**Duck Hunt**

Duck Hunt is implemented in python and uses the pygame library. It is a simple 2-D point and click shooting game. Ducks fly across the screen from left to right. The player has an infinite amount of shots to hit the duck on the screen. However, if a duck makes it across the screen, the game will end and the iconic dog pops out of the grass to indicate the game is over. The player has two options once the game ends. The player can hit the up arrow key on the keyboard to restart the game or hit the escape key to end the game. Credit to Johann Gomes for the original game code in which our game is based off of. The orginal game code can be found on pygame.org.

**Appendix**

**#**-------------------------------------------------------------------------------

# Author: Khashi Xiong

# Team: eyeCU

# Duck Hunt

**#**-------------------------------------------------------------------------------

import pygame

from pygame.locals import\*

from sys import exit

from random import randint

pygame.init()

ccount = 100

# Define black **and** white

black = (0, 0, 0)

white = (255, 255, 255)

# Set screen size to 640x480 **and** go fullscreen

screen = pygame.display.set\_mode((640, 480),pygame.FULLSCREEN)

# Window Title

pygame.display.set\_caption("eyeCU Duck Hunt")

# Mouse cursor position variables

x\_position = 0

y\_position = 0

# Mouse cursor click variable

x\_click = 0

y\_click = 0

# Duck position variables

x\_duck = 0

y\_duck = randint(10, 350)

points = 0

speed = 1

missed = False

notmissed = True

restart = False

# Player starts

pygame.mixer.init(44100, -16, 2, 1024)

# Music volume

pygame.mixer.music.set\_volume(0.8)

game\_running = True

**while** game\_running:

**for** event in pygame.event.get():

**if** event.type == pygame.QUIT:

game\_running = False

elif event.type == pygame.KEYDOWN:

**if** event.key == pygame.K\_ESCAPE:

game\_running = False

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

restart = True

# Used by the reticle

elif event.type == MOUSEMOTION:

x\_position, y\_position = pygame.mouse.get\_pos()

# This is used to **register** the mouse click

elif event.type == MOUSEBUTTONDOWN:

x\_click, y\_click = pygame.mouse.get\_pos()

ccount = 0

# Starting position of the reticle

position = (x\_position - 50, y\_position - 50)

# Move the duck forward

x\_duck += 1

# Defines the end position **for** the duck to end the game

**if** x\_duck \* speed > 640 **and** **not** missed:

x\_duck = 0

y\_duck = randint(10, 350)

# Play game over music **if** duck is missed

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

pygame.mixer.music.play()

missed = True

notmissed = False

# Make the background black

screen.fill(black)

pygame.mouse.set\_visible(False)

# Render the other screen images

screen.blit(pygame.image.load("background.png"), (0, 0))

screen.blit(pygame.font.SysFont("tahoma", 20).render("Points: " + str(points), True, white), (450, 10))

# Threshold **for** duck to be in the reticle to be a hit

**if** notmissed:

ccount = ccount + 1

**if** x\_click in range(x\_duck \* speed - 20, x\_duck \* speed + 20) **and** y\_click in range(y\_duck - 30, y\_duck + 30):

# Play hit audio

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

pygame.mixer.music.play()

# Increase point by 1 **if** duck is hit

points += 1

# New duck position

x\_duck = 0

y\_duck = randint(10, 350)

# Draw the **new** duck

screen.blit(pygame.image.load("duck.gif"), (x\_duck \* speed, y\_duck))

**if** ccount > 30 :

x\_click = 0

y\_click = 0

**if** missed:

# If the duck is missed, then load the dog image

x\_duck = -50

y\_duck = -50

screen.blit(pygame.image.load("dog.gif"), (320, 300))

# Tell player to restart **or** quit program

screen.blit(pygame.font.SysFont("tahoma", 20).render("Press UP to Restart or ESC to Quit", True, white), (175, 175))

# Initiate restart **if** up is pressed

**if** restart:

missed = False

notmissed = True

points = 0

x\_duck = 0

y\_duck = randint(10, 350)

restart = False

screen.blit(pygame.image.load("reticle.gif").convert(), position)

pygame.display.update()

pygame.quit()