

As per the assignment I have created Seven python files named **Q2.py** and **Q3.py**.
This report contains a detailed explanation of my code.

Description of Q2.py :

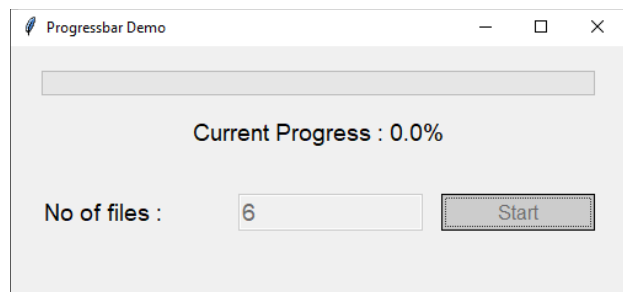
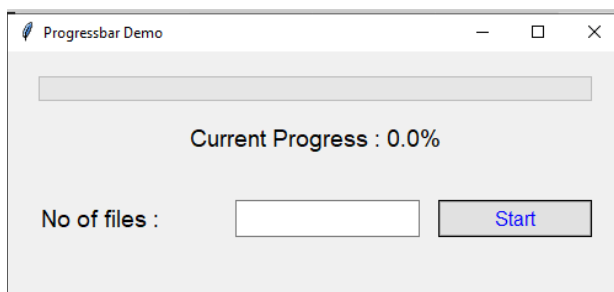
The **Q2.py** file contains one method named **start()**.

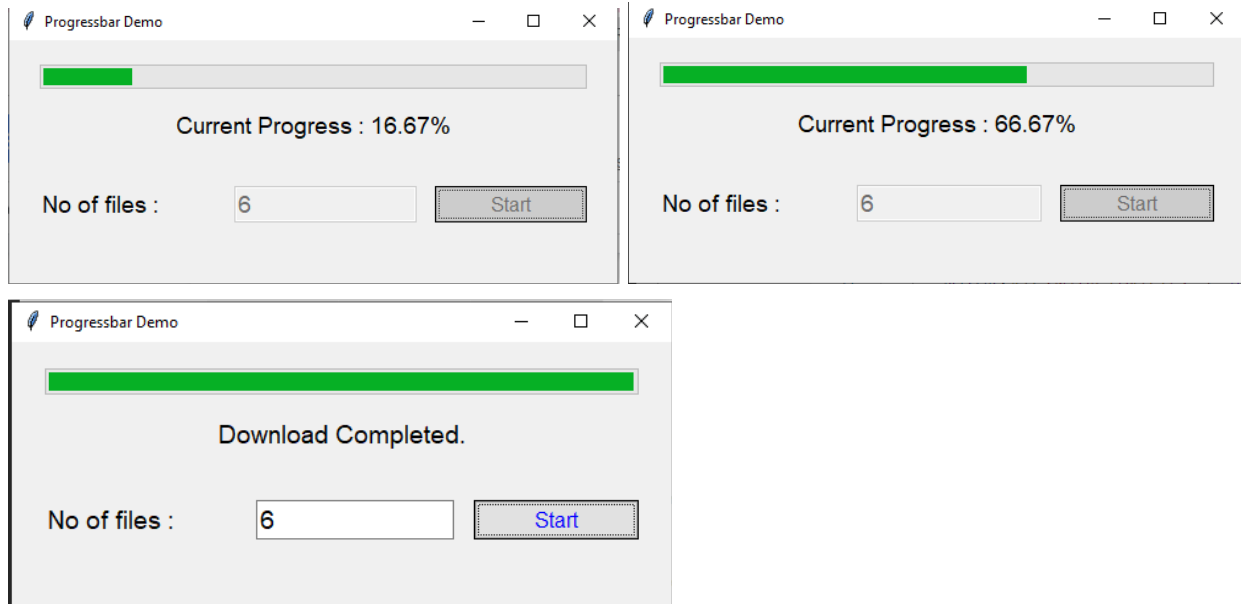
- In Q2.py , I created a progressbar with python tkinter .
- A window is made using the TK() function of tkinter .
- Window title is set using the title() method .
- Window size is set using the geometry() method.
- A progress bar is created using the Progressbar() method from tkinter.ttk .
- Progressbar is placed horizontally and in determined mode .
- Took string variable percent to output progress percentage .
- Took string variable no_of_files_text to create a label denoting the no of files to be downloaded before an entry box.
- Created labels to show the texts of percent and no_of_files_text .
- Created an entry to take input from the user that how many files to be downloaded .
- Created a button named start , to start the process . start() method is invoked on start button click to progress the progressbar .
- Placed all things in the correct place using the place() method .

start() method :

- As the start button clicked , the percent variable is set to "Current Progress : 0.0%".
- no_of_files is captured using get() method on no_of_files_entry .
- Disable the state of button and no_of_files_entry field .
- For each start() method invoked, set the bar value to 0.
- Update the window .
- Take a loop to check when all file downloads complete .
- Sleep for random amount of time , which denotes time required for file download
- Update the bar value with the correct percentage of downloads done .
- Set percent variable with correct percentage value .
- Update window to see changes on screen
- When all files are downloaded , break the loop
- Set the percentage variable to "Download Completed ." , after the downloads finished.
- Set the state of the button and no_of_files_entry to normal .

Test Cases outputs:





Description of Q3.py :

The Q3.py file contains one class named **BarSquareGame()**.

- In Q3.py , I created a **BarSquareGame** with python tkinter.
- The BarSquareGame class contains various methods .

__init__(self) method:

- **__init__(self)** invoked as the BarSquareGame class object is created .
- A window to hold the canvas created using Tk() method .
- A canvas to draw rectangles is created using Canvas() method.
- A bar and a box is created in canvas using create_rectangle() method .
- The bar and box is placed at appropriate position using move() method
- The canvas is packed to ensure all drawings are shown properly.
- For starting the initial freefall I used the after() method.
- Bind moveBar() functions with the bar rectangle which works on button click and drag .
moveBar() method is used to move the bar on canvas .
- Bind binder() to box which works on button click on box. binder() method is used to bind the moveBox() method with the box.
- Bind freefall_buttonrelease() with canvas on button release to freefall the box on button release.
- Bind freefall() with canvas on <<collision>> , a custom event to freefall the box on collision.
- stop_box denotes whether the state of the box is stopped or not.
- ffv denotes whether the box is free falling or not.
- ii is used to track initial freefall.
- fall is used to track whether the box is falling or not .
- bc , lc , rc denotes bottom . left and right collision
- At the end , start the mainloop for the window .

binder(self, event) method:

- binder() is the function for binding mouse movement with the box.
- Binds box with moveBox function using tag_bind() method.

top_collision(self) method:

- top_collision() function is used for finding top collision.
- Find box and bar position which will be used to find collisions.
- Find the top collision and stop the box.

moveBox(self ,event) method:

- moveBox() function responsible for box movement in canvas .
- Change ii to 1 when button click on box happens , it ends the initial freefall.
- Know the box and bar position using coords()method .
- If the mouse position is in the canvas and it doesn't overlap the bar , the moveBox() method moves the bar accordingly .
- Find the right collision and set rc value to True. And generate collision events if the box is already not falling. Otherwise don't generate collision events for right collision and increment ffv . collision event will handle the free fall on collision property of the box.
- Find the left collision and set lc value to True. And generate collision events if the box is already not falling. Otherwise don't generate collision events for left collision and increment ffv . collision event will handle the free fall on collision property of the box.
- Find the bottom collision and set bc value to True. And generate collision events if the box is already not falling. Otherwise don't generate collision events for bottom collision and increment ffv . collision event will handle the free fall on collision property of the box.

moveBar(self ,event) method:

- moveBar() function responsible for bar movement in canvas .
- bar movement possible only within window
- If the bar does not collide with the box , place the bar at the appropriate location with the mouse click event location .
- If the bar stopped the box previously and now it is moved to different location , freefall the box by generating collision event
- set stop_box to false to ensure the box is not in stop state so that the freefall can happen correctly .
- Generate collision event to freefall the box when bar is moved in such a place that is does not block the box's freefall.

Initial_freefall(self) method:

- initial_freefall() function responsible for initial free fall of box in canvas .
- Set ii to 0 , ii denotes the initial freefall , it is used to check when the initial freefall breaks.
- Find box position using coords() method.
- Loop through , while the box does not touch maximum y boundary fall the box or any special case happens when the loop breaks.

- If the top collision happens, stop the box above the bar and break the loop.
- Break the loop if ii is incremented or box stopped.
- Change box position by incrementing its y value only in loop to give the visual effect of box freefalling.
- Sleep time is used for a smooth freefall.
- update changes in the window using update() method.

freefall(self , event) method:

- freefall() function responsible for free fall of box in canvas when collision happens .
- set ffv to 1 , ffv denotes the freefall state , it is used to check when the freefall breaks.
- set the fall to true , fall= true denotes the box is in the falling state.
- Find box position using coords() method.
- Loop through , while the box does not touch maximum y boundary fall the box or any special case happens when the loop breaks.
- Change box position by incrementing its y value only in loop to give the visual effect of box freefalling.
- Sleep time is used for a smooth freefall.
- update changes in the window using update() method.
- Break the loop if ffv incremented or box stopped and set fall to false , as the box stopped and break
- When the loop ends , set fall to false .

freefall_buttonrelease(self , event) method:

- freefall_buttonrelease() function responsible for free fall of the box in canvas when button release happens .
- Only if the box is not already on falling state freefall on button release. This is checked by using the fall variable .
- set ffv to 1 , ffv denotes the freefall state , it is used to check when the freefall breaks.
- Find box position using coords() method.
- Loop through , while the box does not touch maximum y boundary fall the box or any special case happens when the loop breaks.
- If top collision happens, stop the box above the bar and break the loop
- Break the loop if ffv is incremented or box stopped.
- Change box position by incrementing its y value only in loop to give the visual effect of box freefalling.
- Sleep time is used for a smooth freefall.
- update changes in the window using update() method.
- Break the loop if ffv incremented or box stopped and set fall to false , as the box stopped and break

Test Cases outputs:

Please run Q3.py to see the output and interactivensess of the box and bar . I have attached some screenshots below.

