**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Старший преподаватель департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. В. Патнюхин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **КОМПЬЮТЕРНАЯ СТРАТЕГИЧЕСКАЯ ИГРА "NERV" НА UNITY**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.01-01** **ТЗ-1-ЛУ** | | |
|  |  | |
| Исполнитель  студент группы БПИ175  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И. О. Балбин/  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | |
|  | | |
|  | |  |

**Москва 2019**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU. 17701729.04.01-01 81 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЧЕСКОГО ДЕТЕКТИРОВАНИЯ**  **ШАХМАТНЫХ ФИГУР ДЛЯ ЗАПИСИ ХОДОВ**  **Пояснительная записка**  **RU. 17701729.04.01-01 81 01-1**  **Листов 16** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2019**

Оглавление

[1. ВВЕДЕНИЕ 4](#_Toc21952633)

[1.1. Наименование программы 4](#_Toc21952634)

[1.2. Краткая характеристика области применения программы 4](#_Toc21952635)

[2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ 5](#_Toc21952636)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕТРИСТИКИ 6](#_Toc21952637)

[3.1. Постановка задача на разработку программы 6](#_Toc21952638)

[3.2. Описание алгоритмов и функционирование программы 6](#_Toc21952639)

[3.2.1 Детектирование доски 6](#_Toc21952640)

[3.2.2 Обрезание шахматной доски из видеопотока 7](#_Toc21952641)

[3.2.2 Детектирование ходов 8](#_Toc21952642)

[3.2.2.1 Определение руки над шахматной доской 8](#_Toc21952643)

[3.2.2.2 Определение шахматного хода 10](#_Toc21952644)

[3.2.3 Video stream 11](#_Toc21952645)

[3.2.4 Сервер и Web приложение 11](#_Toc21952646)

[3.3. Описание и обоснование метода входных данных. 12](#_Toc21952647)

[3.3.1 Описание метода организации входных и выходных данных 12](#_Toc21952648)

[3.3.2 Обоснование метода организации входных и выходных данных. 12](#_Toc21952649)

[3.4. Описание и обоснование метода выбора технических и программных средств 12](#_Toc21952650)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 13](#_Toc21952651)

[4.1 Предполагаемая потребность 13](#_Toc21952652)

[4.2 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 13](#_Toc21952653)

[Приложение 2 15](#_Toc21952654)

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы: «Приложение для автоматического детектирования шахматных фигур для записи ходов» («Application for automatic detection of chess pieces to record moves»).

## Краткая характеристика области применения программы

Программа предназначена для игроков в шахматы. С его помощью игроки смогут в автоматическом режиме записать партию в шахматной нотации. Это позволит игрокам анализировать сыгранные партии и не отвлекаться на их запись вручную во время самой игры.

# ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Программа выполнена в рамках темы курсовой работы — «Приложение для автоматического детектирования шахматных фигур для записи ходов», в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия».

Основанием для разработки является приказ от ХХ.ХХ.2017 г. № ХХХ «ХХХХ» декана факультета компьютерных наук НИУ ВШЭ.

# ТЕХНИЧЕСКИЕ ХАРАКТЕТРИСТИКИ

## 3.1. Постановка задача на разработку программы

Программа должна обеспечить возможность выполнение следующих функций:

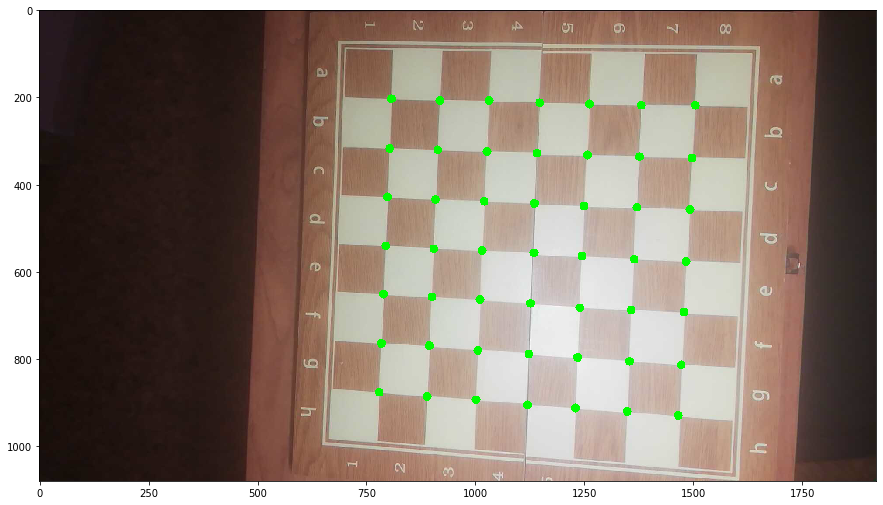
1. Подключение IP-видеокамеры по ее IP адресу
2. Автоматически детектировать доску по первому кадру из видеопотока с подключенной IP камеры
3. Детектировать начало и конец хода на шахматной доску
4. Детектировать изменение шахматной позиции и записывать ход

## 3.2. Описание алгоритмов и функционирование программы

Общий алгоритм детектирование шахматной доски и ходов представлен следующий диаграммой (рис. 1)

## 3.2.1 Детектирование доски

Для детектирования доски используется следующий алгоритм используется алгоритм из OpenCV – findChessBoardCorner. который возвращает внутреннюю углы на шахматной доске (рис. 1)



*Рис. 1 Зеленые точки – найденные внутренние углы на шахматной доске*.

При этом необходимо, чтобы доска была достаточно освещена (при условии недостаточной освещенности возникают шумы, которые мешают работе алгоритма).

Далее по полученной сетке возвращается восстанавливаются углы шахматной доски следующим кодом:

min\_x, max\_x = res[:, 0].min(), res[:, 0].max()

min\_y, max\_y = res[:, 1].min(), res[:, 1].max()

dx = (max\_x - min\_x) / 6

dy = (max\_y - min\_y) / 6

board\_corner = np.array([

[min\_x - dx, max\_y + dy],

[min\_x - dx, min\_y - dy],

[max\_x + dx, max\_y + dy],

[max\_x + dx, min\_y - dy],

])

res = np.vstack((res, board\_corner))

Минусы данного алгоритма, следующие:

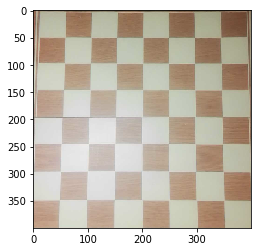
1. Нестабилен к шумам. Способен детектировать только пустую шахматную доску.
2. Способен детектировать только пустую шахматную доску.

Алгоритмы, более стабильные к шумам и способные находить доски с поставленным на нее фигурами описаны здесь <https://github.com/Elucidation/ChessboardDetect> и здесь <https://www.researchgate.net/publication/328461364_Chessboard_and_chess_piece_recognition_with_the_support_of_neural_networks>.

## 3.2.2 Обрезание шахматной доски из видеопотока

С помощью найденных углов шахматной доски getPerspectiveTransform из opencv мы высчитываем матрицу преобразование, и далее к каждому кадру ее применяем.

Пример на рис. 2



*Рис. 2 Обрезанная доска из рис.1*

Далее подразумевается, что при считывании кадра мы его обрезаем по шахматной доске

И преобразовываем в серую школы (RGB->GRAY SCALE).

## 3.2.2 Детектирование ходов

Для детектирование шахматных ходов используется алгоритм поиска руки над шахматной доской.

Мы храним изображение с текущей позицией без руки (frame\_last), и далее итеративно:

1. Считываем текущий кадр (frame)
2. С помощью алгоритма поиска руки, определяем есть ли она над доской
3. Если рука была над доской и пропала – значит совершился ход, иначе ход не был совершен
4. Если ход был совершен, то
   1. мы повторно считываем текущий кадр (frame)
   2. Определяем по frame и frame\_last какой ход был совершен
   3. Обновляем frame\_last – frame\_last = frame

Записываем ход

Алгоритм на python:

frame\_last = next(Camera.frame\_real())

frame\_last = cv2.cvtColor(frame\_last, cv2.COLOR\_BGR2GRAY)

hand\_on\_last = False

**for** frame **in** Camera.frame\_real():

frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

hand\_on\_cur = Camera.\_hand\_on(frame, frame\_last)

hand\_on\_change = hand\_on\_cur ^ hand\_on\_last

hand\_on\_last = hand\_on\_cur

# print("hand on" if hand\_on\_cur else "hand off")

**if** hand\_on\_change **and** **not** hand\_on\_cur:

frame = next(Camera.frame\_real())

frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

cur\_move = Camera.\_get\_move(frame, frame\_last)

**print**(cur\_move)

Camera.board.push(cur\_move)

## 3.2.2.1 Определение руки над шахматной доской

Для детектирования руки над шахматной доской:

1. высчитаем абсолютную разницу между frame и frame\_last - diff.
2. Бинаризуем diff по threshold=150
3. С помощью морфологический операций с прямоугольным ядром чистим шумы:
   1. diff = erode(diff, kernel)
   2. diff = dilate(diff, kernel)
4. Опредялем есть ли рука над доской по проценту ненулевых элементов в отчищенной от шумов diff
5. *Рис. 3 Слева – отчищенный от шумов diff; справа – то же изображение в gray\_scale. В левом верхнем углу присутствие\отсутствие руки над доской.*

|  |  |
| --- | --- |
|  |  |
|  |  |

*Рис. 3 Слева – отчищенный от шумов diff; справа – то же изображение в gray\_scale. В левом верхнем углу присутствие\отсутствие руки над доской.*

Алгоритм на python:

**def** getFrameDiff(frame1, frame2):

diff = cv2.absdiff(frame1, frame2)

diff = cv2.threshold(diff, 10, 150, cv2.THRESH\_BINARY)[1]

kernel = np.ones((5, 5,))

diff = cv2.erode(diff, kernel)

diff = cv2.dilate(diff, kernel)

**return** diff

**def** handOn(frame\_cur, frame\_last, thr = 0.1):

diff = getFrameDiff(frame\_cur, frame\_last)

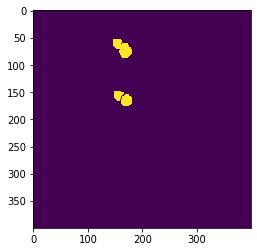
size = diff.shape[0]

**return** (diff != 0).sum() / (size\*size) > thr

## 3.2.2.2 Определение шахматного хода

Для определения хода мы высчитаем отчищенную от шумов разницу между frame и last\_frame.

Далее по полученной разницу определяем в каких клетках произошли изменения (diff[i][j] > 0). Если в клетке произошли значительные изменения ( >0.1), значит эта клетка задействован в ходе.



*Рис. 4 diff при хождение пешкой d2-d4. Желтые пятна – ненулевые значения, фиолетовый цвет - нули*

Алгоритм на python:

**def** getMove(frame\_cur, frame\_last, board, thr = 0.1):

diff = getFrameDiff(frame\_cur, frame\_last)

step = diff.shape[0] // 8

change\_square = []

**for** i **in** range(8):

**for** j **in** range(8):

sub\_diff = diff[step\*i:step\*(i+1),step\*j:step\*(j+1)]

**if** (sub\_diff != 0).sum()/(step\*step) > thr:

change\_square.append([i, j])

moves = []

**for** j, i **in** change\_square:

**for** l, k **in** change\_square:

moves.append(chess.Move(i + j\*8, k + l\*8))

res\_move = None

**for** move **in** moves:

**for** beaut\_move **in** board.legal\_moves:

# if move in board.legal\_moves:

**if** beaut\_move == move:

res\_move = beaut\_move

**break**

**if** res\_move **is** **not** None:

**break**

**return** res\_move

# 3.2.3 Video stream

За Video stream отвечает класс BaseCamera, в котором каждый необходимый видеопоток обрабатывается в отдельном потоке в отдельной корутине.

# 3.2.4 Сервер и Web приложение

Сервер реализован на Flask в соответствующей ему архитектуре .

В файле app.py хранятся url-ы и их обработчики. В папке templates лежат все необходимые html-шаблоны.

# 3.3. Описание и обоснование метода входных данных.

## 3.3.1 Описание метода организации входных и выходных данных

Входными данными является клавиатура, нажатие копок мыши и видеопоток с внешней IP камеры в формате ‘\*.jpg’ или передачи по протоколу RTSP.

## 3.3.2 Обоснование метода организации входных и выходных данных.

Для взаимодействия и управление программой пользователю необходимы клавиатура и мышь. А без видеопотока с шахматной доской невозможно узнать информацию о ней.

# 3.4. Описание и обоснование метода выбора технических и программных средств

Для использования web – приложения необходим только доступ к браузеру, на любом устройстве, с техническим средствам позволяющем ему работать без перебоев с запущенным браузером.

Для сервера необходимы:

* ОС: Ubuntu server
* Видеокарта: nvidia geforce gtx 1060 6gb
* Процессор: intel Core i5-6400
* Оперативная память: 8 Гб

# ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## 4.1 Предполагаемая потребность

Данный продукт предназначен для автоматической записи шахматной партии в шахматную нотацию.

Он может использоваться как в “домашних” партий, так и в турнирах.

## 4.2 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

На рынке существует несколько решений:

* Электронные шахматные доски DGT с записью ходов, один комплект (фигуры и доска) стоят от 20’000 рублей.
* Программные opensource решения, не являющиеся коммерческими продуктами (Пример: <https://github.com/maciejczyzewski/neural-chessboard>, <https://github.com/jonkoi/chess_detection> ).

Мое же решение имеет ряд преимуществ, по сравнению с существующими решениями:

* Низкая стоимость, по сравнению с DGT досками (всю стоимость образует устройство для трансляции (андроид телефон) и механизм для его крепежа над доской). При наличии же андроид-телефона, стоимость будет составлять только механизм для крепежа устройства
* Гибкость решения. Приложение сможет работать с широким спектром моделей досок и шахматных фигур
* Удобный интерфейс, по сравнению с opensource решениями. Позволит использовать приложение без специфических знаний.

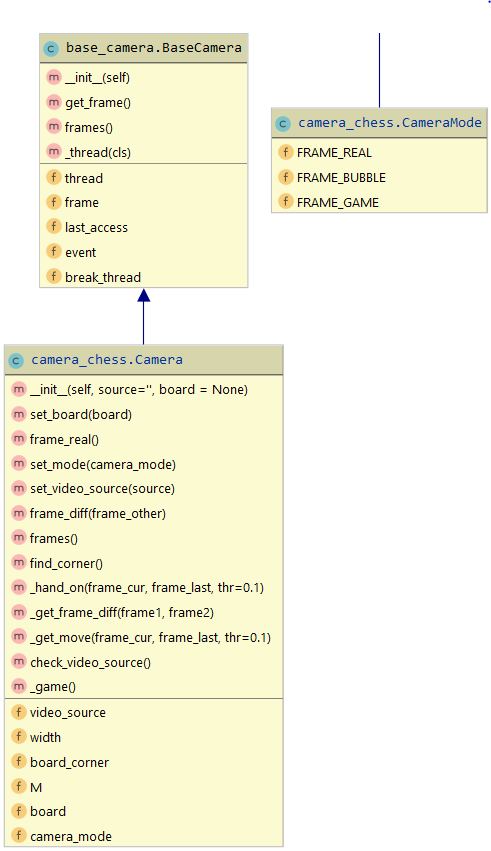
# Приложение 1

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК издательство стандартов, 2001.
10. ГОСТ Р 7.02-2006 Консервация документов на компакт-дисках. Общие требования. – М.: ИПК Издательство стандартов, 2006.
11. ГОСТ 18300-87 Спирт этиловый ректификованный технический. Технические условия. – М.: ИПК Издательство стандартов, 1997.
12. ГОСТ 9805-84. Спирт изопропиловый. Технические условия. – М.: ИПК Издательство стандартов, 1984.
13. ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом. //Единая система программной документации. – М.: ИПК
14. <https://github.com/maciejczyzewski/neural-chessboard>, <https://github.com/jonkoi/chess_detection>
15. <https://www.researchgate.net/publication/328461364_Chessboard_and_chess_piece_recognition_with_the_support_of_neural_networks>

# Приложение 2

**Диаграммы классов программы**

****

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |