2)

pygame.display.update()

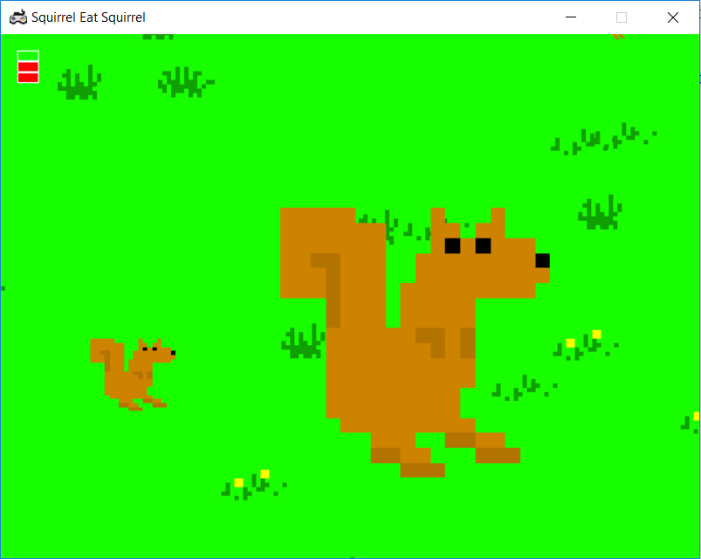
FPSCLOCK.tick(FPS)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**RESULTS**

**CONCLUSION AND FUTURE SCOPE**

We successfully completed the project and created a game using python, pygame on computer graphics. Here we used libraries like: random, sys, time, math, pygame. User can move the squirrel using directional keys, the squirrel can eat smaller squirrels to increase its size. If it collides with a larger squirrel its health will decrease. You win, once you reach the omega size and restart the game by pressing r.

For future scope, we can make it multiplayer, can develop it as web application and android.

**REFERENCES**

<https://www.pygame.org/wiki/tutorials>

<https://www.pygame.org/news>

<https://pythonprogramming.net/pygame-python-3-part-1-intro/>