Technical Report
COMP1100 Assignment 1

Jacob Bos
ANU u7469354

March 31, 2022

Introduction

This Technical report documents the structure of the assignment solution and offers a reflective analysis of design choices made and the results and structure of the testing regeme. The Documentation section will purely describe the design of the solution. The analysis section will asses the design choices made and the specifications and assumptions that led to them. Finally the Testing section documents the structure of testing.

Documentation

Part 1 consists of three functions. The first function of Part 1 `toolToLabel` in `model View` case matches an input of a tool, ignoring the other tool properties to return a string output of instructions to the user on tool use. The second function `nextColor` in `module Controller` uses case matching to cycle through colours according to the following order

```
 Black  -> Red -> Orange -> Yellow -> Green -> Blue -> Purple -> White -> Black
```

Finally, the functon `nextTool` in `module Controller` cycles a particular imput of an empty tool to the next tool in the sequence also with empty parameters. It also uses case matching.

Part 2 Contains four functions the first of which, `colourNameToColour` in `module View` which case matches elements of the type `ColourName` and returns the same colour in the type of `Colour` which codeworld can interpret. The second function `shapeToPicture` takes the information kept within the type `Shape` and converts it to a codeworld `Picture` that can be printed to the display. Where most inputs were case matched to equivilent the specifications of `Rectangle` were through some vector algebra converted to a `solidPolygon` of four points. The `Cap` is a combination of the codeworld `clipped` and `circle` functions transposed as the user specifies. The third function `colourShapeToPicture` takes input of the type `colourShape` and returns the associated shape with the given colour in the type `Picture`. Finally, the function `colourShapesToPicture` recursivly runs through an input of type `[Shape]` and returns each member composed together as a composite `Picture`. It returns an error message for unexpected inputs.

Part 3

Reflection

Testing

Part 1 Part 1 composed of the functions `toolToLabel` `nextTool` and `nextColour` was tested using the provided cabal test run under the command `cabal v2-test`. It passed `1 of 1 test suites`. Further simple tests were condcuted within development based on calling function inputs in the terminal to ensure the case matching was working in correspondence to the intended inputs.

Part 2

Part 3