Week 4 Challenges

Challenge 1

Round the and sum

Given a **list** of numbers create a program that can round each the number to the nearest multiple of 10 and then add up all the rounded numbers.

15 rounds up to 20.
12 rounds down to 10.

Your input must be a **list** of 3 to 10 numbers of any size

Example:

[16, 17, 18] → 60
(12, 13, 14, 3) → 30
(6, 4, 4, 23, 16) → 50

Challenge 2

Using trinket and turtle commands create a simple user interface that will get something (anything) visual to happen on the screen when you touch a particular key on the key board AND also when you click the mouse on a certain area of the screen.

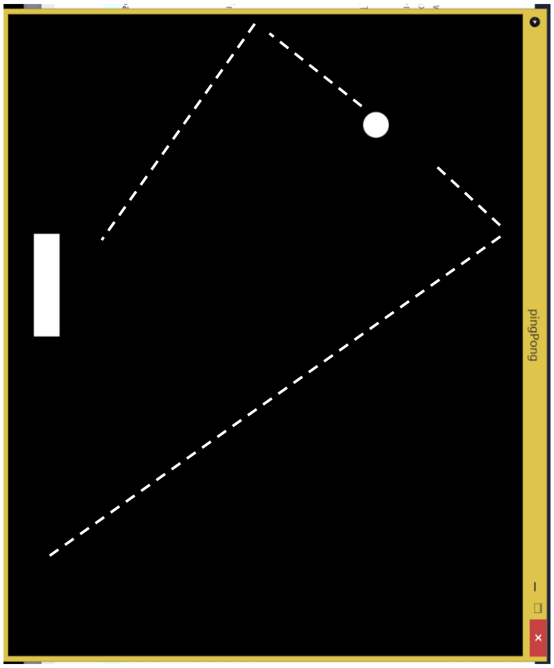
Example:

If the user clicks on a certain area of the screen the turtle will go to that point.

If the user presses the letter "b" he turtle will go back to his starting position.

You can use this example as your tasks if you wish.

Challenge 3



Make a pong game using turtle graphics

Feel free to look at previously constructed pong games/code to develop your game but the following features **must** be in your game:

- You must have horizontal orientation (paddle moves up and down the scree)
- Ball should rebound off 3 out of 4 walls
- If the ball touches the space/wall that the player is defending then (game over).
- The ball's speed will increase after every 4 times the ball touches the players paddle.

Challenge 4 (please contact Cain with questions or clarification)

Cain’s cool coding c(h)allenges

the Check Sum

Here at Walzl inc. were looking to get into the baking industry as a financial crisis is as good as ever to open new and costly financial endeavours. Specifically, we’re looking to create a credic card: Waz-merican Express, or WamEx for short. A hallmark of any good credit card is a check-sum, an algorithm that when the numbers of the credit card are put through return a factor of 10. The algorithm in question is Luhn’s Algorithm which goes as follows given this credit card number:

4536 0004 2073 5206

Step one:
starting from the right, take every second digit and multiply it by 2 (double it)

8,6,0,0,4,14,10,0

Step two:

Take each digit from the product of doubling and add them together (so that means if it’s a two digit number, treat each digit as its own number ie. 12 = 1 + 2)

8 + 6 + 0 + 0 + 4 + 1 + 4 + 1 + 0 + 0 = 24

Step three:

Now take that sum (24) and add it to the numbers that you didn’t do anything with (the Unhighlighted ones)

24 + 5 + 6 + 0 + 4 + 0 + 3 + 2 + 6 = 50

Step four:

Now, given your answer (50), if it is evenly divisible by 10, meaning if you did ANS % 10 (% being the modulo operator which gives you the remainder of a division), it equals zero. If it doesn’t equal zero and the answer from the algorithm does not equal a factor of 10, the card number is fraudulent, which is no good for an up and coming bank.

Recap:

Obtain card number from user

Enter it through Luhn’s Algorithm

Check if its evenly divisible by 10 (use modulo (%) function)

Say if the card number is valid or invalid

Card Numbers: https://developer.paypal.com/docs/payflow/payflow-pro/payflow-pro-testing/#

Ignore any warnings, its fine,,,,

BONUS:

You can identify what type of card each one is based on the length of the card number and the starting digits.

CARD NUMBER LENGTHS:

Visa: 13,16 digits

WazMerican Express (WamEx): 15 digits

MasterCard: 16 digits

CARD NUMBER STARTING DIGITS:

Visa: 4 (ie. 4003600000000014)

WamEx: 34,37 (ie. 340000000030402, 3700030137423)

MasterCard: 51, 52, 53, 54, 55 (you get the idea)

Ok great that's it