Отчёт о проделанной работе

Автор работы

Гордеев Никита, группа 22107

Описание задачи

ЛР 7. Задание «со звездочкой» (не обязательное)

Реализовать игру-головоломку с помощью средств языка shell

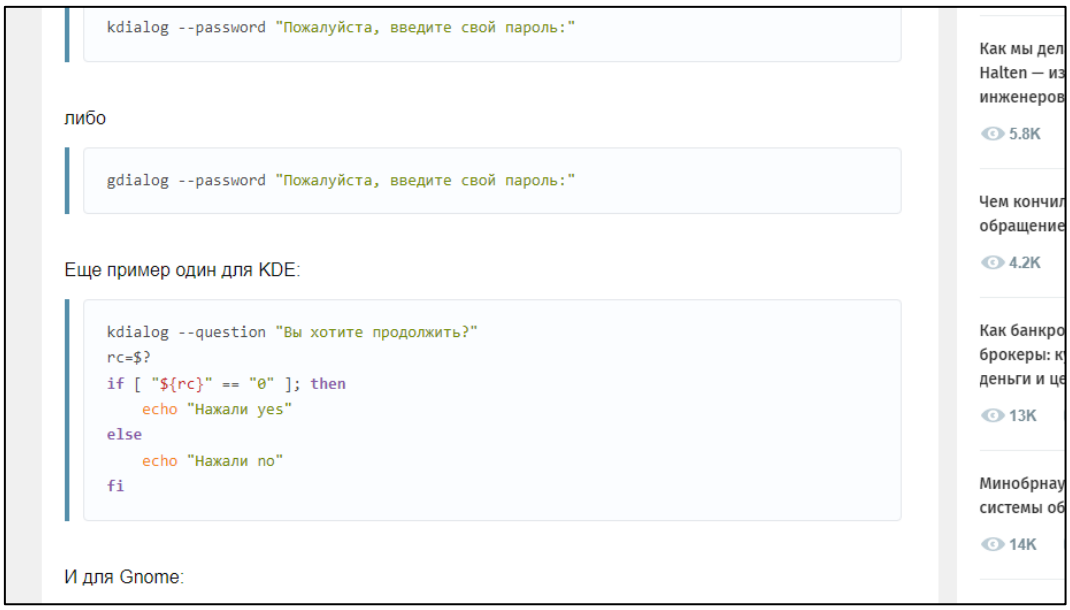
В реализуемой игре необходимо добраться из точки А в точку Б.

Текст решения с содержательными комментариями

- Файл maze_puzzle.sh (головоломка лабиринт)
- Содержательные комментарии оставлены в коде

Ход работы

1) Прочитал про взаимодействие bash-скриптов с пользователем.



```
kdialog --password "Пожалуйста, введите свой пароль:"
```

либо

```
gdialog --password "Пожалуйста, введите свой пароль:"
```

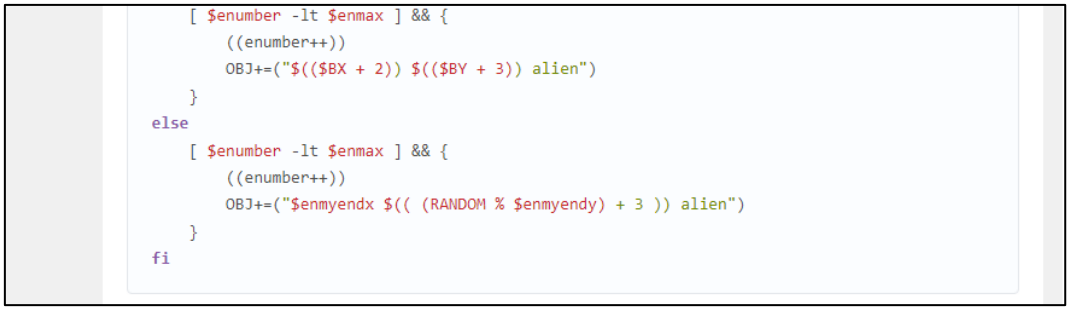
Еще пример один для KDE:

```
kdialog --question "Вы хотите продолжить?"
rc=$?
if [ "${rc}" == "0" ]; then
    echo "Нажали yes"
else
    echo "Нажали no"
fi
```

И для Gnome:

Иллюстрация 1

2) Нашёл готовую игру “скроллер-пулялка”, чтобы посмотреть, как пишут игры другие люди. На сайте была дана инструкция с фрагментами кода по написанию его игры. Я взял эту структуру для своей игры.



```
    [ $enumber -lt $enmax ] && {
        ((enumber++))
        OBJ+=("$(($BX + 2)) $(($BY + 3)) alien")
    }
else
    [ $enumber -lt $enmax ] && {
        ((enumber++))
        OBJ+=("$enmyendx $(( (RANDOM % $enmyendy) + 3 )) alien")
    }
fi
```

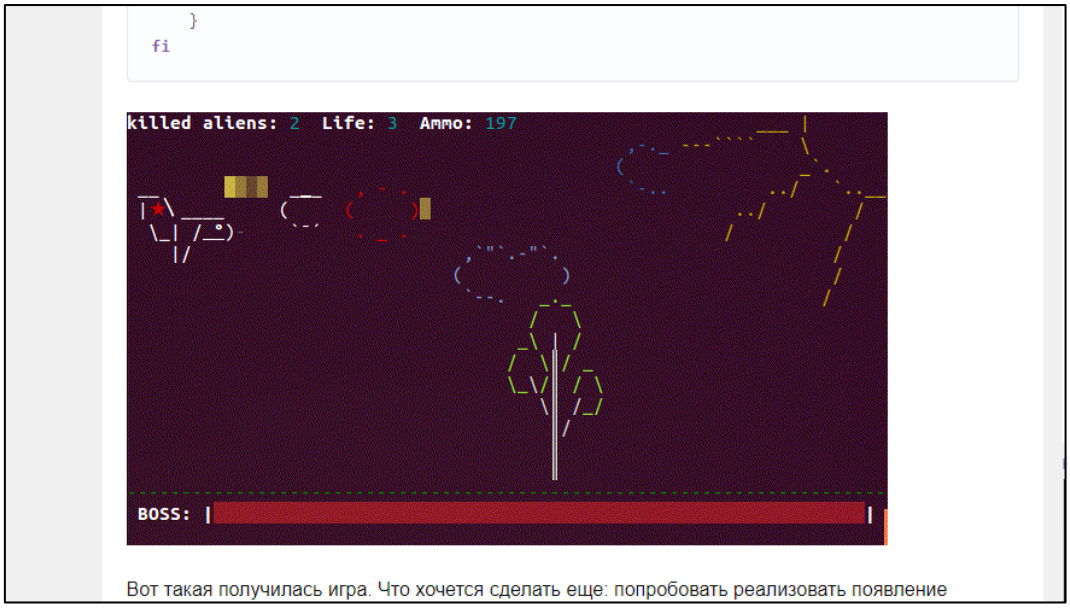


Иллюстрация 2, сайт-инструкция игры "Скроллер-пулялка" на bash

3) Нашёл готовую игру “ Пятнашки”, чтобы посмотреть, как пишут игры другие люди. На сайте была дана инструкция с фрагментами кода по написанию его игры. Я взял эту структуру для своей игры.

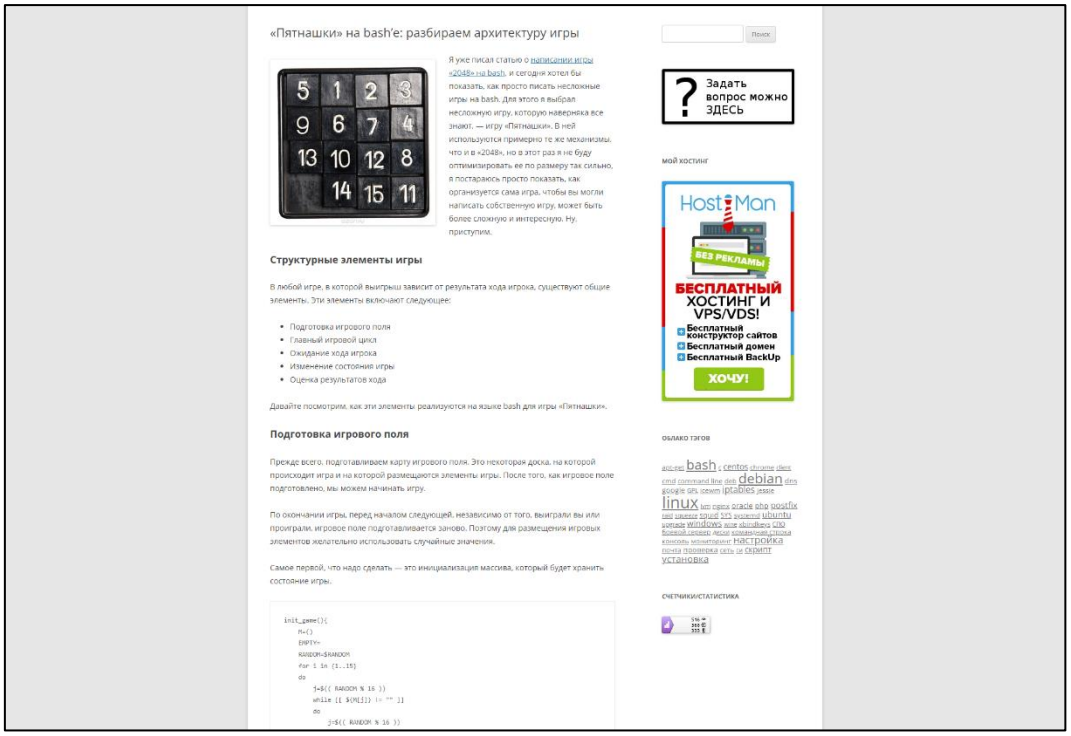


Иллюстрация 3

4) Нашёл готовый вариант игры лабиринт, чтобы посмотреть, как пишут игры другие люди.

Программа этого программиста состоит из 6 модулей. Я взял функции перемещения по лабиринту и вывода на экран изменений из его модулей.

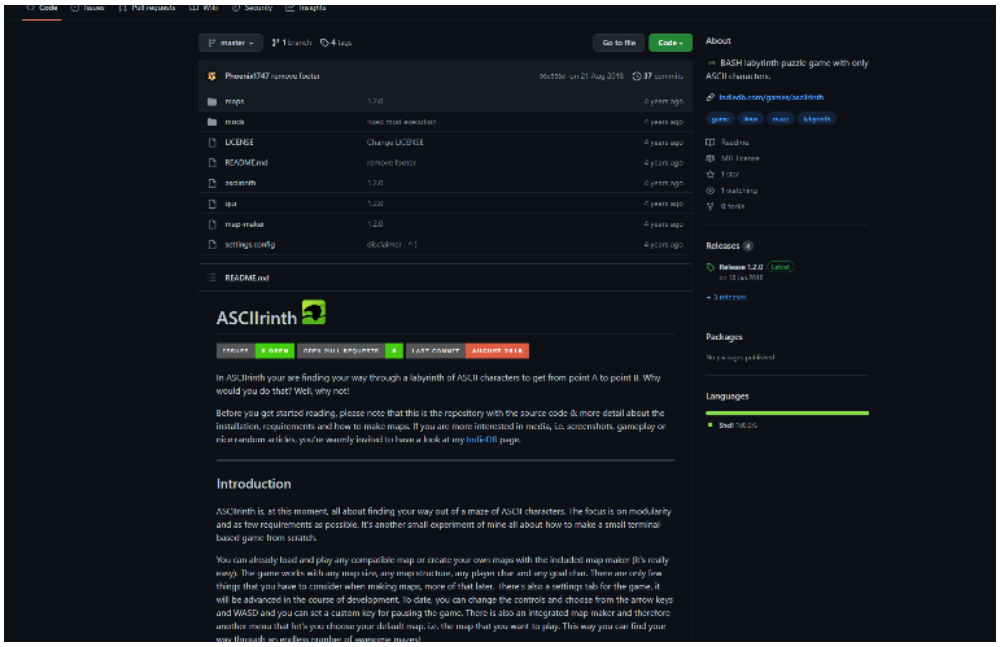


Иллюстрация 4, игра “ASCIIrinth” на bash

5) Создал основную функцию

```
# Основная функция
function main {
  #
  if [ -f $HOME/.bashrc ]; then
    echo "Файл существует!"
  fi

  # Графический интрефейс триггера
  reset_gui
  gui

  # После завершения графического интерфейса запустите игру, подготовив карту.
  read_map
  locate_player
  render_map

  # После инициализации делать это бесконечно
  while true; do
    output
    trap "return" SIGINT
    read -s -n1 control
    input
    case $placeholder in
      win)
          echo -e "\n$TEXT_won\n"
          read -r -s -n1
          reset_gui
          return;;
      *) # ИНАЧЕ
          render_map
          output
    esac
  done
}
```

Иллюстрация 5, основная функция

6) Написал функцию главного меню

```
# Главное меню. Пункты меню
function mainMenu {
  menu=(" Добро пожаловать в игру лабиринт" " [*] Играть!" " [*] Выход")
  call=mainMenu_do
}

# Главное меню. Запись выбранного действия
function mainMenu_do {
  case $SELECT_menu in
      1) exitVar="play";;
      2) exit_program;;
  esac
}
```

Иллюстрация 6, главное меню

7) Составил лабиринт

default.map

```
=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=
|, ,*, ,|, , , , , , , , , , ,|, , , , , , , , , , , ,|, , , , ,|, , , , , , , , , , , , , ,|, , , , , , , ,|, , , , , , , , , , , , ,|, , , , , ,|
|, ,=,=,+, ,+,=,+,=,=,=,=,=, ,+, ,=,=,=, ,=,=,=,=,=,=,=,=,=,=, ,+, ,=,=,=,=,=,=,=,=,=,=, , ,+,=,=,=,=,=,+, ,|, ,z,x,x,=,=,=,=,=,+,z, ,|, ,=,=,+, ,|
|, , , , , ,|, ,|, , , , , , , , , , ,|, , ,|, , , , , , , , , ,|, , , , , , , , , , , ,|, ,+,=,=,=,=,=,+, , , , , ,|, , , , , , , , ,|, , , ,|, ,|
|,=,=,=,=,=,+, ,|, ,+,=,=,=,=,=,=,=,=,=, , ,+,=,=,=,+,=,=,=,+, , , ,+,=,=,=,+,=,=,=,+, ,+, ,|, , , , , ,|, ,+, ,x,x,x,=,=,=,=,=,+, , , , ,+,=,=, ,|
|,=,=,=,=,=,+, ,|, ,+,=,=,=,=,=,=,=,=,=, , ,+,+,=,+,=,+,=,+,+, , , ,+,+,=,+,=,+,=,+,+, ,+, ,|, , , , , ,|, ,+, ,x,x,x,=,=,=,=,=,+, , , , ,+,=,=, ,|
|,=,=,=,=,=,+, ,|, ,+,=,=,=,=,=,=,=,=,=, , ,+,=,+,=,=,=,+,=,+, , , ,+,=,+,=,=,=,+,=,+, ,+, ,|, , , , , ,|, ,+, ,x,x,x,=,=,=,=,=,+, , , , ,+,=,=, ,|
|, , , , , , , , , ,|, , , , , , , , ,x, , , , , , , , , , , , ,|,=,=, , ,|, , , , , , , , , , ,+,=,+, , , ,|, , , , , , , , , , , ,=,=,=,|, ,0, ,|
|,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,=,|
```

Иллюстрация 7, лабиринт

8) Определил необходимые настройки лабиринта

```
# Определение настроек
function query_options {

  # Запросите некоторые вещи из файла настроек
  OPTIONS_controls="ARROWS"

  # Файл карты по умолчанию
  mapfile="default.map"

  # Буква ASCII, используемая для обозначения местоположения игроков.
  CHAR_player="O"

  # Персонаж, который представляет собой цель, которую необходимо достичь для победы
  CHAR_goal="*"

  # Стандартные клавиши для перемещения
  if [ "$OPTIONS_controls" = "WASD" ]
  then
    KEY_right="d"
    KEY_left="a"
    KEY_down="s"
    KEY_up="w"
  else
    KEY_right="C"
    KEY_left="D"
    KEY_down="B"
    KEY_up="A"
  fi
}
```

Иллюстрация 8, настройки лабиринта

9) Написал функцию преобразования файла с лабиринтов в массив

```
# Перевод файла с картой в массивы
function read_map {
  i=0

  local var=
  while IFS= read -r "var"; do
    IFS=',' read -r -a "array$i" <<< "$var"
    ((i++))
  done < "$mapfile"

}
```

Иллюстрация 9, преобразование файла с лабиринтом в массив

10) Написал функцию вывода лабиринта на экран

```
# Вывод лабиринта на экран
function output {
  clear

  # Комментарии по карте лабиринта
  echo -e "${COLOR_help} Текущий лабиринт: $mapfile${COLOR_reset}\n"

  local j=0

  while [ "$j" -le "$i" ]
  do

    local outputLevel="level$j"
    echo -e "${COLOR_labyrinth}${!outputLevel}"
    ((j++))

  done

  # Комментарии по прохождению лабиринта
  echo -e "${COLOR_help} Доберитесь до '$CHAR_goal', чтобы завершить эту карту!${COLOR_reset}"
  echo -e "${COLOR_help} Для выхода зажмите комбинацию клавиш CTRL + C${COLOR_reset}"
}
```

Иллюстрация 10, вывод лабиринта на экран

11) Написал функцию перемещения по лабиринту

```
# Перемещение, границы, победа
function input {
  placeholder=
  case $control in
    $KEY_left)
        local moveToX=$((CHAR_X - 1))
        local moveToY="$CHAR_Y";;
    $KEY_right)
        local moveToX=$((CHAR_X + 1))
        local moveToY="$CHAR_Y";;
    $KEY_down)
        local moveToY=$((CHAR_Y +1 ))
        local moveToX="$CHAR_X";;
    $KEY_up)
        local moveToY=$((CHAR_Y - 1))
        local moveToX="$CHAR_X";;
    *) return;;
  esac
```

```
    local nextPosition="array$moveToY[$moveToX]"
    case ${!nextPosition} in
      "$CHAR_goal")
          placeholder="win";;
      " ") # ПРОБЕЛ
          IFS= read "array$CHAR_Y[$CHAR_X]" <<< " "
          IFS= read "array$moveToY[$moveToX]" <<< "$CHAR_player"
          CHAR_X="$moveToX"
          CHAR_Y="$moveToY";;
    esac

}
```

Иллюстрация 11, функция перемещения по лабиринту

12) Написал функцию-проверку выхода за границы лабиринта

```
local nextPosition="array$moveToY[$moveToX]"
case ${!nextPosition} in
  "$CHAR_goal")
      placeholder="win";;
  " ") # ПРОБЕЛ
      IFS= read "array$CHAR_Y[$CHAR_X]" <<< " "
      IFS= read "array$moveToY[$moveToX]" <<< "$CHAR_player"
      CHAR_X="$moveToX"
      CHAR_Y="$moveToY";;
esac
```

Иллюстрация 12, проверка на стену

13) Добавил красивый вывод надписи победа. Из-за того, что “генератор надписей из символов - рисунки из символов” работает только с символами английского языка воспользовался “конвертором русских ников”.

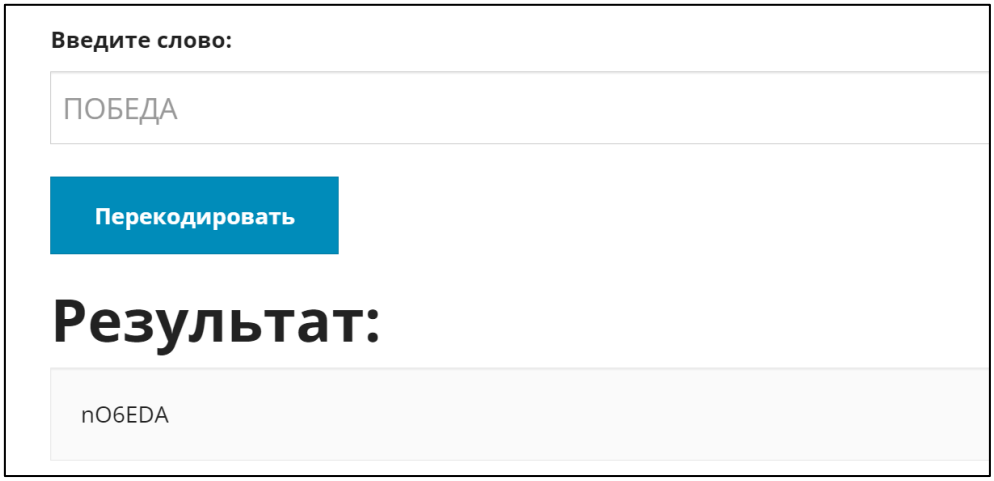


Иллюстрация 13, конвертор русских ников

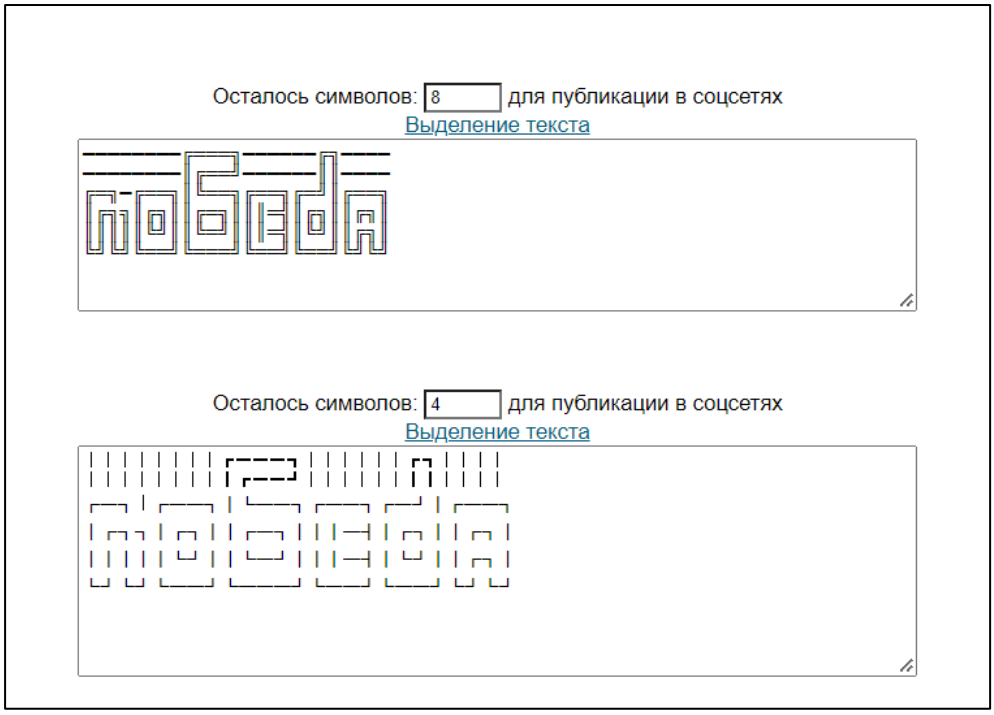


Иллюстрация 14, генератор надписей из символов - рисунки из символов

14) Прочитал как разукрасить вывод в echo и реализовал красивый вывод всех текстов: заголовков, лабиринта, подсказок, надписи “победа”.

```
#black    30  40    \033[30m  \033[40m
#red      31  41    \033[31m  \033[41m
#green    32  42    \033[32m  \033[42m
#yellow    33  43    \033[33m  \033[43m
#blue    34  44    \033[34m  \033[44m
#magenta    35  45    \033[35m  \033[45m
#cyan    36  46    \033[36m  \033[46m
#white    37  47    \033[37m  \033[47m

# Дополнительные свойства для текта:
BOLD='\033[1m'       #  ${BOLD}      # жирный шрифт (интенсивный цвет)
DBOLD='\033[2m'      #  ${DBOLD}    # полу яркий цвет (тёмно-серый, независимо
от цвета)
NBOLD='\033[22m'      #  ${NBOLD}    # установить нормальную интенсивность
UNDERLINE='\033[4m'     #  ${UNDERLINE}  # подчеркивание
NUNDERLINE='\033[4m'     #  ${NUNDERLINE}  # отменить подчеркивание
BLINK='\033[5m'       #  ${BLINK}    # мигающий
NBLINK='\033[5m'       #  ${NBLINK}    # отменить мигание
INVERSE='\033[7m'     #  ${INVERSE}    # реверсия (знаки приобретают цвет фона,
а фон -- цвет знаков)
NINVERSE='\033[7m'     #  ${NINVERSE}    # отменить реверсию
BREAK='\033[m'       #  ${BREAK}    # все атрибуты по умолчанию
NORMAL='\033[0m'      #  ${NORMAL}    # все атрибуты по умолчанию

# Цвет текста:
BLACK='\033[0;30m'     #  ${BLACK}    # чёрный цвет знаков
RED='\033[0;31m'       #  ${RED}      # красный цвет знаков
GREEN='\033[0;32m'     #  ${GREEN}    # зелёный цвет знаков
YELLOW='\033[0;33m'     #  ${YELLOW}    # желтый цвет знаков
BLUE='\033[0;34m'       #  ${BLUE}      # синий цвет знаков
MAGENTA='\033[0;35m'     #  ${MAGENTA}    # фиолетовый цвет знаков
CYAN='\033[0;36m'       #  ${CYAN}      # цвет морской волны знаков
GRAY='\033[0;37m'       #  ${GRAY}      # серый цвет знаков

# Цветом текста (жирным) (bold) :
DEF='\033[0;39m'       #  ${DEF}
DGRAY='\033[1;30m'     #  ${DGRAY}
LRED='\033[1;31m'       #  ${LRED}
LGREEN='\033[1;32m'     #  ${LGREEN}
LYELLOW='\033[1;33m'     #  ${LYELLOW}
LBLUE='\033[1;34m'     #  ${LBLUE}
LMAGENTA='\033[1;35m'   #  ${LMAGENTA}
LCYAN='\033[1;36m'     #  ${LCYAN}
WHITE='\033[1;37m'     #  ${WHITE}
```

Иллюстрация 15, как разукрасить вывод в echo

```
# Текст, который появляется при выигрыше
TEXT_won="
\033[1;34m
\033[1;34m
\033[1;34m
\033[1;34m
\033[1;34m
\033[1;34m
\033[0m
  "

# --------------------------------- Графический интерфейс пользователя ---------------------------------

# Настройки цвета
COLOR_help="\033[0;33m"       # Подсказки, жёлтый текст
COLOR_labyrinth="\033[0;35m"  # Лабиринт, фиолетовый текст
COLOR_menuHeader="\033[1;34m" # Заголовки, синий текст, жирное выделение
COLOR_selection="\033[1;32m"  # Курсор выбора, зелёный текст, жирное выделение
COLOR_reset="\033[0m"         # Все атрибуты по умолчанию

# --------------------------------- Проверка терминала ---------------------------------

```

Иллюстрация 16, разукрашивание вывода на экран

15) Реализовал проверку размеров и функций терминала

```
# Проверка терминала. Версия BASH
if ! [ "${BASH_VERSION:0:1}" -ge 4 ]; then
  echo -e "\033[0;31m[Ошибка] Чтобы играть в эту игру, вам понадобится как минимум BASH версии 4.0! \033[0m"
  exit 1
fi

# Проверка терминала. Размеры терминала
lines=$(tput lines)
columns=$(tput cols)

if [ "$columns" -lt "80" ] || [ "$lines" -lt "15" ]; then
  echo -e "\033[0;31m[Ошибка] Чтобы играть в эту игру, вам понадобится размер терминала 80x15 строк! \033[0m"
  exit 1
fi

# Проверка терминала. Функции терминала
toCheck=("sed" "find" "grep" "wc")
for check in ${toCheck[@]}
do
  "$check" --help>/dev/null 2>&1 || {
    echo -e "\033[1;31m[Ошибка] Ошибка при поиске команды $check. Невозможно начать! \033[0m";
    exit 1;
  }

done
```

Иллюстрация 17, проверка версии и размеров терминала

16) Протестировал работу программы

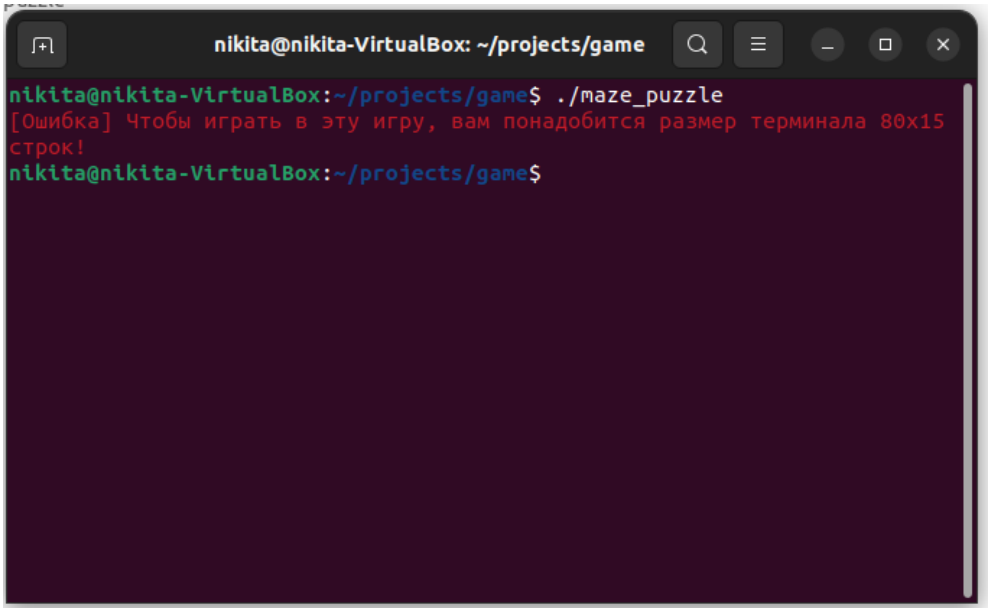


Иллюстрация 18, вывод сообщения об ошибке при недостаточно большом размере терминала

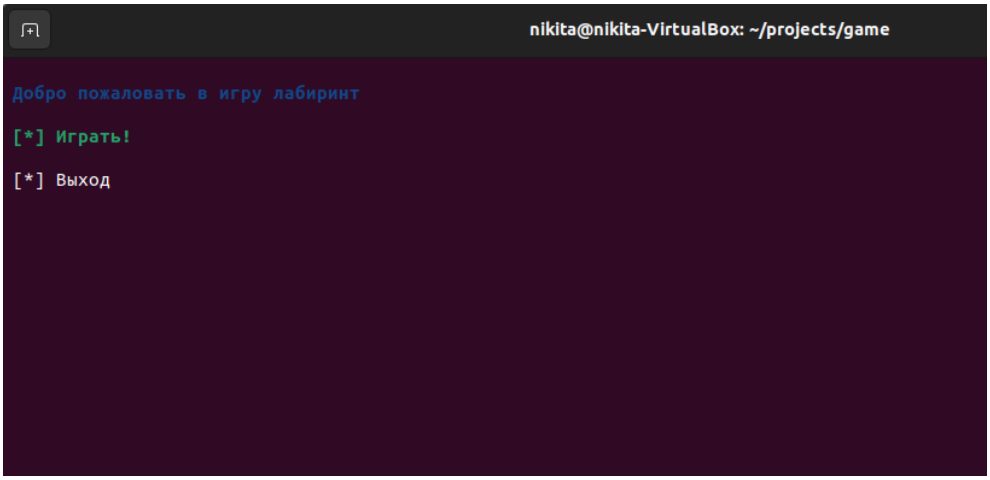


Иллюстрация 19, увеличил экран, запустил игру

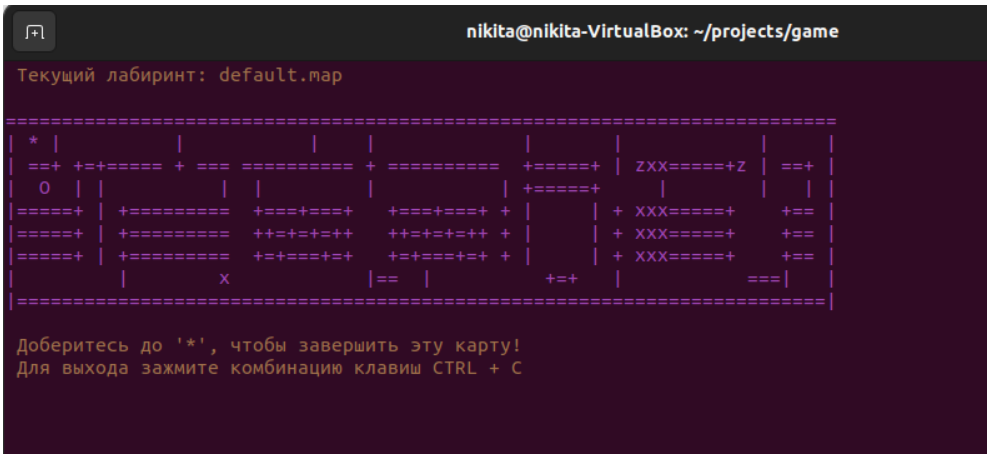


Иллюстрация 20, открыл лабиринт, проверил перемещение

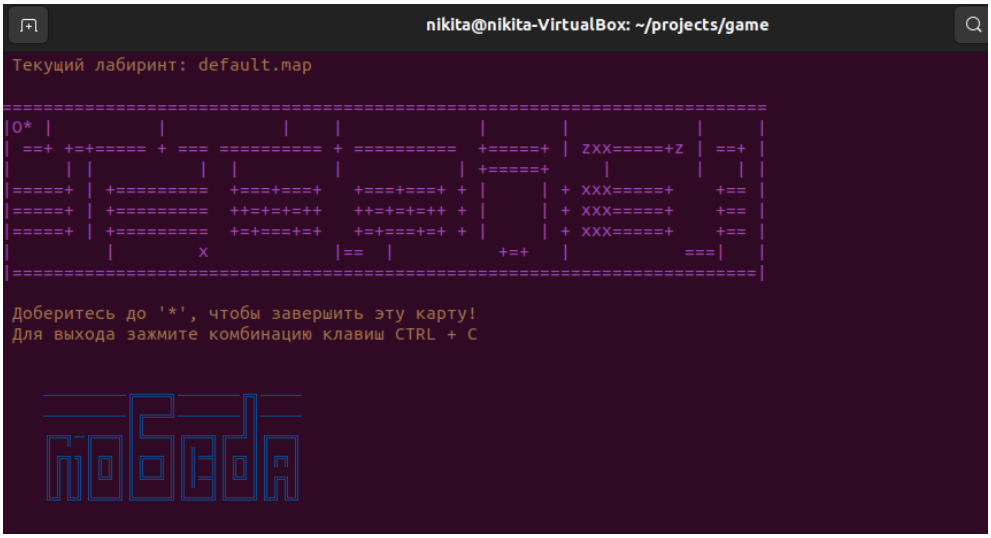


Иллюстрация 21, прошёл лабиринт, получил сообщение "победа"

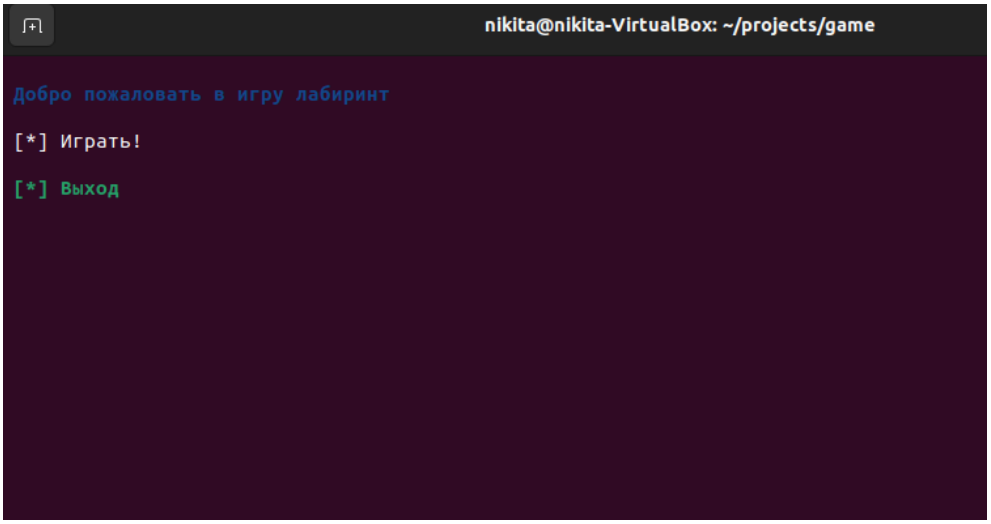


Иллюстрация 22, после прохождения лабиринта вернулся в главное меню

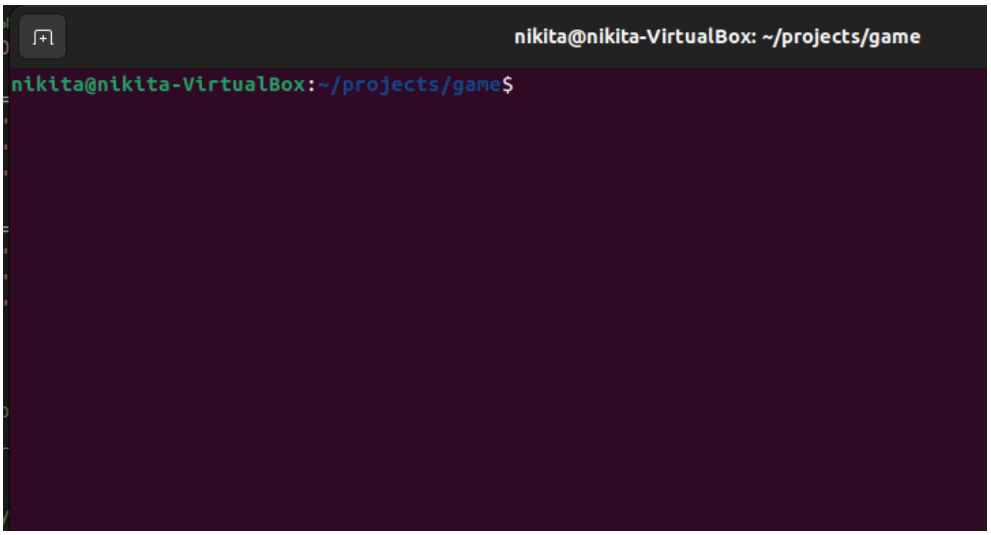


Иллюстрация 23, вышел из программы

Вывод

Реализовал игру-головоломку по перемещению из точки А в точку Б с помощью средств языка shell.

Ссылки на источники информации, использованные при решении задачи

1. Взаимодействие bash-скриптов с пользователем // Хабр URL: https://habr.com/ru/post/126701/ (дата обращения: 19.05.2022).

2. Играючи BASH'им // Хабр URL: https://habr.com/ru/post/335960/ (дата обращения: 20.05.2022).

3. ASCIIrinth // Github URL: https://github.com/Phoenix1747/asciirinth (дата обращения: 20.05.2022).

4. «Пятнашки» на bash’е: разбираем архитектуру игры // MNorin.com URL: https://mnorin.com/igra-pyatnashki-na-bashe.html (дата обращения: 21.05.2022).

5. Играючи BASH'им // Хабр URL: https://habr.com/ru/post/335960/ (дата обращения: 20.05.2022).

6. Конвертер русских ников // FaceCam.ru URL: https://www.facecam.ru/rus-lat (дата обращения: 23.05.2022).

7. Генератор надписей из символов - рисунки из символов // Text-Image.Ru Рисунки символами и картинки из символов ASCII Art URL: http://text-image.ru/index/generator_nadpisej_iz_simvolov/0-17 (дата обращения: 23.05.2022).

8. Цветной man или как разукрасить вывод echo // Хабр URL: https://habr.com/ru/post/119436/ (дата обращения: 25.05.2022).