

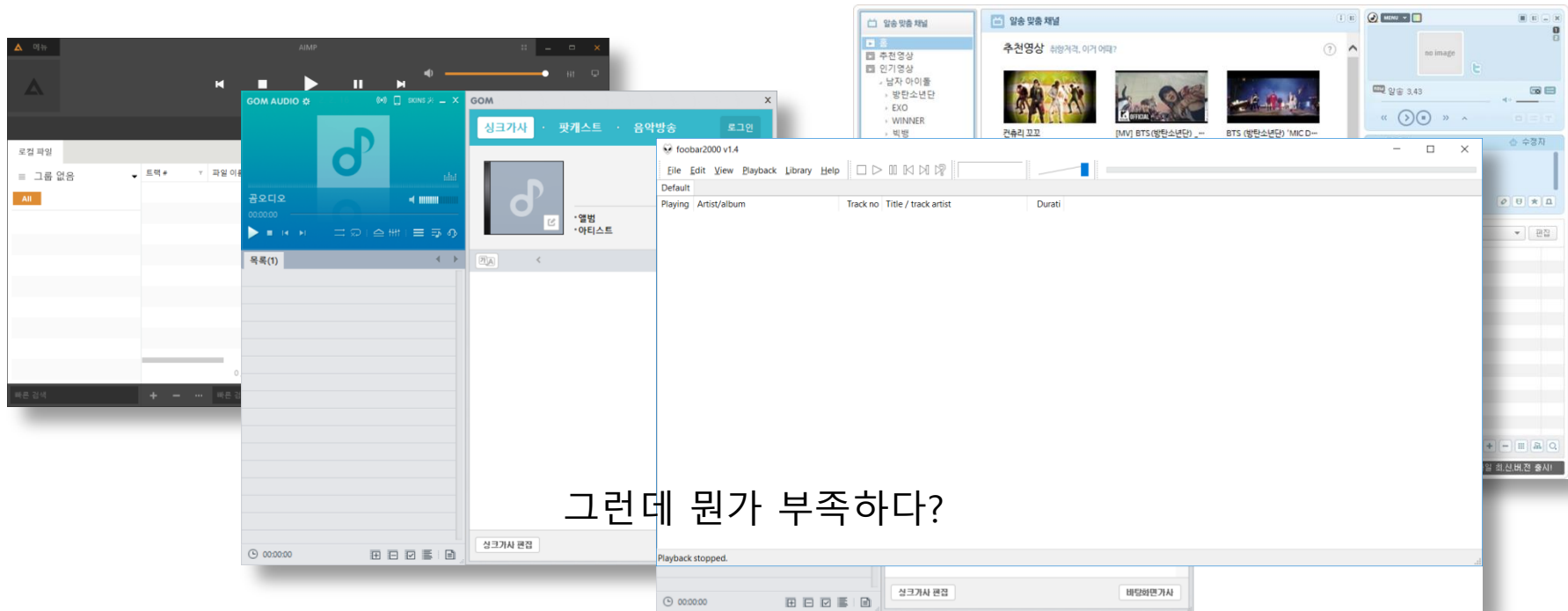
오디오 라이브러리 CSCore를 사용한 음악 플레이어 개발

3학년 의반 박현우

개발 동기



시중에 존재하는 다양한 음악 플레이어들



개발 동기



기존의 음악 플레이어의 아쉬운 점들

빈약한 디자인

음악 재생과 상관없는 잡다한 기능이 많아 무거움

번잡한 광고

가사를 지원하지 않음

구조가 직관적이지 못해 사용하기 불편함

등등등...

하나가 충족되는 건 다른 하나가 없고...

그럼 직접 만들어보자!

개발 계획



빈약한 디자인

- C#/WPF에 Material 디자인 라이브러리를 사용하여 개발

음악 재생과 상관없는 잡다한 기능이 많아 무거움

- 음악 플레이어 본연에 충실하게 필요한 기능만 개발

번잡한 광고

- 광고를 넣지 않고 깔끔하게 개발

가사를 지원하지 않음

- PHP + Slim 프레임워크로 가사 서버 개발

구조가 직관적이지 못해 사용하기 불편함

- 사용자가 알아보기 쉽게 직관적으로 컨트롤을 배치하여 개발

나만의 음악 플레이어, 'Enoz'



재생목록 가사 설정

폴더

오디오 앨범

검색...

Music

재생목록

재생목록 1

+

| | | |
|---|--|------|
| Yonde Mirakurun (Instrumental) | Majokko Mirakurun (CV Ayana Taketatsu) | 3:33 |
| Pyua Pyua Mirakurun (Instrumental) | Majokko Mirakurun (CV Ayana Taketatsu) | 3:26 |
| Shiawase Gift | Akari Akaza (CV Shiori Mikami) | 3:55 |
| Akaza gekijō | Akari Akaza (CV Shiori Mikami) | 4:00 |
| Shiawase Gift (Instrumental) | Akari Akaza (CV Shiori Mikami) | 3:54 |
| Akaza gekijō (Instrumental) | Akari Akaza (CV Shiori Mikami) | 3:59 |
| Ano ne~tsu! | Chinatsu Yoshikawa (CV Rumi Ookubo) | 4:13 |
| Otome biyori | Chinatsu Yoshikawa (CV Rumi Ookubo) | 4:06 |
| Ano ne~tsu! (Instrumental) | Chinatsu Yoshikawa (CV Rumi Ookubo) | 4:13 |
| Otome biyori (Instrumental) | Chinatsu Yoshikawa (CV Rumi Ookubo) | 4:05 |
| Hashaide mitari Nayande mitari Koisuru Otome wa Yuruyuri | Kyoko Toshino (CV Yuka Ootsubo) | 3:46 |
| Hallelujah Itsumo | Kyoko Toshino (CV Yuka Ootsubo) | 4:18 |
| Hashaide mitari Nayande mitari Koisuru Otome wa Yuruyuri (Instrumental) | Kyoko Toshino (CV Yuka Ootsubo) | 3:46 |
| Hallelujah Itsumo (Instrumental) | Kyoko Toshino (CV Yuka Ootsubo) | 4:16 |
| E kagen YUI kagen | Yui F unami (CV Minami Tsuda) | 3:42 |

胸がとけ
舞臺がとけ
がさがとけ

Hashaide mitari Nayande
Yuru Yuri no Uta 2 Character S

0:42 3:45

‘Enoz’의 특징



C# 오디오 라이브러리 CScore 사용



C#으로 작성된 고급 오디오 라이브러리로 오디오 재생 / 기록에서 오디오 스트림 / 파일 인코딩 / 디코딩 / 인코딩에 이르기까지 수많은 실시간 오디오 데이터 처리(예 : 재생 중 사용자 정의 효과 적용, 시각화 생성 등)를 제공한다.

```
132 public async void OpenAudio(Audio audio)
133 {
134     StopPlayback();
135
136     if (CurrentAudio != null)
137         CurrentAudio.IsOpened = false;
138
139     audio.IsOpened = true;
140
141     CurrentAudio = audio;
142     AudioChanged?.Invoke(this, new AudioChangedEventArgs(CurrentAudio));
143
144     CurrentLyrics = null;
145     Lyrics = null;
146
147     Lyrics = await GetLyricsByFile(CurrentAudio.FileName);
148
149     Output.Load(audio.FileName);
150
151     RaisePropertyChanged("Length");
152     RaisePropertyChanged("LongLength");
153
154     Output.Volume = Volume;
155     StartVisualization?.Invoke(this, EventArgs.Empty);
156     TogglePlayPause();
157 }
158
159 public void StopPlayback()
160 {
161     if (!Output.IsStopped)
```


‘Enoz’의 특징



데이터베이스 시스템 SQLite 사용



서버가 아니라 응용 프로그램에 넣어 사용하는 비교적 가벼운 데이터베이스 시스템. 구글 안드로이드 운영체제에 기본으로 탑재되어있기도 하다. Enoz는 재생 목록에 SQLite를 사용함으로써 몇 백개의 음악 파일도 빠르게 읽어올 수 있고 음악 검색 또한 빠르게 할 수 있다.


```
157
158 public BackgroundWorker Inserts(DirectoryInfo directory)
159 {
160     BackgroundWorker worker = new BackgroundWorker();
161     worker.WorkerReportsProgress = true;
162     worker.WorkerSupportsCancellation = true;
163     worker.DoWork += async (sender, e) =>
164     {
165         try
166         {
167             SQLiteCommand cmd = new SQLiteCommand($"CREATE TABLE {directory.Name.Replace(" ", "_")} (Title VARCHAR, Artist VARCHAR, All
168             await cmd.ExecuteNonQueryAsync();
169             cmd.Dispose();
170
171             int i = 0;
172
173             string[] extensions = OutputSource.SupportedFiles.Split('|')[1].Split(';');
174             FileInfo[] files = directory.GetFiles("*.");
175             int count = files.Length;
176             foreach (FileInfo file in files)
177             {
178                 if (worker.CancellationPending)
179                 {
180                     e.Cancel = true;
181                     return;
182                 }
183
184                 if (extensions.Contains($"{file.Extension}"))
185                     Insert(directory.Name, file.FullName, AudioDb.Folder);
```

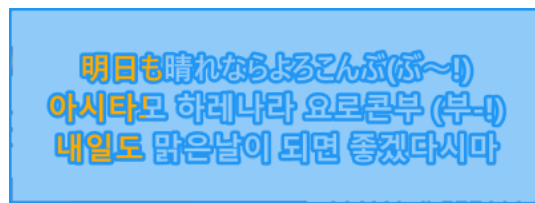
‘Enoz’의 특징



뛰어난 가사 지원



자체 가사 서버나 Irc 파일에서 가사를 받아
와 플레이어 창과 모니터 화면에 표시함. 또
외부 플러그인을 추가하면 다양한 가사 서버
에서 가사를 받아올 수 있음.



가사창에 노래방 효과가 적용된다!

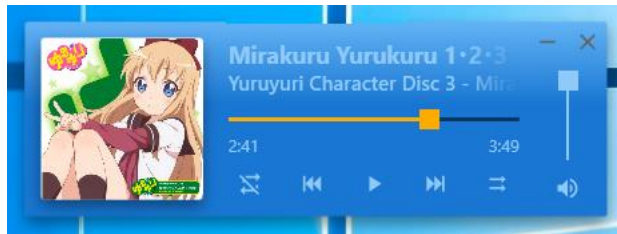
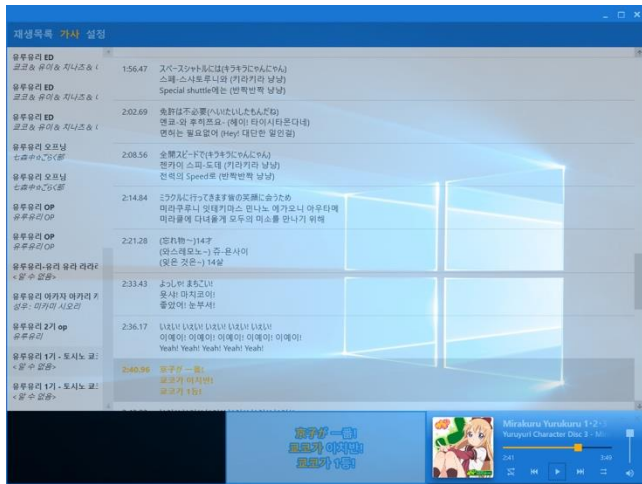
```
53
54 private void Source_LyricsChanged(object sender, LyricsChangedEventArgs e)
55 {
56     Dispatcher.Invoke(DispatcherPriority.ApplicationIdle, new Action(() =>
57     {
58         LinearGradientBrush brush = Tag as LinearGradientBrush;
59         brush.GradientStops[1].Offset = brush.GradientStops[2].Offset = 0;
60
61         if (e.NewLyrics != null)
62         {
63             int index = e.Document.Lyrics.IndexOf(e.NewLyrics);
64             karaokeItems.ItemsSource = e.NewLyrics.Content.Split(new[] { Environment.NewLine }, StringSplitOptions.RemoveEmptyEntries);
65             _sectionMilliseconds = ((index < e.Document.Lyrics.Count - 1 ? e.Document.Lyrics[index + 1].Timestamp : Source.Length) - e
66             _currentLyrics = e.NewLyrics;
67         }
68     }));
69 }
70
71 private void Source_PositionChanged(object sender, PositionChangedEventArgs e)
72 {
73     Dispatcher.Invoke(DispatcherPriority.ApplicationIdle, new Action(() =>
74     {
75         if (_currentLyrics != null)
76         {
77             LinearGradientBrush brush = Tag as LinearGradientBrush;
78             brush.GradientStops[1].Offset = brush.GradientStops[2].Offset = Math.Min(Math.Max((e.NewPosition - _currentLyrics.Timestamp
79         }
80     }));
81 }
82
83 private void Source_AudioChanged(object sender, AudioChangedEventArgs e)
```

‘Enoz’의 특징



도킹·언도킹으로 미니플레이어 전환

우측 하단의 플레이어를 드래그해서 직관적으로 간편하게 미니플레이어로 전환이 가능



```
89 private void MouseMove(object sender, MouseEventArgs e)
90 {
91     if (IsDocked)
92         return;
93
94     if (Mouse.LeftButton == MouseButtonState.Pressed)
95     {
96         PlayerView view = _regionManager.Regions["PlayerRegion"].Views.FirstOrDefault() as PlayerView;
97         Point point = Mouse.GetPosition(null);
98         Vector diff = point - _startPoint;
99
100         if (Math.Abs(diff.X) > 20 || Math.Abs(diff.Y) > 20)
101         {
102             _regionManager.Regions["PlayerRegion"].Deactivate(view);
103
104             Window window = new Window()
105             {
106                 Title = "ENOZ Mini Player",
107                 WindowStyle = WindowStyle.None,
108                 ResizeMode = ResizeMode.NoResize,
109                 AllowsTransparency = true,
110                 Background = Brushes.Transparent,
111                 TaskbarItemInfo = Application.Current.MainWindow.TaskbarItemInfo,
112                 SizeToContent = SizeToContent.WidthAndHeight,
113                 Content = new ContentControl() { Content = view, Width = 350, Height = 117.6, Margin = new Thickness(10) },
114             };
115             window.MouseLeftButtonDown += delegate
116             {
117                 window.DragMove();
118             }
119         }
120     }
121 }
```

앞으로의 계획



디자인 개선

- 지금까지 개발한 것 중에서는 가장 디자인이 좋지만, 여전히 발전해야 할 부분이 많다. 따라서 UI 디자인을 배워 더 뛰어난 음악 플레이어로 발전시키고자 한다.

가사 서버 추천

- 정식으로 배포하면 낱시 가사나, 오타와 번역 오류 등이 있는 가사들이 업로드될 것이다. 따라서 사용자의 추천수가 높은 가사에 우선순위를 부여하는 시스템을 개발하고자 한다.

사용자 설정

- 아직 사용자 설정을 구현하지 못했으므로 빠른 시일 내에 개발하고자 한다.

앞으로의 계획



태그 에디터

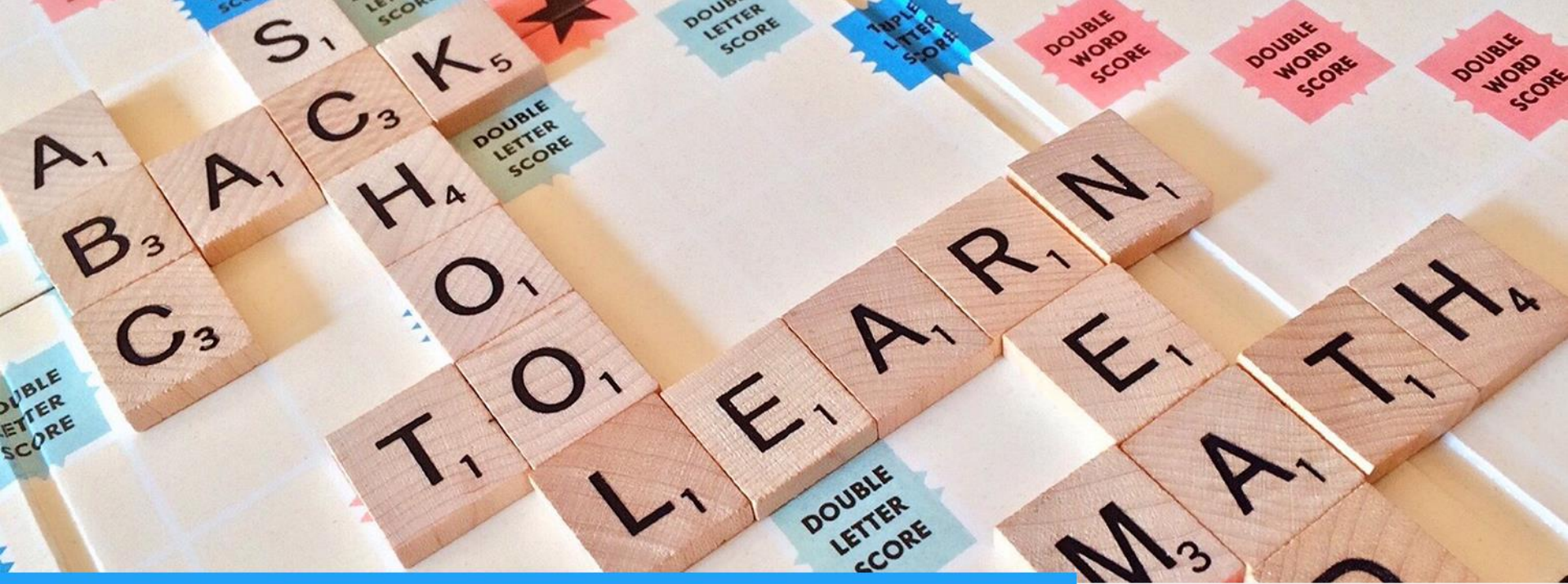
- 쉽게 음악의 태그를 수정할 수 있는 에디터를 개발하고자 한다.

가사 에디터

- lrc파일의 가사를 수정할 수 있는 에디터를 개발하고자 한다.

음악 추천 서비스

- 빅데이터와 인공지능을 배워 전국의 사용자들의 평가와 현재 사용자의 기분을 파악해 가장 적절한 음악을 추천해주는 서비스를 개발하고자 한다.



감사합니다