VG100 — Introduction to Engineering

Project 1 Report (Team 6)

Rubric

- Game Design (10 pts)
- Code Quality (50 pts)
- Readme (15 pts)
- Personal work (20 pts)

1 Game Design

Not included in this report.

2 Code Quality

Your total score of this part is 44/50.

All related information is listed below:

1 point(s) deduction, useless let in, in file View.elm, lines 129-130.

```
129         list = if model.active > 0 then [ first , second ]
130               else [ first ]
```

2 point(s) deduction, duplicate code and usage of consinuous if, in file View.elm, lines 149-215.

```
149         List.map (\ball -> if ball.dx > 8
150                           then renderImage 200 200 "./assets/shadow_right.png"
151                               ( ball.x + 200 ) ( ball.y ) 1 (tanBall ball + 180)
152                           else if ball.dx == 1
153                           then renderImage 200 200 "./assets/shadow_right.png"
154                               ( ball.x + 180 ) ( ball.y - 20 ) 1 (tanBall ball + 180)
155                           else if ball.dx == 2
156                           then renderImage 200 200 "./assets/shadow_right.png"
157                               ( ball.x + 160 ) ( ball.y - 40 ) 1 (tanBall ball + 180)
158                           else if ball.dx == 3
```

...

```
206                               renderImage 200 200 "./assets/shadow_left.png"
207                               ( ball.x - 32 ) ( ball.y + 13 * ball.dy ) 1 (tanBall ball)
208                           else
209                               if ball.x > 0
210                               then
211                                   renderImage 200 200 "./assets/shadow_left.png"
212                                   ( ball.x - 30 * ball.dx ) ( ball.y + 40 * ball.dy ) 1 (tanBall ball)
213                               else
214                                   renderImage 200 200 "./assets/shadow_right.png"
215                                   ( ball.x + 200 ) ( ball.y ) 1 (tanBall ball + 180)
```

1 point(s) deduction, duplicate code and usage of consinuous if, in file View.elm, lines 312-404.

```
312         [ style "color" "#bdc3c7"
313         , style "font-weight" "300"
314         , style "line-height" "1"
315         , style "margin" "30px 0 0"
316         ]
317         [ text txt ]
318
319 renderNum : Int -> Html Msg
320 renderNum n =
321     div
```

...

```
395                 _ ->
396                     ("", Noop)
397     in
398     button
399         [ class "button_choice"
400         , style "left" "180px"
401         , style "background-position-x" "right"
402         , onClick msg
403         ]
404         [ text str ]
```

1 point(s) deduction, duplicate code and usage of consinuous if, in file Subscriptions.elm, lines 35-62.

```
35         "E" ->
36             Decode.succeed ( EasterEgg "E" )
37         "X" ->
38             Decode.succeed ( EasterEgg "X" )
39         "P" ->
40             Decode.succeed ( EasterEgg "P" )
41         "D" ->
42             Decode.succeed ( EasterEgg "D" )
43         "H" ->
44             Decode.succeed ( EasterEgg "H" )
```

...

```
53         "O" ->
54             Decode.succeed ( EasterEgg "O" )
55         "N" ->
56             Decode.succeed ( EasterEgg "N" )
57         "R" ->
58             Decode.succeed ( EasterEgg "R" )
59         "A" ->
60             Decode.succeed ( EasterEgg "A" )
61         "Y" ->
62             Decode.succeed ( EasterEgg "Y" )
```

1 point(s) deduction, duplicate code and usage of consinuous if, in file Collision.elm, lines 25-48.

```
25         topHelper = List.filter (\brick -> x >= Tuple.first brick.pos && x <= Tuple.first brick.pos + 100
26                 && y >= Tuple.second brick.pos - 10 && y <= Tuple.second brick.pos )
27
28         botHelper = List.filter (\brick -> x >= Tuple.first brick.pos && x <= Tuple.first brick.pos + 100
```

```
29                  && y >= Tuple.second brick.pos + 20 && y <= Tuple.second brick.pos + 30)
30
31          sideHelperL = List.filter (\brick -> x >= Tuple.first brick.pos - 10 && x <= Tuple.first brick.pos
32                  && y >= Tuple.second brick.pos && y <= Tuple.second brick.pos + 20 )
33
34          sideHelperR = List.filter (\brick -> x >= Tuple.first brick.pos + 100 && x <= Tuple.first brick.pos + 110
35                  && y >= Tuple.second brick.pos && y <= Tuple.second brick.pos + 20 )
36
37          cornerHelperTL = List.filter (\brick -> x <= Tuple.first brick.pos && x>= Tuple.first brick.pos - 10
38                  && y >= Tuple.second brick.pos - 10 && y <= Tuple.second brick.pos
39                  && (x - Tuple.first brick.pos) ^ 2 + ( y - Tuple.second brick.pos ) ^ 2 <= 100)
40          cornerHelperTR = List.filter (\brick -> x <= Tuple.first brick.pos + 110 && x>= Tuple.first brick.pos + 100
41                  && y >= Tuple.second brick.pos - 10 && y <= Tuple.second brick.pos
42                  && (x - Tuple.first brick.pos - 100) ^ 2 + ( y - Tuple.second brick.pos ) ^ 2 <= 100)
43          cornerHelperLL = List.filter (\brick -> x <= Tuple.first brick.pos && x>= Tuple.first brick.pos - 10
44                  && y >= Tuple.second brick.pos + 20 && y <= Tuple.second brick.pos + 30
45                  && (x - Tuple.first brick.pos) ^ 2 + ( y - Tuple.second brick.pos - 20 ) ^ 2 <= 100)
46          cornerHelperLR = List.filter (\brick -> x <= Tuple.first brick.pos + 110 && x>= Tuple.first brick.pos + 100
47                  && y >= Tuple.second brick.pos + 20 && y <= Tuple.second brick.pos + 30
48                  && (x - Tuple.first brick.pos - 100 ) ^ 2 + ( y - Tuple.second brick.pos - 20 ) ^ 2 <= 100)
```

2 point(s) bonus, some documentations found.

.

2 point(s) deduction, useless zip file in repo.

3 Readme

Not included in this report.

4 Personal work

Not included in this report.