

[C# .NET Framework] 데스크탑 기반 애플리케이션 제작 프로젝트

코딩을 위한 타자 연습 게임

SpeedyCoding

KOSTA 265기 최혜린

1. 아이디어 스케치

1) 게임 기획 - 튜토리얼

1. 시작 버튼 : 게임 시작
2. 종료 버튼 : 게임 종료,
Total Score 출력
3. Mode : Easy_짧은 문자열,
Hard_코드 일부
4. Speed : 1~5 설정 가능
5. 맞힌 문자열마다 + 10점
6. 오타 발생,
장애물에 문자열이 닿을 시
배터리 - 20%, 점수 - 10점

2) 화면 구성 요소

① 배터리

: 5회의 오답 기회 -> 게임 종료

② 메인 보드

: 랜덤 글자 출력, 장애물 배치

③ 텍스트 박스

: 사용자의 단어 입력 받음

④ Total Score 기록

: 정답, 오답 개수/ 누적 점수
: 평균 타수/ 연습 시간 / 정확도

⑤ 사용자 개별 설정 요소

: Mode - Easy / Hard

: Obstacle - 난이도 세분화를
위한 중간 장애물(검정 버그
이미지)

: Speed - 속도 조정

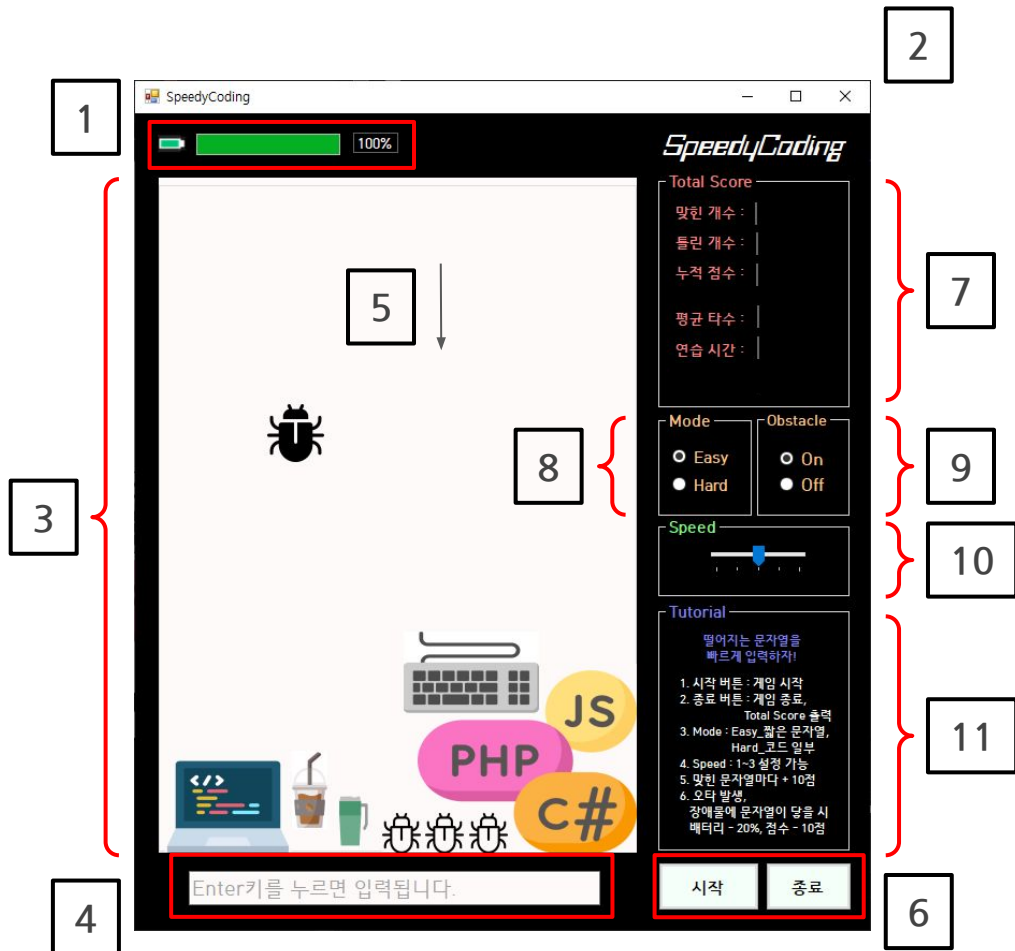
⑥ 게임 시작/종료 버튼

: 시작 ↔ 일시 정지 전환 가능

: 종료 ↔ 초기화 전환 가능

구현 화면

5



2. 품 구성하기

번호	컨트롤 종류	속성	설정 값
1-1	Picture Box	(Image)	
1-2	Progress Bar	(Name)	pgBarBattery
		(Min / Max)	0 / 5
1-3	Label	(Name)	lblBattery
2	Form	(Name)	Form
3	Group Box	(Name)	grpMain
4	Text Box	(Name / Text)	txtBoxTyping / Enter키를 누르면 입력됩니다.
5	Label	(Name / Visible)	lblWords / False
6-1	Button	(Name / Text)	btnStart / 시작 (↔ 일시 정지)
6-2		(Name / Text)	btnEnd / 종료 (↔ 초기화)

[메인 보드 내 장애물 배치 : 충돌 시 고려할 변]

3 - 1) 메인 보드 중간 : 상, 하, 좌, 우

3 - 2) 메인 보드 하단 : 상, 좌, 우

3 - 3, 4, 5) 메인보드 하단 : 상

3-1	Picture Box	(Name)	bBugImg
3-2		(Name)	keyboardImg
3-3		(Name)	languageImg
3-4		(Name)	laptopImg, coffeelmg, tumblerImg, wBugImg

2. 품 구성하기

번호	컨트롤 종류	속성	설정 값
7	Group Box	(Name / Text)	grpScore / Total Score
7-1	Label	(Name / Text)	lblCorrect / 맞힌 개수
7-2		(Name / Text)	lblWrong / 틀린 개수
7-3		(Name / Text)	lblTotal / 누적 점수
7-4		(Name / Text)	lblAvg / 평균 타수
7-5		(Name / Text)	lblTime / 연습 시간
8	Group Box	(Name / Text)	grpMode / Mode
8-1	Radio Button	(Name / Text)	rbtnEasy / Easy
8-2	Radio Button	(Name / Text)	rbtnHard / Hard
		Checked	False

9	Group Box	(Name / Text)	grpbBug / Obstacle
9-1	Radio Button	(Name / Text)	rbtnbOn / On
9-2	Radio Button	(Name / Text)	rbtnbOff / Off
		Checked	False

10	Group Box	(Name / Text)	grpSpeed / Speed
10-1	TrackBar	(Name)	tkBarSpeed
		(Min / Max)	1 / 5

11	Group Box	(Name / Text)	grpTutorial / Tutorial
11-1	Label	(Name)	lblTutorial

3. 코드 작성하기

```
namespace 코딩타자연습게임
{
    참조 4개
    public partial class Form : System.Windows.Forms.Form
    {
        Random rnd = new Random();
        Stopwatch sw = new Stopwatch();

        int battery = 100;
        int errorCount = 0;

        string eMode, hMode;
        string[] eMList, hMList;
        string eMWord, hMWord;

        int fCorrect = 0;
        int fWrong = 0;
        int fTotal = 0;

        double correctCharCount = 0;
        double avgSpeed = 0;

        참조 1개
        public Form()
        {
            InitializeComponent();
        }
    }
}
```

▼ Form_Load 이벤트 처리

- 화면 로드 시 기본 설정
- 리소스 > 파일 : 단어 txt 불러오기

◀ 변수 설정

```
참조 1개
private void Form_Load(object sender, EventArgs e)
{
    Size = new Size(700, 800); //게임 전체 영역 설정
    lblWords.Visible = false; //폼이 로드 되었을 때에는 글자가 보이지 않음
    rbtnOn.Checked = true; //화면 중간 검정 버그 이미지 : 기본 장애물로 설정
    rbtnEasy.Checked = true; //Mode 초기값 : Easy
    tkBarSpeed.Value = 3; //Speed 초기값 : 3
    pgBarBattery.Value = 5; //총 5번의 오타 기회
    lblBattery.Text = $"{battery}%";

    //단어 파일 불러오기
    //Mode : Easy
    eMode = Properties.Resources.easywords_C_;
    eMList = eMode.Split('\n');

    //Mode : Hard
    hMode = Properties.Resources.hardwords_C_;
    hMList = hMode.Split('\n');
}
```

◀ Easy Mode 용

◀ Hard Mode 용

▼ 랜덤 단어 생성 함수 : Mode에 따른 2가지 버전

- 랜덤 생성 => 수직 하강하는 단어 박스에 입력
- 단어 박스의 x좌표 랜덤 출력 & 영역 제한

//단어 생성 함수

참조 3개

private void makeWords()

```
{
    if (rbtnEasy.Checked)
    {
        //Mode : Easy - 랜덤으로 단어 선택
        int eMIndex = rnd.Next(0, eMList.Length);
        eMWord = eMList[eMIndex];

        lblWords.Text = eMWord.ToString();

        //단어의 x좌표가 부모 컨테이너인 grpMain 영역 밖을 넘어가지 않게 설정
        lblWords.Left = rnd.Next(grpMain.Width - lblWords.Width);
        //grpMain 상단에서부터 단어의 y좌표가 시작되게 함.
        lblWords.Top = grpMain.Top;
    }
    else
    {
        //Mode : Hard - 랜덤으로 단어 선택
        int hMIndex = rnd.Next(0, hMList.Length);
        hMWord = hMList[hMIndex];

        lblWords.Text = hMWord.ToString();

        int maxWidth = Math.Max(grpMain.Width - lblWords.Width, 0); //둘 중 더 큰 값을 변수에 저장
        lblWords.Left = rnd.Next(maxWidth);
        lblWords.Top = grpMain.Top;
    }
}
```



◀ Easy Mode 용

◀ Hard Mode 용

[예상 개선점]

- 랜덤 생성 단어의 길이와 메인 보드의 가로 길이 간 구체적 조정
- 단어 파일 및 배열 내 결측치 확인



```
//단어를 담은 label 컨트롤의 높이 만큼 단어를 계속 떨어뜨림
//장애물 충돌 시 Total Score에 반영, 새 단어 생성
참조 1개
private void tmrWords_Tick(object sender, EventArgs e)
{
    lblWords.Visible = true;
    lblWords.Top += lblWords.Height;

    if (rbtnb0n.Checked)
    {
        if (CheckCollision(bBugImg))
        {
            CalCollision();
        }
        else
        {
            CollisionOdds();
        }
    }
    else
    {
        CollisionOdds();
    }
}
```

[타이머 이용]
1.CheckCollision
2.CollisionOdds
3.CalCollision

```
참조 1개
private bool CheckCollision(Control obstacle)
{
    int bBLeft = bBugImg.Left;
    int bBRight = bBugImg.Right;
    int bBTop = bBugImg.Top;

    int lBlWLeft = lblWords.Left;
    int lBlWRight = lblWords.Right;
    int lBlWBottom = lblWords.Bottom;

    //lblWords와 bBugImage
    return (bBRight >= lBlWLeft && bBLeft <= lBlWRight && bBTop <= lBlWBottom);
}
```

```
참조 2개
private void CollisionOdds()
{
    //단어와 장애물 간 충돌 계산
    int kLeft = keyboardImg.Left;
    int kRight = keyboardImg.Right;
    int kTop = keyboardImg.Top;

    int lTop = laptopImg.Top;
    int lgTop = languageImg.Top;
    int cTop = coffeelmg.Top;
    int tTop = tumblerImg.Top;

    int lBlWLeft = lblWords.Left;
    int lBlWRight = lblWords.Right;
    int lBlWBottom = lblWords.Bottom;

    //grpMain 하단의 장애물들
    if (kRight >= lBlWLeft && kLeft <= lBlWRight && kTop <= lBlWBottom
        || lgTop <= lBlWBottom || lTop <= lBlWords.Bottom
        || cTop <= lBlWords.Bottom || tTop <= lBlWords.Bottom)
    {
        CalCollision();
    }
}
```

[메인 보드 내 장애물]
: 충돌 시 고려할 번

3 - 1)
메인 보드 중간 : 상, 하, 좌, 우
3 - 2)
메인 보드 하단 : 상, 좌, 우
3 - 3, 4, 5) 메인보드 하단 : 상

```
참조 2개
private void CalCollision()
{
    fWrong += 1;
    lblWrong.Text = fWrong.ToString();
    fTotal -= 10;
    lblTotal.Text = fTotal.ToString();
    lblWords.Text = "";
    makeWords();
}
```

3-1 ▶

◀ 3-2

◀ 3-3

3-4

3-5

참조 1개

```
private void btnStart_Click(object sender, EventArgs e)
{
    //일시 정지 버튼을 누른 상태
    if (tmrWords.Enabled)
    {
        tmrWords.Stop();

        //연습 시간 기록 일시 정지
        sw.Stop();

        txtBoxTyping.Text = "";
        this.ActiveControl = txtBoxTyping;
        txtBoxTyping.Focus();
        btnStart.Text = "시작";

        CalRecords();
        lblWords.Visible = false;
    }
    //시작 버튼을 누른 상태
    else
    {
        txtBoxTyping.Text = "";
        this.ActiveControl = txtBoxTyping;
        txtBoxTyping.Focus();

        makeWords();
        tmrWords.Start();

        //연습 시간 기록 시작
        sw.Start();

        //게임이 시작되면 시작 버튼이 일시 정지 버튼으로 바뀜
        btnStart.Text = "일시 정지";
    }
}
```

참조 1개

```
private void btnEnd_Click(object sender, EventArgs e)
{
    if (tmrWords.Enabled)
    {
        tmrWords.Stop();
        btnEnd.Text = "초기화";
        txtBoxTyping.Text = "";
        lblWords.Visible = false;

        CalRecords();
    }
    else
    {
        //전체 값 초기화
        txtBoxTyping.Text = "";
        this.ActiveControl = txtBoxTyping;
        txtBoxTyping.Focus();
        lblWords.Visible = false;

        fCorrect = 0;
        fWrong = 0;
        fTotal = 0;
        correctCharCount = 0;
        avgSpeed = 0;
        lblCorrect.Text = "0";
        lblWrong.Text = "0";
        lblTotal.Text = "0";
        lblTime.Visible = false;
        lblAvg.Visible = false;

        pgBarBattery.Value = 5;
        errorCount = 0;
        battery = 100;
        lblBattery.Text = $"{battery}%";

        sw.Reset();

        btnStart.Text = "시작";
        //초기화 버튼을 누르면 해당 버튼 텍스트가 종료 버튼으로 바뀜
        btnEnd.Text = "종료";
    }
}
```

참조 3개

private void CalRecords()

```
{
    //연습 시간 출력
    sw.Stop();
    TimeSpan elapsed = sw.Elapsed; //시간 기록을 TimeSpan(시간 간격)으로 얻음
    double minutes = elapsed.Minutes;
    double seconds = elapsed.Seconds;
    lblTime.Visible = true;
    lblTime.Text = minutes + "분 " + seconds + "초";

    //평균 타수 출력
    double TotalMinutes = elapsed.TotalMinutes;
    if (TotalMinutes > 0)
    {
        avgSpeed = correctCharCount / TotalMinutes;
        lblAvg.Visible = true;
        lblAvg.Text = $"{avgSpeed:F2} 타";
    }
}
```




```
//txtBoxTyping에 이벤트 발생 시
참조 1개
private void txtBoxTyping_TextChanged(object sender, EventArgs e)
{
    txtBoxTyping.ForeColor = Color.Black;
}
```

```
참조 1개
private void txtBoxTyping_KeyDown(object sender, KeyEventArgs e)
{
    if(e.KeyCode == Keys.Enter)
    {
        // 떨어지던 문자열과 입력 문자열 비교 (대소문자 구분)
        string typedWord = txtBoxTyping.Text;
        string displayedWord = lblWords.Text.Trim(); // 표시된 단어에서 앞뒤 공백 제거

        if (string.Compare(typedWord, displayedWord) == 0)
        {
            fCorrect += 1;
            lblCorrect.Text = fCorrect.ToString();
            fTotal += 10;
            lblTotal.Text = fTotal.ToString();

            lblWords.Text = ""; //두 문자열이 일치할 때 랜덤 단어 초기화
            makeWords(); //새로운 단어 생성

            //옳게 입력한 단어에 한해서 타수 기록
            correctCharCount += typedWord.Length;
        }
        else
        {
            fWrong += 1;
            lblWrong.Text = fWrong.ToString();
            fTotal -= 10;
            lblTotal.Text = fTotal.ToString();
        }
        txtBoxTyping.Text = "";
    }
    txtBoxTyping.Focus();
}
```



```
//좌상단 프로그레스바를 미로 활용, 5번의 오타 기회 제공
참조 1개
private void lblWrong_TextChanged(object sender, EventArgs e)
{
    errorCount++;
    //틀린 개수가 쌓일 때마다 좌상단 프로그레스바의 Value를 5에서부터 1씩 감소
    if (errorCount <= 4)
    {
        if (pgBarBattery.Value > 0)
        {
            pgBarBattery.Value -= 1;
        }
        else
        {
            pgBarBattery.Value = 0;
            tmrWords.Stop();
            lblWords.Visible = false;
        }

        if (battery >= 20)
        {
            battery -= 20;
        }
        lblBattery.Text = $"{battery}%";
    }
    else
    {
        lblBattery.Text = "Game Over!!";
        pgBarBattery.Value = 0;
        tmrWords.Stop();
        lblWords.Visible = false;
        CallRecords();
        btnEnd.Text = "초기화";
    }
}
```

//속도 설정

참조 1개

```
private void tkBarSpeed_Scroll(object sender, EventArgs e)
{
    if (rbtnEasy.Checked)
    {
        if (tkBarSpeed.Value == 1)
        {
            tmrWords.Interval = 1000;
        }
        else if (tkBarSpeed.Value == 2)
        {
            tmrWords.Interval = 800;
        }
        else if (tkBarSpeed.Value == 3)
        {
            tmrWords.Interval = 600;
        }
        else if (tkBarSpeed.Value == 4)
        {
            tmrWords.Interval = 400;
        }
        else if (tkBarSpeed.Value == 5)
        {
            tmrWords.Interval = 200;
        }
    }
}
```

◀ Easy Mode 용

```
else
{
    if (tkBarSpeed.Value == 1)
    {
        tmrWords.Interval = 1000;
    }
    else if (tkBarSpeed.Value == 2)
    {
        tmrWords.Interval = 800;
    }
    else if (tkBarSpeed.Value == 3)
    {
        tmrWords.Interval = 600;
    }
    else if (tkBarSpeed.Value == 4)
    {
        tmrWords.Interval = 400;
    }
    else if (tkBarSpeed.Value == 5)
    {
        tmrWords.Interval = 200;
    }
}
```

◀ Hard Mode 용

감사합니다