

CS 112 Project Assignment: Whack-a-Prof

Overview

The purpose of this project is to gain experience using classes and the Python graphical user interface (GUI) toolkit -- Tkinter.

In the project you will use Python to create a small video game inspired by the Whack-A-Mole game. <http://en.wikipedia.org/wiki/Whack-a-mole>. In this game pictures will appear at random locations in the window. As they appear you must “whack” them by clicking on the picture using the mouse. The faster you click the higher you will score in the game.

This project is due 5/4/2009 at 11:55PM for everyone in CS112. Programming projects are considered individual efforts, therefore no sharing of code and/or discussion of algorithms or problem solution is allowed with anyone except lab GTA, UTA or instructor.

For this project you may also use resources from the following websites in addition to textbook and Blackboard discussion groups:

<http://www.ffmpeg.org/>

<http://www.pythonware.com/library/tkinter/introduction/index.htm>

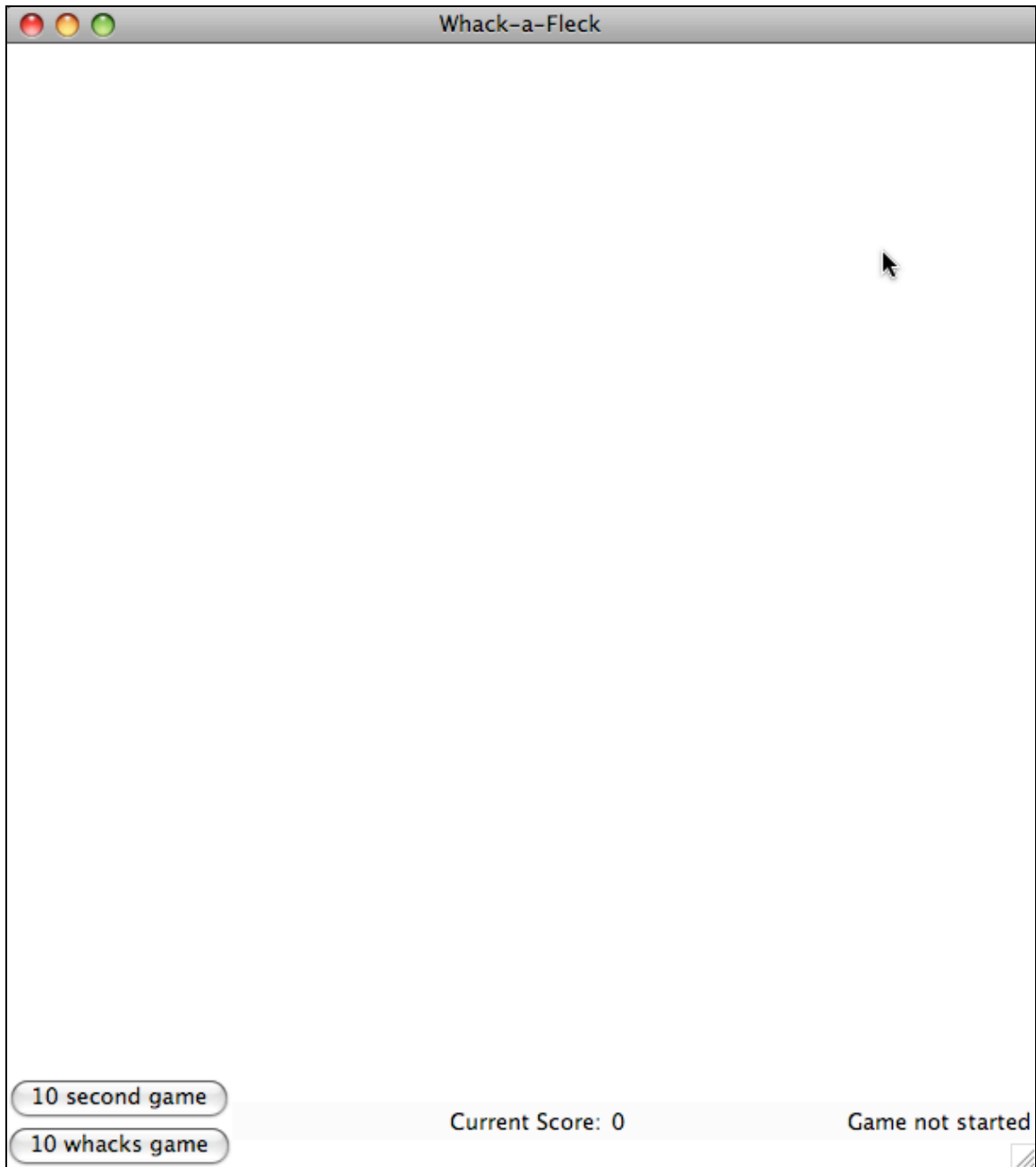
Requirements

- The source code for this project must be submitted in two files:
 - **Target.py** --- this module defines the Target class.
 - **WhackAProf.py** – this module uses the Target.py class to create and manage the game. You will need to use functions within in this class to achieve the goal.
- The source code file must contain a **file header** formatted as previously specified.
- The source code file should use **self-documenting code** and additional comments (as required) to improve code readability.
- **Picture file:** You must submit the picture you used so that your code runs when we download it!

Procedure

Screen Layout

Your game screen should be laid out exactly as shown below. Note: You can change the title to “Whack-a-<<whoever you want>>”



The canvas should be 600 pixels wide by 600 pixels tall. Then the bottom panel (which holds the buttons and text) can be the default size automatically created by Python.

Game Play

The game can be played in two different modes. In one mode you attempt to get 10 points in the shortest possible time, and in another mode you attempt to get the best score you can within 10 seconds.

10 Second game – When the user clicks the “10 second game” button, the game should begin. The 10 second game should have a timer at the bottom right (which replaces the “Game not started” label and counts to 10, updating each second.

While the counter is counting, a picture should randomly appear in the top portion of the screen (on the canvas). Everytime the picture is clicked using the mouse, the score should increase by one and the picture should reappear somewhere else on the screen.

After 10 seconds the game ends:

- the current picture should automatically disappear
- no further scoring should be possible
- the text in the bottom right should change to “GAME OVER Time: 10”

10 Whacks game – When the user clicks the “10 whacks game” button, the game should begin. A timer should appear at the bottom right and increment every second. A picture should randomly appear in the top portion of the screen (on the canvas). Everytime the picture is clicked using the mouse the score should increase by one and the picture should reappear somewhere else on the screen.

After the player scores 10 points the game ends:

- the current picture should automatically disappear
- no further scoring should be possible
- the text in the bottom right should change to “GAME OVER Time: XX”
(where XX is the time it took to score 10 points)

10 second game

Current Score: 10

GAME OVER Time:12

10 whacks game

File: If you want to whack Prof. Fleck, here is a picture:

<http://cs.gmu.edu/~dfleck/classes/cs112/spring09/projects/whackaprof/fleck.gif>

Coding Requirements

For this project you MUST use at least one class for the Target.

Target class – this represents the thing you try to click “the prof”.

- It should have attributes for x,y location, the current image
- It should have at least a method to determine if the target was hit (by the mouse click)
- You may have other methods and attributes, but must have at least those.

Other code for the project may use classes, or not... but still must use functions appropriately.

Bonus Points

If you want to get bonus points on this assignment you can add in any or all of the following requirements.

- Animate the targets so they appear to rise up from the canvas. You can do this by creating a series of pictures where each picture shows the thing a little bit more out of the ground. Do that, and put a picture of something as the background of the game. See this for some inspiration:
<http://www.addictinggames.com/whackamole.html> +7 points
-

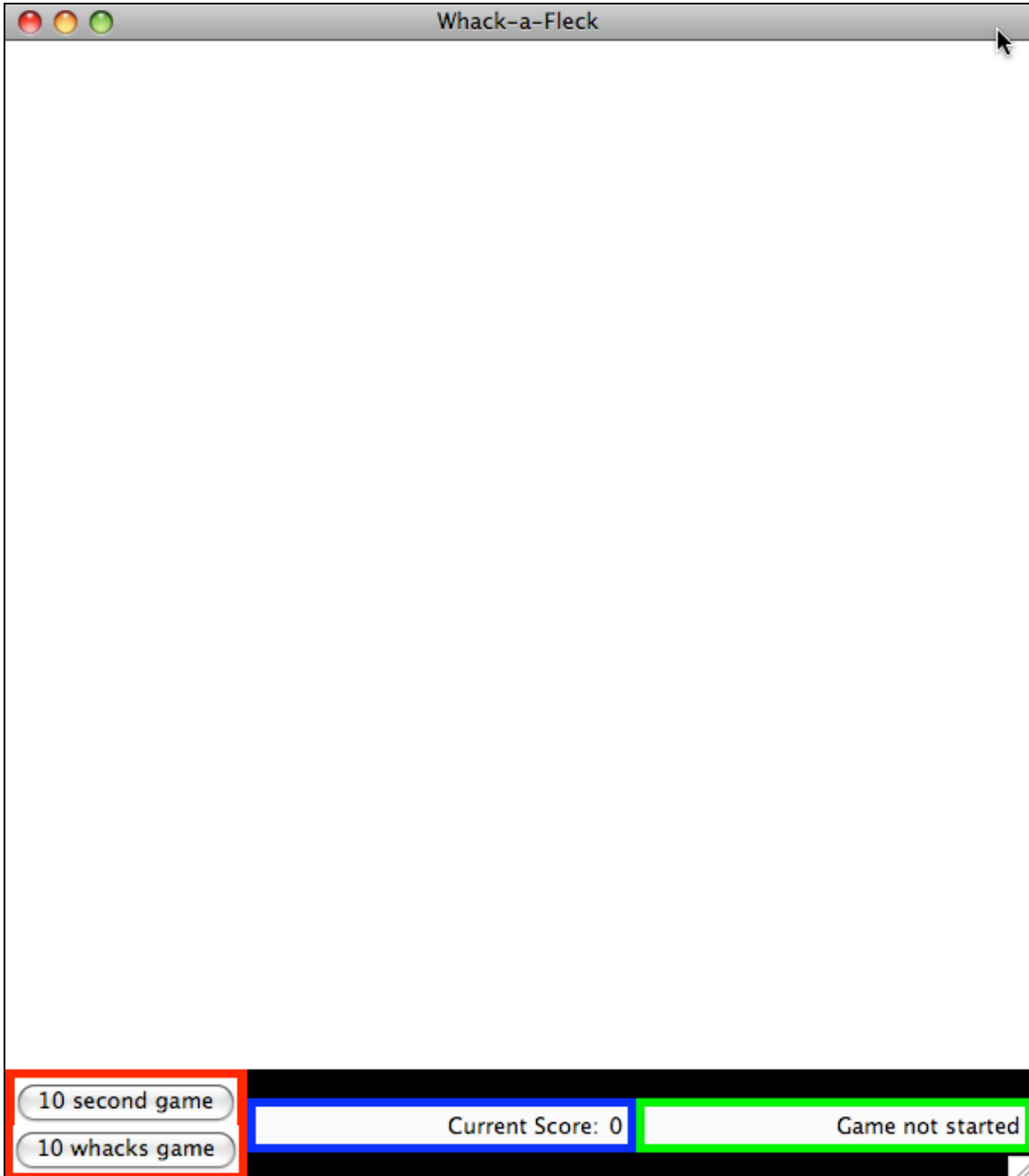
Hints

This hint is only relevant is using the “pack” LayoutManager. If you use Grid, you don’t really need these frames.

Frame layout is one of the challenging things when doing GUI layout. Because frames are invisible, to “see” them you can set a border and a background.

```
aFrame = Frame(parent,bd=5,background="red")
```

Doing this to all frames can help you understand what’s going on in your GUI. Here is a screenshot of Frames I used... you can use other layouts, but the end result should look the same as the original screen shot at the beginning of the assignment.



Grading Rubric

	Excellent (85% or higher)	Average (60% or higher)	Needs Improving (Less than 60%)	Points Possible
Submission Instructions	<ul style="list-style-type: none"> Both filename and header meet the stated specifications 	<ul style="list-style-type: none"> Either is missing or has incorrect/missing information 	<ul style="list-style-type: none"> Both are missing or have incorrect details 	3
GUI Layout	<ul style="list-style-type: none"> The GUI is exactly as shown in the screen shot 	<ul style="list-style-type: none"> The GUI has all required components, but layout is not correct. 	<ul style="list-style-type: none"> The GUI is missing components 	15
Use of functions	<ul style="list-style-type: none"> Functions are used appropriately Comments are meaningful and professional. Code is clean and easy to read. Uses meaningful variable names 	<ul style="list-style-type: none"> Functions are used, but are not clear Comments are not helpful for understanding the code Variable names are not meaningful 	<ul style="list-style-type: none"> Code is written without appropriate use of functions Code is very difficult to read and understand due to poor coding style (missing comments/self-documenting code) 	4
Use of classes	<ul style="list-style-type: none"> The Target is a class with at least the specified methods and attributes. 	<ul style="list-style-type: none"> The Target is a class, but is missing attributes or methods needed. 	<ul style="list-style-type: none"> Target is not a class or used as a class. 	9
Game Play	Both game modes work as specified.	The game modes are both present, but do not function appropriately.	Game modes are missing or at least one is non-functional.	9
Bonus Points	Targets animate rising out of the screen and there is a screen background	None given	None given	(+7)
Final Score				40 (+7)