Set SCREEN\_WIDTH to 800

Set SCREEN\_HEIGHT to 600

Set BACKGROUND\_COLOR to (0, 0, 0)

Set FROG\_COLOR to (0, 255, 0)

Set CAR\_COLOR to (255, 0, 0)

Set CAR\_WIDTH to 50

Set CAR\_HEIGHT to 30

Set CAR\_SPEED to 7

Set CAR\_GAP to 200

Initialize pygame

Set screen to display with dimensions (SCREEN\_WIDTH, SCREEN\_HEIGHT)

Set window caption to "Frogger"

Initialize clock

Define class Frog:

Initialize width to 30

Initialize height to 30

Set x to half of SCREEN\_WIDTH minus half of width

Set y to SCREEN\_HEIGHT minus height

Set color to FROG\_COLOR

Define method draw():

Draw rectangle on screen with color, position (x, y), and size (width, height)

Define method move(dx, dy):

Add dx to x

Add dy to y

If x is less than 0, set x to 0

Else if x is greater than SCREEN\_WIDTH - width, set x to SCREEN\_WIDTH - width

If y is less than 0, set y to 0

Else if y is greater than SCREEN\_HEIGHT - height, set y to SCREEN\_HEIGHT - height

Define class Car:

Define method \_\_init\_\_(x, y):

Initialize width to CAR\_WIDTH

Initialize height to CAR\_HEIGHT

Set x to input x

Set y to input y

Set color to CAR\_COLOR

Set speed to CAR\_SPEED

Define method draw():

Draw rectangle on screen with color, position (x, y), and size (width, height)

Define method move():

Add speed to x

If x is greater than SCREEN\_WIDTH, set x to -width

Create Frog object

Create empty list cars

Repeat 3 times:

Create Car object with random x position and y position based on loop index multiplied by CAR\_GAP

Append Car object to cars list

Set running to True

While running is True:

Fill screen with BACKGROUND\_COLOR

For each event in pygame event queue:

If event is QUIT:

Set running to False

Get pressed keys

If left key is pressed, move frog left

If right key is pressed, move frog right

If up key is pressed, move frog up

If down key is pressed, move frog down

For each car in cars:

Move car

Draw car

Draw frog

For each car in cars:

If frog collides with car:

Reset frog position to center bottom of screen

Update display

Tick clock at 60 frames per second

Define class LblTime:

Define method \_\_init\_\_():

Call parent constructor

Set text to "Time Left: 10"

Set center position to (500, 30)

Define method process():

For each coin in game:

If frog collides with coin:

Play coin sound

Reset coin position

Increase score by 1

Update score label

Update time label with remaining time

If time runs out:

Print final score

Stop game

Define class Instructions:

Define method \_\_init\_\_(score):

Call parent constructor

Set background image

Set instructions text

Set score label

Set play and quit buttons

Define method process():

If quit button clicked, set response to "Quit" and stop scene

If play button clicked, set response to "Play" and stop scene

If up arrow key pressed, set response to "Play" and stop scene

If down arrow key pressed, set response to "Quit" and stop scene

Define main function:

Initialize keepGoing to True

Initialize score to 0

While keepGoing is True:

Show instructions scene and get response

If response is "Play":

Start game

Update score

Else:

Set keepGoing to False

Quit pygame