Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирование (ОАиП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему:

Игровое приложение «Реверси»

Студент: гр. 451005

Грамович В. А.

Руководитель:

Данилова Г. В.

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ПОИТ

––––––––––––––––––––––––

(подпись)

––––––––––––––––– 2015 г.

ЗАДАНИЕ

по курсовому проектированию

Студенту    Грамовичу Владиславу Александровичу––––––––––––––––––––

1. Тема работы   Игровое приложение   «Реверси»

2. Срок сдачи студентом законченной работы–\_\_\_\_\_\_\_\_\_\_\_\_–

3. Исходные данные к работе ———

4. Содержание расчётно-пояснительной записки (перечень вопросов, которые подлежат разработке):

Введение;

1. Анализ предметной области;

2. Разработка программного средства;

3. Тестирование программного средства;

4. Руководство пользователя;

Заключение.

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

1. Схема алгоритма работы системы

6. Консультант по курсовой работе Данилова Г.В.

7. Дата выдачи задания 12.02.2015 г.–––––––––––––––––––––––   –

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и процентом от общего объёма работы):

раздел 1,2 к 20.02.2015 – 15 % готовности работы;

разделы 3, 4 к 15.03.2015 – 30 % готовности работы;

разделы 5, 6, к 07.04.2015 – 60 % готовности работы;

раздел 7, 8, 9 к 27.04.2015 – 90 % готовности работы;

оформление пояснительной записки и графического материала к 05.05.2015- 100 % готовности работы.

Защита курсового проекта с 06 мая 2015.

РУКОВОДИТЕЛЬ - Данилова Г.В.

(подпись)

Задание принял к исполнению –––\_\_\_\_–– Грамович В. А.12.02.2015 г.

(дата и подпись студента)

**СОДЕРЖАНИЕ**

Введение 4

1 Анализ предметной области 5

1.1 Обзор аналогов 5

1.2 Постановка задачи 7

2 Разработка программного средства 8

2.1 Игровое меню 8

2.2 Размерность поля 8

2.3 Объекты игрового поля 10

2.4 Инверсия фишек 11

2.5 Блокировка фишек 13

2.6 Реализация искуственного интеллекта 14

2.7 Режим игры "Два игрока" 16

2.8 Подсветка возможных вариантов хода 13

3 Тестирование программного средства 19

4 Руководство пользователя 20

4.1 Начало работы 20

Заключение 24

Приложение А. Исходный код программы 26

Приложение Б. Блок-схема функции TStick.Update 59

ВВЕДЕНИЕ

Игры всегда занимали умы людей и будоражили их сознания. Никто точно не может сказать, когда была придумана первая игра, но ясно, что с тех пор, как она появилась, игры стали неотъемлемой частью человеческого существования. В средние века карточные игры развлекали и королей и простой народ. Наиболее популярные и древние интеллектуальные игры это шахматы, домино, карты и игра Го. История многих игр, уходит в глубину веков и окутана легендами и тайнами.

В эру высоких технологий игры стали развиваться ещё интенсивнее. Появление персональных компьютеров позволило автоматизировать большинство игр и создать реалистичные симуляторы. Сейчас уже никого не удивишь возможностью при помощи компьютера вместе с другими людьми проехаться на Формуле 1, принять участие в боевых действиях. Есть также игры, развивающие логическое мышление. Именно такой игрой является Реверси.

Как и многие игры, реверси довольно распространены в Интернете. Однако, отсутствие «культового» статуса, позволяет наткнуться в онлайне на игроков мирового. Практически все уважающие себя гейм-порталы имеют раздел реверси, однако вследствие того, что компьютеры играют намного лучше людей, в Интернете считается хорошим тоном играть только блицы (обычно до двух минут на каждого игрока).

1. анализ предметной области

В данном разделе будет сравнение приложения с аналогами и постановка задач приложения.

* 1. Обзор аналогов

Игра была изобретена в Великобритании в 1880 году и пользовалась большой популярностью, но впоследствии была забыта. Возродили её в Японии, где она в 1971 году под названием отелло(рисунок 1.1) вновь стала популярна. С 1977 года регулярно проводятся чемпионаты мира по игре в реверси.

Реверси является стратегической игрой, схожей с шашками и шахматами. Так же как и в шахматах, принято разделять партию на три части: дебют (начало), миттельшпиль (середина игры) и эндшпиль (концовка). Однако, в отличие от шахмат, количество возможных дебютов здесь намного меньше, и все они легко запоминаются. Все сколько-либо серьёзные игроки знают дебюты на 5-6 ходов вперёд, чтобы избежать заведомо проигрышных ходов на данной стадии. Миттельшпиль, пожалуй, является наиболее «свободной» и одновременно сложной частью игры, когда положение можно либо упрочить, либо изменить в свою пользу. Несмотря на это, многие, казалось бы, проигранные в миттельшпиле партии обретают новые качества при вступлении в конечную стадию игры — эндшпиль. Золотое правило концовки — не спешить и считать. Считать принято фишки, которые результируют конечный исход игры для конкретной тактики. Естественно, количество исходов зависит от того, с какого хода начинать считать, и именно поэтому компьютеры могут позволить себе просчитать все возможные варианты (их, по компьютерным меркам, немного) и всегда выбирают тот, при котором минимизируется результат человека и максимизируются очки компьютера.

Компьютерные программы реверси уже с середины 1990-х годов играют намного сильнее людей. Программа Logistello в 1997 году обыграла чемпиона мира Такэси Мураками 6:0.

В игре существует порядка 1028 позиций, и около 1058 возможных партий.



Рисунок 1.1 - Отелло

Варианты реверси:

Реверси n × n

Игра на поле n × n клеток. От игры 8 × 8 отличается тем, что фишки одного цвета в начале игры ставятся не в шахматном порядке, а рядом. Существуют варианты реверси с размером поля 10 × 10(рисунок 1.2) и больше. Они не отличаются от обычных ничем, кроме размера поля. В целом, варианты размером меньше 8 × 8 не представляют интереса, поскольку являются детерминированными и при идеальной стратегии всегда выигрывает второй игрок (тот, кто ходит вторым).

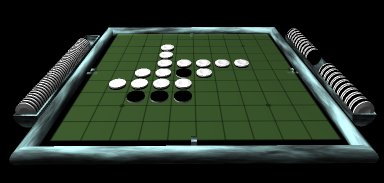


рисунок 1.2 - Реверси 10х10

Антиреверси

Отличается только тем, что при подведении результатов игры выигрывает тот, у кого фишек меньше.

Реверси с чёрной дырой

Отличается только тем, что одна из клеток доски (случайно выбирается в начале игры) помечается как «черная дыра». При этом на неё нельзя сделать ход, и фишки с одной стороны такой клетки не могут захватить фишки с другой

* 1. Постановка задачи

В рамках данного курсового проекта планируется разработать игровое программное средство.

Проанализировав предметную область можно поставить ряд задач:

разработать игровое программное средство «Реверси»;

необходимо реализовать обработку коллизий объектов игрового поля;

реализовать возможность игры против компьютера и против человека;

реализовать возможность подсветки доступных ходов;

разработать возможность выбора размера поля и режима игры;

использовать графические возможности среды разработки Embarcadero RAD Studio XE7;;

создать игровое меню.

При проектировании программы использовать метод нисходящего проектирования: сначала проработать основные концепции программы, затем детально продумать отдельные функциональные составляющие.

Для разработки программного средства использовать язык программирования Delphi. Среда для разработки – Embarcadero RAD Studio XE7. Использование данной среды разработки дает множество возможностей в создании оконных приложений Windows. С помощью данной среды очень удобно и быстро разработать качественное оконное приложение, создать дружественный интерфейс пользователя, обеспечить удобные средства для ввода, редактирования, сохранения и отображения данных.

1. РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА

Для создания программы используется язык программирования Object Pascal и среду разработки Embarcadero RAD Studio XE7. Использование данной среды позволяет разрабатывать качественные приложения для семейства операционных систем Windows. Целью курсовой работы является создать простое в использовании программное средство «реверси».

* 1. Игровое меню

Первым шагом при создании компьютерной игры «Реверси» является формирование визуального представления. Главная задача состоит в создании пользовательского интерфейса, удобного для использования даже самым неопытным пользователем.

В первую очередь создается главное меню, которое содержит в себе все основные компоненты программы. Размер данного окна исходит из размеров внутренних компонентов.

Меню состоит из следующих кнопок и вариантов режима игры:

* «Начать игру» – запускается новая игра;
* «Об авторе» – открывается информация авторе;
* «Справка» – открываться справка (правило игры);
* «Выход из игры» – завершение игры;
* «Выберите размерность поля» – устанавливается размер поля;
* «Выберите режим игры» – устанавливается уровень сложности или режим «Два игрока»
* «Подсвечивать возможные ходы» – устанавливается цвет для доступных ходов

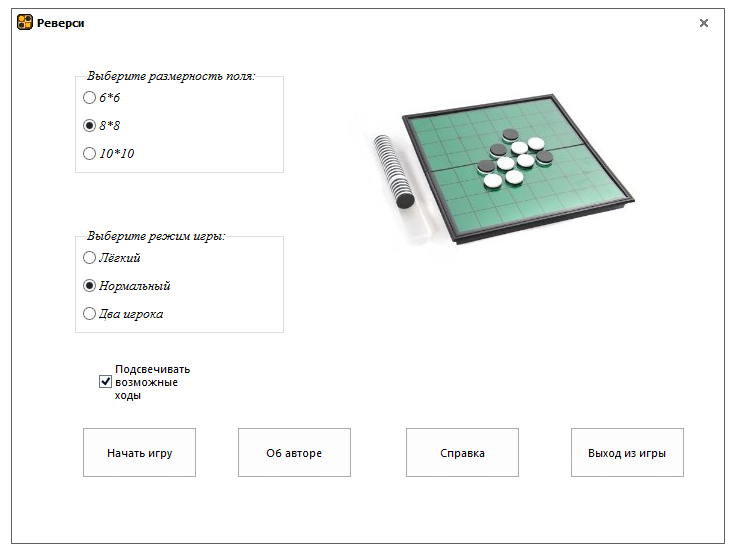
При запуске программного средства появляется главное окно, оно же является главным меню и содержит все выше перечисленные ссылки (рисунок 2.1).

Для создания формы используется компоненты Memo, RadioGroup, Button, CheckBox. При создании формы, компонент Memo скрыт, но по нажатию на некоторые Button он становится видимым.

Главное окно готово.

* 1. Размерность поля

При выборе размерности поя в компоненте Size\_fild пользователь устанавливает начальное количество фишек, а также положение фишек,

Рисунок 2.1 – Главное меню

которые доступны ко времени начала игры.

Ниже приведен код процедуры :

procedure TGame.start\_position;

var

position: integer;

begin

position := n div 2;

chips.images[position, position].Picture.LoadFromFile('./123.jpg');

chips.images[position + 1, position + 1].Picture.LoadFromFile('./123.jpg');

chips.images[position + 1, position].Picture.LoadFromFile('./321.jpg');

chips.images[position, position + 1].Picture.LoadFromFile('./321.jpg');

chips.images[position, position].Enabled := false;

chips.images[position + 1, position].Enabled := false;

chips.images[position, position + 1].Enabled := false;

chips.images[position + 1, position + 1].Enabled := false;

chips.color\_chips[position, position] := 1;

chips.color\_chips[position + 1, position + 1] := 1;

chips.color\_chips[position + 1, position] := 2;

chips.color\_chips[position, position + 1] := 2;

end;

Модуль формируется следующим образом: при начале игры открывается основная форма, в которой изображается само поле и начальные фишки.

* 1. Объекты игрового поля

Ниже приведено описание класса Tchips полями которого являются такие характеристики, как расположение, изображение, количество, подсветка и возможность инверсии объекта. Данный класс используется для поиска и передачи возможных координат, куда может походить искусственный интеллект .

Ниже приведен код описание типа класс:

tchips = record

images: array of array of TImage;

color\_chips: array of array of byte;

count\_b:integer;

count\_w:integer;

number\_of\_inverted:integer;

x\_backlight:integer;

y\_backlight: integer;

end;

Переменная images отвечает за хранение изображений различных режимов фишки, а именно:

Пустая (Рисунок 2.2) - показывает, что фишка не занята;

C:\Users\техно\Desktop\Курсач, только прога\000.jpg

Рисунок 2.2 - Пустая

Белая (Рисунок 2.3) - показывает, что фишка занята первым игроком;

C:\Users\техно\Desktop\Курсач, только прога\321.jpg

Рисунок 2.3 - Жёлтая

Чёрная (Рисунок 2.4) - показывает, что фишка занята вторым игроком или искусственным интеллектом;

C:\Users\техно\Desktop\Курсач, только прога\123.jpg

Рисунок 2.4 - Чёрная

Возможного хода (Рисунок 2.5) - показывает, что эту фишку можно выбрать, если ход возможен и в меню выбран режим «Подсвечивать возможные ходы»;

C:\Users\техно\Desktop\Курсач, только прога\555.jpg

Рисунок 2.5 - Возможность хода

Положение курсора (Рисунок 2.6) - показывает текущее местоположение фишки;

C:\Users\техно\Desktop\Курсач, только прога\6666.jpg

Рисунок 2.6 - Положение курсора

C помощью переменной color\_chips узнаётся цвет фишки; x\_backlight, y\_backlight - указывают положение курсора; для вычисления количества белых и чёрных фишек используются count\_b и count\_w; переменная number\_of\_inverted показывает можно ли поставить фишку в нужное место.

* 1. Инверсия фишек

Фишки на игровом поле инвертируются в восьми направлениях: вверх, вниз, налево, направо, по диагоналям вниз и влево, вверх и влево, вниз и в право, вверх и в лево . Каждый раз инвертируется как минимум одна фишка в заданном направлении. В первую очередь следует установить координаты только что поставленного элемента, а затем получив их, установить положение фишек, которые нужно перевернуть, и инвертировать их.

В качестве примера ниже приведён код инверсии фишек:

по горизонтали:

procedure TGame.reversi\_x(x, y: integer);

var

a, temp: integer;

begin

for i := y - 1 downto 0 do

if (chips.color\_chips[x, i] = chips.color\_chips[x, y]) then

for a := i + 1 to y - 1 do

if chips.color\_chips[x, y] = 1 then

begin

chips.images[x, a].Picture.LoadFromFile('./123.jpg');

chips.color\_chips[x, a] := 1;

end

else

begin

chips.images[x, a].Picture.LoadFromFile('./321.jpg');

chips.color\_chips[x, a] := 2;

end;

end;

В коде, описанном выше, используется 2 цикла: в первом просматриваются элементы и высчитывается координаты точки, до которой будет производиться инвертирование, второй нужен для переворачивания фишек и для установления значения режима фишки.

- по диагонали :

procedure TGame.reversi\_xy(x, y: integer);

var

temp\_x, temp\_y: integer;

begin

temp\_x := x - 1;

temp\_y := y - 1;

while (temp\_x >= 0) or (temp\_y >= 0) do

begin

if (chips.color\_chips[temp\_x, temp\_y] = 0) or

(chips.color\_chips[temp\_x, temp\_y] = chips.color\_chips[x, y]) then

exit

else

dec(temp\_x);

dec(temp\_y);

if chips.color\_chips[temp\_x, temp\_y] = chips.color\_chips[x, y] then

begin

inc(temp\_x);

inc(temp\_y);

while (temp\_x < x) and (temp\_y < y) do

begin

inversy(temp\_x, temp\_y, x, y);

inc(temp\_x);

inc(temp\_y);

end;

end;

end;

end;

Этот алгоритм схож с процедурой инвертирования по недиагональным сторонам, но в нём движение происходит сразу по двум координатам и для инвертирования фишек используется отдельная процедура, а именно inversy(temp\_x, temp\_y, x, y).

Код данной процедуры:

procedure TGame.inversy(temp\_x, temp\_y, x, y: integer);

begin

if chips.color\_chips[x, y] = 1 then

begin

chips.images[temp\_x, temp\_y].Picture.LoadFromFile('./123.jpg');

chips.color\_chips[temp\_x, temp\_y] := 1;

end

else

begin

chips.images[temp\_x, temp\_y].Picture.LoadFromFile('./321.jpg');

chips.color\_chips[temp\_x, temp\_y] := 2;

end;

end;

* 1. Блокировка фишек

В игре необходима процедура блокировки фишек, походя на которые не будет инвертирована ни одна из фишек противника. Она схоже с процедурой инверсии фишек, но блокировка не переворачивает их, а только считает перевёрнутые и, если количество таких фишек равно нулю, то в данное место поставить фишку нельзя.

В качестве примера ниже приведён пример кода подсчёта возможно-инвертируемых фишек:

procedure reversi\_x\_1(x, y: integer);

var

temp: integer;

begin

for i := y - 1 downto 0 do

if (chips.color\_chips[x, i] = 0) then

break

else if (chips.color\_chips[x, i] = chips.color\_chips[x, y]) then

begin

for temp := i + 1 to y - 1 do

inc(chips.number\_of\_inverted);

break;

end;

end;

При помощи этого модуля формируются следующие функции игры:

* блокировка хода;
* искусственный интеллект;
* подсветка возможного хода;
  1. Реализация искусственного интеллекта

Так как компьютерные программы реверси играют намного сильнее людей, целесообразно выбрать для компьютера наиболее простую тактику игры – захват угловых клеток. В режиме игры "Лёгкий" компьютер имеет примитивный искусственный интеллект. Координаты игрового поля выбираются случайным образом: осуществляется просмотр всех свободных вариантов положения фишки, и если есть возможные ходы, то компьютер выбирает место случайным образом.

Код реализации ИИ:

procedure TGame.comp(var x, y: integer);

var

col, row, pos, temp, count: integer;

position: array of tpos;

begin

pos := -1;

setlength(position, pos + 2);

for col := 1 to n do

for row := 1 to n do

begin

if chips.images[col, row].Enabled = True then

begin

chips.number\_of\_inverted := 0;

check\_step(col, row);

if chips.number\_of\_inverted <> 0 then

begin

inc(pos);

setlength(position, pos + 1);

position[pos].x := col;

position[pos].y := row;

end;

end;

end;

randomize;

pos := random(pos);

x := position[pos].x;

y := position[pos].y;

angle(x, y);

end;

Для формирования этого модуля необходимо было использовать класс для хранения и передачи координат, которые выбрал клмпьютер.

Ниже приведен код описание типа класс:

Tpos

tpos = record

x: integer;

y: integer;

end;

В режиме "Нормальный" в приоритете стоят угловые фишки, так как их нельзя инвертировать обратно.

Код усложнения ИИ:

procedure angle(var x, y: integer);

begin

if form1.choose\_difficult.ItemIndex = 1 then

begin

chips.number\_of\_inverted := 0;

Game.check\_step(1, 1);

if (chips.number\_of\_inverted <> 0) and (chips.images[1, 1].Enabled = True)

then

begin

x := 1;

y := 1;

end;

chips.number\_of\_inverted := 0;

Game.check\_step(n, n);

if (chips.number\_of\_inverted <> 0) and (chips.images[n, n].Enabled = True)

then

begin

x := n;

y := n;

end;

chips.number\_of\_inverted := 0;

Game.check\_step(n, 1);

if (chips.number\_of\_inverted <> 0) and (chips.images[n, 1].Enabled = True)

then

begin

x := n;

y := 1;

end;

chips.number\_of\_inverted := 0;

Game.check\_step(1, n);

if (chips.number\_of\_inverted <> 0) and (chips.images[1, n].Enabled = True)

then

begin

x := 1;

y := n;

end;

end;

end;

Благодаря тому, что реализована более простая тактика игры, оценивающий программу сможет не только поучаствовать, но и победить в игре один на один с искусственным интеллектом, что весьма благоприятно скажется на его настроении.

* 1. Режим игры «Два игрока»

Чтобы было удобно играть двум пользователям нужна процедура, которая реализует управление игрой не только через мышь, но и через клавиши. Для этого используем Клавиши-стрелки, при перемещении курсора(рисунок 2.6) и клавишу Enter, для выбора места хода.

Пример кода управления курсором:

procedure up;

begin

backlight;

if chips.x\_backlight = 1 then

chips.x\_backlight := n + 1;

if chips.images[chips.x\_backlight - 1, chips.y\_backlight].Enabled = True

then

begin

if form1.CheckBox1.Checked = false then

backlight\_off;

chips.images[chips.x\_backlight - 1, chips.y\_backlight]

.Picture.LoadFromFile('./6666.jpg');

dec(chips.x\_backlight);

end

else

begin

dec(chips.x\_backlight);

if chips.images[1, chips.y\_backlight].Enabled = false then

begin

if chips.y\_backlight = n then

chips.y\_backlight := 1

else

inc(chips.y\_backlight);

chips.x\_backlight := 1;

end;

backlight;

up;

end;

end;

Часть кода, которая реагирует на события нажатия клавиш:

if (form1.choose\_difficult.ItemIndex = 2) and (flag = 1) then

case Key of

vk\_up:

up;

vk\_down:

down;

vk\_right:

right;

vk\_left:

left;

VK\_RETURN:

click(Sender);

end;

* 1. Подсветка возможных вариантов хода

В этом приложении доступна функция подсветки возможных вариантов ходов (рисунок 2.5), что приводит к более динамичной и более лёгком усвоении игры для новичков.

Код для подсветки ходов:

procedure TGame.backlight;

var

col, row, count, pos, temp, a: integer;

begin

count := 0;

pos := 0;

for col := 0 to n + 1 do

for row := 0 to n + 1 do

begin

if (chips.images[col, row].Enabled = True) and

(chips.color\_chips[col, row] = 0) then

begin

chips.number\_of\_inverted := 0;

check\_step(col, row);

if chips.number\_of\_inverted <> 0 then

begin

chips.images[col, row].Picture.LoadFromFile('./555.jpg');

inc(count);

chips.images[col, row].tag := 2

end

else

begin

chips.images[col, row].Picture.LoadFromFile('./000.jpg');

chips.images[col, row].tag := 1

end;

end;

end;

end;

В коде, описанном выше, используется 2 цикла, в которых процедура проходится по всем свободным элементам и подсвечивает те из них, по нажатию на которые будет доступен код.

1. тестирование программного средства

Тестирование программного средства (ПС) - это процесс выполнения программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Указанный набор называется тестовым. Тестирование программ является одной из , составных частей более общего понятия - «отладка программ». Под отладкой понимается процесс, позволяющий получить программу, функционирующую с требующимися характеристиками в заданной области изменения входных данных .

Основные параметры, определяющие качество ПО, это: надежность, сопровождаемость, практичность, эффективность, мобильность, функциональность. Для достижения наилучших показателей было проведено тестирование, которое описано в следующей таблице:

Таблица 1 - Тестирование и отладка ПС

|  |  |  |  |
| --- | --- | --- | --- |
| **Описание теста** | **Ожидаемый результат** | **Полученный результат** | **Решение проблемы** |
| **1** | **2** | **3** | **4** |
| Ход для обращения фишек по горизонтали | Инвертирование внутренних фишек | Инвертировались лишние фишки | Изменение передаваемых параметров в функцию |
| Ход для обращения фишек по вертикали | Инвертирование внутренних фишек | Инвертировались нужные фишки | Изменение не требуется |
| Ход для обращения фишек по диагоналям | Инвертирование внутренних фишек | Ошибка выхода за пределы массива фишек | Создание добавочного контура из невидимых фишек |
| Ход компьютера | Ход в доступное место | Правильный ход | Изменение не требуется |
| Подсветка доступных ходов | Выделение фишек | Правильное выделение фишек | Изменение не требуется |
| Подсветка текущего местоположения курсора | Выделение выбранной фишки | Закрашивание всех предыдущих координат | Обратно перекрашивать предыдущую координату |

1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ
   1. Начало работы

При запуске программы можно увидеть следующий интерфейс (рисунок 4.1). Выбранный пункт меню подсвечивается путем цвета при наведении курсора. Переход между пунктами меню осуществляется нажатием клавиш «↑» и «↓» . Для начала игры необходимо нажать на клавишу «Enter».

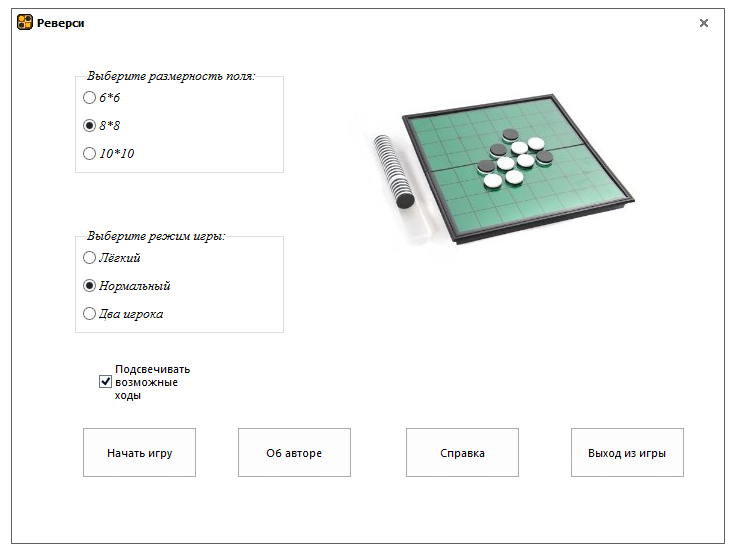


Рисунок 4.1 - Главное меню игрового приложения

После выбора пункта меню «Начать» отображается игровое поле (рисунок 4.2). Игровой процесс запущен. Далее пользователь, нажимая клавишу мыши, выбирает место, куда хочет поставить фишку. Если она недоступна, инверсия не происходит. Если всё-таки элемент доступен, то фишки переворачиваются .

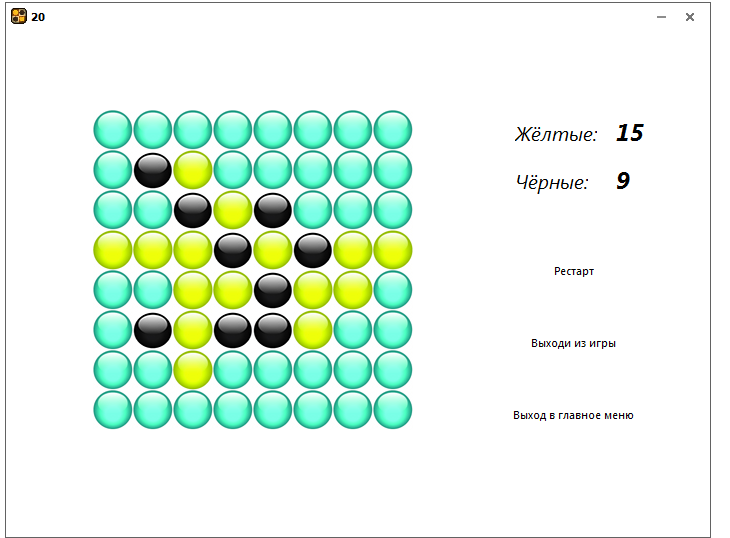


Рисунок 4.2 – Игровое поле

После того, как походил пользователь возможность поставить фишку переходит к компьютеру или же ко второму пользователю. Если пользователь или компьютер не может сделать ход, перевернув хоть одну фишку противника, то происходит передача хода оппоненту, и появляется окно оповещающее пользователю об этом (рисунок 4.3 ).

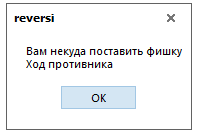


Рисунок 4.3 – Экран Сообщение о передачи хода

Если выбрать в меню режим “Два игрока”, то можно увидеть в правом верхнем углу чей ход в данный момент (рисунок 4.4.)



Рисунок 4.4 – Режим "два игрока"

В этом режиме для второго пользователя управление осуществляется клавишами, изображёнными на рисунке 4.5.

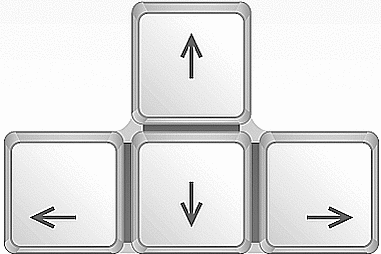


Рисунок 4.5 - «Клавиши управления»

В конце игры появляется сообщение о том, что победил компьютер или один из пользователей.

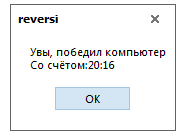
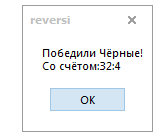


Рисунок 4.6 – Сообщение о конце игры

Для новичков, в данном приложении есть режим «Подсвечивать возможные ходы» , который ускоряет поиск доступного хода и делает легче выбор хода.

В стартовом меня пользователь может выбрать размерность поля, что повлияет на скорость игры, сложность, так как количество возможных комбинаций увеличивается или уменьшается в зависимости от выбора размера. Этот режим служат ещё для того, чтобы разнообразить игру

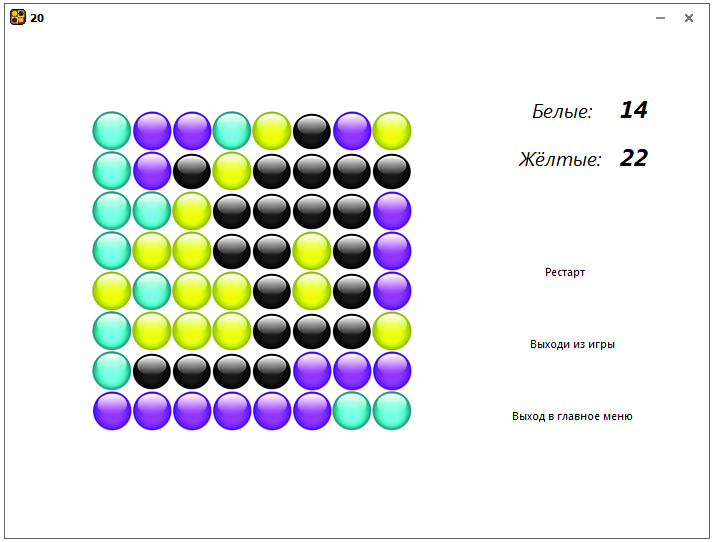


Рисунок 4.7 – Фоны доступных локаций

В пункте меню «Об авторе» вы можете узнать информацию об авторе данного игрового приложения.

Выбрав пункт меню «Справка», вы сможете получить краткую информацию о том, как нужно играть в данную игру.

Для того чтобы выйти из игры, выберите пункт меню «Выход из игры».

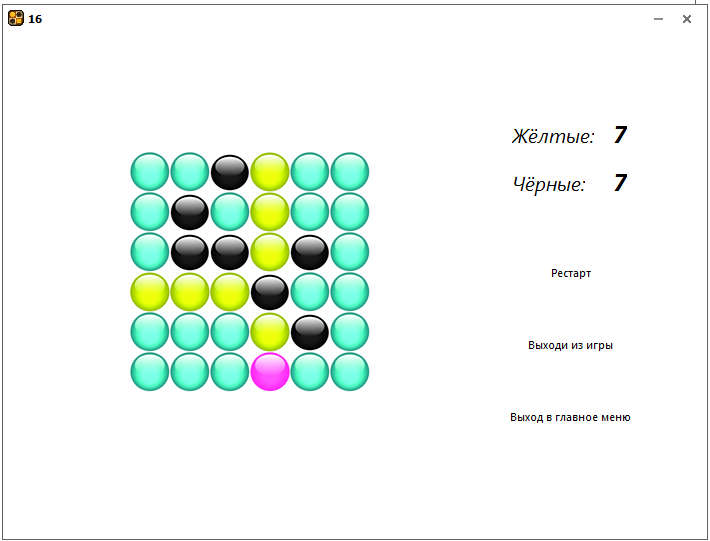


Рисунок 4.8 – Фоны доступных локаций

**ЗАКЛЮЧЕНИЕ**

Конец ХХ – начало ХХI века характеризуется активным внедрением в деятельность человека компьютерных информационных технологий, особенно разработка логических компьютерных игр, что подтверждает актуальность выбранной темы.

При написании курсовой работы было спроектировано и реализовано программное приложение «Компьютерная игра Реверси» .

Программа интуитивно проста и понятна, для ее использования не нужно специального обучения пользователя.

В ходе разработки получены навыки в создании игрового программного средства для операционной системы Windows.

В результате разработки приложения компьютерной игры была достигнута поставленная цель – создана компьютерная игра «Реверси».Для этого были разработаны алгоритмы и программы для реализации данного приложения с использованием современной технологии программирования. Программное приложение реализовано с использованием объектно-ориентированной технологии программирования. Тестирование, разрабатываемого приложения показало работоспособность компьютерной игры, целостность и структурированность.

Предполагаемая область применения компьютерной игры «SUDOKU»: проведение досуга детей в возрасте 12-16 лет, обучающихся второй ступени, средних учебных заведений, например, в рамках внеклассных мероприятий. Игра адресована учителям, учащимся и их родителям, а так же всем тем, кто интересуется развитием логического мышления, тренировки памяти и математических способностей.

Таким образом, поставленные цели и задачи курсовой работы решены, однако может быть усовершенствована и продолжена в других аспектах.

**СПИСОК ИИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Delphi справочник (электронный ресурс). – Электронные данные. – Режим доступа: delphibasics.ru/Class.php
2. Реверси – Википедия (электронный ресурс). – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki/Реверси
3. Графические возможности Делфи (электронный ресурс). – Электронные данные. – Режим доступа: http://delphi7.gym5cheb.ru;
4. Научные исследования компьютерных игр (электронный ресурс). – Электронные данные. – Режим доступа: http://gamestudies.ru
5. В.Фаронов Программирование в Delphi 6 (учебный курс). – М.: Издатель Молгачева С.В., 2001. – 672 с., ил.

Приложение А. Исходный код программы

Приложение Б. Блок-схема функции