Australian National University

ASSIGNMENT COVER SHEET

ANU College of Engineering and Computer Science
Australian National University
Canberra ACT 0200 Australia
www.anu.edu.au

+61 2 6125 5254

Student ID	U6611178		
Name	Yanming Feng (Danny)		
Course Code & Name	COMP1100 - Programming as Problem Solving		
Assignment Topic	Technical Report for Assignment 1		
Lab Time	Monday 16:00 – 18:00		
Tutor	Jack Kelly <jack@jackkelly.name>		
Word count	830	Due Date	20 April 2018
Last Edited	12 April 2018	Extension Granted	

I declare that this work:

- ☐ upholds the principles of academic integrity, as defined in the ANU Policy: Code of Practice for Student Academic Integrity;
- ☐ is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the course outline and/or Wattle site;
- ☐ is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;
- ☐ gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;
- ☐ in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling.

Initials YMF

For group assignments, each student must initial.

Technical Report for Assignment 1

Yanming Feng

1. Program Introduction

This Assignment is to use Haskell and CodeWorld API to let the user draw shape including rectangle, ellipse, polygon and line by typing or Clicking. User can change the colour of their shape and custom the position before they draw. If user what to draw shape by typing, run “cabal run TypeToDraw” command and follow the instruction in it. For drawing by mouse, user should run “cabal run ClickToDraw”, and then type the initial character of the shape and the colour they what to draw. Left click the mouse, drag and release to draw. For polygon, left click the mouse to create the vertex and press space bar to draw.

2. Functions Objectives

1. Function I create

changeTool in *State.hs* : This function I made is called by *handleEvent* in *Event.hs*. It passes the previous state and the shapetool the user wants to change. Then it gives back the new state with new tool but remain other parts unchanged.

passColour in *State.hs* :This function I made is called by *handleEvent* in *Event.hs*. It passes the previous state and the colour the user wants to change. Then it gives back the new state with new colour but remain other parts unchanged.

removeShape in *State.hs* : This function I made is use to remove the first element in the graphic list. It is called by *handleEvent* in *Event.hs* when user press backspace. It can remove the last shape the user drawed.

2. The Important Function I edited

In *View.hs*

shapeToPic : convert the abstract shapes into the picture in CodeWorld API so that it can be printed.

graphicToPic :convert the costum data type graphic to picture in CodeWorld API so that the shape can be actually draw with color and position.

get…Graphic : take the point for position and colour of a shape. Make a “graphic”.

drawNewGraphic : This function is called when mouse release. It take the state and point that finished in State.hs and add it to the graph list

In *Event.hs*

handleEvent : use case to pattern match the mouse or key event (user input). And then run corresponding function.

3. Issues Encountered and Solutions

1. Syntax error. I was suffering from serval syntax errors especially at the beginning for the assignment. And this is the most painful thing thought the assignment. For example, I don't know the difference of upper case and lower case at first. I also don't know the important of putting brackets. I often got the idea but when I wrote it out, a lot of red line were shown on the screen. These are all the Haskell basis. So I asked my friend and tutor for help. I read the craft of functional programming to learn the foundation. What's more, practice makes perfect. I just keep fixing my code again and again.

2. Don't know how to change part of the state (like colour) and remain other parts unchanged. I asked my friend for ideas and he gave me the piazza link https://piazza.com/class/jd9j45dtklj4az?cid=447 Jack Kelly's example inside is really helpful.

4. Major Design Choice and Assumptions

Most of the time I just followed the assignment instructions. My design choice is bottom-up. I created some small function to solve part of the problem. Then called them in other function to rise to more complex things.

5. Improvement

I should make my code more concise. Use 'where' clause to simplify things. Besides, I should make my variable symbol clearer. For example, I sometimes use p for points and sometimes I use (x2,y2) for points. I should write more comments to explain what I'm doing in my code.

6. Test Result

I finished all part A and part B. I run the program to see if it works as required. Also, press 'D' to trace the state is helpful to debug the ClickToDraw in part B. I also use doctest to test my code. I have fixed a little bug in part A when I'm doing part B. My code now passed all the tests above.

```
Dannys-MacBook-Pro:src dannyfeng$ doctest ClickToDraw.hs
Examples: 63  Tried: 63  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest ColourName.hs
Examples: 3  Tried: 3  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest Events.hs
Examples: 63  Tried: 63  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest Graphic.hs
Examples: 3  Tried: 3  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest Shape.hs
Examples: 0  Tried: 0  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest State.hs
Examples: 3  Tried: 3  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest TypeToDraw.hs
Examples: 5  Tried: 5  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest View.hs
Examples: 5  Tried: 5  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest Warmup.hs
Examples: 0  Tried: 0  Errors: 0  Failures: 0
Dannys-MacBook-Pro:src dannyfeng$ doctest WarmupInteraction.h
Examples: 2  Tried: 2  Errors: 0  Failures: 0
```

Figure 1. doctest result