

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» (УрФУ)

Физико-технологический институт

Кафедра «Технической физики»

Оценка

Руководитель курсового

проектирования

Члены комиссии

Дата защиты

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по теме: Разработка логической игры «Переверни фишки»

по дисциплине: Прикладное программирование

Студент: Чистополов Александр Алексеевич

(ФИО) (Подпись)

Группа: ФТ - 230008

Екатеринбург

2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc186227836)

[1 Постановка задачи 4](#_Toc186227837)

[2 Описание алгоритмов программы 5](#_Toc186227838)

[2.1 Словесное описание алгоритмов программы 5](#_Toc186227839)

[2.2 Блок-схемы алгоритмов основных функций 6](#_Toc186227840)

[3 Реализация программы 10](#_Toc186227841)

[3.1 Листинг программы 10](#_Toc186227842)

[3.2 Тестирование функциональных и нефункциональных характеристик программы 15](#_Toc186227843)

[4 Руководство пользователя 19](#_Toc186227844)

[4.1 Назначение программы 19](#_Toc186227845)

[4.2 Запуск программы 19](#_Toc186227846)

[4.3 Выполнение основных функций 19](#_Toc186227847)

[4.4 Завершение программы 20](#_Toc186227848)

[4.5 Пример работы с программой 20](#_Toc186227849)

[Заключение 22](#_Toc186227850)

[Библиографический список 23](#_Toc186227851)

# ВВЕДЕНИЕ

С развитием технологий, человек стал всё больше и больше полагаться не на свои умения и качества, а на помощь этих технологий. Поэтому нам следует не забывать про саморазвитие в том числе и логическое мышление. Существуют множество различных способов тренировать логику так, например, ирга «Переверни фишки».

«Переверни фишки» это простая с виду, но сложная в освоении игра, цель которой заключается в том, чтобы все фишки на поле стали одного цвета. Переворачивая одну фишку, также переворачиваются все соседние с ней, т.е. при изменении цвета угловой фишки меняются три, при перевороте боковой – четыре, а в центре – пять. Для достижения цели недостаточно просто в случайном порядке нажимать на всё подряд, нужно рассчитывать какие фишки перевернуться в следующем ходу и понимать, как это может изменить ситуацию на поле.

# **1 Постановка задачи**

Основной задачей данной курсовой работы является разработка приложения, реализующего игру «Переверни фишки». Для достижения поставленной цели необходимо решить несколько более конкретных задач:

а) Создание поля с фишками двух цветов;

* Реализовать алгоритм для создания поля размером 4×4, случайно заполненного фишками красного или синего цвета.

б) Создание графического интерфейса;

* Разработать понятный графический интерфейс;
* Интерфейс должен включать поле 4×4 и фишки различных цветов.

в) Обработка взаимодействия с игроком;

* Написать функционал, который будет обрабатывать нажатия на фишки, меняя их цвет;
* Реализовать завершение игры с выводом сообщения «Задача решена!».

г) Тестирование и отладка программы;

* Провести тестирование программы для проверки на наличие возможных ошибок;
* Отладить программу, исправив найденные в ходе тестировки ошибки.

д) Создание руководства пользователя.

* Написать подробное руководство для пользователей, в котором будут описаны правила игры, интерфейс приложения и инструкции использованию программы.

# **2 Описание алгоритмов программы**

# **2.1 Словесное описание алгоритмов программы**

При запуске программы перед пользователем появляется окно с игрой. В окне представлены поле фишек разных цветов 4×4 и кнопка перезапуска программы.

Каждая фишке представляет собой кнопку, реализованную с помощью библиотеки tkinter, имеющую свой индекс. Когда пользователь нажимает на кнопку меняется её цвет, а также цвет фишек, расположенных рядом, затем производится проверка на то, сколько фишек одного цвета находится на поле: если их количество равно шестнадцати, то игра останавливается и появляется окно с сообщение «Задача решена!».

В окне «победы» присутствуют две функциональных кнопки: «Перезапуск» и «Выход». При нажатии на «Выход» выполняется внутренняя команда python exit(), которая завершает программу с указанным кодом ошибки (код ошибки 0 означает, что программа была завершена корректно). Если пользователь нажмёт «Перезапуск», то программа закрое все открытые окна и заново запускает игру.

# **2.2 Блок-схемы алгоритмов основных функций**

Блок-схема удобный, компактный и простой способ объяснения алгоритма работы программы. При запуске кода, программа создаёт окно игры и выполняются соответствующие алгоритмы (Рисунок 1). При определённых условия перед пользователем может возникнуть окно «победы», которое реализовано по алгоритму, представленному на рисунке 2.

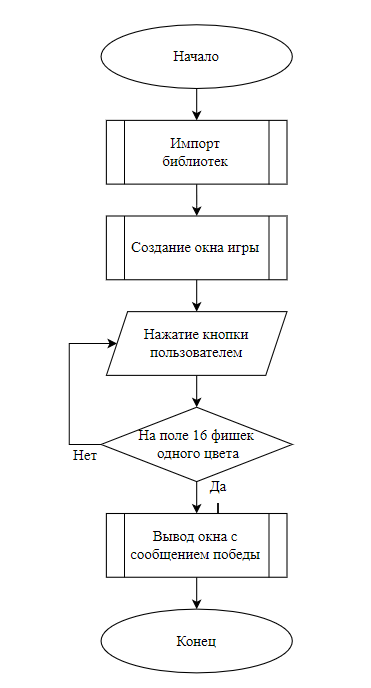
****

Рисунок 1 – Блок-схема алгоритма работы главного окна

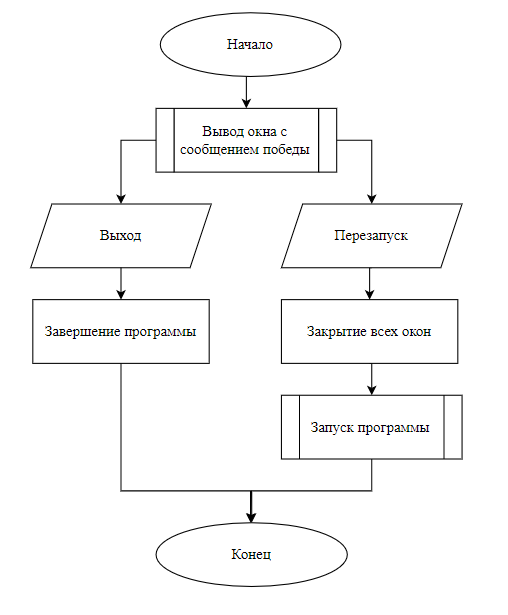
****

Рисунок 2 – Блок-схема алгоритма работы окна «победы»

# **3 Реализация программы**

# **3.1 Листинг программы**

|  |  |
| --- | --- |
| 1 | import tkinter as tk |
| 2 | from random import randint |
| 3 | from time import sleep |
| 4  5  6 | #Main window |
| 7 | def main(): |
| 8 | def wincheck(): |
| 9 | def WinWindow(): |
| 10 | global WinWindow |
| 11  12 | WinWindow = tk.Tk() |
| 13 | WinWindow.geometry('300x150') |
| 14 | WinWindow.resizable(False, False) |
| 15 | WinWindow.title('WIn') |
| 16  17 | Label = tk.Label(WinWindow, |
| 18 | text="Задача решена!", |
| 19 | font=('Arial', '16') |
| 20 | ) |
| 21 | Label.place(x=70, y=10) |
| 22  23 | winrestartB = tk.Button(WinWindow, |
| 24 | bg='White', |
| 25 | width=10, |
| 26 | height=1, |
| 27 | text='Перезапуск', |
| 28 | font=('Arial', '14'), |
| 29 | command=RestartC) |
| 30 | winrestartB.place(x=87, y=50) |
| 31  32 | exitB = tk.Button(WinWindow, |
| 33 | bg='White', |
| 34 | width=10, |
| 35 | height=1, |
| 36 | text='Выход', |
| 37 | font=('Arial', '14'), |
| 38 | command=lambda: exit(0)) |
| 39 | exitB.place(x=87, y=95) |
| 40  41  42 | WinWindow.grab\_set() |
| 43  44  45 | rcount = 0 |
| 46 | bcount = 0 |
| 47 | for i in range(4): |
| 48 | for j in range(4): |
| 49 | v = str(i) + str(j) |
| 50 | a = buttons[buttons\_pos.index(v)] |
| 51  52 | if a['bg'] == 'red': |
| 53 | rcount += 1 |
| 54 | else: |
| 55  56 | bcount += 1 |
| 57 | if rcount == 16: |
| 58 | WinWindow() |
| 59 | elif bcount == 16: |
| 60 | WinWindow() |
| 61  62 | WinWindow() |
| 63  64  65 | def GCC(v): |
| 66 | a = buttons[buttons\_pos.index(v)] |
| 67 | if a['bg'] == 'red': |
| 68 | a['bg'] = 'blue' |
| 69 | else: |
| 70 | a['bg'] = 'red' |
| 71 | try: |
| 72 | a = buttons[buttons\_pos.index(v[0]+str(int(v[1])-1))] |
| 73 | if a['bg'] == 'red': |
| 74 | a['bg'] = 'blue' |
| 75 | else: |
| 76 | a['bg'] = 'red' |
| 77 | except: |
| 78 | pass |
| 79 | try: |
| 80 | a = buttons[buttons\_pos.index(v[0]+str(int(v[1])+1))] |
| 81 | if a['bg'] == 'red': |
| 82 | a['bg'] = 'blue' |
| 83 | else: |
| 84 | a['bg'] = 'red' |
| 85 | except: |
| 86 | pass |
| 87 | try: |
| 88 | a = buttons[buttons\_pos.index(str(int(v[0])-1)+v[1])] |
| 89 | if a['bg'] == 'red': |
| 90 | a['bg'] = 'blue' |
| 91 | else: |
| 92 | a['bg'] = 'red' |
| 93 | except: |
| 94 | pass |
| 95 | try: |
| 96 | a = buttons[buttons\_pos.index(str(int(v[0])+1)+v[1])] |
| 97 | if a['bg'] == 'red': |
| 98 | a['bg'] = 'blue' |
| 99 | else: |
| 100 | a['bg'] = 'red' |
| 101 | except: |
| 102 | pass |
| 103  104 | wincheck() |
| 105  106  107 | def RestartC(): |
| 108 | global WinWindow |
| 109  110 | try: |
| 111 | GameWindow.destroy() |
| 112 | except: |
| 113 | pass |
| 114  115 | try: |
| 116 | WinWindow.destroy() |
| 117 | except: |
| 118 | pass |
| 119  120  121 | main() |
| 122  123  124 | GameWindow = tk.Tk() |
| 125 | GameWindow.geometry('310x370') |
| 126 | GameWindow.resizable(False, False) |
| 127 | GameWindow.title('Game') |
| 128  129 | buttons = [] |
| 130 | buttons\_pos = [] |
| 131 | col = ['red', 'blue'] |
| 132 | col\_list = [] |
| 133  134 | while True: |
| 135 | for i in range(16): |
| 136 | col\_list.append(col[randint(0, 1)]) |
| 137  138 | if col\_list.count('red') % 2 == 0: |
| 139 | break |
| 140 | elif col\_list.count('red') % 2 == 0: |
| 141 | break |
| 142 | else: |
| 143 | col\_list = [] |
| 144  145  146 | GC00 = tk.Button(GameWindow, |
| 147 | bg=col\_list[0], |
| 148 | width=8, |
| 149 | height=4, |
| 150 | command=lambda : GCC('00')) |
| 151 | GC00.grid(row=0, column=0, |
| 152 | padx=5, pady=5) |
| 153 | buttons.append(GC00) |
| 154 | buttons\_pos.append('00') |
| 155  156 | GC01 = tk.Button(GameWindow, |
| 157 | bg=col\_list[1], |
| 158 | width=8, |
| 159 | height=4, |
| 160 | command=lambda : GCC('01')) |
| 161 | GC01.grid(row=0, column=1, |
| 162 | padx=5, pady=5) |
| 163 | buttons.append(GC01) |
| 164 | buttons\_pos.append('01') |
| 165  166  167 | GC02 = tk.Button(GameWindow, |
| 168 | bg=col\_list[2], |
| 169 | width=8, |
| 170 | height=4, |
| 171 | command=lambda : GCC('02')) |
| 172 | GC02.grid(row=0, column=2, |
| 173 | padx=5, pady=5) |
| 174 | buttons.append(GC02) |
| 175 | buttons\_pos.append('02') |
| 176  177  178 | GC03 = tk.Button(GameWindow, |
| 179 | bg=col\_list[3], |
| 180 | width=8, |
| 181 | height=4, |
| 182 | command=lambda : GCC('03')) |
| 183 | GC03.grid(row=0, column=3, |
| 184 | padx=5, pady=5) |
| 185 | buttons.append(GC03) |
| 186 | buttons\_pos.append('03') |
| 187  188  189 | GC10 = tk.Button(GameWindow, |
| 190 | bg=col\_list[4], |
| 191 | width=8, |
| 192 | height=4, |
| 193 | command=lambda : GCC('10')) |
| 194 | GC10.grid(row=1, column=0, |
| 195 | padx=5, pady=5) |
| 196 | buttons.append(GC10) |
| 197 | buttons\_pos.append('10') |
| 198  199  200 | GC11 = tk.Button(GameWindow, |
| 201 | bg=col\_list[5], |
| 202 | width=8, |
| 203 | height=4, |
| 204 | command=lambda : GCC('11')) |
| 205 | GC11.grid(row=1, column=1, |
| 206 | padx=5, pady=5) |
| 207 | buttons.append(GC11) |
| 208 | buttons\_pos.append('11') |
| 209  210  211 | GC12 = tk.Button(GameWindow, |
| 212 | bg=col\_list[6], |
| 213 | width=8, |
| 214 | height=4, |
| 215 | command=lambda : GCC('12')) |
| 216 | GC12.grid(row=1, column=2, |
| 217 | padx=5, pady=5) |
| 218 | buttons.append(GC12) |
| 219 | buttons\_pos.append('12') |
| 220  221  222 | GC13 = tk.Button(GameWindow, |
| 223 | bg=col\_list[7], |
| 224 | width=8, |
| 225 | height=4, |
| 226 | command=lambda : GCC('13')) |
| 227 | GC13.grid(row=1, column=3, |
| 228 | padx=5, pady=5) |
| 229 | buttons.append(GC13) |
| 230 | buttons\_pos.append('13') |
| 231  232  233 | GC20 = tk.Button(GameWindow, |
| 234 | bg=col\_list[8], |
| 235 | width=8, |
| 236 | height=4, |
| 237 | command=lambda : GCC('20')) |
| 238 | GC20.grid(row=2, column=0, |
| 239 | padx=5, pady=5) |
| 240 | buttons.append(GC20) |
| 241 | buttons\_pos.append('20') |
| 242  243  244 | GC21 = tk.Button(GameWindow, |
| 245 | bg=col\_list[9], |
| 246 | width=8, |
| 247 | height=4, |
| 248 | command=lambda : GCC('21')) |
| 249 | GC21.grid(row=2, column=1, |
| 250 | padx=5, pady=5) |
| 251 | buttons.append(GC21) |
| 252 | buttons\_pos.append('21') |
| 253  254  255 | GC22 = tk.Button(GameWindow, |
| 256 | bg=col\_list[10], |
| 257 | width=8, |
| 258 | height=4, |
| 259 | command=lambda : GCC('22')) |
| 260 | GC22.grid(row=2, column=2, |
| 261 | padx=5, pady=5) |
| 262 | buttons.append(GC22) |
| 263 | buttons\_pos.append('22') |
| 264  265  266 | GC23 = tk.Button(GameWindow, |
| 267 | bg=col\_list[11], |
| 268 | width=8, |
| 269 | height=4, |
| 270 | command=lambda : GCC('23')) |
| 271 | GC23.grid(row=2, column=3, |
| 272 | padx=5, pady=5) |
| 273 | buttons.append(GC23) |
| 274 | buttons\_pos.append('23') |
| 275  276  277 | GC30 = tk.Button(GameWindow, |
| 278 | bg=col\_list[12], |
| 279 | width=8, |
| 280 | height=4, |
| 281 | command=lambda : GCC('30')) |
| 282 | GC30.grid(row=3, column=0, |
| 283 | padx=5, pady=5) |
| 284 | buttons.append(GC30) |
| 285 | buttons\_pos.append('30') |
| 286  287  288 | GC31 = tk.Button(GameWindow, |
| 289 | bg=col\_list[13], |
| 290 | width=8, |
| 291 | height=4, |
| 292 | command=lambda : GCC('31')) |
| 293 | GC31.grid(row=3, column=1, |
| 294 | padx=5, pady=5) |
| 295 | buttons.append(GC31) |
| 296 | buttons\_pos.append('31') |
| 297  298  299 | GC32 = tk.Button(GameWindow, |
| 300 | bg=col\_list[14], |
| 301 | width=8, |
| 302 | height=4, |
| 303 | command=lambda : GCC('32')) |
| 304 | GC32.grid(row=3, column=2, |
| 305 | padx=5, pady=5) |
| 306 | buttons.append(GC32) |
| 307 | buttons\_pos.append('32') |
| 308  309  310 | GC33 = tk.Button(GameWindow, |
| 311 | bg=col\_list[15], |
| 312 | width=8, |
| 313 | height=4, |
| 314 | command=lambda : GCC('33')) |
| 315 | GC33.grid(row=3, column=3, |
| 316 | padx=5, pady=5) |
| 317 | buttons.append(GC33) |
| 318 | buttons\_pos.append('33') |
| 319  320 | restartB = tk.Button(GameWindow, |
| 321 | bg='White', |
| 322 | width=10, |
| 323 | height=1, |
| 324 | text='Перезапуск', |
| 325 | font=('Arial', '14'), |
| 326 | command=RestartC) |
| 327 | restartB.place(x=90, y=325) |
| 328  329  330 | GameWindow.mainloop() |
| 331 | main() |

# **3.2 Тестирование функциональных и нефункциональных характеристик программы**

Функциональное тестирование направлено на проверку того, выполняет ли приложение заявленные функции. В первую очередь следует проверить корректность отображения поля, т.е. как отображаются фишки и не возникает ли проблемы с их расположением относительно друг друга и границ экрана (Рисунок 3).

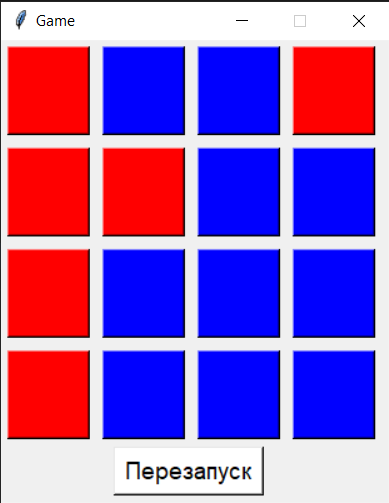


Рисунок 3 – Иллюстрация главного окна программы

Затем следует проверить, работу вывод окна «победы», когда все фишки станут одного цвета (Рисунок 4).

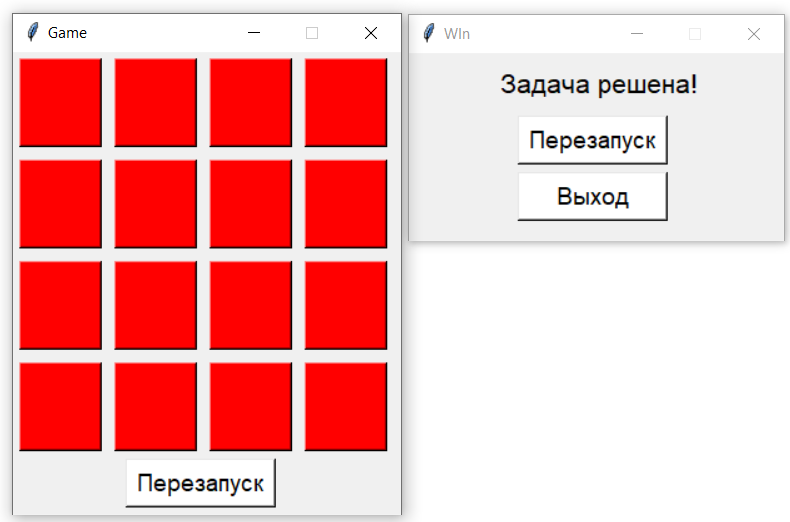
****

Рисунок 4 – Главное окно и окно «победы»

# **4 Руководство пользователя**

# **4.1 Назначение программы**

Игра «Переверни фишки» направлена на развитие логического мышления пользователя. Разработка стратегии для решение такой задачи на первый взгляд кажется простым, но на деле это очень сложная задача, ведь требуется не просто путём переворота фишек закрасить всё поле, но и не попасть в «ловушку» ⎯ ситуацию, когда задача становиться не решаемой.

# **4.2 Запуск программы**

1. Убедитесь, что у вас установлен Python и откройте терминал в соответствующей программе.

2. Сохраните код программы в файл с расширением «.py», например, main.py. Обратите внимание, что для выполнения файлов зачастую недопустимо появление любых букв кроме латиницы или пробелов на пути к расположению файла;

3. Откройте терминал или командную строку;

4. Перейдите в каталог, где сохранен файл программы;

5. Запустите программу с помощью команды «python (имя файла)».

# **4.3 Выполнение основных функций**

При запуске программы появляется окно, в котором расположены 16 фишек разных цветов. При нажатии на одну из них, её цвет и цвет всех соседних фишек по вертикали и горизонтали меняются на противоположный.

В случае если пользователь попал в «ловушку» или просто хочет сыграть заново, изменив начальное положение фишек, требуется нажать кнопку «Перезапуск».

После решения задачи у пользователя есть два выбора. Сыграть снова нажав кнопку «Перезапуск» или завершить программу нажав кнопку «выход».

# Заключение

В ходе выполнения курсовой работы по теме разработка логической игры «Переверни фишки» была разработана программа с применением графического интерфейса на базе языка программирования python и встроенной библиотеки tkinter. Были изучены различные новые методы и функции данной библиотеки. Также были выполнены все поставленные задачи и достигнута основанная цель проекта – разработка программы и выполнение графического интерфейса.

Игра «Переверни фишки» действительна уникальна, она помогает развить логическое и стратегическое мышление при помощи очень простой задачи. Сталкиваясь с подобным человек не всегда способен быстро найти выход из сложившейся ситуации, но при должном умении и желании найти выход можно всегда.

# Библиографический список

1. Mathcha [Электронный ресурс] // Онлайн WYSIWYG редактор: https://www.mathcha.io/− Дата обращения 11.01.2025

2. Руководство по Tkinter // Metanit [Электронный ресурс] // Онлайн блог о программировании: https://metanit.com/python/tkinter/− Дата обращения 28.11.2024

3. Draw.io [Электронный ресурс] // Облачная платформа для создания визуального контента: https://www.drawio.com/ − Дата обращения 11.01.2025

4. Pythonru [Электронный ресурс] //Форум и электронная школа по языку программирования Python: https://pythonru.com – Дата обращения 23.12.2024