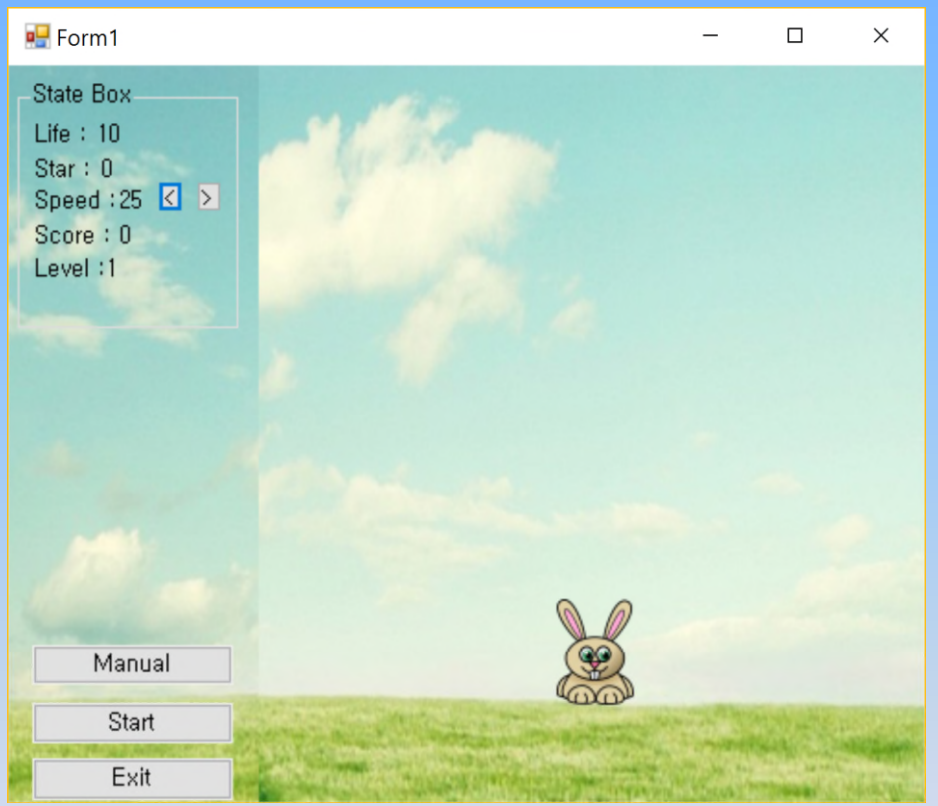
소프트웨어 실습 3
과제3 보고서

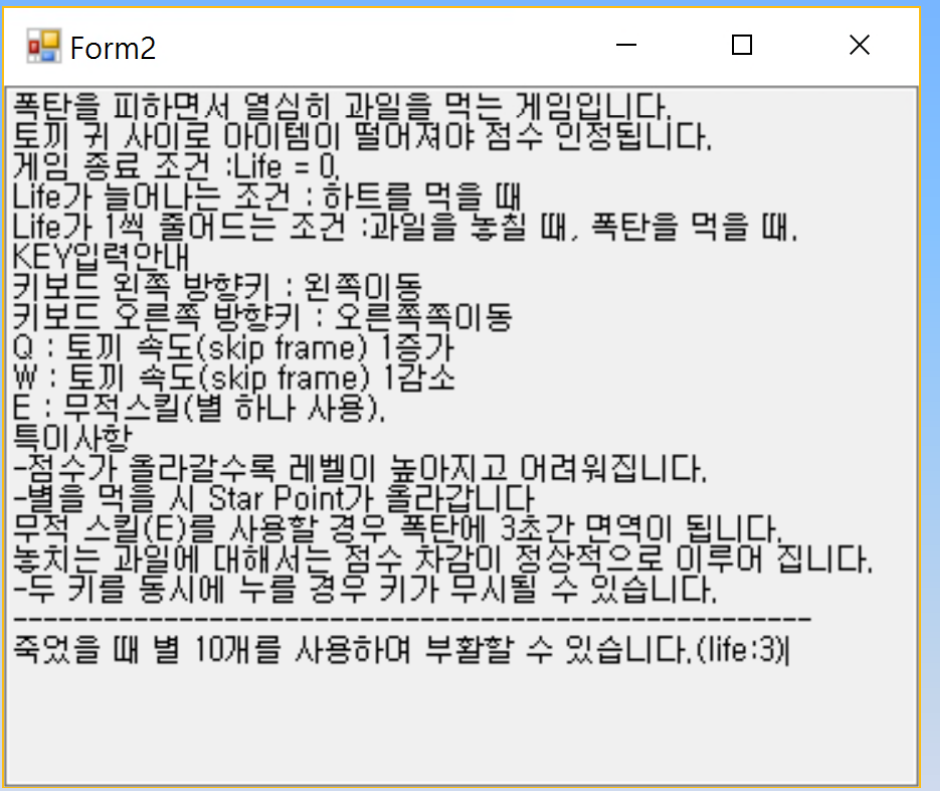
2013726011 최은주

받을 수 있다고 생각하는 점수
- 필수구현 8점
- 추가요소 1점 (부활)

실행화면

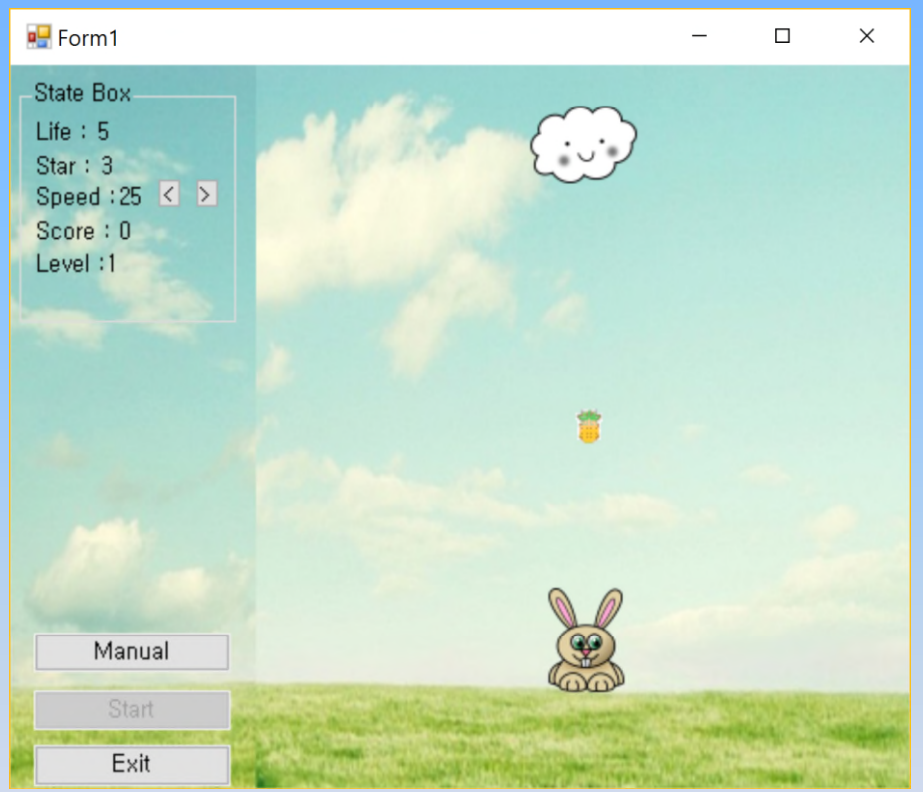


<시작화면>

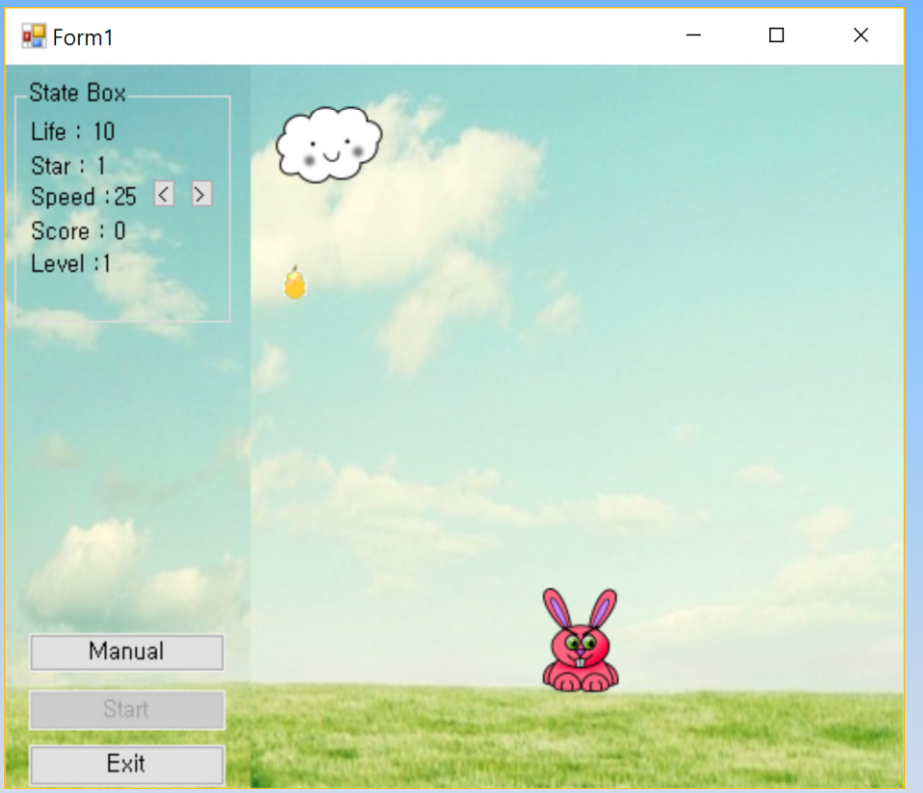


<매뉴얼화면>

실행화면

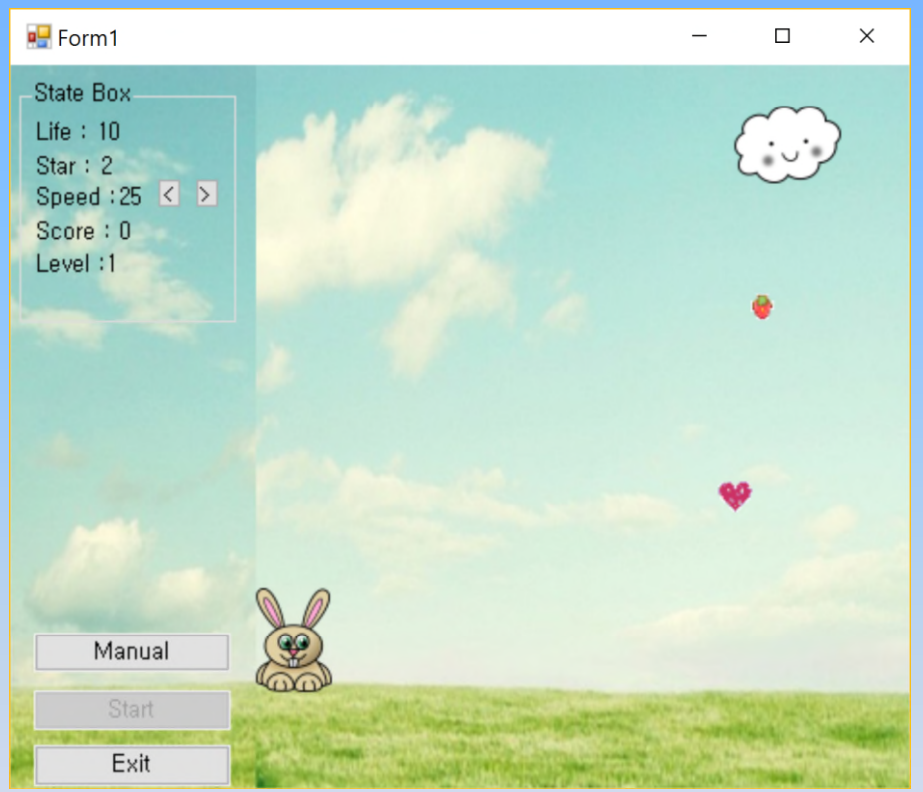


Start 버튼을 누르면
게임이 시작됩니다.

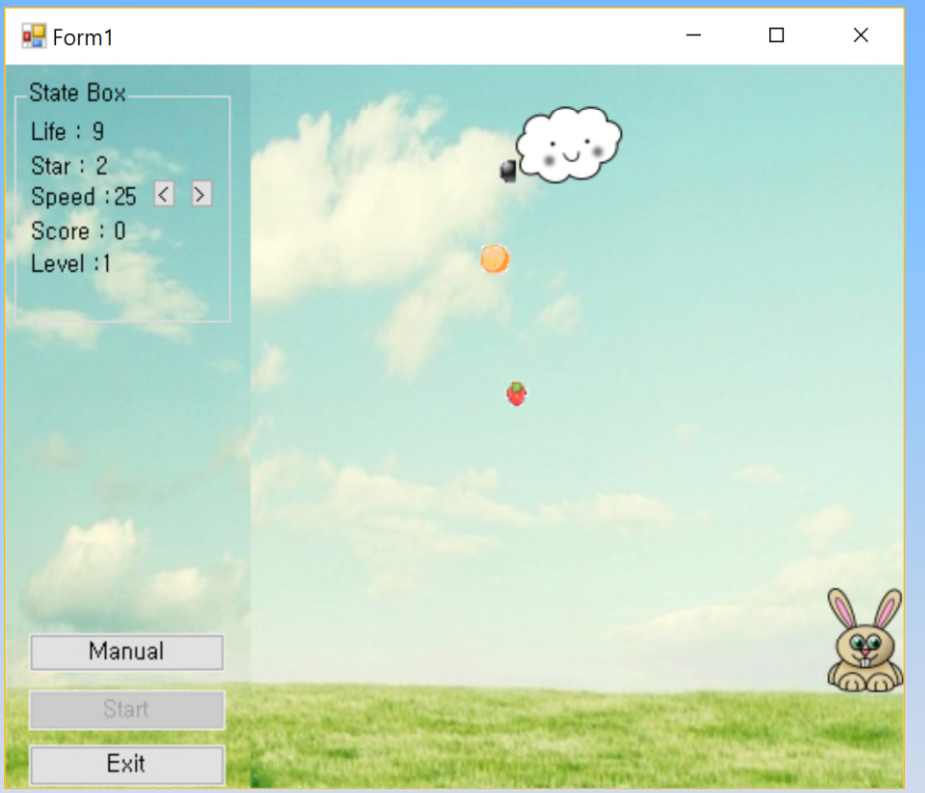


E 버튼을 누르면
3초간 무적 모드가 됩니다.

실행화면

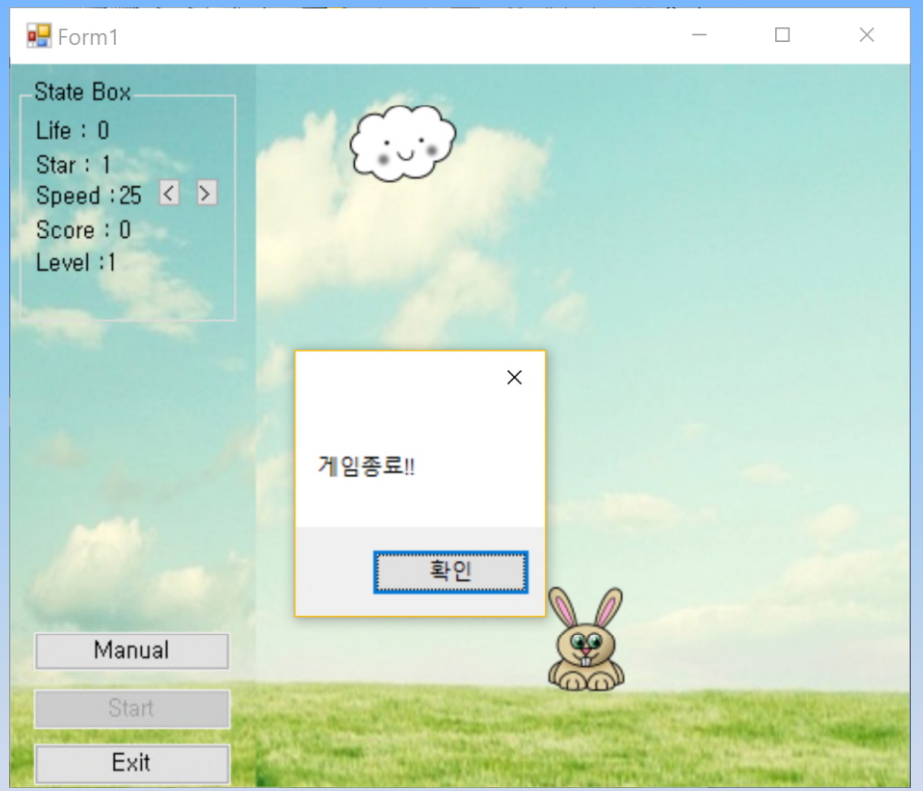


왼쪽 벽에 닿으면
더 이상 왼쪽으로 갈 수 없습니다.

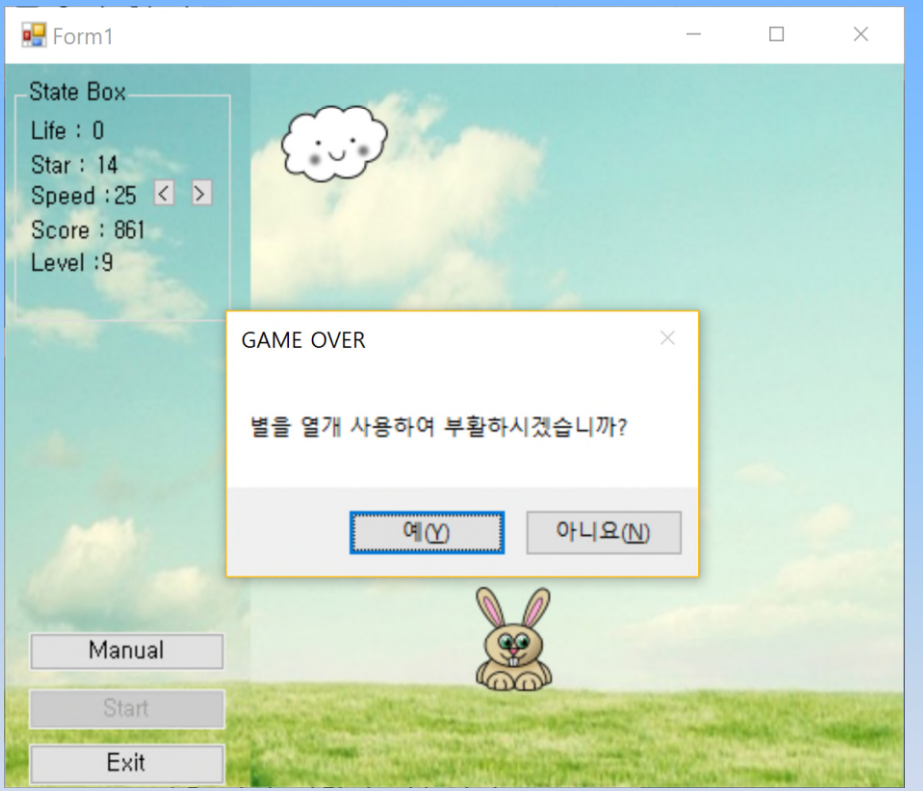


오른쪽도 같은 방법으로 더 이상
오른쪽으로 갈 수 없습니다.

실행화면



Life가 0이 되면
게임종료 창이 뜨게 됩니다.



별의 개수가 10개 이상인 경우
10개로 부활이 가능합니다.
(추가요소 1점)

코드설명

1. Form1_Load

```
private void Form1_Load(object sender, EventArgs e)
{
    groupBox1.BackColor = Color.Transparent;
    groupBox1.Parent = pictureBox1;

    rabbit.BackColor = Color.Transparent;
    rabbit.Parent = pictureBox2;

    cloud.BackColor = Color.Transparent;
    cloud.Parent = pictureBox2;
}
```

▪프로그램 실행 시 가장 먼저 실행되는 Form1_Load함수

▪점수판과 토끼, 구름의 BackColor를 pictureBox의 그림으로 설정합니다.

코드설명

2. 속도 조절 버튼

```
private void slow_Click(object sender, EventArgs e)
{
    if (speed.Text != "1")
    {
        speedNum -= 1;
        speed.Text = speedNum.ToString();
    }
}

private void fast_Click(object sender, EventArgs e)
{
    speedNum += 1;
    speed.Text = speedNum.ToString();
}
```

- '<'버튼 클릭 시 속도가 1보다 크거나 같으면 1 줄입니다.

- '>'버튼 클릭 시 속도를 1 올립니다.

코드설명

3. 토끼 움직이기

```
private void rabbit_PreviewKeyDown(object sender, PreviewKeyDownEventArgs e)
{
    MovePicBox(sender, e);
}
public void MovePicBox(object sender, EventArgs e)
{
    if (e is KeyEventArgs)
    {
        KeyEventArgs ke = (KeyEventArgs)e;
        MovePicKey(ke.KeyCode);

    }
    else if (e is PreviewKeyDownEventArgs)
    {
        PreviewKeyDownEventArgs ke = (PreviewKeyDownEventArgs)e;
        MovePicKey(ke.KeyCode);
    }
}
```

- 토끼 picture box에 포커스가 맞춰져 있을 때 방향키를 누르면 PreviewKeyDown 이벤트가 발생합니다.

- MovePicBox에서 MovePicKey 함수를 실행합니다.

코드설명

4. 키 입력 처리 함수

(1) 왼쪽 방향키

```
private void MovePicKey(Keys ke)
{
    this.Invoke(new MethodInvoker(() =>
    {

        if (ke == Keys.Left)
        {
            if (rabbit.Location.X != 0)
            {
                if (rabbit.Location.X - speedNum >0)
                {
                    rabbit.Location = new Point(rabbit.Location.X - speedNum, rabbit.Location.Y);
                }
                else if (rabbit.Location.X - speedNum < 0)
                {
                    rabbit.Location = new Point(0, rabbit.Location.Y);
                }
                else
                {
                    rabbit.Location = new Point(0, rabbit.Location.Y);
                }
            }
        }
```

- 토끼가 picturebox2의 가장 왼쪽자리에 있을때와 그렇지 않을 때로 구분하여 움직입니다.

- 새로 이동할 위치가 가장 왼쪽 자리가 아닌 경우 speedNum만큼 이동합니다,

- 새로 이동할 위치가 picturebox2의 범위를 벗어나는 경우 0으로 이동시킵니다.

- 아예 토끼가 가장자리에 있을 경우 움직이지 않습니다.

코드설명

4. 키 입력 처리 함수

(2) 오른쪽 방향키

```
else if (ke == Keys.Right)
{
    if (rabbit.Location.X != 0)
    {
        if (rabbit.Location.X + speedNum < pictureBox2.Size.Width - rabbit.Size.Width)
        {
            rabbit.Location = new Point(rabbit.Location.X + speedNum, rabbit.Location.Y);
        }
        else
        {
            rabbit.Location = new Point(pictureBox2.Size.Width - rabbit.Size.Width, rabbit.Location.Y);
        }
    }
    else
    {
        rabbit.Location = new Point(rabbit.Location.X + speedNum, rabbit.Location.Y);
    }
}
```

- 왼쪽과 같은 방법으로 구현합니다.

코드설명

4. 키 입력 처리 함수

(3) ‘Q’, ‘W’ 키

```
else if (ke == Keys.W)
{
    speed.Focus();
    speedNum += 1;
    speed.Text = speedNum.ToString();
    rabbit.Focus();
}
else if (ke == Keys.Q)
{
    if (speed.Text != "1")
    {
        speed.Focus();
        speedNum -= 1;
        speed.Text = speedNum.ToString();
        rabbit.Focus();
    }
}
```

- ‘W’ 키를 누르면 속도가 1 증가하고, ‘Q’ 키를 누르면 속도가 1 감소 합니다.
- 그 다음에 speed 라벨로 되어있던 포커스를 토끼로 옮겨줍니다.

코드설명

4. 키 입력 처리 함수

(4) ‘E’ 키

```
else if (ke == Keys.E)
{
    if (starNum >= 1)
    {
        star.Focus();
        starNum -= 1;
        star.Text = starNum.ToString();
        rabbit.Focus();
        rabbit.Image = global::_HW03_2013726011최은주.Properties.Resources.화난토끼;
        isMad = true;
        timer1.Start();
    }
}
```

- ‘E’ 키를 누를 경우 3초간 무적토끼 모드가 실행됩니다.
- 별의 개수를 하나 줄이고 토끼의 사진을 화난토끼로 바꿔줍니다.
- 타이머를 실행시킵니다.

코드설명

5. 무적 토끼 해제 timer1_Tick

```
private void timer1_Tick(object sender, EventArgs e)
{
    this.Invoke(new MethodInvoker(delegate ()
    {
        rabbit.Image = global::_HW03_2013726011최은주.Properties.Resources.토끼;
    }));
    isMad = false;
    timer1.Stop();
}
```

- 3초가 지난 뒤 실행 되는 timer1_Tick함수입니다.
- 이미지를 다시 일반 토끼로 바꿔줍니다,
- 폭탄을 먹었을 때 화난 토끼인지 확인하는 is_Mad변수를 false로 바꿔준 뒤 timer를 멈춰줍니다.

코드설명

6. 게임 종료 함수

```
private void button5_Click(object sender, EventArgs e)
{
    if(isgaming != false)
    {
        m_Thread.Abort();
        this.Close();
    }
    else
    {
        this.Close();
    }

}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    if (isgaming != false)
    {
        m_Thread.Abort();
    }
}
```

- Exit버튼을 눌렀을 때 게임이 실행중이라면 구름을 움직이게 하는 쓰레드를 종료시킨 후 닫습니다.

- 폼에서 X버튼을 눌렀을 때 또한 같은 방법으로 확인합니다.

코드설명

7. 게임 시작 함수

```
private void start_Click(object sender, EventArgs e)
{
    isgaming = true;
    levelNum = 1;
    scoreNum = 0;
    lifeNum = 10;
    starNum = 2;
    level.Text = levelNum.ToString();
    score.Text = scoreNum.ToString();
    life.Text = lifeNum.ToString();
    star.Text = starNum.ToString();
    cloud.Visible = true;
    rabbit.Focus();
    start.Enabled = false;
    m_Thread = new Thread(new ThreadStart(moving_Cloud));
    m_Thread.Start();
}
```

- Start버튼을 누르면 발생하는 이벤트 함수입니다.

- 게임이 시작 되었다는 표시인 isgaming을 true로 바꿔주고, state box의 값들을 초기화 시켜줍니다.

- 구름의 visible을 true로 바꾸어 나타나게 한 뒤, 구름이 움직이도록 thread를 시작합니다.

코드설명

8. moving_Cloud 함수

```
private void moving_Cloud()
{
    while(true)
    {
        Thread.Sleep(300-levelNum*20);
        int n = random.Next(1, 10);

        int moving = random.Next(10, 20);
```

- 구름이 계속 움직여야 되므로 while(true)를 사용합니다.

- 구름의 속도를 조절하기 위해 Thread.Sleep을 사용합니다.

- 랜덤함수로 n값을 만들어서 왼쪽으로 갈 것인지 오른쪽으로 갈 것인지 결정합니다.

- 랜덤함수로 moving값을 만들어서 얼마나 이동할지 결정합니다.

코드설명

8. moving_Cloud 함수

```
this.Invoke(new MethodInvoker(() =>
{
    if (n % 2 == 0)      //오른쪽
    {
        if(cloud.Location.X != 0 )
        {
            if(cloud.Location.X + moving + levelNum < pictureBox2.Size.Width - cloud.Size.Width)
            {
                cloud.Location = new Point(cloud.Location.X + moving + levelNum, cloud.Location.Y);
            }
            else
            {
                cloud.Location = new Point(pictureBox2.Size.Width-cloud.Size.Width, cloud.Location.Y);
            }
        }
        else
        {
            cloud.Location = new Point(cloud.Location.X + moving + levelNum, cloud.Location.Y);
        }

    }
```

- 구름은 이 함수에서 만들어진 것이 아니기 때문에 invoke를 사용합니다.
- n이 짝수인 경우 입니다.
- 구름이 왼쪽 가장자리에 있는 경우가 아니라면 옮겨질 위치가 어디냐에 따라 구름이 이동할 곳이 정해집니다.

코드설명

8. moving_Cloud 함수

```
else    //왼쪽
{
    if (cloud.Location.X != 0)
    {
    if (cloud.Location.X - moving - levelNum > 0)
    {
        cloud.Location = new Point(cloud.Location.X - moving - levelNum, cloud.Location.Y);
    }
    else if (rabbit.Location.X-moving-levelNum <0)
        {
            cloud.Location = new Point(0, cloud.Location.Y);
        }
    }
    else
    {
        cloud.Location = new Point(cloud.Location.X + moving + levelNum, cloud.Location.Y);
    }
}
```

- 왼쪽으로 가는 경우에도 오른쪽과 비슷한 방법으로 코딩합니다.

코드설명

8. moving_Cloud 함수

```
if (levelNum == 10)
{
    if (n > 3)
    {
        m_Thread2 = new Thread(new ThreadStart(fallen_Items));
        m_Thread2.Start();
    }

}
else
{
    if (n > 5)
    {
        m_Thread2 = new Thread(new ThreadStart(fallen_Items));
        m_Thread2.Start();
    }
}
```

- 난이도 조절을 위해 level이 10일 때와 그렇지 않을 때 아이템이 만들어지는 확률을 조절합니다.

코드설명

8. moving_Cloud 함수

```
if (lifeNum <= 0)
{
    m_Thread.Abort();
    life.Focus();
    lifeNum = 0;
    life.Text = lifeNum.ToString();

    if (starNum >= 10)
    {
        if (MessageBox.Show("별을 열개 사용하여 부활하시겠습니까?", "GAME OVER", MessageBoxButtons.YesNo)==DialogResult.Yes)
        {
            life.Focus();
            lifeNum = 3;
            life.Text = lifeNum.ToString();
            star.Focus();
            starNum -= 10;
            star.Text = starNum.ToString();
            rabbit.Focus();
            m_Thread = new Thread(new ThreadStart(moving_Cloud));
            m_Thread.Start();

        }
```

- life가 0이 되었을 때 별이 10개 이상이면 yes를 클릭해 부활 할 수 있습니다.

- 별을 10개가 깎이고 life는 3늘어난 상태로 재시작합니다.

코드설명

8. moving_Cloud 함수

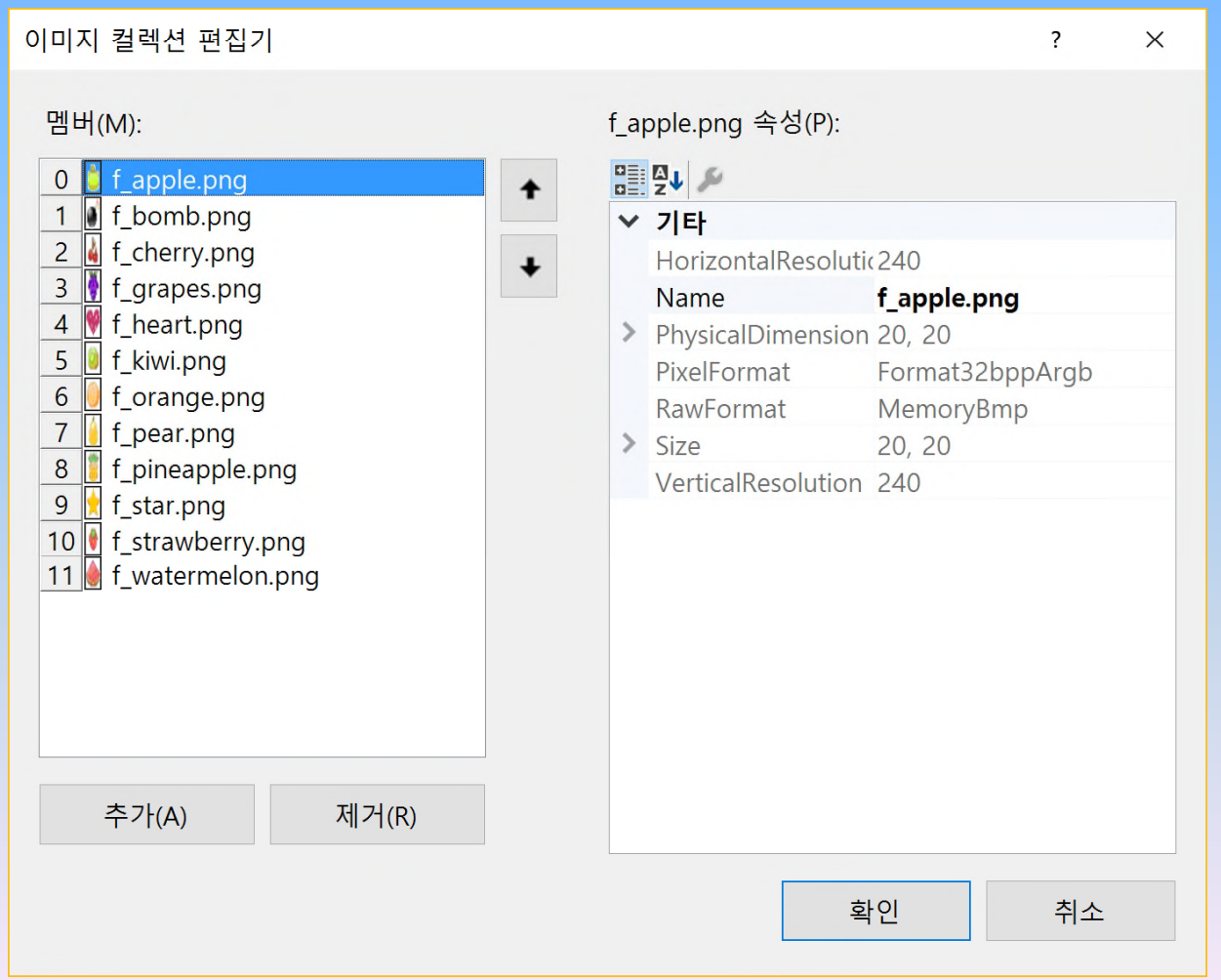
```
    else
    {
        life.Focus();
        lifeNum = 0;
        life.Text = lifeNum.ToString();
        MessageBox.Show("게임종료!!!");

        start.Enabled = true;
    }
}
else
{
    life.Focus();
    lifeNum = 0;
    life.Text = lifeNum.ToString();
    MessageBox.Show("게임종료!!");
    start.Enabled = true;
}
```

- no를 클릭한 경우나 별이 10개 미만인 경우 바로 게임 종료 팝업창이 뜨고 게임이 종료됩니다.

코드설명

8. fallen_Items 함수



- 구름에서 아이템이 떨어지게 하기 위해서 먼저 imagelist에 아이템을 넣어 놓습니다.

-점수판-

사과	5
체리	10
포도	15
키위	20
오렌지	20
배	15
파인애플	10
딸기	5
수박	1

코드설명

8. fallen_Items 함수

```
private void fallen_Items()
{
        int itemNum = random.Next(0, 12);

        PictureBox fallenItems = new PictureBox();
    this.Invoke(new MethodInvoker(() =>
        {
            fallenItems.Parent = pictureBox2;

            fallenItems.Width = 20;
            fallenItems.Height = 20;

            fallenItems.Image = imageList1.Images[itemNum];
            fallenItems.BackColor = Color.Transparent;
            fallenItems.Tag = itemNum;

            fallenItems.Location = new Point(cloud.Location.X, cloud.Location.Y);
        }));
```

- 아이템을 픽쳐박스에 넣기 위해 무작위로 숫자를 하나 선택합니다.

- 픽쳐박스를 하나 생성 하고 이미지리스트에 들어있는 이미지를 삽입합니다.

- 처음 픽쳐박스의 위치는 구름의 현재 위치로 설정합니다.

코드설명

8. fallen_Items 함수

```
int i = 1;
while (fallenItems.Location.Y < 350)
{
    Thread.Sleep(50);
    this.Invoke(new MethodInvoker(() =>
    {
        fallenItems.Location = new Point(fallenItems.Location.X, fallenItems.Location.Y + i);
        i = i+1;
```

- 아이템이 pictureBox의 풀밭근처인 350에 위치할때까지 내려갑니다.
- 가속도를 위해 변수 i를 설정하고 i값 만큼 fallenItems가 떨어집니다.

코드설명

8. fallen_Items 함수

```
if(rabbit.Bounds.IntersectsWith (fallenItems.Bounds))
{
    if(itemNum == 0)
    {
        fallenItems.Location = new Point(fallenItems.Location.X, fallenItems.Location.Y+100);
        fallenItems.Dispose();
        score.Focus();
        scoreNum += 5;
        score.Text = scoreNum.ToString();
        rabbit.Focus();
        isEat = true;
    }
```

- 아이템이 rabbit picturebox의 범위 안에 포함 되는 경우 점수 계산을 합니다.
- itemNum이 폭탄, 하트, 별이 아닌 경우 점수표의 점수만큼 score가 증가합니다.
- 아이템의 위치를 100증가시켜 화면에서 사라지게 한 뒤 리소스를 해제합니다.

코드설명

8. fallen_Items 함수

```
else if(itemNum == 1)   //폭탄
{
    if(isMad == true)
    {
        fallenItems.Location = new Point(fallenItems.Location.X, fallenItems.Location.Y + 100);
        fallenItems.Dispose();
        life.Focus();
        life.Text = lifeNum.ToString();
        rabbit.Focus();
        isEat = true;
    }
    else
    {
        fallenItems.Location = new Point(fallenItems.Location.X, fallenItems.Location.Y + 100);
        fallenItems.Dispose();
        life.Focus();
        lifeNum -= 1;
        life.Text = lifeNum.ToString();
        rabbit.Focus();
        isEat = true;
    }

}
```

- 떨어지는 아이템이 폭탄인 경우입니다.
- 토끼가 무적토끼인 경우 폭탄을 먹어도 life가 줄어들지 않습니다.
- 먹었는지 판단하기 위해 isEat을 true로 바꿔줍니다.
- 토끼가 무적토끼가 아닌경우 폭탄을 먹을 때 life가 1 줄어듭니다.

코드설명

8. fallen_Items 함수

```
else if (itemNum == 4)       //하트
{
    fallenItems.Location = new Point(fallenItems.Location.X, fallenItems.Location.Y + 100);
    fallenItems.Dispose();
    life.Focus();
    lifeNum += 1;
    life.Text = lifeNum.ToString();

    rabbit.Focus();
    isEat = true;
}
```

- 먹은 아이템이 하트인 경우 life가 1 늘어납니다.

```
else if (itemNum == 9)  //별
{
    fallenItems.Location = new Point(fallenItems.Location.X, fallenItems.Location.Y + 100);
    fallenItems.Dispose();
    star.Focus();
    starNum += 1;
    star.Text = starNum.ToString();

    rabbit.Focus();
    isEat = true;
}
```

- 먹은 아이템이 별인 경우 star가 1 늘어납니다.

코드설명

8. fallen_Items 함수

```
if (isEat == false)
{
    if (itemNum == 1 || itemNum == 4 || itemNum ==9)
    {
        fallenItems.Dispose();
    }
    else
    {
        fallenItems.Location = new Point(fallenItems.Location.X, fallenItems.Location.Y + 100);
        fallenItems.Dispose();
        life.Focus();
        lifeNum -= 1;
        life.Text = lifeNum.ToString();
        if (lifeNum <= 0)
        {
            life.Focus();
            lifeNum = 0;
            life.Text = lifeNum.ToString();
        }
        rabbit.Focus();
    }
}
isEat = false;
```

- 아이템을 먹지 못했을 때 아이템이 별이거나 하트이거나 폭탄이면 life의 값을 바꾸지 않습니다.

- 세가지 아이템이 아닌 경우 life가 1 줄어듭니다. isEat을 다시 false로 바꿔줍니다.

코드설명

8. fallen_Items 함수

```
this.Invoke(new MethodInvoker(() =>
{
    if (scoreNum>levelNum*100)
    {
        levelNum++;
        if(levelNum > 10)
        {
            levelNum = 10;
        }
        level.Focus();
        level.Text = levelNum.ToString();
        rabbit.Focus();
    }
}));
```

- 레벨 판정 부분입니다.
- score가 100점 쌓일 때마다 level이 증가합니다.
- Level이 10이 되면 더 이상 증가하지 않습니다.

고찰

이번 과제가 처음 나왔을 때는 이 과제가 과연 내가 할 수 있는 과제인가 의문이 들었습니다. 그래도 만들고 나니 기본적으로 수업을 들으면 충분히 구현 할 수 있는 과제였다고 생각합니다.

처음에 과제를 시작했을 때 구름과 토끼가 벽을 뚫고 사라져버리는 현상이 있었습니다. Speed를 높이면 토끼가 벽 너머로 사라져버려서 굉장히 당황했습니다. 범위를 값으로 지정해줘서 생기는 문제였습니다. 범위를 숫자가 아닌 pictureBox의 size와 cloud또는 rabbit의 size로 지정해주니 벽을 뚫고 사라지는 문제를 해결 할 수 있었습니다.

토끼가 아이템을 먹었을 때 while문을 탈출 할 수 있도록 break를 걸어주려고 했는데 탈출이 안돼서 isEat이라는 변수로 먹었는지 먹지 않았는지 판정하는 방법을 사용했습니다.