SUT specification (Homework)

The SUT specification consists of the functionality of each tab that should be tested. There are 18 tests in total. Each bullet point represents a test. General info should not be tested, however, should be taken into account for better understanding when testing each bullet point.

For the report make sure you name your tests and classes according to each bullet point. >>>> Otherwise you will get 0 points! <<<< For example, the first class will be “Folders” and the first test there “test_apps_delete”.

Tab 1 (Folders):
General:
- There are 3 folders with different colors. Each color represents the folders’ capacity. Blue should be able to hold 2 apps, orange - 0 and green - 1.
- Right-clicking an app or a folder opens a small context menu.

Functionality to test:
- Dragging an icon to the folder makes it disappear from the screen. ["test_apps_delete"]
- Dragging a folder over a folder doesn’t make anything disappear. The dragged folder should appear to be on top of another folder. ["test_move_folders"]
- The context menu should always stay within the borders of the “toy” application. (Test each border). ["test_menu"]

Tab 2 (Calculator):
Functionality to test:
- The black output rectangle must be empty when nothing was typed. When something was typed, the rectangle must show everything that was typed. The rectangle must display 20 characters at the most. ["test_calc_black"]
- The purple output rectangle must show “0” when nothing was typed. When something is typed, the rectangle must output the answer (e.g. typing “2+2” outputs “4”). When an expression is not full (e.g. incomplete expression “2*3+” or empty/erased) rectangle must output the last answer (e.g in the case of expression “2*3+” the answer is “6”). ["test_calc_purple"]
- When typing an expression that is mathematically impossible, the purple output rectangle must show “Error”. ["test_calc_err"]
Tab 3 (PinCode):
Functionality to test:

- For each number pressed (max is 4), an “X” should appear in the output rectangle. ["test_pin_x"]
- When 4 numbers are pressed and the password is correct (“1234”), the pin code system should be replaced by an empty screen with the message “Welcome, *user*!” (where *user* is a random name). ["test_pin_unlock"]
- When 4 numbers are pressed and the password is wrong, an icon or closed lock appears on top of the output rectangle. ["test_pin_lock"]

Tab 4 (Converter):
General:

- The converter consists of 2 text fields: regular text and unicode.
- When text is typed in the left field it is converted and displayed in the right field.
- Regular text and unicode can be swapped places by pressing the “<->” button.

Functionality to test:

- The regular text should be translated to Unicode in real-time. ["test_text2uni"]
  
  Note: The only thing you need to test is if it looks like a Unicode. I.e. there are only numbers showing. There is no need to test if the conversion is correct.

- You should always be able to type in the left field, but not in the right field. ["test_fields"]
- When you swap the fields labeled “Unicode” and “Text” by pressing the swap arrow (“<->”), then the swap arrow should stay in the same place. In addition, the input and output text fields should not change their position and shape. ["test_switch"]

  Typing in the unicode field should translate int sequences to regular text. When a non-integer is typed, the field labeled “Text” should display the text “Error! Not unicode”. ["test_uni2text"]

Tab 5 (CatFlower):
General:

- On the right of the scene, there is a “field of cats”.
- The width of the “field of cats” is 125 and the height is the same as the tab’s height. It is always located on the right border of the tab.
- Currently, the game has 3 “flower lines”. This number might be changed later by the developers.

Functionality to test:
• When you click on a cat in the “field of cats” and then click on the rock tile (grey square on the horizontal green line) or the green line where the tile is, the chosen cat should be placed there. ["test_place_cat"]

• Cats placed on the “flower line” move equal distance each second. ["test_cat_moves"]

• When the walking cat touches a flower, the flower should disappear from the screen. ["test_flower_gone"]

• Only when all “flower lines” are filled with at least one cat, a confirmation (checkmark) should appear at the top of the screen. ["test_cats_win"]

• There are always 2 cats at the “field of cats” (a cat that is put on the line should be immediately replaced by another cat in the same place). ["test_cat_backup"]

Note: only cats can appear at the ‘field of cats” and there are only 3 types of cats.