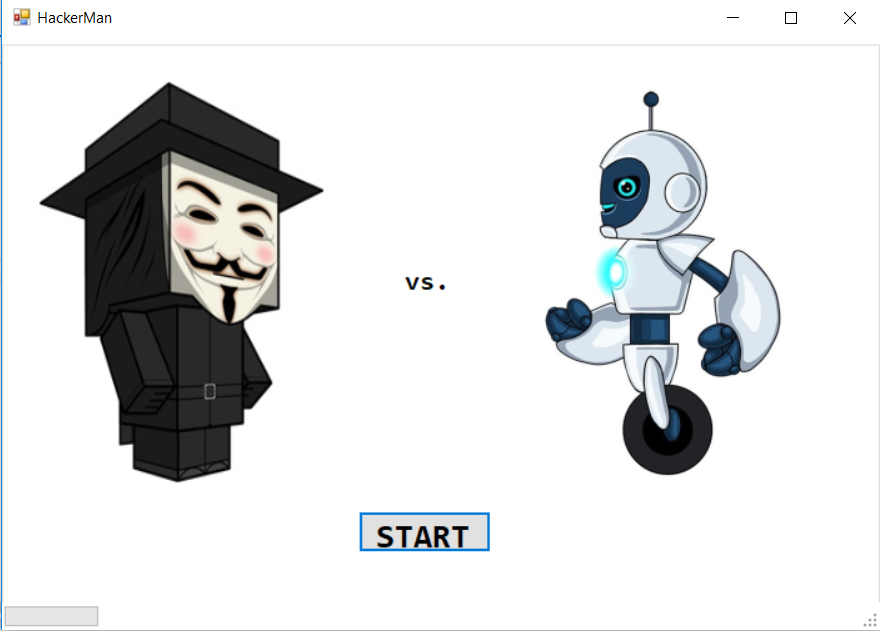
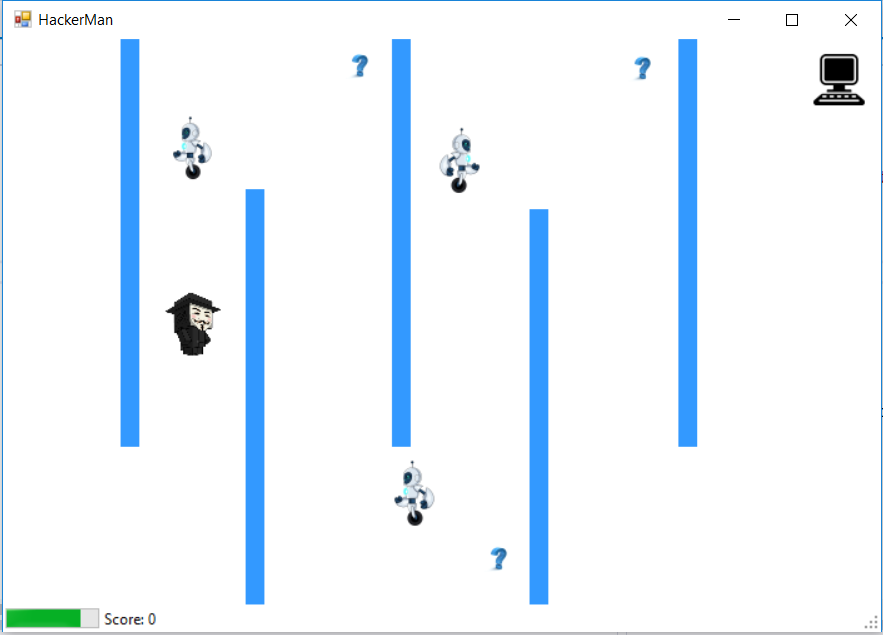
HackerManGame

Во оваа проектна задача се обидовме да составиме едноставна игра во која играчот со помош на стрелките преку тастатура го движи карактерот низ интерактивна мапа каде целта е да стигне до компјутер кој е поставен на одредено место, со цел да продолжи на наредното ниво. Играта почнува со клик на копчето старт што означува почеток на негативците да се движат, времето да почне со истекување, и да се појават одредени “checkpoint” локации на мапата кои треба да бидат посетени и одговорени за да се соберат поени и да може да се активира компјутерот. Играчот има 2:30 минути да го измине нивото и да ги одговори прашањата, притоа внимавајќи да не дојде во контакт со негативците кои се движат на одредени места на мапата. Играта исто така може да се паузира.

Почетната состојба на играта изгледа вака:



За време на играње изгледа вака:



Доколку играчот дојде во контакт со негативците се појавува дијалог прозор за избор дали ќе сака да започне од почеток, доколку избере да, тогаш карактерот се враќа во почетната состојба, тајмерот и мапата се ресетираат.

Начинот на кој е реализирана играта се состои од група на слики сместени во PictureBox и нивно придвижување низ екранот. Сликите се сместени во Resources фајлот кој се наоѓа кај изворниот код од играта, од каде се повикуваат. Главниот карактер се придвижува со KeyDown и KeyUp евентите во главната форма. Кодовите за нив се:

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

{

if (e.KeyCode == Keys.Left)

{

goleft = false;

}

if (e.KeyCode == Keys.Right)

{

goright = false;

}

if (e.KeyCode == Keys.Down)

{

godown = false;

}

if (e.KeyCode == Keys.Up)

{

goup = false;

}

}

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

{

timer1.Start();

timer2.Start();

if (e.KeyCode == Keys.Left)

{

goleft = true;

HackerManPic.Image = Properties.Resources.HackermanPicLeft;

poml = true; pomd = false; pomr = false; pomu = false;

}

if (e.KeyCode == Keys.Left&& e.KeyCode==Keys.Right)

{

goleft = false; goright = false;

poml = false; pomd = false; pomr = false; pomu = false;

}

if (e.KeyCode == Keys.Right)

{

goright = true;

HackerManPic.Image = Properties.Resources.HackermanPicRight;

poml = false; pomd = false; pomr = true; pomu = false;

}

if (e.KeyCode == Keys.Down)

{

godown = true;

HackerManPic.Image = Properties.Resources.HackermanPicLeft;

poml = false; pomd = true; pomr = false; pomu = false;

}

if (e.KeyCode == Keys.Up)

{

goup = true;

HackerManPic.Image = Properties.Resources.HackermanPicRight;

poml = false; pomd = false; pomr = false; pomu = true;

}

if (e.KeyCode == Keys.P)

{

pauseGame();

}

if (e.KeyCode == Keys.S)

{

startGame();

}

}

За карактерот да се придвижува исто е потребно тоа да биде реализирано и во главниот Timer, со timer\_Tick event каде се повикуваат методите CharacterMove(); и HackerManMoves.

Во кодот за CharacterMove се сместени условите за карактерот да не може да излегува надвор од границите на мапата, тоа е направено со дефинирање на големината на мапата и проверка дали карактерот се стреми да излезе од неа, доколку карактерот дојде до границите на мапата и сака да продолжи неговата брзина се намалува на 0 и се враќа пред границата, со што се добива изглед како карактерот да се удира и одбива од краевите на мапата.

public void CharacterMove()

{

int desno = 650;

int dole = 400;

int gore = 0;

int levo = 0;

if (HackerManPic.Location.X < levo)

{

HackerManPic.Left = HackerManPic.Left + 1;

speed = 0;

}

else if (HackerManPic.Location.X > desno)

{

HackerManPic.Left = HackerManPic.Left - 1;

speed = 0;

}

else if (HackerManPic.Location.Y < gore)

{

HackerManPic.Top = HackerManPic.Top + 1;

speed = 0;

}

else if (HackerManPic.Location.Y > dole)

{

HackerManPic.Top = HackerManPic.Top - 1;

speed = 0;

}

else

speed = 5;

}

Исто така со цел карактерот де се однесува “нормално” додаден е и услов во главниот тајмер кој проверува дали карактерот има намера да премине низ ѕидовите на мапата, доколку тоа е точно, карактерот повторно се враќа во состојбата во која бил пред да проба да мине низ ѕидот и му е одземена брзината. Еве го кодот за условот и кодот за другиот метод за движење.

if (Stone1.Bounds.IntersectsWith(HackerManPic.Bounds) || Stone2.Bounds.IntersectsWith(HackerManPic.Bounds) || Stone3.Bounds.IntersectsWith(HackerManPic.Bounds) || Stone4.Bounds.IntersectsWith(HackerManPic.Bounds) || Stone5.Bounds.IntersectsWith(HackerManPic.Bounds))

{

if (poml)

{

HackerManPic.Left = HackerManPic.Left + 1;

speed = 0;

}

if (pomr)

{

HackerManPic.Left = HackerManPic.Left - 1;

speed = 0;

}

if (pomu)

{

HackerManPic.Top = HackerManPic.Top + 1;

speed = 0;

}

if (pomd)

{

HackerManPic.Top = HackerManPic.Top - 1;

speed = 0;

}

}

else

{

speed = 5;

}

}

public void HackerManMoves()

{

toolStripStatusLabel1.Text = "Score: " + score;

if (goleft)

{

HackerManPic.Left -= speed;

}

if (goright)

{

HackerManPic.Left += speed;

}

if (goup)

{

HackerManPic.Top -= speed;

}

if (godown)

{

HackerManPic.Top += speed;

}

}

Во главниот тајмер исто така се наоѓа и кодот кој служи за проверка дали карактерот е во интеракција со некој од објектите на мапата, како што се негативците, checkpoint-ите и крајната цел, компјутерот. Доколку е, тогаш се извршуваат одредени настани во зависност од тоа со што дошол во контакт карактерот. Вториот тајмер се состои од код во кој се иницијализира ProgressBar објектот и лабела која го покажува моменталниот резултат. Тие се сместени во StatusStrip објект. Во прилог е кодот за првиот тајмер:

if (Enemy1Pic.Bounds.IntersectsWith(Stone1.Bounds))

{

Enemy1Pic.Image = Properties.Resources.EnemyRight;

enemy1 = -enemy1;

}

if (Enemy1Pic.Bounds.IntersectsWith(Stone3.Bounds))

{

Enemy1Pic.Image = Properties.Resources.EnemyLeft;

enemy1 = -enemy1;

}

if (Enemy2Pic.Bounds.IntersectsWith(Stone2.Bounds))

{

Enemy2Pic.Image = Properties.Resources.EnemyRight;

enemy2 = -enemy2;

}

if (Enemy2Pic.Bounds.IntersectsWith(Stone4.Bounds))

{

Enemy2Pic.Image = Properties.Resources.EnemyLeft;

enemy2 = -enemy2;

}

if (Enemy3Pic.Bounds.IntersectsWith(Stone3.Bounds))

{

Enemy3Pic.Image = Properties.Resources.EnemyRight;

enemy3 = -enemy3;

}

if (Enemy3Pic.Bounds.IntersectsWith(Stone5.Bounds))

{

Enemy3Pic.Image = Properties.Resources.EnemyLeft;

enemy3 = -enemy3;

}

foreach (Control x in this.Controls)

{

if (x is PictureBox && x.Tag == "enemy")

{

if (x.Bounds.IntersectsWith(HackerManPic.Bounds))

{

HackerManPic.Left = 0;

HackerManPic.Top = 5;

score = 0;

timer1.Stop();

System.Windows.Forms.DialogResult BYesNo = MessageBox.Show(" You lost\n Do you want to play again? ", "Game Over", MessageBoxButtons.YesNo);

if (BYesNo == System.Windows.Forms.DialogResult.Yes)

{

vreme = 150;

GBStart.Visible = true;

GBStart.Enabled = true;

godown = false;

goleft = false;

goright = false;

goup = false;

score = 0;

this.Controls.Add(PBQuestion1);

this.Controls.Add(PBQuestion2);

this.Controls.Add(PBQuestion3);

}

if (BYesNo == System.Windows.Forms.DialogResult.No)

{

this.Close();

}

}

}

if(x is PictureBox && x.Tag == "question")

{

if (((PictureBox)x).Bounds.IntersectsWith(HackerManPic.Bounds))

{

timer1.Stop();

timer2.Stop();

Random r = new Random();

int number = r.Next(0, 2);

string str = Question[number];

if (number == 0)

{

if (MessageBox.Show("Question 1", str, MessageBoxButtons.YesNo) == System.Windows.Forms.DialogResult.No)

{

score += 10;

this.Controls.Remove(x);

godown = false;

goleft = false;

goright = false;

goup = false;

}

else

{

score -= 5;

this.Controls.Remove(x);

godown = false;

goleft = false;

goright = false;

goup = false;

}

}

if (number == 1)

{

if (MessageBox.Show("Question 2", str, MessageBoxButtons.YesNo) == System.Windows.Forms.DialogResult.Yes)

{

score += 10;

this.Controls.Remove(x);

godown = false;

goleft = false;

goright = false;

goup = false;

}

else

{

score -= 5;

this.Controls.Remove(x);

godown = false;

goleft = false;

goright = false;

goup = false;

}

}

if (number == 2)

{

if (MessageBox.Show("Question 3", str, MessageBoxButtons.YesNo) == System.Windows.Forms.DialogResult.No)

{

score += 10;

this.Controls.Remove(x);

}

else

{

score -= 5;

this.Controls.Remove(x);

}

GBStart.Visible = false;

GBStart.Enabled = false;

timer1\_Tick(sender, e);

timer1.Start();

godown = false;

goleft = false;

goright = false;

goup = false;

}

}

timer1.Start();

timer2.Start();

}

if (x is PictureBox && x.Tag == "computer")

{

if (((PictureBox)x).Bounds.IntersectsWith(HackerManPic.Bounds))

{

timer1.Stop();

timer2.Stop();

if (score > 0)

{

System.Windows.Forms.DialogResult YesNo = MessageBox.Show("Play next level", "You win", MessageBoxButtons.OK);

if (YesNo == System.Windows.Forms.DialogResult.Yes)

{

GBStart.Visible = true;

GBStart.Enabled = true;

}

}

else

{

MessageBox.Show("You haven't gathered enough points for next level", "Not quite there yet", MessageBoxButtons.OK);

timer1.Start();

timer2.Start();

HackerManPic.Left = 0;

HackerManPic.Top = 5;

}

}

}