

```

import Khanr144.StartPage
import Khanr144.LearningModule
import Khanr144.PracticeModule
import Khanr144.Calculator

```

```

myShapes model =

```

```

    case model.state of

```

```

        Main -> {-This is the main starting page with the Trigflix logo and three buttons to go to the
learning, practice and graphing calculator module-}

```

```

        [ Khanr144.StartPage.myShapes model |>group
        , group

```

```

        [
            roundedRect 40 12 1
                |> filled (rgb 220 26 41)
                |> move (0,-25),
            text "Practice"
                |> bold
                |> sansserif
                |> filled white
                |> move (-21,-53)
                |> scale 0.5

```

```

        ] |> notifyTap ToPracticeModule {-This transition takes us to practice module from the
main starting page-}
        , group

```

```

        [
            roundedRect 40 12 1
                |> filled (rgb 220 26 41)
                |> move (-50,-25),
            text "Learn"
                |> bold
                |> sansserif
                |> filled white
                |> move (-115,-53)
                |> scale 0.5

```

```

        ] |> notifyTap ToLearningModule {-This transition takes us to learning module from the
main starting page-}
        , group

```

```

        [
            roundedRect 40 12 1
                |> filled (rgb 220 26 41)
                |> move (50,-25),
            text "Graphing"
                |> bold
                |> sansserif
                |> filled white
                |> move (97,-58.5)
                |> scale 0.4,
            text "Calculator"
                |> bold
                |> sansserif
                |> filled white
                |> move (95,-73)
                |> scale 0.4

```

```

        ] |> notifyTap ToCalculator {-This transition takes us to graphing calculator module
from the main starting page-}

```

```

        ]
        Learn -> {-This state is the learning module-}

```

```
[ Khanr144.LearningModule.myShapes model.l1model |> group |> GraphicSVG.map
L1MSG
  , group
    [roundedRect 20 7 2.5
      |> filled (hsl (degrees 0) 0.629 0.218)
      |> addOutline (solid 0.2) white
    , text "Back"
      |> centered
      |> size 6
      |> filled white
      |> move(0, -2)
    ]
    |> move (-62, -50)
    |> notifyTap LearnToMain {-This transition takes us from learning module to the main
start page-}
    |> if model.l1model.screen == 1 then makeTransparent 1 else (move (-130,-45))
    {-The back button will only appear if the 'screen' variable in the imported learning module
is 1 otherwise it will be shifted out of the screen-}
  ]
```

Practice -> {-This state is the practice module-}

```
[ Khanr144.PracticeModule.myShapes model.l2model |> group |> GraphicSVG.map
L2MSG
  , group
    [rect 16 9
      |> filled black
      |> move(-2,0),
      triangle 9
      |> filled black
      |> rotate (degrees 60)
      |> move (-10,0)
    ]
    |> scale 0.8
    |> move(-80,-45)
    |> notifyTap PracticeToMain {-This transition takes us from practice module to the
main start page-}
    |> if model.l2model.screen == 1 then makeTransparent 1 else (move (-130,-45))
    {-The back button will only appear if the 'screen' variable in the imported practice module
is 1 otherwise it will be shifted out of the screen-}
  ]
```

Calculator -> {-This state is the graphing calculator module-}

```
[ Khanr144.Calculator.myShapes model.l3model |> group |> GraphicSVG.map L3MSG
  , group
    [rect 16 9
      |> filled black
      |> move(-2,0),
      triangle 9
      |> filled black
      |> rotate (degrees 60)
      |> move (-10,0)
    ]
    |> scale 0.8
    |> move(-70,-45)
    |> notifyTap CalculatorToMain {-This transition takes us from graphing calculator
module to the main start page-}
    |> if model.l3model.screen == 1 then makeTransparent 1 else (move (-130,-45))
    {-The back button will only appear if the 'screen' variable in the imported graphing
calculator module is 1 otherwise it will be shifted out of the screen-}
  ]
```

{-declaring and naming all the transitions and msgs-}

type Msg = Tick Float GetKeyState

| ToLearningModule

| LearnToMain

| ToPracticeModule

| PracticeToMain

| ToCalculator

| CalculatorToMain

| L1MSG Khanr144.LearningModule.Msg {-variants created to help import other modules-}

| L2MSG Khanr144.PracticeModule.Msg

| L3MSG Khanr144.Calculator.Msg

{-declaring and naming all the states-}

type State = Main

| Learn

| Practice

| Calculator

{-updating the model using the transitions and msgs mentioned above-}

update msg model =

case msg of

Tick t _ ->

{ model | time = t }

ToLearningModule ->

case model.state of

Main ->

{ model | state = Learn }

otherwise ->

model

LearnToMain ->

case model.state of

Learn ->

{ model | state = Main }

otherwise ->

model

ToPracticeModule ->

case model.state of

Main ->

{ model | state = Practice }

otherwise ->

model

PracticeToMain ->

case model.state of

Practice ->

{ model | state = Main }

otherwise ->

model

ToCalculator ->

case model.state of

Main ->

{ model | state = Calculator }

otherwise ->

model

CalculatorToMain ->

```
case model.state of
  Calculator ->
    { model | state = Main }
```

```
otherwise ->
  model
```

```
L1MSG lmsg -> { model | l1model = Khanr144.LearningModule.update lmsg model.l1model
}
```

```
L2MSG lmsg -> { model | l2model = Khanr144.PracticeModule.update lmsg model.l2model
}
```

```
L3MSG lmsg -> { model | l3model = Khanr144.Calculator.update lmsg model.l3model
}
```

{-declaring the types for all variables-}

```
type alias Model =
```

```
{ time : Float
, state : State
, l1model : Khanr144.LearningModule.Model
, l2model : Khanr144.PracticeModule.Model
, l3model : Khanr144.Calculator.Model
}
```

{-initializing all the variables-}

```
init : Model
```

```
init = { time = 0
```

```
  , state = Main {-initial state will be the Main state-}
```

```
  {-imported modules will be initialized to their initial definitions that is done in their own
```

```
seperate module-}
```

```
  , l1model = Khanr144.LearningModule.init
```

```
  , l2model = Khanr144.PracticeModule.init
```

```
  , l3model = Khanr144.Calculator.init
```

```
}
```

```
main = gameApp Tick { model = init, view = view, update = update, title = "Game Slot" }
```

```
view model = collage 192 128 (myShapes model)
```