**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 2](#_Toc26290430)

[**1. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ** 3](#_Toc26290431)

[1.1 Постановка задачи 3](#_Toc26290432)

[1.2 Выбор язык программирования 3](#_Toc26290433)

[**КОНСТРУКТОРСКАЯ ЧАСТЬ** 4](#_Toc26290434)

[**ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ** 9](#_Toc26290435)

[**2 ИНСТРУКЦИЯ ПОЛЬЗОВАНИЯ** 9](#_Toc26290436)

[2.1 Скачивание игры «PacMan» 9](#_Toc26290437)

[2.2 Установка 9](#_Toc26290438)

[2.3 Запуск игры «PacMan» 9](#_Toc26290439)

[2.4 Инструкция самой игры 9](#_Toc26290440)

[**ЗАКЛЮЧЕНИЕ** 11](#_Toc26290441)

[**СПИСОК ЛИТЕРАТУРЫ** 11](#_Toc26290442)

[**ПРИЛОЖЕНИЕ** 12](#_Toc26290443)

# **ВВЕДЕНИЕ**

Темой данного курсового проекта является разработка компьютерной игры «PacMan». При написании данной программы будет использован такой язык программирования как C# и платформы Visual Studio. Данное приложение является компьютерной игрой, в которую можно поиграть на досуге.

Пояснительная записка состоит из трех глав. В первой рассматриваются вопросы, связанные с материалом, который был изучен в ходе выполнения курсовой работы. Во второй описаны принципы работы основных частей программы. В третьей перечислены системные требования к персональному компьютеру, а также инструкция пользователя.

Целью выполнения настоящей курсовой работы является создание программного продукта, предназначенного для развлечения пользователей, совершенствования их реакции. В рамках данной курсовой работы реализуется всем хорошо известная игра «PacMan». Таким образом, задачей в данном случае является разработка и реализация алгоритма данной игры.

Предметом курсового проекта является разработка приложения «PacMan», которое позволит с интересом убить время, а также позволит провести его с пользой, так как оно развивает внимание и реакцию во время игры.

Для достижения поставленной цели были сформулированы и решены следующие задачи:

1. Изучить принципы и правила игры “PacMan”.

2. Выбрать технологии реализации создаваемой системы.

3. Реализовать основные модули системы.

4. Сформировать документацию. Объектом в данной работе является «PacMan».

# **1. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ**

## 1.1 Постановка задачи

Задачей этого курсового проекта является написание компьютерной игру «PacMan». Основой задачей в игре является создание колобка, который будет поедать жёлтые шарики на игровом поле. Управление в игре производится с помощью клавиатуры, это позволяет изменять направление движения колобка.

Кроме этого, в данной игре надо было реализовать вражеских сущностей, которые мешают PacMan «спокойно жить».

## 1.2 Выбор язык программирования

При решении конкретной прикладной задачи в большинстве случаев язык и среда программирования не выбираются, а задаются заказчиком, начальником и т.п. В том же редком случае, когда возможен выбор, исходить, на мой взгляд, следует из следующих условий (в порядке приоритета):

1. Характера самой задачи и технических требований;

2. Наработанного инструментария и имеющихся для данной среды библиотек;

3. Имеющихся в языке и среде программирования инструментальных средств. По набору понятий языки прежде всего подразделяются на высоко- и низкоуровневые.

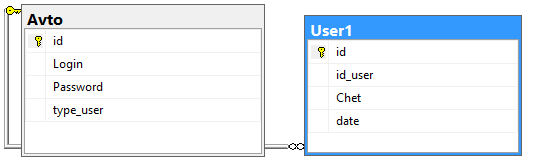
C# —объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к С++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков С++, Java, Delphi, Модула и Smalltalk — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++).

В результате проведенного анализа и того факта, что некоторые навыки по написанию программ на языке программирования C# у нас уже имеются, можно сделать вывод о том, что наиболее подходящим языком программирования является С#.

# **КОНСТРУКТОРСКАЯ ЧАСТЬ**

Для создания данного приложения понадобилось использовать приложение Visual Studio и SQL Server. В приложении придется отображать и изменять данные то, то для этого нужно, создать несколько классов. Были созданы три основных класса, которые отвечают за контроль, отображение и логику игры. За контроль отвечает класс – Controller\_MainForm. За отображение отвечает класс –View. Логика приложения реализована в классе GameLogic. За изменение направление движения PacMan будет отвечать, такое событие как KeyDown. Событие будет отслеживать нажатие пользователем клавишей ←, ↑, →, ↓, которое будет изменять направление движения «PacMan». Для этого был создан специальный класс, в котором происходит смена статуса движения.

SQL База для приложение «PacMan»



Листинг 1. Класс, где перечислены направления движения

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Left) next = 1;

if (e.KeyCode == Keys.Right) next = 2;

if (e.KeyCode == Keys.Up) next = 3;

if (e.KeyCode == Keys.Down) next = 4;

if (e.KeyCode == Keys.Escape) Close();

temp = next;

}

Также создан специально класс для самого колобка, где описаны его поля, изображения для него, которые меняются циклически во время движения.

Листинг 2. Класс PacMan

private void powermod\_Tick(object sender, EventArgs e)

{

tresec = false;

if (!g1mangiato)

{

v1 = false;

ghost1puomangiare = true;

Supermod1 = false;

ghost1velocity = 2;

}

if (!g2mangiato)

{

v2 = false;

ghost2puomangiare = true;

Supermod2 = false;

ghost2velocity = 2;

}

if (!g3mangiato)

{

v3 = false;

ghost3puomangiare = true;

Supermod3 = false;

ghost3velocity = 2;

}

if (!g4mangiato)

{

v4 = false;

ghost4puomangiare = true;

Supermod4 = false;

ghost4velocity = 2;

}

Supermod = false;

powermod.Enabled = false;

powermod1.Enabled = false;

if (prec1 == 1 && !g1mangiato) { if (ghost1.Left % 2 == 0) leftghost1 = ghost1velocity; ghost1.Image = Properties.Resources.rdx; }

if (prec1 == 2 && !g1mangiato) { if (ghost1.Left % 2 == 0) leftghost1 = -ghost1velocity; ghost1.Image = Properties.Resources.rsx; }

if (prec1 == 3 && !g1mangiato) { if (ghost1.Top % 2 == 0) topghost1 = -ghost1velocity; ghost1.Image = Properties.Resources.rup; }

if (prec1 == 4 && !g1mangiato) { if (ghost1.Top % 2 == 0) topghost1 = ghost1velocity; ghost1.Image = Properties.Resources.rdown; }

}

Создан специально класс для вражеских сущностей, где описаны поля, изображения. Листинг 3. Класс Dispose

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Codice generato da Progettazione Windows Form

/// <summary>

/// Metodo necessario per il supporto della finestra di progettazione. Non modificare

/// il contenuto del metodo con l'editor di codice.

/// </summary>

private void InitializeComponent()

{

this.components = new System.ComponentModel.Container();

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));

this.timer1 = new System.Windows.Forms.Timer(this.components);

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.label3 = new System.Windows.Forms.Label();

this.timer2 = new System.Windows.Forms.Timer(this.components);

this.timer3 = new System.Windows.Forms.Timer(this.components);

this.timer4 = new System.Windows.Forms.Timer(this.components);

this.timer5 = new System.Windows.Forms.Timer(this.components);

}

В приложении создан специальный класс, который отвечает за визуализацию, благодаря которому пользователю становится понятно, как запускать игру и управлять ею. В нём срабатывают события, которые позволяют колобку плавно передвигаться по игровому полю, поедать шарики и двигаться вражеским сущностям.

Листинг 5. Класc

View partial class View : UserControl

{

GameLogic gameLogic;

public View(GameLogic gameLogic)

{

InitializeComponent();

this.gameLogic = gameLogic;

gameLogic.RedrawNeeded += new GameLogic.RedrawNeededEventHandler(model\_RedrawNeeded); gameLogic.SharikRemoved += new GameLogic.SharikRemovedEventHandler(model\_SharikRemoved); gameLogic.New\_game += new GameLogic.New\_Game(NewGame\_Shariki); InitializeLabels();

}

}

Самым важным классом приложения, является класс, где реализована вся логика игры – это класс GameLogic. Исходный код данного класса приведен в приложении 1.

Созданы методы, которые отвечают за движение колобка, вражеских сущностей. Листинг 7. Метод Run private void Run() Игровое поле представлено в виде двумерного массива. Создан метод, который позволяет сбрасывать все параметры игры и позволяет осуществлять запуск новой игры.

Листинг 8. Метод NewGame

internal void NewGame()

{

pacman = new Pacman();

sound\_Start = new SoundPlayer(Properties.Resources.start\_song);

if (status\_sound)

{

sound\_Start.Load();

sound\_Start.Play();

}

gameStatus = GameStatus.stoping;

Shariki = new <List>();

for (int i = 0; i < 13; i++)

{

for (int j = 0; j < 17; j++)

{

if (pole\_mass[i, j] == 0)

{

Shariki.Add(new Point(j \* 25, i \* 25));

}

}

}

OnNew\_game();

timer.Interval = 1;

timer.Elapsed -= new System.Timers.ElapsedEventHandler(timer\_Elapsed);

timer.Elapsed += new System.Timers.ElapsedEventHandler(timer\_Elapsed);

enemy = new Enemy();

enemy\_2 = new Enemy();

enemy\_3 = new Enemy();

rnd = new Random(DateTime.Now.Millisecond);

Walls = new List();

for (int i = 0; i < 13; i++)

{

for (int j = 0; j < 17; j++)

{

if (pole\_mass[i, j] == 1)

{

Walls.Add(new Point(j \* 25, i \* 25));

}

}

}

Shariki\_New = new List();

}

}

Так же были созданы дополнительные Formi:

Первая Forma1 авторизации Login и Password

|  |  |
| --- | --- |
|  | // Получаем логин и пароль в переменные сразу чистим краевые пробелы  string login = textBox1.Text.Trim();  string pass = textBox2.Text.Trim();  if (login == "" || pass == "")  {  MessageBox.Show("Вы не ввели данные для входа");  }  else  {  SqlConnection connect = new  SqlConnection(Program.st\_connect);  // Открыть подключение  connect.Open();  //строка запроса на поиск юзера  string s = "select \* from Avto where Login = '" + login + "' and Password= '" + pass + "'";  //Делаем запрос к БД  SqlCommand command = new SqlCommand(s, connect);  //Получаем данные, которые вернул зарос  SqlDataReader reader = command.ExecuteReader();  if (reader.HasRows)  {  //получаем одну строку из набора данных  reader.Read();  //Получаем данные, в скобках это номер столбцов в запросе!!!  Program.id\_user = reader.GetInt32(0);  Program.type\_user = reader.GetBoolean(3);  if (Program.type\_user == false)  {  //MessageBox.Show("Простой юзер");  Form1 f = new Form1();  f.Show();  }  else  {  //MessageBox.Show("AdmiN");  Form3 f = new Form3();  f.Show();  }  }  else  {  MessageBox.Show("Пользователь не найден!!!");  }  // Закрыть подключение  connect.Close();  } |

На Form2 переход только для Admina, где Администратор сможет увидит таблицу рекордов игроков и специальные кнопки для добавления, удаления и ЛК пользователя.

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

Рисунок ….

# **ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ**

# **2 ИНСТРУКЦИЯ ПОЛЬЗОВАНИЯ**

## 2.1 Скачивание игры «PacMan»

Скачать игру можно по ссылке «https://github.com/Drakylo/Game-Pacman»

## 2.2 Установка

Дважды кликаем на скаченную иконку «PacMan». Проводим установку. Как сделать установку приложения показано в **Приложении 2.**

## 2.3 Запуск игры «PacMan»

После запуска игры, необходимо авторизация. Если аккаунт существует, то можно сразу начать играть. Если его нет, то нажать на кнопку «Зарегистрироваться». Как регистрироваться показано в **Приложении 3**

## 2.4 Инструкция самой игры

Приложение открывается двойным щелчком по иконке программы. После запуска программы загружается основное окно программы (Рисунок 1)



Рисунок 1 – Стартовая картинка

Для того чтобы начать играть надо нажать на кнопку «Play Game», где визуализировано игровое поле (Рисунок 2):

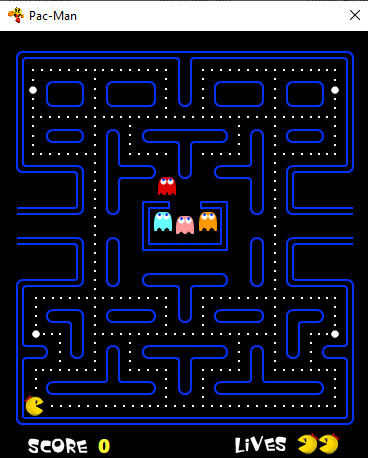


Рисунок 2 – Игровое поле приложения

После этого происходит запуск игры, начинается движение «Приведения». Движение «PacMan», происходит с помощью кнопок на клавиатуре пользователем. Основная задача пользователя, играя игру, собрать все шарики на игровом поле, при этом не пересекаясь с врагами, которые двигаются произвольно, мешая «PacMan» победить.

В нижней части окна находится «Score» и «Lives». «Score» считает количество собранных шариков, а «Lives» это количество жизни «PacMan» (Рисунок 3)

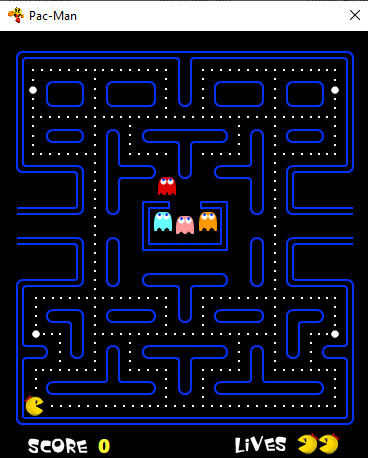


Рисунок 3 Информационное поле

# **ЗАКЛЮЧЕНИЕ**

При разработке игрового приложения были реализованы следующие задачи: Удобный интерфейс Возможность осуществлять управление движения «PacMan» с помощью клавиатуры Осуществление анимации колобка Создание «приведений», которые двигаются произвольно Добавление звуковых эффектов в игре Поедание пакманом шариков на игровом поле «Прозрачные стены» для игрового поля – т.е. когда сущность подходит к границе поля, она не отходит в сторону, а выходит с противоположной стороны поля. Объекты при движении смещаются на пиксель для создания эффекта плавного перемещения.

В рамках данной работы была разработана компьютерная игра, в которую можно поиграть в свободное время, приукрасить свой досуг. Так же при создании проекта были изучены новые элементы управления и свойства языка C#.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Фролов А. В., Фролов Г. В. Язык C#. Самоучитель – M.: ДИАЛОГ-МИФИ, 2003. – 560 с.

2. Лабор В.В. Си Шарп: создание приложений для Windows – Мн.: Харвест, 2003. – 384 с.

3. Разработка Windows-приложений на Microsoft Microsoft Visual C# 2003 /Пер. с англ. – M.: Издательско-торговый дом <Русская Редакция>, 2003. – 512 с.

4. Коннолли Томас, Бегг Карелин. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание. : Пер. с англ. — М. : Издательский дом "Вильяме", 2003. — 1440 с. : ил. — Парал. тит. англ.

# **ПРИЛОЖЕНИЕ**

**Приложение 1**

public partial class Form1 : Form

{

//dichiarazione variabili

public int leftghost1, leftghost2, leftghost3, leftghost4, left;

public int topghost1, topghost2, topghost3, topghost4, top;

public int next, direzione, temp = 1;

public bool pacturn, ghost1turn, ghost2turn, ghost3turn, ghost4turn, start, ghost1puomangiare, ghost2puomangiare, ghost3puomangiare, ghost4puomangiare, g1mangiato, g2mangiato, g3mangiato, g4mangiato;

public bool dir1, dir2, dir3, dir4, c, Supermod, Supermod1, Supermod2, Supermod3, Supermod4, v1, v2, v3, v4, tresec = true;

public int random1, random2, random3, random4, vita = 3, tic4, tic5, tic6, point;

private void label253\_Click(object sender, EventArgs e)

{

}

private void label255\_Click\_1(object sender, EventArgs e)

{

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

DateTime dateTimeVariable = DateTime.Now;

string date1 = dateTimeVariable.ToString("yyyy-MM-dd");

string s = "INSERT INTO User1 " +

"(id\_user, Chet, date) VALUES" +

"('" + Program.id\_user + "'," +

"'" + score.Text.ToString() + "'," +

"'" + date1 + "')";

SqlCommand command = new SqlCommand(s, conn);

SqlDataReader reader = command.ExecuteReader();

conn.Close();

}

private void Form1\_Load(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "SELECT Login FROM Avto WHERE id= " + Program.id\_user;

//Делаем запрос к БД через адаптер, что бы потом поместить даные в DATASET

SqlDataAdapter adap = new SqlDataAdapter(s, conn);

//создаем пустой датасет

DataSet ds = new DataSet();

//соединяем датасет с нашим набором данных

adap.Fill(ds);

//соединяем датасет с визуальным компонентом

dataGridView1.DataSource = ds.Tables[0];

label253.Text = dataGridView1[0, 0].Value.ToString();

conn.Close();

SqlConnection conn1 = new SqlConnection(Program.st\_connect);

conn1.Open();

string s1 = "SELECT TOP 1(Chet) FROM User1 ORDeR BY Chet DESC ";

//Делаем запрос к БД через адаптер, что бы потом поместить даные в DATASET

SqlDataAdapter adap1 = new SqlDataAdapter(s1, conn1);

//создаем пустой датасет

DataSet ds1 = new DataSet();

//соединяем датасет с нашим набором данных

adap1.Fill(ds1);

//соединяем датасет с визуальным компонентом

dataGridView2.DataSource = ds1.Tables[0];

label254.Text = dataGridView2[0, 0].Value.ToString();

conn1.Close();

}

public int prec1, prec2, prec3, prec4, startdirection = 1, ghost1velocity = 2, ghost2velocity = 2, ghost3velocity = 2, ghost4velocity = 2;

public Random rand = new Random();

public List<object> oggetti = new List<object>();

//fine dichiarazione variabili

//inizializzazione della Form1

public Form1()

{

InitializeComponent();

oggetti.Add(label4);

oggetti.Add(label5);

oggetti.Add(label6);

oggetti.Add(label7);

oggetti.Add(label8);

oggetti.Add(label9);

oggetti.Add(label10);

oggetti.Add(label11);

oggetti.Add(label12);

oggetti.Add(label13);

oggetti.Add(label14);

oggetti.Add(label15);

oggetti.Add(label16);

oggetti.Add(label17);

oggetti.Add(label18);

oggetti.Add(label19);

oggetti.Add(label20);  
 }

private void button1\_Click(object sender, EventArgs e)

{

start = true;

life();

points();

panel1.Visible = false;

resetall();

}

//направления движения

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Left) next = 1;

if (e.KeyCode == Keys.Right) next = 2;

if (e.KeyCode == Keys.Up) next = 3;

if (e.KeyCode == Keys.Down) next = 4;

if (e.KeyCode == Keys.Escape) Close();

temp = next;

}

//stile bottone1

private void button1\_MouseLeave(object sender, EventArgs e)

{

button1.Image = Properties.Resources.playgame;

}

//stile bottone1

private void button1\_MouseEnter(object sender, EventArgs e)

{

button1.Image = Properties.Resources.playgame2;

}

//timer1

private void timer1\_Tick(object sender, EventArgs e)

{

if (pacman.Enabled)

{

direction();

freedirection();

}

if (ghost1.Enabled || ghost2.Enabled || ghost3.Enabled || ghost4.Enabled)

{

ghost();

}

if (c) control();

supermod();

points();

collision();

}

//timer2

private void timer2\_Tick(object sender, EventArgs e)

{

label3.Visible = false;

timer1.Enabled = true;

timer2.Enabled = false;

c = true;

}

//timer3

private void timer3\_Tick(object sender, EventArgs e)

{

if (ghost2.Top == 180) startdirection = 2;

if (ghost2.Top == 192) startdirection = 1;

if (startdirection == 1)

{

ghost2.Top--;

ghost4.Top--;

}

if (startdirection == 2)

{

ghost2.Top++;

ghost4.Top++;

}

}

//timer4

private void timer4\_Tick(object sender, EventArgs e)

{

tic4++;

if (ghost3.Top > 150 && tic4 > 100) { prec1 = 3; ghost3.Top--; }

if (ghost3.Top == 150) timer4.Enabled = false;

}

//timer5

private void timer5\_Tick(object sender, EventArgs e)

{

tic5++;

if (ghost2.Left > 153 && ghost2.Left < 176 && tic5 > 300) ghost2.Left++;

if (ghost2.Top > 150 && ghost2.Left == 176) { ghost2.Top--; }

if (ghost2.Top == 150) { timer6.Enabled = true; timer5.Enabled = false; }

}

//timer6

private void timer6\_Tick(object sender, EventArgs e)

{

tic6++;

if (ghost4.Left > 176 && ghost4.Left < 200 && tic6 > 100) ghost4.Left--;

if (ghost4.Top > 150 && ghost4.Left == 176) { ghost4.Top--; }

if (ghost4.Top == 150) timer6.Enabled = false;

}

private void powermod\_Tick(object sender, EventArgs e)

{

tresec = false;

if (!g1mangiato)

{

v1 = false;

ghost1puomangiare = true;

Supermod1 = false;

ghost1velocity = 2;

}

if (!g2mangiato)

{

v2 = false;

ghost2puomangiare = true;

Supermod2 = false;

ghost2velocity = 2;

}

if (!g3mangiato)

{

v3 = false;

ghost3puomangiare = true;

Supermod3 = false;

ghost3velocity = 2;

}

if (!g4mangiato)

{

v4 = false;

ghost4puomangiare = true;

Supermod4 = false;

ghost4velocity = 2;

}

Supermod = false;

powermod.Enabled = false;

powermod1.Enabled = false;

if (prec1 == 1 && !g1mangiato) { if (ghost1.Left % 2 == 0) leftghost1 = ghost1velocity; ghost1.Image = Properties.Resources.rdx; }

if (prec1 == 2 && !g1mangiato) { if (ghost1.Left % 2 == 0) leftghost1 = -ghost1velocity; ghost1.Image = Properties.Resources.rsx; }

if (prec1 == 3 && !g1mangiato) { if (ghost1.Top % 2 == 0) topghost1 = -ghost1velocity; ghost1.Image = Properties.Resources.rup; }

if (prec1 == 4 && !g1mangiato) { if (ghost1.Top % 2 == 0) topghost1 = ghost1velocity; ghost1.Image = Properties.Resources.rdown; }

if (prec2 == 1 && !g2mangiato) { if (ghost2.Left % 2 == 0) leftghost2 = ghost2velocity; ghost2.Image = Properties.Resources.adx; }

if (prec2 == 2 && !g2mangiato) { if (ghost2.Left % 2 == 0) leftghost2 = -ghost2velocity; ghost2.Image = Properties.Resources.asx; }

if (prec2 == 3 && !g2mangiato) { if (ghost2.Top % 2 == 0) topghost2 = -ghost2velocity; ghost2.Image = Properties.Resources.aup; }

if (prec2 == 4 && !g2mangiato) { if (ghost2.Top % 2 == 0) topghost2 = ghost2velocity; ghost2.Image = Properties.Resources.adown; }

if (prec3 == 1 && !g3mangiato) { if (ghost3.Left % 2 == 0) leftghost3 = ghost3velocity; ghost3.Image = Properties.Resources.vdx; }

if (prec3 == 2 && !g3mangiato) { if (ghost3.Left % 2 == 0) leftghost3 = -ghost3velocity; ghost3.Image = Properties.Resources.vsx; }

if (prec3 == 3 && !g3mangiato) { if (ghost3.Top % 2 == 0) topghost3 = -ghost3velocity; ghost3.Image = Properties.Resources.vup; }

if (prec3 == 4 && !g3mangiato) { if (ghost3.Top % 2 == 0) topghost3 = ghost3velocity; ghost3.Image = Properties.Resources.vdown; }

if (prec4 == 1 && !g4mangiato) { if (ghost4.Left % 2 == 0) leftghost4 = ghost4velocity; ghost4.Image = Properties.Resources.gdx; }

if (prec4 == 2 && !g4mangiato) { if (ghost4.Left % 2 == 0) leftghost4 = -ghost4velocity; ghost4.Image = Properties.Resources.gsx; }

if (prec4 == 3 && !g4mangiato) { if (ghost4.Top % 2 == 0) topghost4 = -ghost4velocity; ghost4.Image = Properties.Resources.gup; }

if (prec4 == 4&& !g4mangiato) { if (ghost4.Top % 2 == 0) topghost4 = ghost4velocity; ghost1.Image = Properties.Resources.gdown; }

}

//timer7

private void timer7\_Tick(object sender, EventArgs e)

{

timer7.Enabled = false;

panel1.Visible = true;

attendo.Enabled = true;

}

// хп

private void life()

{

if (vita == 3)

{

pictureBox1.Visible = true;

pictureBox2.Visible = true;

}

if (vita == 2)

{

pictureBox1.Visible = true;

pictureBox2.Visible = false;

}

if (vita == 1)

{

pictureBox1.Visible = false;

pictureBox2.Visible = false;

}

if (vita == 0)

{

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

DateTime dateTimeVariable = DateTime.Now;

string date1 = dateTimeVariable.ToString("yyyy-MM-dd");

string s = "INSERT INTO User1 " +

"(id\_user, Chet, date) VALUES" +

"('" + Program.id\_user+ "'," +

"'" + score.Text.ToString()+ "'," +

"'" + date1 + "')";

SqlCommand command = new SqlCommand(s, conn);

SqlDataReader reader = command.ExecuteReader();

conn.Close();

}

}

**Приложение 2**

Установка приложения на свой ПК

Дважды клик на иконку «PacMan», затем откроется окно. Необходимо выбрать язык установки. Рисунок …

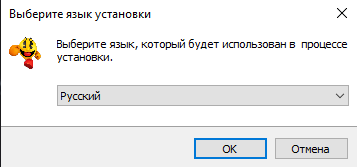


Рисунок …Выберите язык установки

Выбрать дополнительные задачи, которые должны выполняться при установке «PacMan», после этого нажать «Далее» Рисунок …

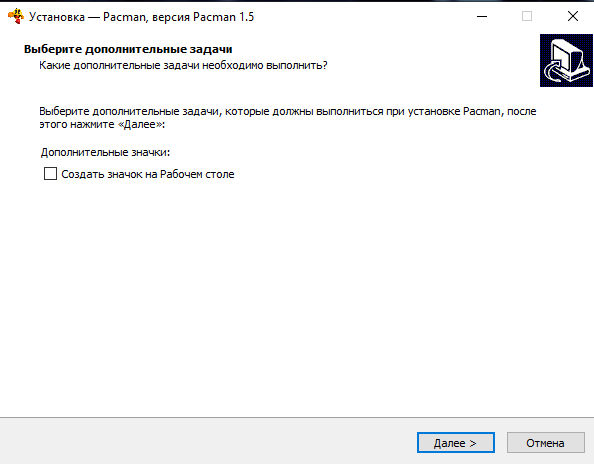


Рисунок … Выбрать дополнительные задачи

Нажмите «Установить», чтобы продолжить установку игры «PacMan», или «Назад», если вы хотите просмотреть или изменить опции установки. Рисунок …

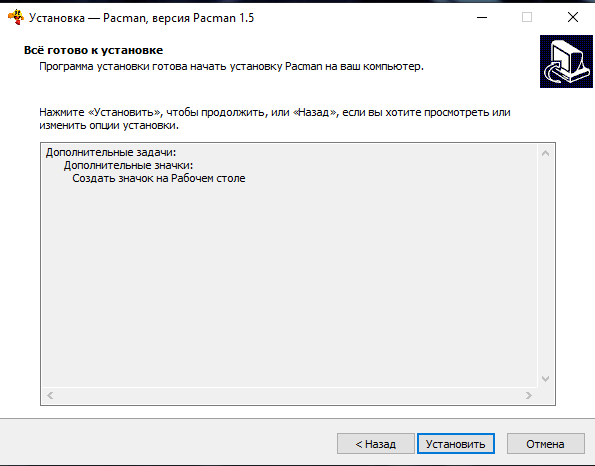
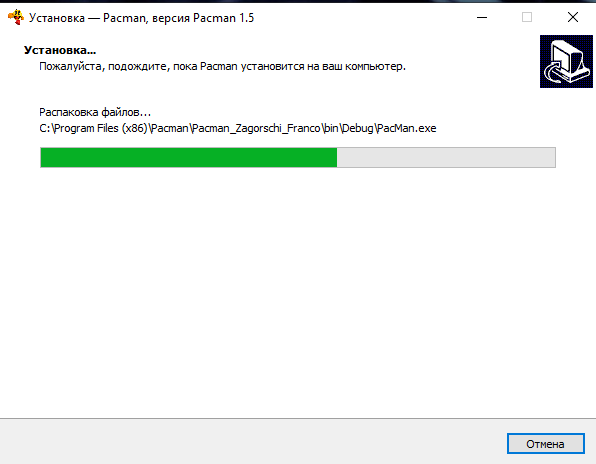


Рисунок … Подготовка установки игры

Подождите, пока «PacMan» установиться на ваш компьютер. Рисунок…



Рисунок… Установка игры

Завершение установки игры «PacMan», приложение можно запустить с помощью соответствующего значка (кликнуть 2 раза по нему) или поставить галочку, чтобы приложение запустилась автоматически, после нажатие кнопки «Завершить», показано на Рисунке …

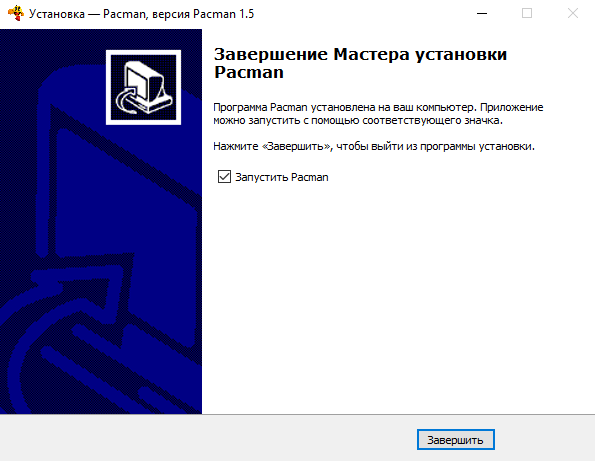


Рисунок … Завершение установки игры «PacMan»

**Приложение 3**

Для того чтобы начать играть в «PacMan», необходимо зарегистрироваться. Для этого нажимаем кнопку «Зарегистрироваться». Рисунок …

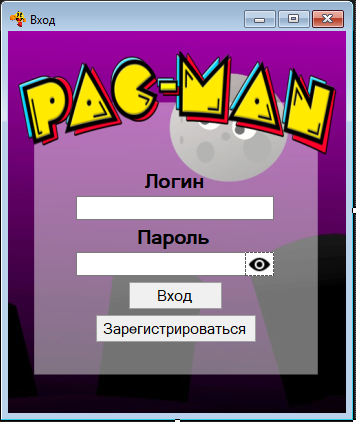


Рисунок … Главное окно входа

После того как нажали на кнопку «Зарегистрироваться», выходит следующие окно. Там необходимо ввести «Логин» и «Пароль». Рисунок …

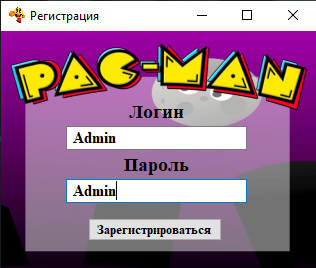


Рисунок … Форма регистрации

Нажать на кнопку «Зарегистрироваться» и можно входить в игру и играть.