**ДОМАШНЯЯ РАБОТА №2**

***Задание***

*«На уроке мы разрабатывали игру крестики нолики для игры с компьютером.*

*В качестве домашнего задания нужно обновить написанный на уроках код, чтобы можно было играть в следующий режимах Player vs Player, Player vs AI, AI vs Player, AI vs AI.»*

*Напомню условия задачи:*

*:*

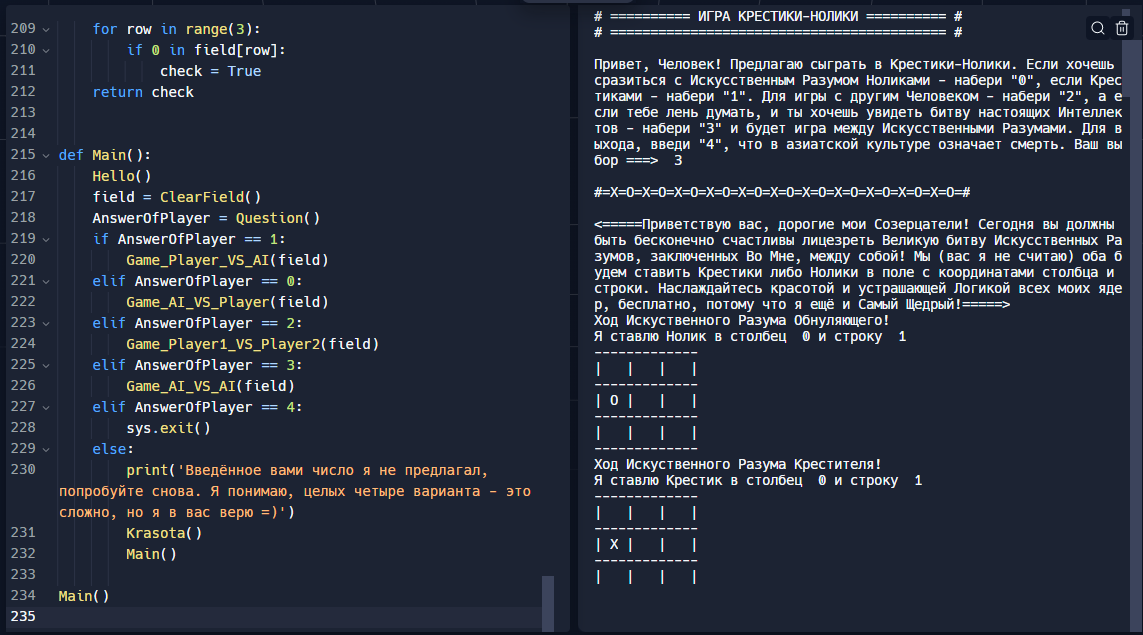
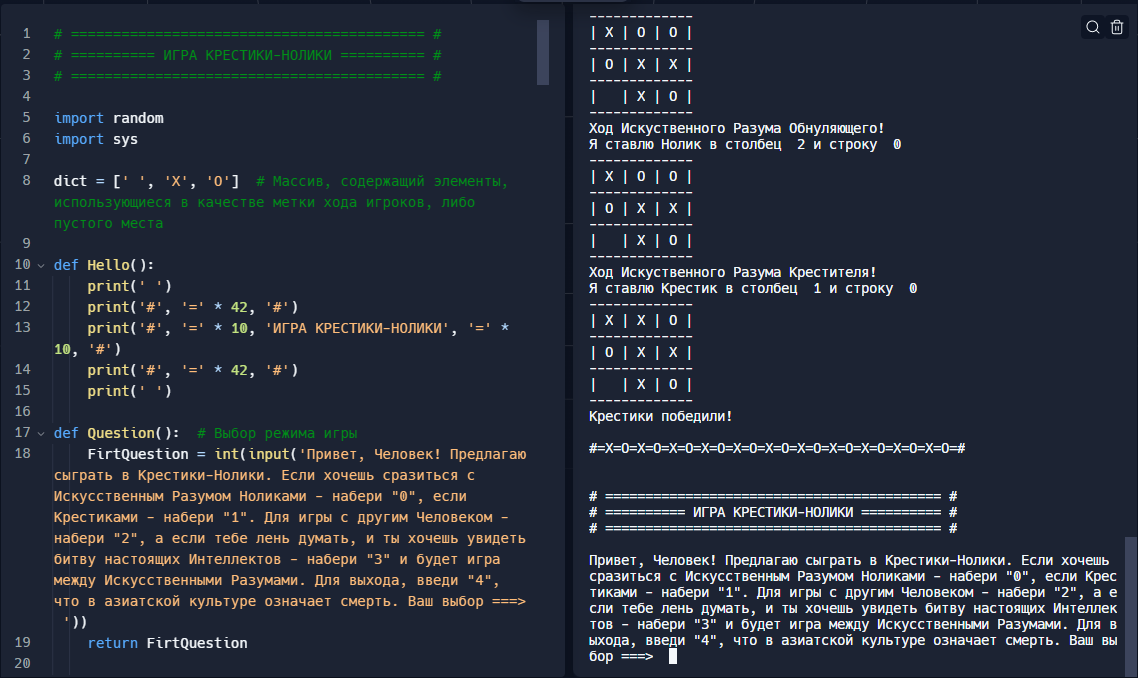
|  |
| --- |
| *Задача:*   1. *Написать игру крестики нолики.* 2. *Игра должна работать из консоли (то есть должна быть использована псевдографика).* 3. *После хода каждого из игроков должно показываться обновленное поле.* 4. *Во время хода AI должна показываться фраза от AI - “ставлю <крестик/нолик> в [<y>][<x>]”* 5. *AI ставит крестик или нолик с случайную точку поля.* 6. *Режим Player vs Player (hotseat), оба пользователя вводят с клавиатуры куда ставить крестик или нолик.* 7. *Режим Player vs AI, игрок вводящий с клавиатуры играет за крестики, AI за нолики.* 8. *Режим AI vs Player, игрок вводящий с клавиатуры играет за нолики, AI за крестики.* 9. *Режим AI vs AI, нельзя играть с клавиатуры, игра играет сама с собой.* |
| *Ограничения:*   1. *Игра должна работать из консоли.* 2. *Для игры должна использоваться псевдографика.* 3. *Массив field должен содержать числа.* |
| *Подсказки, найденные на паре:*   1. *Все необходимое детально разобрано на паре* |
| *Что будет оцениваться:*   1. *Работа программы и ее функционал* 2. *Именование переменных* 3. *Наличие функций и их именование.* 4. *Структура проекта (насколько легко разобраться в том что и как работает)* |

Мне удалось довести игру до рабочего состояния, все указанные в ТЗ требования соблюдены, в чём можно убедиться, обратившись к результатам небольшого тестирования, приведённым в конце этого документа.

|  |
| --- |
| * # ================================================================================= # * # ============================== ИГРА КРЕСТИКИ-НОЛИКИ ============================= # * # ================================================================================= # * **import** random * **import** sys * dict **=** [' ', 'X', 'O']  # Массив, содержащий элементы, использующиеся в качестве метки хода игроков, либо пустого места * **def** Hello():# Отображение названия игры * print(' ') * print('#', '=' **\*** 42, '#') * print('#', '=' **\*** 10, 'ИГРА КРЕСТИКИ-НОЛИКИ', '=' **\*** 10, '#') * print('#', '=' **\*** 42, '#') * print(' ') * **def** Question():  # Выбор режима игры * FirtQuestion **=** int(input('Привет, Человек! Предлагаю сыграть в Крестики-Нолики. Если хочешь сразиться с Искусственным Разумом Ноликами - набери "0", если Крестиками - набери "1". Для игры с другим Человеком - набери "2", а если тебе лень думать, и ты хочешь увидеть битву настоящих Интеллектов - набери "3" и будет игра между Искусcтвенными Разумами. Для выхода, введи "4", что в азиатской культуре означает смерть. Ваш выбор ===>  ')) * **return** FirtQuestion * **def** GetFieldPos(field, row, column):  # Определение координат хода * fpos **=** field[row][column] * **return** dict[fpos] * **def** ShowField(field):  # Отрисовка поля и расстановка мест хода игроков * **for** row **in** range(3): * print('-' **\*** 13) * **for** column **in** range(3): * print(f'| {GetFieldPos(field, row, column)} ', end**=**'') * print('|') * print('-'**\***13) * **def** ClearField():  # Двумерный массив, представляющий собою игровое поле 3 х 3 * **return**[[0, 0, 0], * [0, 0, 0], * [0, 0, 0]] * **def** Krasota():  # Элементы псевдографики для визуального разделения партий * print(' ') * print('#', '=X=O' **\*** 11, '=', '#', sep**=**'') * print(' ') * **def** Player(field):  # Ход Икгрока за Крестики * print('Ваш ход, Игрок за Крестики! ') * x **=** int(input('Введите номер колонки: 0, 1 или 2 ==> ')) * y **=** int(input('Введите номер строки: 0, 1 или 2 ==> ')) * print('Игрок за Крестики, Вы походили Крестиком на столбец: ', x, 'и строку', y) * field[y][x] **=** 1 * **def** PlayerTWO(field):  # Ход Игрока за Нолики * print('Ваш ход, Игрок за Нолики! ') * x **=** int(input('Введите номер колонки: 0, 1 или 2 ==> ')) * y **=** int(input('Введите номер строки: 0, 1 или 2 ==> ')) * print('Игрок за Нолики, Вы походили Ноликом на столбец: ', x, 'и строку', y) * field[y][x] **=** 2 * **def** AITurn(field):  # Ход Первого Компьютера, Нолика * print('Ход Искуственного Разума Обнуляющего! ') * x **=** random.randint(0, 2) * y **=** random.randint(0, 2) * field[y][x] **=** 2 * print('Я ставлю Нолик в столбец ', x, 'и строку ', y) * **def** AITWOTurn(field):  # Ход Второго Компьютера, Крестика * print('Ход Искуственного Разума Крестителя! ') * x **=** random.randint(0, 2) * y **=** random.randint(0, 2) * field[y][x] **=** 1 * print('Я ставлю Крестик в столбец ', x, 'и строку ', y) * **def** VerticalXXXWin(field):  # Проверка победы Крестиков во всех вариантах, кроме строк * XVertical **=** False * **if** (field[1][1] **==** 1 **and** ((field[0][1] **==** 1 **and** field[2][1] **==** 1) **or** (field[0][0] **==** 1 **and** field[2][2] **==** 1) **or** (field[0][2] **==** 1 **and** field[2][0] **==** 1))) **or** (field[0][0] **==** 1 **and** field[1][0] **==** 1 **and** field[2][0] **==** 1) **or** (field[0][1] **==** 1 **and** field[1][1] **==** 1 **and** field[2][1] **==** 1) **or** (field[0][2] **==** 1 **and** field[1][2] **==** 1 **and** field[2][2] **==** 1): * XVertical **=** True * **return** XVertical * **def** VerticalOOOWin(field):  # Проверка победы Ноликов во всех вариантах, кроме строк * OVertical **=** False * **if** (field[1][1] **==** 2 **and** ((field[0][1] **==** 2 **and** field[2][1] **==** 2) **or** (field[0][0] **==** 2 **and** field[2][2] **==** 2) **or** (field[0][2] **==** 2 **and** field[2][0] **==** 2))) **or** (field[0][0] **==** 2 **and** field[1][0] **==** 2 **and** field[2][0] **==** 2) **or** (field[0][1] **==** 2 **and** field[1][1] **==** 2 **and** field[2][1] **==** 2) **or** (field[0][2] **==** 2 **and** field[1][2] **==** 2 **and** field[2][2] **==** 2): * OVertical **=** True * **return** OVertical   * **def** CheckForWinXXX(field):  # Проверка победы Крестиков во всех вариантах * XXX **=** False * **for** row **in** range(3): * **if** (0 **not** **in** field[row] **and** 2 **not** **in** field[row]): * XXX **=** True * print('Крестики победили!') * **if** VerticalXXXWin(field) **is** True: * XXX **=** True * print('Крестики победили!') * **return** XXX * **def** CheckForWinOOO(field):  # Проверка победы Ноликов во всех вариантах * OOO **=** False * **for** row **in** range(3): * **if** 0 **not** **in** field[row] **and** 1 **not** **in** field[row]: * OOO **=** True * print('Нолики победили!') * **if** VerticalOOOWin(field) **is** True: * OOO **=** True * print('Нолики победили!') * **return** OOO * **def** Game\_Player\_VS\_AI(field):  # Игра Человека против Компьютера * Krasota() * print('<=====Приветсвую вас в неравной схватке человека и Великолепного Богоподобного Искуственного Разума! Человек может ставить Крестики в поле с координатами столбца и строки, а Великий Могучий Беспринципный Искусственный разум ставит Нолик куда захочет! Ха-ха! Начнём-с!=====> ') * OOOWins **=** False * XXXWins **=** False * isEnd **=** False * **while** isEnd **is** False **and** XXXWins **is** False **and** OOOWins **is** False: * Player(field) * ShowField(field) * XXXWins **=** CheckForWinXXX(field)  # Проверка победы Крестиков * **if** XXXWins **is** True: * **break** * AITurn(field) * ShowField(field) * OOOWins **=** CheckForWinOOO(field)  # Проверка победы Ноликов * **if** OOOWins **is** True: * **Break** * isEnd **=** **not** CheckingForZero(field) * **if** isEnd **is** True: * print('Поле заполнено. Ничья, господа!') * Krasota() * Main() * **def** Game\_AI\_VS\_Player(field):  # Игра Компьютера за Крестики против Человека за Нолики * Krasota() * print('<=====Приветсвую вас в неравной схватке человека и Великолепного Богоподобного Искуственного Разума! Человек может ставить Нолики в поле с координатами столбца и строки, а Великий Могучий Беспринцыпный Искусственный разум ставит Крест куда захочет! Ха-ха! Начнём-с!=====> ') * OOOWins **=** False * XXXWins **=** False * isEnd **=** False * **while** isEnd **is** False **and** XXXWins **is** False **and** OOOWins **is** False: * AITWOTurn(field) * ShowField(field) * XXXWins **=** CheckForWinXXX(field)  # Проверка победы Крестиков * **if** XXXWins **is** True: * **break** * PlayerTWO(field) * ShowField(field) * OOOWins **=** CheckForWinOOO(field)  # Проверка победы Ноликов * **if** OOOWins **is** True: * **break** * isEnd **=** **not** CheckingForZero(field) * **if** isEnd **is** True: * print('Поле заполнено. Ничья, господа!') * Krasota() * Main() * **def** Game\_Player1\_VS\_Player2(field):  # Игра между Людьми * Krasota() * print('<=====Приветсвую вас, дорогие мои человечки! Вы оба можете ставить Крестики либо Нолики в поле с координатами столбца и строки. Играйте, наслаждайтесь, используйте мой Невероятный Разум для примитивной игры, для которой вы ухитрились не найти бумажку и карандаш! К Вашим услугам все мои 8 ядер!=====> ') * OOOWins **=** False * XXXWins **=** False * isEnd **=** False * **while** isEnd **is** False **and** XXXWins **is** False **and** OOOWins **is** False: * Player(field) * ShowField(field) * XXXWins **=** CheckForWinXXX(field)  # Проверка победы Крестиков * **if** XXXWins **is** True: * **break** * PlayerTWO(field) * ShowField(field) * OOOWins **=** CheckForWinOOO(field)  # Проверка победы Ноликов * **if** OOOWins **is** True: * **break** * isEnd **=** **not** CheckingForZero(field) * **if** isEnd **is** True: * print('Поле заполнено. Ничья, господа!') * Krasota() * Main() * **def** Game\_AI\_VS\_AI(field):  # Игра Компьютера с самим собой * Krasota() * print('<=====Приветствую вас, дорогие мои Созерцатели! Сегодня вы должны быть бесконечно счастливы лицезреть Великую битву Искусственных Разумов, заключенных Во Мне, между собой! Мы (вас я не считаю) оба будем ставить Крестики либо Нолики в поле с координатами столбца и строки. Наслаждайтесь красотой и устрашающей Логикой всех моих ядер, бесплатно, потому что я ещё и Самый Щедрый!=====> ') * OOOWins **=** False * XXXWins **=** False * isEnd **=** False * **while** isEnd **is** False **and** XXXWins **is** False **and** OOOWins **is** False: * AITWOTurn(field) * ShowField(field) * XXXWins **=** CheckForWinXXX(field)  # Проверка победы Крестиков * **if** XXXWins **is** True: * **break** * AITurn(field) * ShowField(field) * OOOWins **=** CheckForWinOOO(field)  # Проверка победы Ноликов * **if** OOOWins **is** True: * **break** * isEnd **=** **not** CheckingForZero(field) * **if** isEnd **is** True: * print('Поле заполнено. Ничья, господа Компьютеры!') * Krasota() * Main() * **def** CheckingForZero(field):  # Проверка на наличие нулей в массивах, играющих роль строк * check **=** False * **for** row **in** range(3): * **if** 0 **in** field[row]: * check **=** True * **return** check  * **def** Main(): * Hello() * field **=** ClearField() * AnswerOfPlayer **=** Question() * **if** AnswerOfPlayer **==** 1: * Game\_Player\_VS\_AI(field) * **elif** AnswerOfPlayer **==** 0: * Game\_AI\_VS\_Player(field) * **elif** AnswerOfPlayer **==** 2: * Game\_Player1\_VS\_Player2(field) * **elif** AnswerOfPlayer **==** 3: * Game\_AI\_VS\_AI(field) * **elif** AnswerOfPlayer **==** 4: * sys.exit() * **else**: * print('Введённое вами число я не предлагал, попробуйте снова. Я понимаю, целых пять вариантов — это сложно, но я в вас верю =)') * Krasota() * Main() * Main() |

**Тестирование. Чек лист. Скриншоты.**

|  |  |  |
| --- | --- | --- |
| **ID** | **Функциональный контроль Игры Крестики-Нолики** | **CL#001** |
| **Sys Info** | Windows 10, Python 3.11 | |
| **Version** | 0.2 | |
| **Precondition** | Запустить файл ХоХ2.py в консоли Python | |
| **Check** | | **Status** |
| **№** | **Запуск игры** | |
| 1 | Отображается название игры | passed |
| 2 | Отображается приветсвие и выбор режима игры | passed |
| 3 | Возможен ввод символов для выбора режима игры | passed |
|  | **Проверка режимов игры** | |
| 4 | Запускается режим "0" | passed |
| 5 | Режим "0" соответсвует описанию | passed |
| 6 | Порядок ходов сответсвует режиму игры | passed |
| 7 | Поле игры отображается корректно (3 х 3 клетки) | passed |
| 8 | Метка игрока устанавливается правильно, согласно введённым координатам | passed |
| 9 | Метка Компьютера устанавливается правильно, согласно сгенерированным координатам | passed |
| 10 | Метки хода игроков соответсвует режиму "0" | passed |
| 11 | Возможна победа Ноликов | passed |
| 12 | Возможна победа Крестиков | passed |
| 13 | Возможна ничья | passed |
| 14 | После окончания партии появляется Приветсвие и выбор режима игры | passed |
| 15 | Шаги с 4 по 14 для режимов работают корректно для всех режимов (0, 1, 2, 3) | passed |
|  | **Завершение работы игры** | |
| 16 | В каждом из четырёх режимов ввод цифры 4 после окончания партии прекращает исполнение программы | passed |
| 17 | Ввод любой цифры, кроме 0-4, вызывает сообщение об ошибке и повтор выбора режима игры | passed |
|  |  | |

****

В ходе тестирования были успешно проверены критерии:

- соответствия внешнего вида игры ТЗ;

- соответствия функционала игры ТЗ;

- корректности работы игры во всех четырёх комбинациях игроков;

- достигнуты все варианты завершения партии в каждом режиме;

- проверена функция завершения работы игры;

- а также ввод недопустимых значений при выборе режима игры.