



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
М. В. ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики
Кафедра алгоритмических языков

Отчёт о выполнении задания практикума
«Блэкджек»

Студент 325 группы
Доманчук А.

Москва, 2022

1 Постановка задачи

Блэкджек — классическая и одна из самых популярных карточных игр в казино.

В блэкджеке ваша цель как игроку — обыграть дилера. В игре используется от одной до восьми колод. Значения очков каждой карты: от 2 до 10, у туза — 1 или 11 (11 пока общая сумма не больше 21, далее 1), у т. н. картинок (король, дама, валет) — 10. Каждому игроку в начале игры раздаётся по две карты, дилер тоже получает две но одну от игроков он скрывает. Далее по кругу каждый игрок может набирать карты пока не захочет остановиться либо пока не перебьёт 21. После того как каждый игрок закончил набирать карты, дилер открывает вторую карту и набирает карты пока у него не будет 17.

- Если у игрока больше очков чем у дилера или дилер превысил 21, он победил и ему выплачивается его удвоенная ставка.
- Если у игрока и дилера число очков на руках равное, то такая ситуация называется "пуш". В такой ситуации все остаются при своих ставках, никто не выигрывает и не проигрывает.
- Если игрок превысил 21 или у него меньше очков чем у дилера, игрок проиграл и теряет свою ставку.

1.1 Базовые требования

- Интерфейс для игры в блэкджек
- Работа с бюджетом игроков и дилера
- Работа с механиками блэкджека
- Поддержка локального мультиплеера

2 Модули проекта

Проект состоит из следующих частей:

- **MysteriousConstans.hs** - константы для настроек игры
- **Lib.hs** - основной модуль игры, запуск приложения
- **ImageLoader.hs** - модуль, отвечающий за загрузку изображений до начала работы игры
- **Handlers.hs** - модуль, отвечающий за все взаимодействия с приложением, в нем также находится движок игры
- **Rendering.hs** - модуль, отвечающий за отрисовку всех объектов игры
- **State.hs** - модуль, отвечающий за инициализацию всех объектов игры и операции с картами игры

- **Types.hs** - модуль, содержащий объявления всех типов, используемых в программе

В модуле **MysteriousConstans.hs** описаны следующие константные значения:

cardPNGNames - список названий файлов, котоыре относятся к картам

xButtonOffset - описывает где находятся кнопки относительно центра экрана

xSize - описывает размер объектов в игре

bufferBetweenCards - расстояние между картами в рендеринге. При отрицательном значении карты ставятся друг на друга. В реальных казино так не делают, из ставят отдельно.

xCardsCenter - центр рендеринга карт для каждого игрока

В модуле **Types.hs** объявлены следующие типы:

Slider - структура, задающая все значения на слайдере: позиция "шарика на слайдере шаг слайдера, минимальные и максимальные значения

Player - содержит основную информацию о игроке: о его руке, балансе, текущей ставки, позиции, закончил ли он и выиграл ли он. *Игрок тоже дилер: просто баланс у него бесконечный*

State - состояния игры: фаза ставок, фаза игры, окончание игры, фаза когда игроку нужно принять какие-то действия.

Images - изображения игры

Deck - общая колода для всех игроков. Из неё в процессе игры берутся карты.

randomGen - Сид для случайной генерации, которая передаётся в функцию перемешивания карт.

selectedPlayer - Игрок, который ходит на данный момент.

Card - структура карты: имеет **cardRank** - от 2 до 10, валет, королева, король.
cardSuit - 4 карточные масти.

В модуле **Lib.hs** описаны следующие функции:

window - создание окна игры

handleInput - основная функция, отвечающая за реагирования за нажатие игрока на кнопки, меняет состояние игры.

В модуле **ImageLoader.hs** описаны следующие функции:

loadImages - загружает изображения используя библиотеку Gloss Juicy

Images - структура в которой хранятся уже загруженные изображения, передаётся в состояние игры (**BlackjackGame**)

loadCards - загружает все картинки карт в игре

В модуле **Handlers.hs** описаны следующие функции:

handleInput, **handleClick** - основные функции которая решает что делать при нажатии на кнопку / слайдер, возможны данные случаи:

- **hittedHitButton** - добавляет для выбранного игрока карту, используя **addCardsToPlayer**
- **hittedStandButton** - передаёт ход следующему игроку, если такого нет, то дилеру. Делает много проверок: вызывает функции **passToNextPlayerIfCurrentFinished**, **passToDealerIfAllFinished**

- **hittedDoubleButton** - добавляет одну карту, удваивает ставку и выполняет действия **hittedStandButton**.
- **hittedNewGameButton** - завершает игру. Убирает у всех игроков и у дилера карты, обновляет статусы и балансы игроков. Переводит на экран, где делаются ставки.
- **hittedSlider** - меняет значение на слайдере.

checkIfBust - вызывается при каждом нажатии на кнопку "Stand". Тем самым, проверяет превысил ли игрок 21.

passToNextPlayerIfCurrentFinished - передаёт ход следующего игроку если игрок закончил свой ход.

passToDealerIfAllFinished - передаёт ход дилеру если все игроки закончили.

determineWinner - вызывается в конце игры. Смотрим каждого игрока и даёт ему статус: получил блэкджек / победил / пуш / проиграл.

handEndGameBets - вызывается в конце игры. Делаем все действия со ставками: выигравшими игрока в 2 раза больше выплачивает, победившим в блэкджеке - 3/2.

addCardsToPlayer - добавляет карты игроку из общей колоды. Если колода опустошится, то её нужно перемешать.

checkButtonHit - проверяет в какую зону нажал игрок

getHitbox - вспомогательная функция для **checkButtonHit**. Возвращает "хитбокс" какой-то определенной кнопки.

updateSliderData - обновляет данные на слайдере при нажатии на слайдер.

Вспомогательные функции: **handleSliderClick**, **moveSliderBall**.

В модуле **Rendering.hs** описаны следующие функции:

drawScreen - отрисовка всех элементов на игровом поле

drawPlayerSelected - отрисовка "жетона" текущего игрока, помечено красным цветом.

drawEndGameStatus - отрисовка игры в конечном состоянии. Показывается кто выиграл, кто проиграл и кто превысил 21.

drawCards - рисовка карт. Зона рисовки отдельна для каждого игрока и дилера.

drawPlayerCardCount - рисует "очки" игрока. Следует обратить внимание, что очки в режиме *soft*. Это означает, что тузы считаются за 11, если это возможно. Однако это не означает, что игрок не может брать карты. Если туз в виде 11 превысит 21, то значение туза будет 1.

drawSlider - рисует слайдер для ставок.

В модуле **State.hs** описаны следующие функции:

initGameState - функция для создания игры с изначальными данными и загруженными изображениями.

createGameStateWith - инициализирует состояние игры *BlackjackGame*.

shuffle - функция для перемешки колоды.

getNumberByHand - по картам возвращает количество "очков".

countAcesAs - вспомогательная функция для **getNumberByHand**. Считает тузы в колоде за 1 или 11.

getNumberByCard - вспомогательная функция для **getNumberByHand**. Возвращает в зависимости от карты её значение.

3 Используемые библиотеки

При реализации использовались следующие библиотеки:

- **gloss** - графический интерфейс и обработка внешних событий
- **gloss-juicy** - дополнение к **gloss**. Помогает загружать изображения в виде *.png* и *.jpg*.
- **random** - генерация случайных чисел

4 Сценарии работы с приложением

При запуске требуется только указать число. Это число - количество игроков в игре. Данное значение на данный момент может быть от 1 до 3.