Lecture Week 8

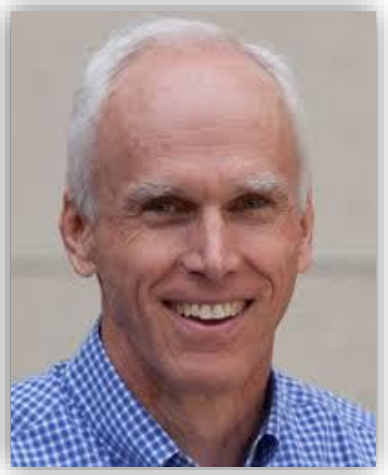
Graphics and User Interaction (Tkinter)

Dr Stewart Blakeway
Lecturer in Computer Science

COMP16321
Introduction to Programming 1

Intro

tkinter



- Tkinter is the standard GUI library for Python.
- Tkinter provides a fast and easy way to create GUI applications.
- Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.
- Tk was developed as a GUI extension for the Tcl scripting language by John Ousterhout (1991)
- bindings for lots of other programming languages have also been developed (Perl, Ada, Ruby, Lisp…)

Steps in creating a GUI

1. Import tkinter (and the widgets that you want)
2. Create a main window (and configure it)
3. Add your widgets
4. Enter a main event loop to manage your triggers (i.e. button press)

Creating a Window

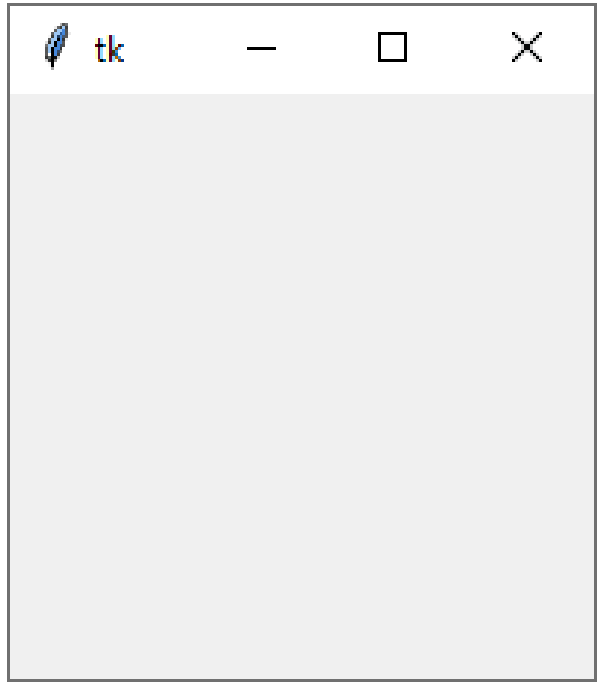
```
from tkinter import Tk

window = Tk()
window.mainloop()
```

We import Tk from the tkinter module

We create an instance of the Tk class

We enter a main loop (this displays the window)



Configuring the Window

We create a new function for the configuration of the window.

```
from tkinter import Tk
# All my functions go here
window = Tk()
configure_window()
# Any code not within a function will go here
# Mostly stuff that will be accessed by more
# than one function
window.mainloop()
```

Configuring the Window

```
def configure_window():
    window.geometry("800x600")
    window.configure(background='#b3ffff')
    window.title("M  A  T  H  S     Q  U  I  Z ")
```

Conclusion

- Imported TkInter
- Created a simple GUI
- Used some methods to change the appearance

- Also, discussed that mainloop is a blocking method

End of Section

Time to take a break! Make a cup of tea, think about what we have discussed. Come back when you are refreshed and ready

Dr Stewart Blakeway
Lecturer in Computer Science

COMP16321
Introduction to Programming 1

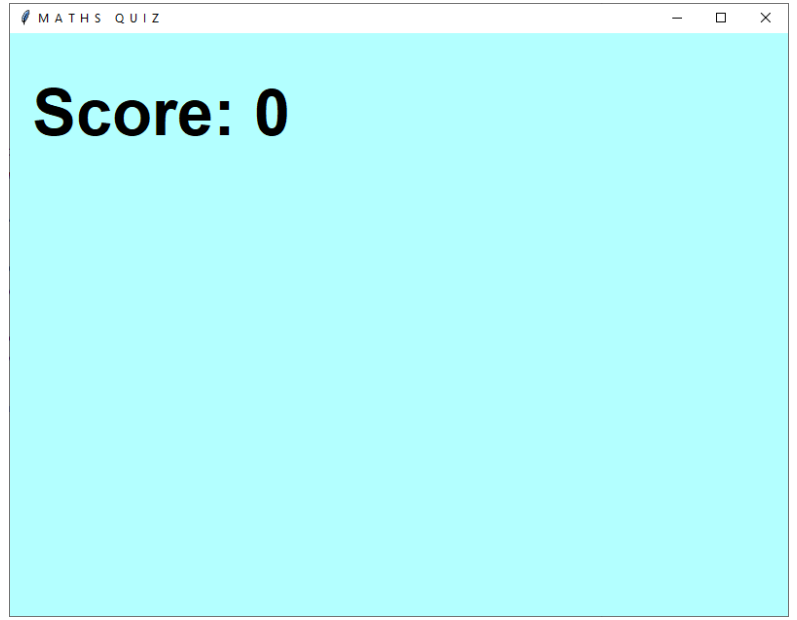
Start of Next Section

Hope that you enjoyed your break (and your tea).
Let's pick up where we left off.

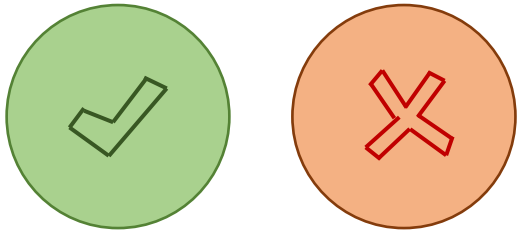
Dr Stewart Blakeway
Lecturer in Computer Science

COMP16321
Introduction to Programming 1

The Global Stuff



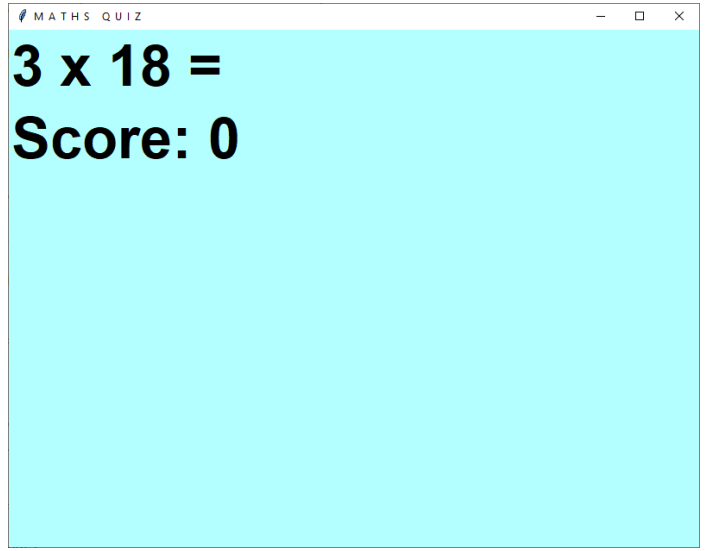

```
user_answer = IntVar()
btn = None
current_score = 0
score = Label(window, text="Score: 0", font=("Arial
Bold",50), background='#b3ffff')
score.grid(column=2, row=8, columnspan=4, sticky="E")
correct=PhotoImage(file="correct.png")
incorrect=PhotoImage(file="incorrect.png")
```



Next we create a game loop

```
def game_loop():
    num1 = rand(1,20)
    num2 = rand(1,20)
    operator = rand(0,2)
    display_question(num1, num2, operator)
```

Then display the question




```
def display_question(num1, num2, operator)
  operators = ["+", "x", "-"]
  question = str(num1) + " " +
   operators[operator] + " " + str(num2) + " ="
  question = Label(window, text=question,
   font=("Arial Bold",50), background='#b3ffff')
  question.grid(column=1, row=1, columnspan=6,
   sticky="W")
```

Back to the game loop

```
def game_loop():
  num1 = rand(1,20)
  num2 = rand(1,20)
  operator = rand(0,2)
  display_question(num1, num2, operator)
  answer = calculate_answer(num1, num2, operator)
```

Calculate the Answer

```
def calculate_answer(num1, num2, operator):
    if operator == 0:
        return num1 + num2
    elif operator == 1:
        return num1 * num2
    else:
        return num1 - num2
```

Back to the game loop

```
def game_loop():
  num1 = rand(1,20)
  num2 = rand(1,20)
  operator = rand(0,2)
  display_question(num1, num2, operator)
  answer = calculate_answer(num1, num2, operator)
  answers = add_incorrect_answers(answer)
```

Generate Some Incorrect Answers

```
answers = []
answers.append(answer)
for i in range(4):
      while True:
            r = rand(-6, 6)
            ans = answer + r
            if ans not in answers:
                  answers.append(ans)
                  break
return answers
```

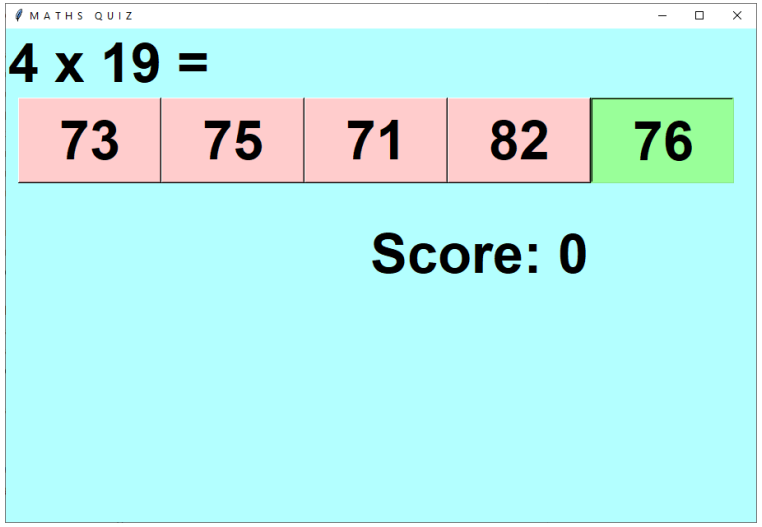
Back to the game loop

```
def game_loop():
  num1 = rand(1,20)
  num2 = rand(1,20)
  operator = rand(0,2)
  display_question(num1, num2, operator)
  answer = calculate_answer(num1, num2, operator)
  answers = add_incorrect_answers(answer)
  create_answer_radiobuttons(answers)
```

create_answer_radiobuttons

```
def create_answer_radiobuttons(answers):
    ans = []
    for i in range(len(answers)):
        ans.append(Radiobutton(window, text=answers[i],
                               value=answers[i],
                               font=("Arial Bold",50),
                               indicatoron=0,
                               variable=user_answer,
                               bg='#ffcccc’,
                               selectcolor='#99ff99’,
                               width=4))
...
```

create_answer_radiobuttons




```
def create_answer_radiobuttons(answers):
    ...
    shuffle(ans)
    padding = Label(window, text=" ", font=("Arial Bold",25),
                    background='#b3ffff')
    padding.grid(column=1, row=3)
    for i in range(5):
        ans[i].grid(column=i+2, row=3)
    padding.grid(column=1, row=4)
```

End of Section

Don’t take a break!

Stewart messed up big time! No it wasn't OBS – It was his code ☺

Do you know where he messed up? Try to figure it out before continuing.
Message on the forum if you worked it out. Just don’t tell his boss!

Dr Stewart Blakeway
Lecturer in Computer Science

Start of Next Section

I knew it was an infinite loop! Somewhere…

Dr Stewart Blakeway
Lecturer in Computer Science

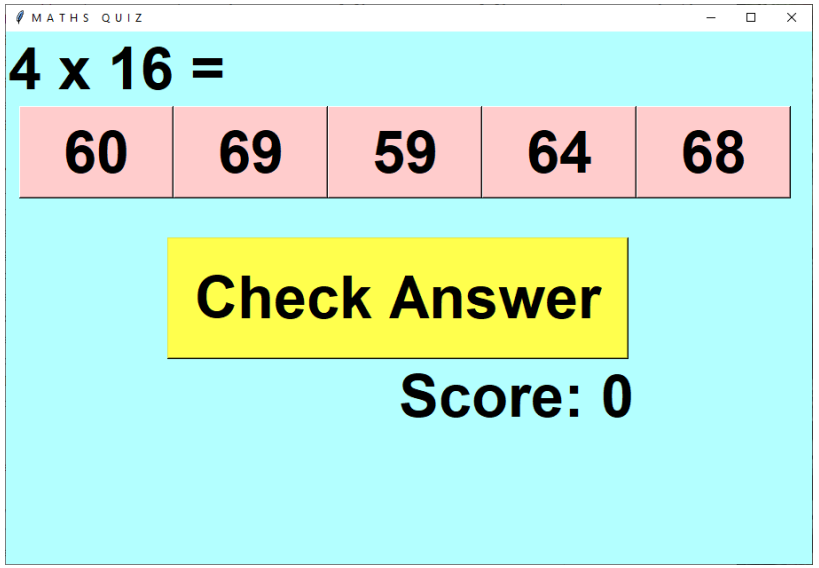
COMP16321
Introduction to Programming 1

Back to the game loop

```
def game_loop():
    num1 = rand(1,20)
    num2 = rand(1,20)
    operator = rand(0,2)
    display_question(num1, num2, operator)
    answer = calculate_answer(num1, num2, operator)
    answers = add_incorrect_answers(answer)
    create_answer_button(answer)
```

create_answer_button

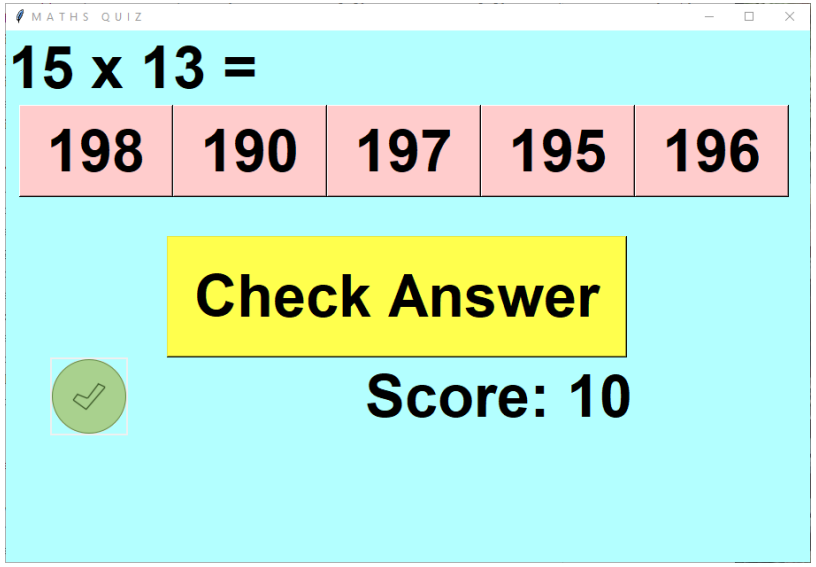
```
def create_answer_button(answer):
    global btn
    btn = Button(window, text="Check Answer",
                 command=lambda:clicked(answer),
                 background='#ffff4d', font=("Arial Bold",50) )
    btn.grid(column=0, row=5, columnspan = 7)
```



clicked

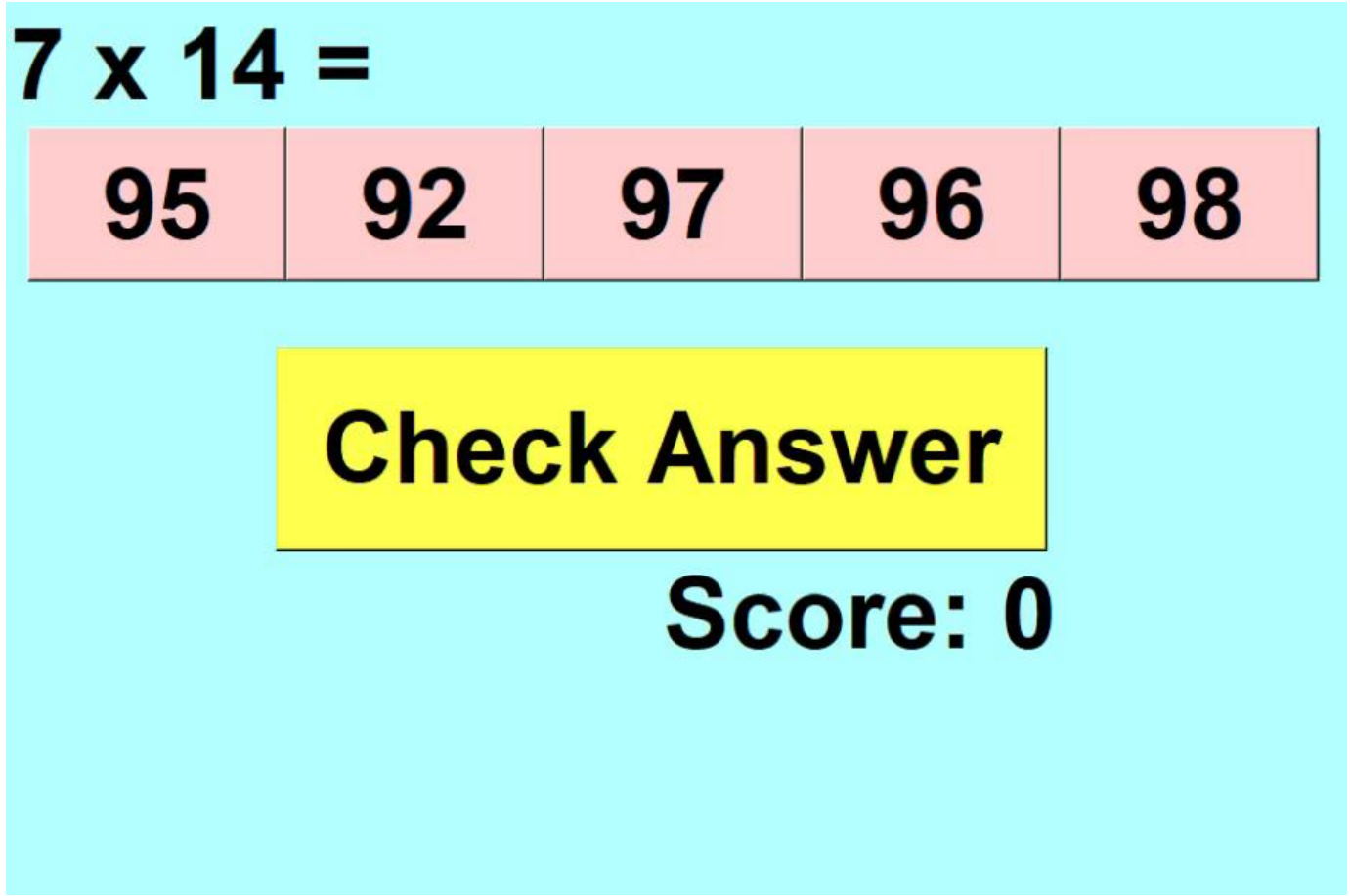
```
def clicked(answer):
    if (user_answer.get() == answer):
        btn.configure(text=" Correct ")
        ans_label = Label(window,image=correct)
        global current_score
        current_score += 10
        score.config(text="Score: "+
                     str(current_score))
...
```

clicked




```
...
    else:
        ans_label = Label(window,image=incorrect)
        btn.configure(text="Incorrect")
    ans_label.grid(column=1, row=8, columnspan=2)
    window.after(1000, game_loop)
```

The final Game



Summary

- Introduced the tkinter library
- Created a simple maths game consisting of a Tk root object (the window)
  - The game will generate a random question, i.e. 3 x 20 =
  - The game will calculate the correct answer and four incorrect answers
  - The user selected an answer and clicked a button to check
  - If the user was correct the score will increased
- Add various widgets to the window
  - Labels
  - Radio Button
  - Standard Button
  - An image

We will build on our knowledge of tkinter in the labs where you create a simple OXO game.

In the live coding session we will create a simple Simon memory game.