Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

КУРСОВАЯ РАБОТА

По дисциплине «Функциональное программирование»

Выполнил
студент гр. 3530904/80007 Пономарев С.П.

Руководитель Лукашин А.А.

16 декабря 2019 г.

Санкт-Петербург
2019

Оглавление

Введение:

Функциональное программирование — парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании).

Противопоставляется парадигме императивного программирования, которая описывает процесс вычислений как последовательное изменение состояний (в значении, подобном таковому в теории автоматов). При необходимости, в функциональном программировании вся совокупность последовательных состояний вычислительного процесса представляется явным образом, например, как список.

Функциональное программирование предполагает обходиться вычислением результатов функций от исходных данных и результатов других функций, и не предполагает явного хранения состояния программы. Соответственно, не предполагает оно и изменяемость этого состояния (в отличие от императивного, где одной из базовых концепций является переменная, хранящая своё значение и позволяющая менять его по мере выполнения алгоритма).

Описание задачи.

Реализация очередной вариации игры «змейка» применяя подходы функционального программирования. На примере языка Haskell.

Описание решения:
Весь код программы располагается в main: отрисовка окна,

Отрисовка змейки идет в функции render.
Управление идет змейкой идет с помощью keyboardMouse.

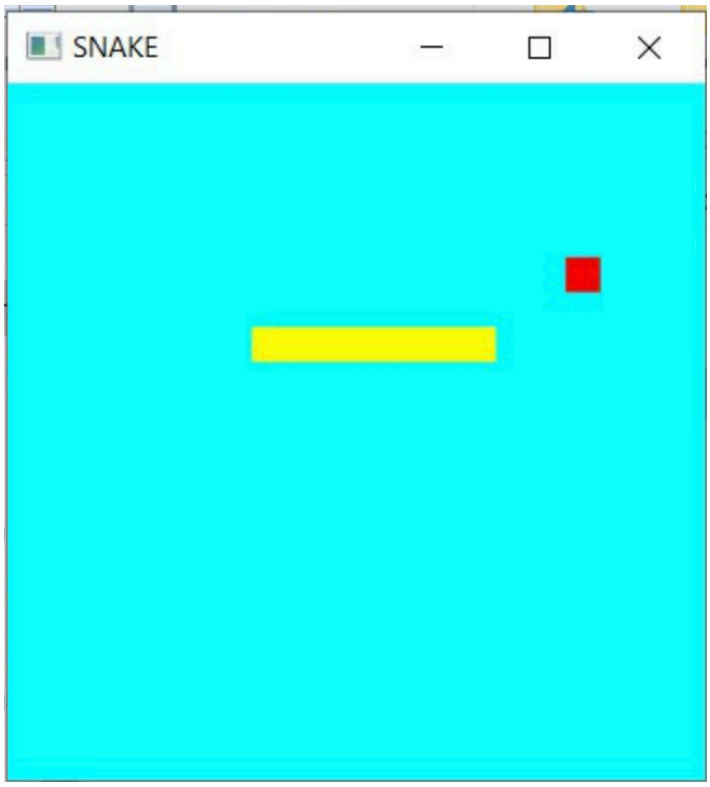
Игра идет до тех пор, пока змейка не заденет своё тело или же не соприкоснется со границей поля. Управление идет за счет клавиш: up, down, right, left.

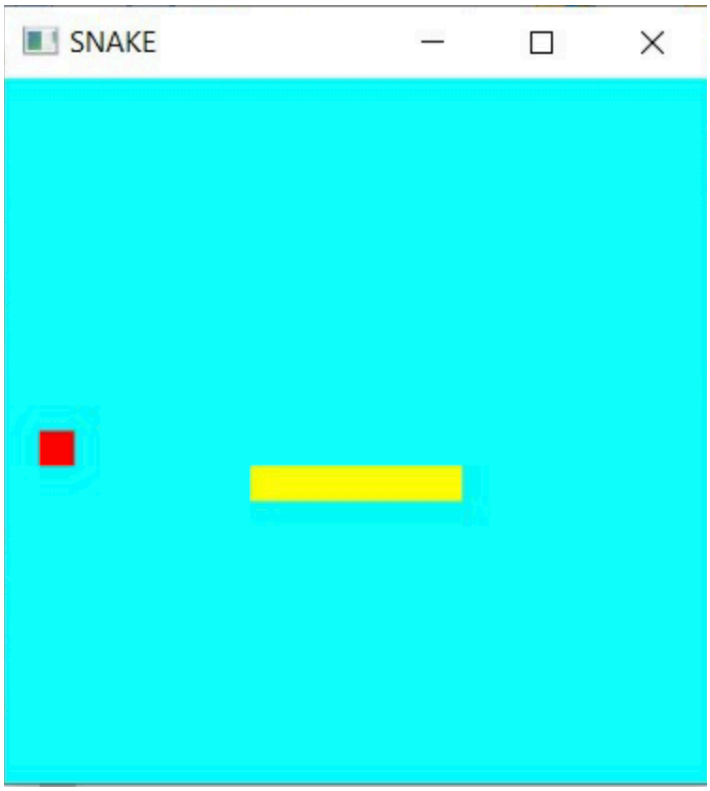
```
main :: IO () main = do

(_progName, _args) <- getArgsAndInitialize
_window <- createWindow "SNAKE"
gameState <- newIORef startGS
displayCallback $= display gameState keyboardMouseCallback $= Just
(keyboardMouse gameState) addTimerCallback startGap (update gameState)

mainLoop
```

Скриншоты





Вывод

В результате проделанной курсовой работы были улучшены навыки программирования на функциональном языке программирования, а также произошло ознакомление с GLUT на Haskell.

Приложение. Код программы.

https://github.com/mycelium/hsse-fp-2019-2/blob/3530904/80007_ponomarev-stepan/tasks/ haskell/course_project/snake.hs