**Egg Drop**

**Game Design Document**

**Briana Franklin**

**Overview**

“Egg Drop” is a basic 2D arcade-style game.

The player Henrietta the Hen will move left and right along the bottom of the screen with arrow keys, catching eggs falling from above. Each egg caught added to the player’s scores. Missed eggs will reset at the top. The background features a grassy field, blue sky and flower or two for visual flair. The game is timed for 10 seconds. Once the game is complete it will display your score. You will be given the option to Play or Quit.

**State Transition Diagram**

**A diagram of a diagram

AI-generated content may be incorrect.**

Begin: This will be were the game starts and open the “Intro” screen. In the “Intro” the player will be able to read the instructions on how to play, view their score, chose to play or quit.

Play button will allow the player to move to the “Play” Screen

Player will control Henrietta the Hen to catch the eggs in the basket and earn 1 point per egg caught.

Once the timer runs out, the game will end and transition back to the “Intro” screen, displaying the player final score.

Quit button will end the game

**The Instructions Scene**

**A diagram of a diagram

AI-generated content may be incorrect.**

**Visual elements**

* instructions- multiLabel
* prevScore – Label
* btnPlay – Button
* btnQuit – Button

Other Attributes

prevScore – int showing the last scored on Game screen

response- two buttons “Play” and “Quit”

**Init(sore):**

Set image to gamebackground.jpg

Set response to “Play”

Create lblinstructions Multilabel

Add textLines containing instructions

Set instructions top center

Set instructions size

Copy score parameter to prevScore attribute

Creal lblScore

Set text to “Last Score: {prevScore}”

Set bottom center

Create btnPlay

Set text to “Play”

Set bottom left center

Creat btnQuit

Set text to “Quit”

Set bottom right center

Add lblInstructions, lblScore, btnPlay, and btnQuit to sprites

**What will happen**

process():

If quit button is pressed:

Set response to “Quit”

Stop the Scene

If play button is pressed

Set response to “Play”

Open The Game Class

**The Game class**

A white background with text and images

AI-generated content may be incorrect.

**Visual Attributes**

* henrietta- Henrietta class
* Eggs- Egg class
* lblScore- LBLScore class
* lblTime- LblTimeclass

**Non-sprite Assets**

* timer- stock so
* score- int containing the score

**Initializer will create all need components:**

Init:

Set image to Game Background.jpg

Create timer

Set timer’s total time to 10

Set score to zero

Initialize sndThump to egg sound effect

Create instance of Henrietta -> henrietta

Create list of (15) Egge instances -> eggs

Create instance of LblScore -> lblScore

Create instance of LblTime -> lblTime

Add henrietta, eggs, lblScore, lbltime, to sprites

**What will happen**

process:

For each egg in the egg list:

If egg collides with henrietta:

Play the egg collision sound (sndThump)

Rest egg

Add one to the score

Update lblScore to indicate the new score

Update lblTimer wit the current time left

If the time left is less than zero:

Stop the scene

**Components**

**Henrietta**

init():

* Image “henrietta.png”
* Bottom Center

process():

* Up, Down, Left Right Arrows allows the chicken to move around the board

**Egg**

init():

* Image “egg.png”
* Call reset

reset():

* Top of screen
* Random positon
* Random speed

checkBounds():

if egg hit bottom

Call reset ()

**LblScore**

Label top left of the board

Add 100 points everytime an egg is caught

**LblTime**

Label top right of the screen

Time starts at 30

When time hits 0, Game will end and go back to instruction screen

**Milestones**

1. Put the background on the screen
2. Add basic sprite
3. Add keyboard motion to Charlie
4. Add single coin with reset, falling and boundary behaviors
5. Add collison effect between Charlie an coin, sound effect
6. Modifu for multiple (ren coins) including collision behavior
7. Add scorekeeping, timing and appropriate labels
8. Add instructions class and state transition

**Asset Plan**

Game Background.jpg

**A grass and clouds in the sky

AI-generated content may be incorrect.**

Made in Canva Design

Henrietta The Hen.png

A cartoon of a bird holding a basket

AI-generated content may be incorrect.

Made in Canva Design

Egg.png

A white circle in a black background

AI-generated content may be incorrect.

Made in Canva Design

**Code**

**import simpleGE, pygame, random**

**class Game(simpleGE.Scene):**

**def \_\_init\_\_(self):**

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

**self.lblScore = LblScore()**

**self.score = 0**

**self.timer = lblTimer()**

**self.timer.totalTime = 30**

**self.hen = Hen(self)**

**self.eggs = []**

**for i in range(10):**

**self.eggs.append(Egg(self))**

**self.sprites = [self.hen, self.eggs, self.lblScore, self.lblTimer]**

**def process(self):**

**for egg in self.eggs:**

**if self.hen.collidesWith(egg):**

**egg.reset()**

**self.score += 100**

**self.lblScore.text = f"Score: {self.score}"**

**self.lblTime.text = f"Time Left: {self.timer.getTimeLeft()}"**

**if self.timer.getTimeLeft() < 0:**

**print(f"Score: {self.score}")**

**self.stop()**

**class Instructions(simpleGE.Scene)**

**def \_init\_(self, prevScore):**

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

**self.prevScore = prevScore**

**self.setImage("Game Background.png")**

**self.response = "Quit"**

**self.directions = simleGE.Multilabel()**

**self.directions.textLines = [**

**"Use the left and right arrow keys to",**

**"allow Henrietta the Hen to catch the eggs",**

**"See how many eggs you can catch",**

**"before the time runs out"]**

**self.directions.center = (320, 240)**

**self.directions.size = (500, 250)**

**self.btnPlay = simpleGE.Button()**

**self.btnPlay.text = "Play"**

**self.btnPlay.center = (100,400)**

**self.btnQuit = simpleGe.Button()**

**self.btnQuit.text = "Quit"**

**self.btnPlay.center = (500, 400)**

**self.lblScore = simpleGe.Label()**

**self.lblScore.text = "Last score: 0"**

**self.lblScore.center = (320, 400)**

**self.lblScore.text = f"Last score: {self.prevScore}"**

**self.sprites = [self.directions, self.btnPlay, self.btnQuit, self.lblScore]**

**def setPrevScore(self, prevScore):**

**self.prevScore = prevScore**

**self.lblScore.text = f"Last score: {self.prevScore}"**

**def process(self):**

**if self.btnPlay.clicked:**

**self.response = "Play"**

**self.stop()**

**if self.btnQuit.clicked:**

**self.response = "Quit"**

**self.stop()**

**class Hen(simpleGE.Sprite):**

**def \_\_init\_\_(self, scene):**

**super().\_\_init\_\_(scene)**

**self.setImage("henriettathehen.png")**

**self.position = (320, 240)**

**def process(self):**

**if self.isKeyPressed(pygame.K\_LEFT):**

**self.x -= 5**

**if self.isKeyPressed(pygame.K\_RIGHT):**

**self.x += 5**

**if self.isKeyPressed(pygame.K\_UP):**

**self.y -= 5**

**if self.isKeyPressed(pygame.K\_DOWN):**

**self.y += 5**

**class Egg(simpleGE.Sprite):**

**def \_\_init\_\_(self, scene):**

**super().\_\_init\_\_(scene)**

**self.setImage("Egg.png")**

**self.reset()**

**def reset(self):**

**self.x = random.randint(0, self.screenWidth)**

**self.y = random.randint(0, self.screenHeight)**

**self.dx = random.randint(-3, 3)**

**self.dy = random.randint(-3, 3)**

**class LblScore(simpleGE.Label):**

**def \_\_init\_\_(self):**

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

**self.text = "Score: 0"**

**self.center = (100, 100)**

**class LblTimer(simpleGE.Label):**

**def \_init\_(self):**

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

**self.text = "Time left: 10"**

**self.center = (500, 400)**

**def main():**

**keepGoing = True**

**lastScore = 0**

**while keepGoing:**

**instructions = Instructions(lastScore)**

**instructions.setPrevScore(lastScore)**

**instructions.start()**

**if instructions.response == "Play":**

**game = Game()**

**game.start()**

**lastScore = game.score**

**else:**

**keepGoing = False**

**print(instructions.response)**

**game = Game()**

**game.start()**

**if \_\_name\_\_ == "\_\_main\_\_":**

**main()**