Keeping Score

We want to track the number of apples consumed by the worm and display this value as the player’s score.

We’ll need to add code to count every time an apple is eaten. Then we’ll need to add code to display the current score every time the screen is re-drawn.

Counting the Apples

We’ll track the apple count in WormData.py, where most of the game related information is handled. First, add to the newGame() method so that we have a variable that counts the score, and set the initial score to 0. It does matter where this line appears in this method.

```python
self.score  = 0
```

We need to count everytime the worm eats an apple. This will be in the evolve() method, because the evolve method moves the worm, and checks for collisions between the worm’s head and the apple. Find the line that grows the tail of the player. Add this line right after it, at the same indentation level:

```python
self.score += 1
```

Done. We can now track the score easily.

Displaying the Score

Ok, time to show the user their score. This is also done in WormData.py, where the main screen drawing occurs. Find the draw() method.

In this method, after the background is drawn, but before the worm and apple are drawn, add these lines to draw the score:

```python
score_txt = "Apples: " + str(self.score)
score_txt
```

The first line creates a text message, and the second line causes it to be drawn, 10 pixels from the left of the screen, and 20 pixels from the top of the screen.

Other Thoughts

- If you want to draw the score somewhere else, just change the numbers.
- If you want the text to be centered, instead of left justified, use drawTextCenter instead of drawTextLeft.
- If you want a different size of text, change the size of the font created in the __init__ method.
- If you want a different color of text, change the color in __init__.
- If you end up with multiple ways to score, you can define the number of points per each scoring event by added different numbers to the score.
- Use your imagination to make the game more interesting.