PIC 16 Final Project: Powell Cat

Rui Yan, Fanfei Li, Xue Xia

Our game "Powell Cat" is based on classic snake game, but with many new and creative rules. (YouTube demo: https://youtu.be/li-Hd-VYLqc)

Game background:

The Powell Cat has been a staple to all students that have frequented by the Powell Library at UCLA over the past years, quickly becoming a student favorite. It was adopted by a Powell Librarian and considered as a UCLA mascot, like Bruin Bear.

Game rules:

- Along the road, the cat can collect books to get points. The more books collect, the higher the points. But the bombs are randomly moving, and collision needs to be avoided.
- The cat has three parts: head, body, and tail. After collecting a book, the length of the cat's body will increase by one, i.e. the score will increase by one.
- In each difficulty level, the speeds of both the cat and the bomb will change. The speed is low in easy mode, mediate in median mode and high in hard mode.
- The player can control the "Cat" to move in four different directions: up, down, left, right. (Note that if the cat goes into the opposite direction of its movement, it will collide this its own body and die.)
- The game will end in the following circumstances: (1) the cat hits an edge; (2) the cat hits his own body; (3) that cat hits a bomb; (4) game player manually quits the game; (5) the cat collects as many books as possible and wins the game.

Pseudocode:

```
build a class Bomb
#define the initial values
   import the background image
   set the width of the background image
   set the height of the background image
   define x position of the bomb
   define y position of the bomb
   define x speed of the bomb
   define y speed of the bomb
   set default speed to zero
#define the function set_speed that takes a parameter speed
```

```
    set x speed equal to the parameter speed
    set y speed equal to the parameter speed


build a class CattGUI
#define the initial values in the __init__ function
    set the unit length to 20
    set the difficulty to false
    set the pausing test to true
    ##initiate the window
        define the size of the window
        define the title of the window
        import the background image
        set the width of the background image
        set the height of the background image
    ##define the initial variables
        set the initial frequency of the bomb occurrence
        set the initial speed of the cat
        set the initial number of bombs
        declare the new game function
    ##initiate the canvas
        set the size of the canvas
        add the background image to the canvas
        ##initiate the start button
        add the start button to the canvas

#define the start function
    stop the pause of the game
    change the “start” button to the “pause” button
    call the animate function

#define the pause function
    change the “start” button to the “resume” button
    pause the game

#define the popup function which pops the window of setting difficulty level
    add the “easy” button to the popup window
    add the “medium” button to the popup window
    add the “hard” button to the popup window

#define the newgame function which initiates a new game
    ##set the group of coordinates that the food and each cat segment will be in
    ##initiate the start button again when renew the game
        set the “new game” button back to the “start” button
        set the score to 0
        update the score on canvas
    ##set size of each 'step' when the cat moves
    ##set a list of the coordinates of each cat segment to make the food not falling in this list
        set the coordinate by the x coordinate and y coordinate
        set the tail variable as the first variable
            in the coordinate array
        set the body variable as the second variable in the coordinate array
    ##initial coordinates of the first cat segment
        set the x coordinate of the first cat segment
        set the y coordinate of the first cat segment
    ##declare a list of cat segments
    ##call the load_images function
    ##form the most initial cat (with only three segments)
        add the tail segment with the tail picture
        add the body segments with the body pictures
        add the head segments with the head pictures
    ## bind the moving directions with the keys
        bind the up moving buttons
        bind the down moving buttons
        bind the right moving buttons
        bind the left moving buttons
    ##call set_difficulty _parameters function
    ##call the food function to create foods
    ##call the create_bomb function to create bombs
    ##call the bomb_run() to move bombs

#define the animate function which moves the cat
    if the game is not paused:
        record the coordinates of the old head
        change the coordinate of the cat’s head to make it move
        ##test the conditions that can make the game end:
            define game over as False
        ##check if the cat collides the bombs
            for bombs:
                get the coordinates of the segments
                get the coordinates of the head
                get the coordinates of the tail
                if the cat body coordinates fall on the bomb
                    change game over to True
                    break out of the loop
                if the head coordinates fall on the bomb
                 or the head gets to the edges:
                    show a popup window with game over and the score
                    change the button to the “new game” button
        ##check if the head coordinates fall in the food:
            add one to the score
            remove the food from the canvas
            call the food function
            delete the last segment
            add the new segment
            update score on canvas
    else:
        delete the last segment
        add the new segment
        change the image of the segments

#define the create_bomb function to create bombs
```

```
define a bomb list to store all bomb objects
while length of the bomb list not equal to number of bombs: ## generates new bombs
    create bomb objects from the Bomb class
    if bomb does not collide the cat:
        set the speed of bombs calling set_speed in bomb class
        append a new bomb to the bomb list
        show the bomb on canvas

#define the bomb_run function which randomly moves bombs
    for bomb:
        show the move on canvas
        set the position variable equal to the position on canvas
        if the bomb hits the upper or lower edge:
        change the speed of the y direction
        if the bomb hits right or left edge:
        change the speed of the x direction

#define the recall function which starts a new game
    remove everything from the canvas
    add the background image to the canvas
    call the newgame function

#define the food function which generates books
    generate the x coordinate of the food
    generate the y coordinate of the food
    if the food is not in the cat:
        draw the food on the canvas
    else:
        call the food function

#define the up function
    change the x direction to 0
    change the y direction to positive

#define the down function
    change the x direction to 0
    change the y direction to negative

#define the right function
    change the y direction to 0
    change the x direction to positive

#define the left function
    change the y direction to 0
    change the x direction to negative

#define the command of the "easy" button: closeandeasy
    call the set_difficulty_parameters function as easy mode
    destroy the popup window

#define the command of the "medium" button: closeandmedium
```

```
    call the set_difficulty_parameters function as medium mode
    destroy the popup window

#define the command of the “hard” button: closeandhard
    call the set_difficulty_parameters function as hard mode
    destroy the popup window

#define the set_difficulty_parameters function:
    if the variable is easy:
        set the bomb number
        set the speed
        set the difficulty level
    if the variable is medium:
        set the bomb number
        set the speed
        set the difficulty level
    if the variable is hard:
        set the bomb number
        set the speed
        set the difficulty level
        set the set difficulty variable to true

#define the load_images function
    ##loads images from the disk
    load the body image
    load the head image
    load the tail image
    load the book image
    load the bomb image
```