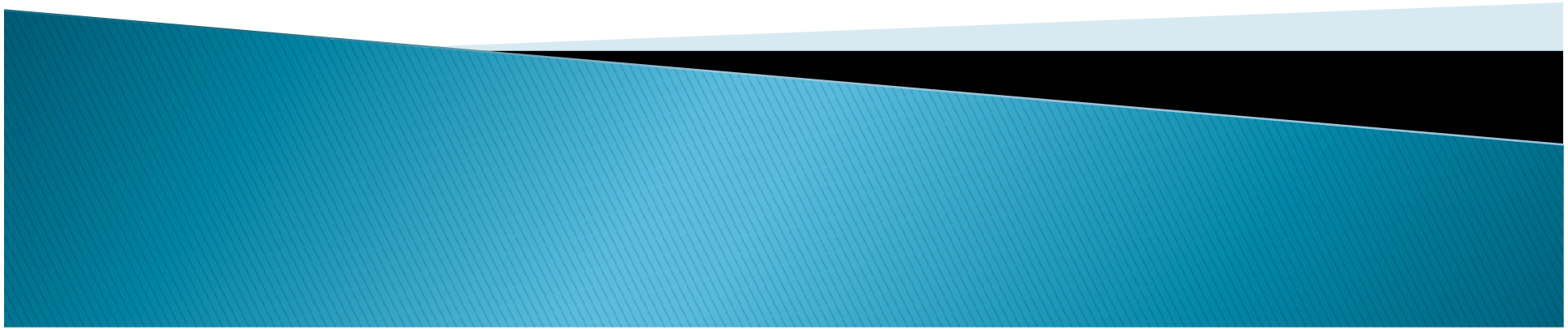
C#でゲームを作ろう2017

第3回　担当:ten



自己紹介

- ID:ten
- 京都大学工学部情報学科2回生
  - 計算機科学コース
- 第40代会長
- ゲーム制作して競プロしてる
- パズルと音ゲー

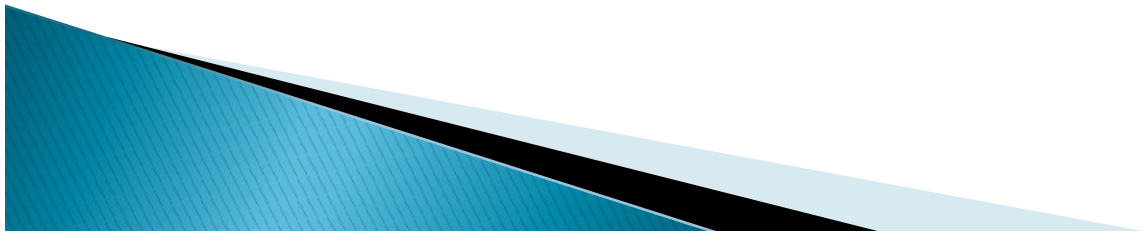
Twitter

Slackとか

自己紹介をしましょう（短縮版）

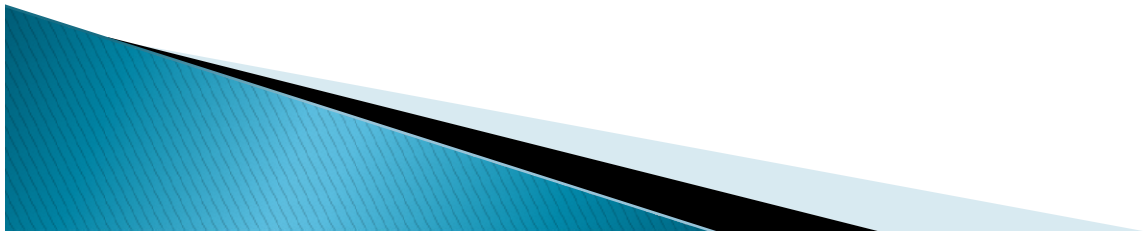
- KMC-ID (or 本名)
- 所属

- ※すべて任意です



ゲームを考えてきてね

- 夏休み後あたりに発表会をする予定
- 夏休み前に内容を終えるので、そこから自分のゲームを作ってもらう予定
- というわけでゲームの内容を考えてきてください
- いわゆる「３分ゲー」のようなもので大丈夫です

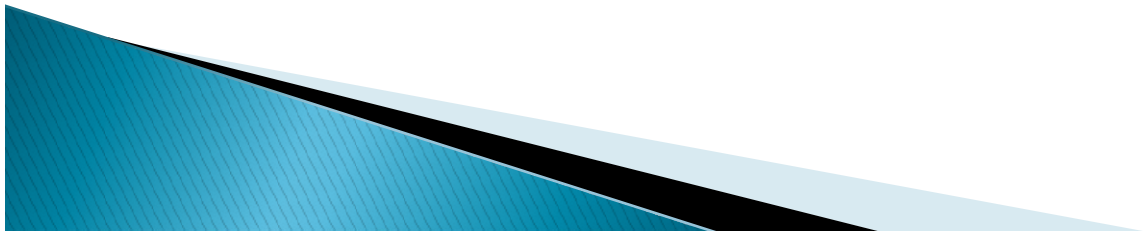


Slack

- #csgame
- 質問があればここで　or　口頭で

- この時間中の質問はいつでも受け付けています
- どんどん質問しましょう

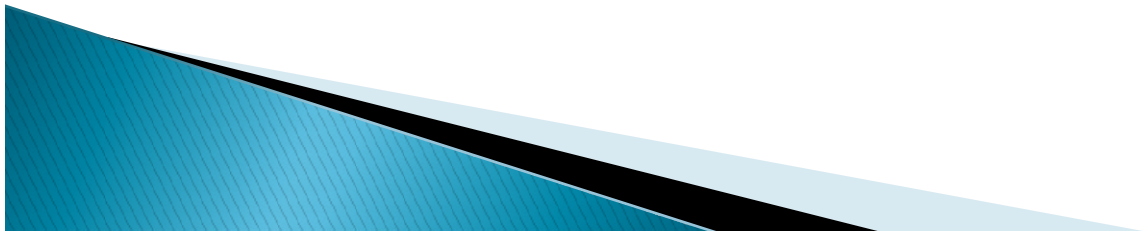
- 忘れたらスライドとかどんどん見よう
- 調べよう



Github

- https://github.com/kmc-jp/csgame2017

- これからはここにスライドとかコードとかを載せていきます

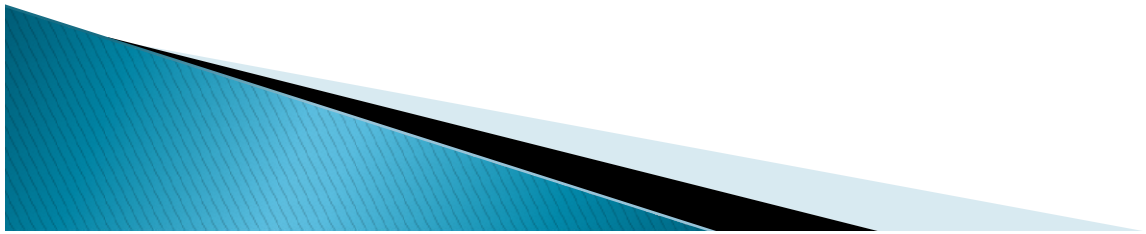


前回のまとめ


MONOGAME

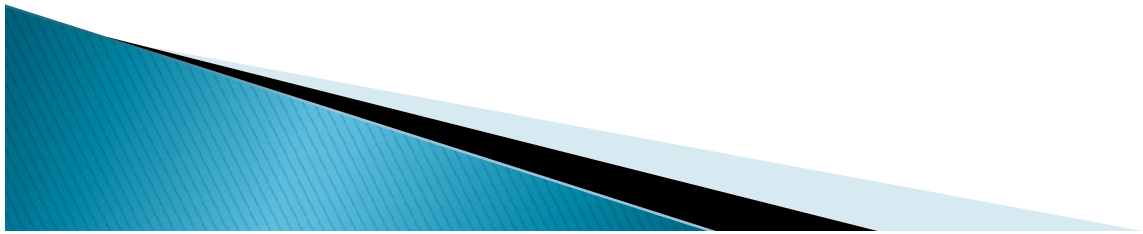
Monogameつらい

- 導入部分が死ぬほどつらい
- これを毎年続けるのか
- 無理



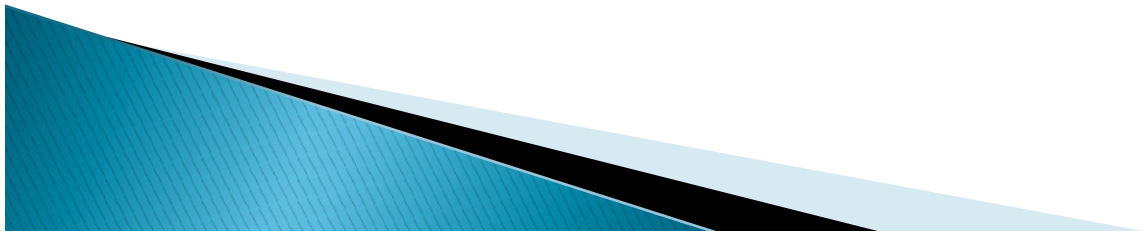
ゲームエンジン変えます

- 最初から変えてればよかったんや



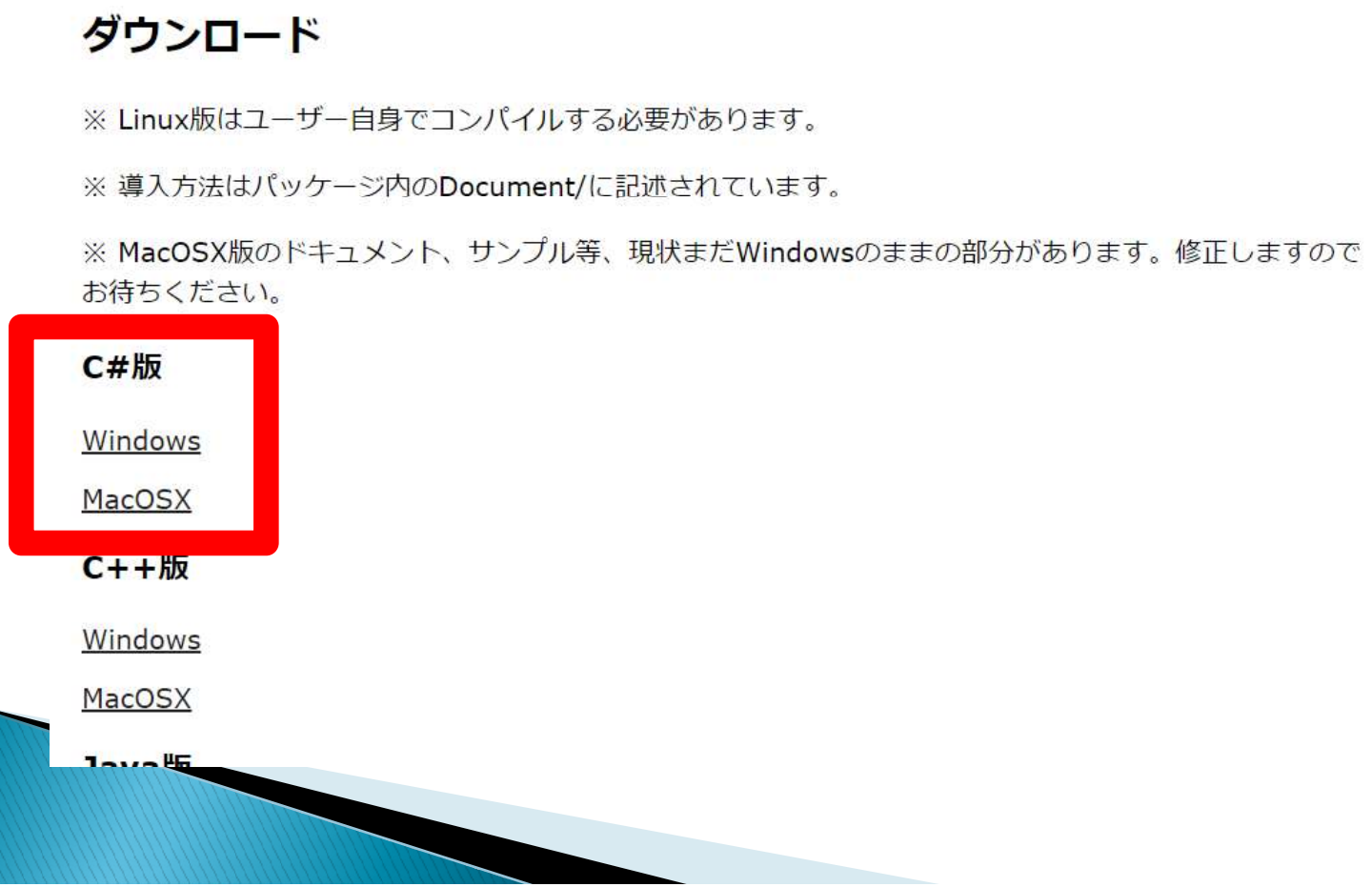
Altseed

- https://altseed.github.io/index.html
- 公式ページ
- このページにあるチュートリアルや
  ZIPファイル同梱のリファレンスを参考にしよう



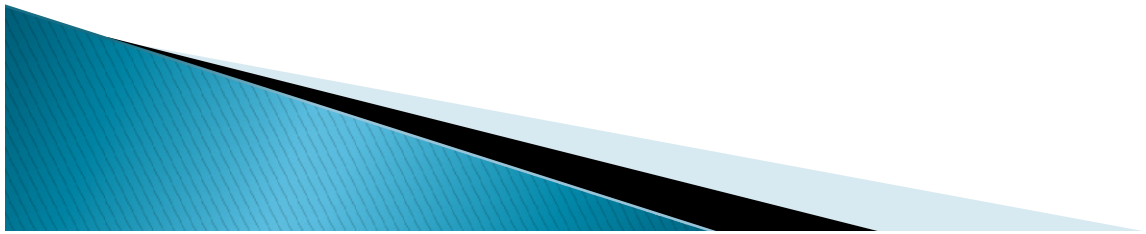
Altseedの導入

- ダウンロードページから「C#」版を選択
- ZIPファイルを好きなところに解凍

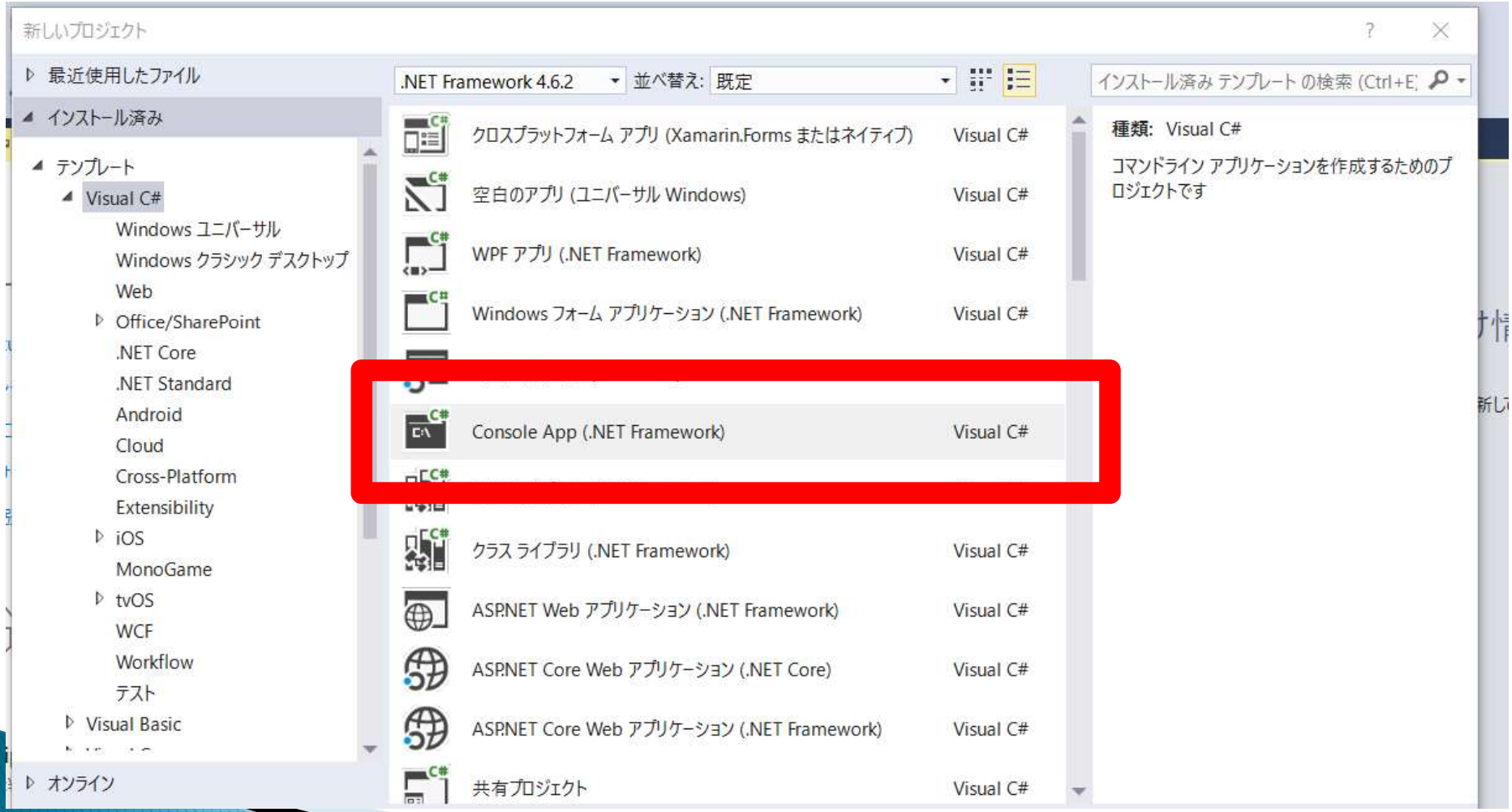


Altseedの導入

- Visual Studio
- 新しいプロジェクトをつくる

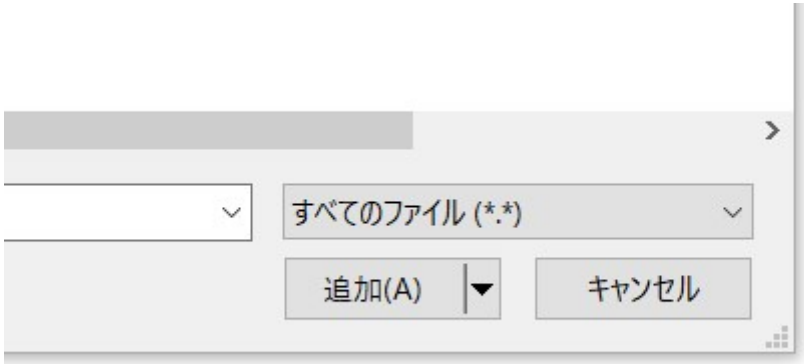


Altseedの導入

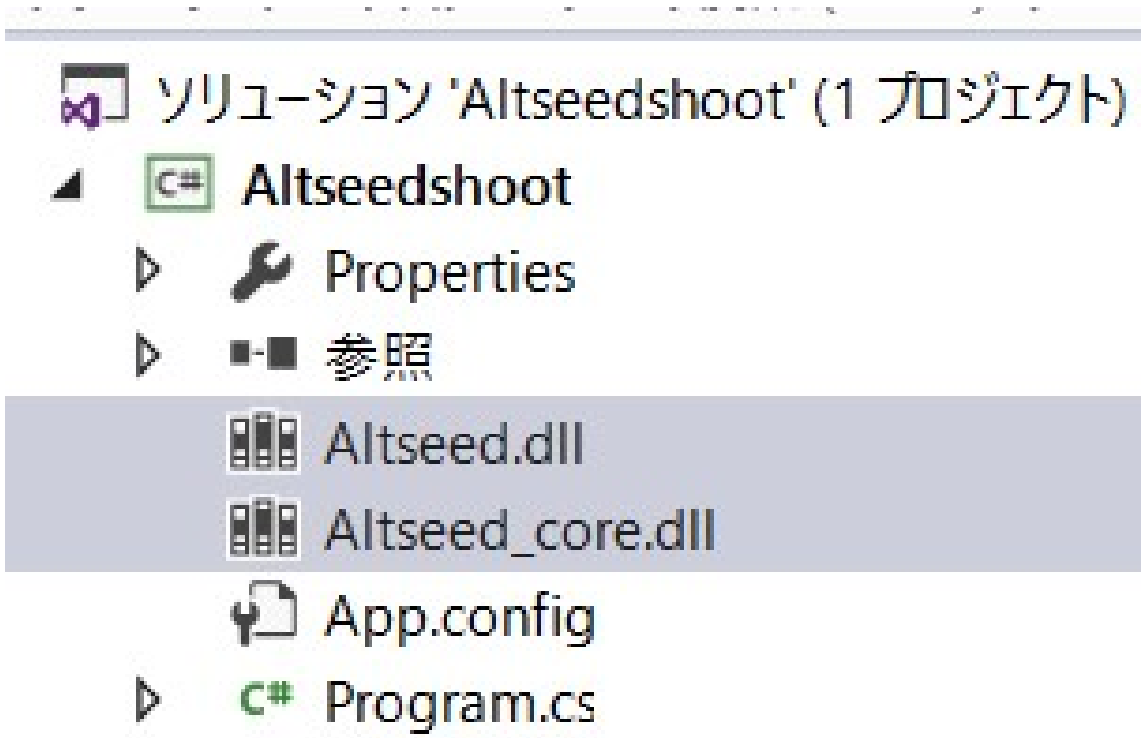

新しいプロジェクト
最近使用したファイル
インストール済み
テンプレート
Visual C#
Windows ユニバーサル
Windows クラシック デスクトップ
Web
Office/SharePoint
.NET Core
.NET Standard
Android
Cloud
Cross-Platform
Extensibility
iOS
MonoGame
tvOS
WCF
Workflow
テスト
Visual Basic
オンライン
.NET Framework 4.6.2
並べ替え: 既定
インストール済み テンプレート の検索 (Ctrl+E
クロスプラットフォーム アプリ (Xamarin.Forms またはネイティブ) Visual C#
空白のアプリ (ユニバーサル Windows) Visual C#
WPF アプリ (.NET Framework) Visual C#
Windows フォーム アプリケーション (.NET Framework) Visual C#
Console App (.NET Framework) Visual C#
クラス ライブラリ (.NET Framework) Visual C#
ASP.NET Web アプリケーション (.NET Framework) Visual C#
ASP.NET Core Web アプリケーション (.NET Core) Visual C#
ASP.NET Core Web アプリケーション (.NET Framework) Visual C#
共有プロジェクト Visual C#
種類: Visual C#
コマンドライン アプリケーションを作成するためのプロジェクトです

Altseedの導入

- 「Altseed.dll」「Altseed_core.dll」をプロジェクトに入れる
  - プロジェクト名を右クリック→追加→既存の項目
- dllファイルはさっきダウンロードしたファイルの中の「Runtime」内にあります
- 見つからないときは右下を「すべてのファイル」に

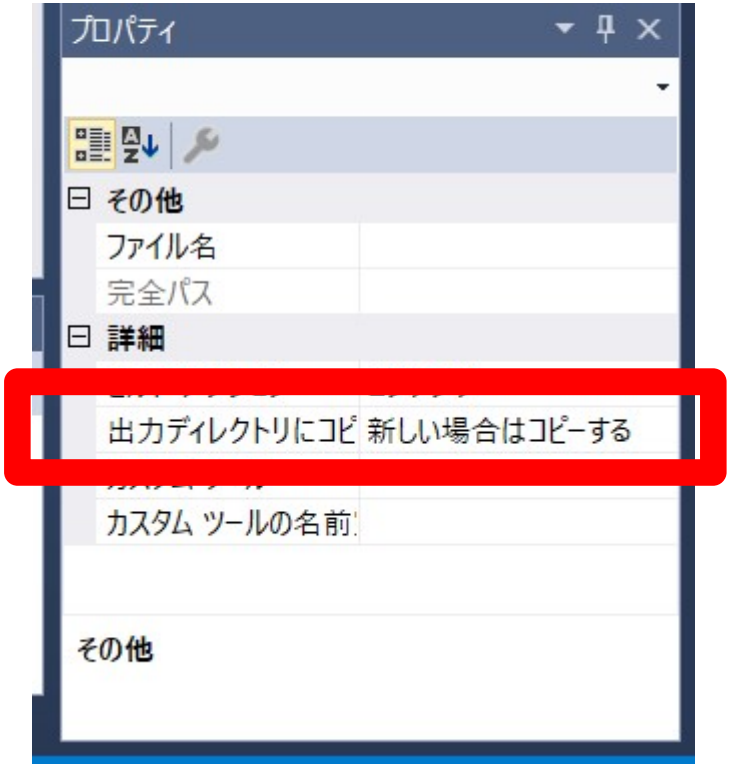


Altseedの導入


ソリューション 'Altseedshoot' (1 プロジェクト)
Altseedshoot
Properties
参照
Altseed.dll
Altseed_core.dll
App.config
Program.cs

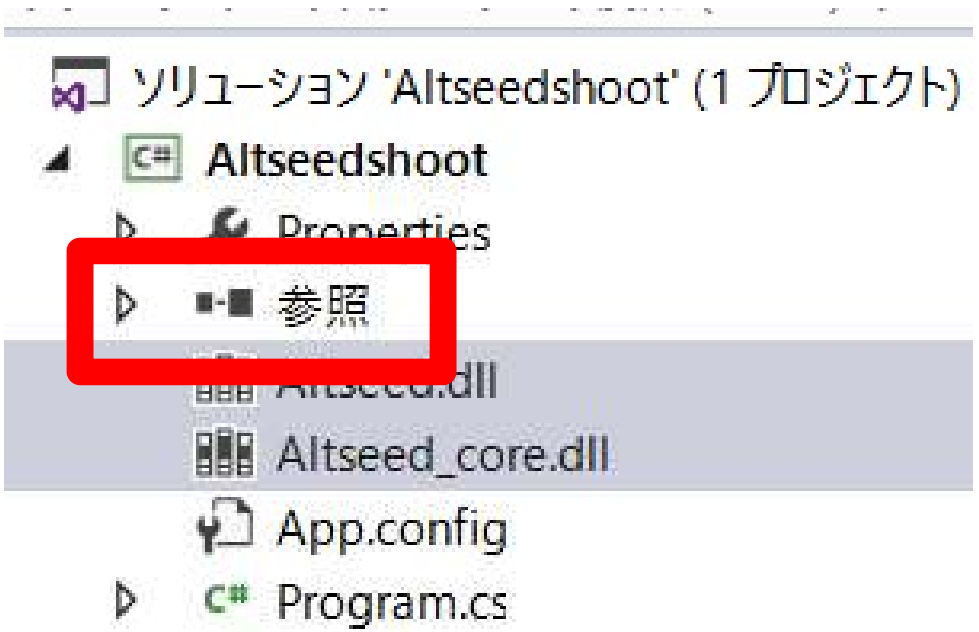
Altseedの導入

- 例によってプロパティ
- 「新しい場合はコピーする」に変更



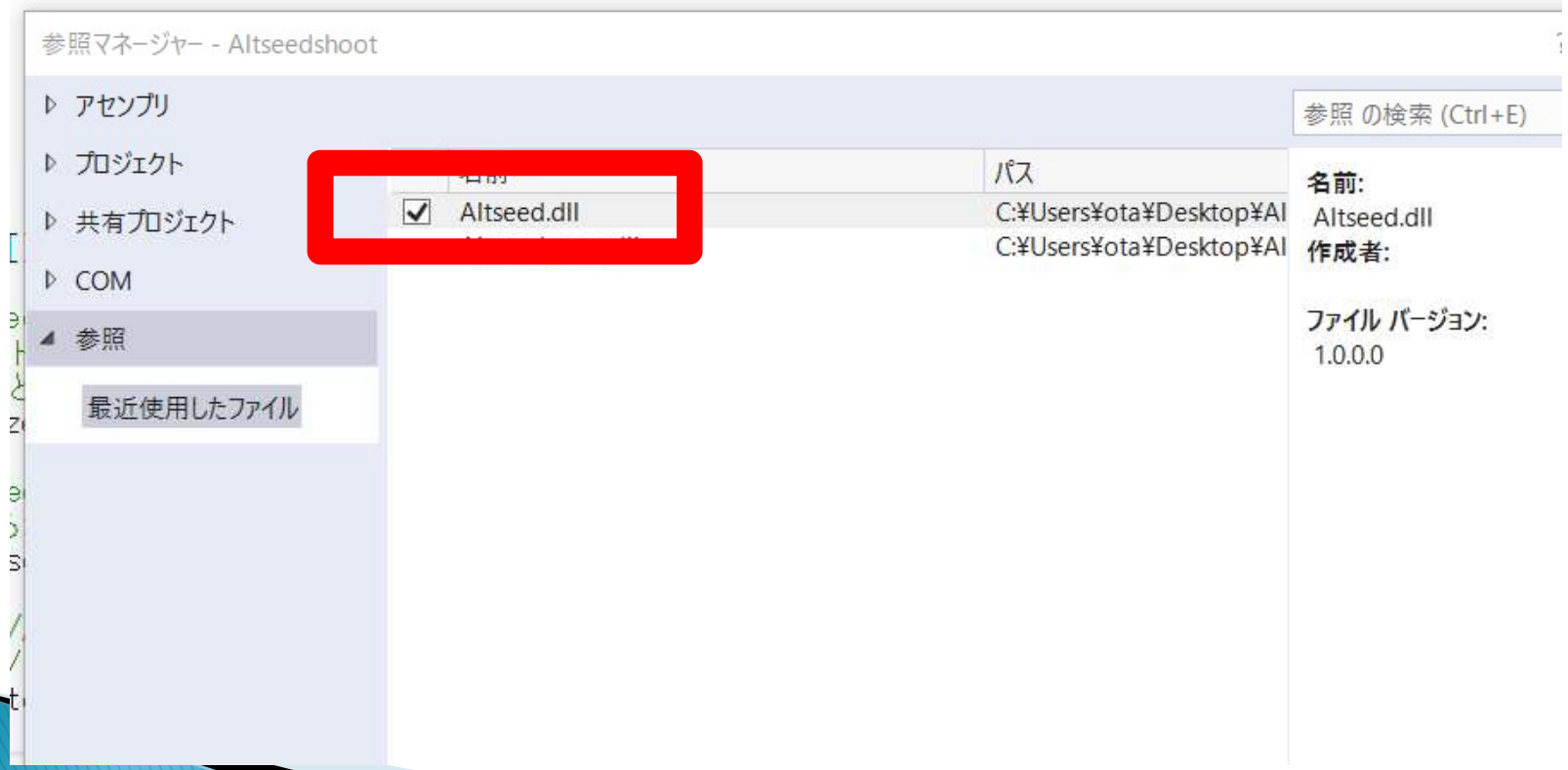
Altseedの導入

- 参照を右クリック
- 「参照の追加」



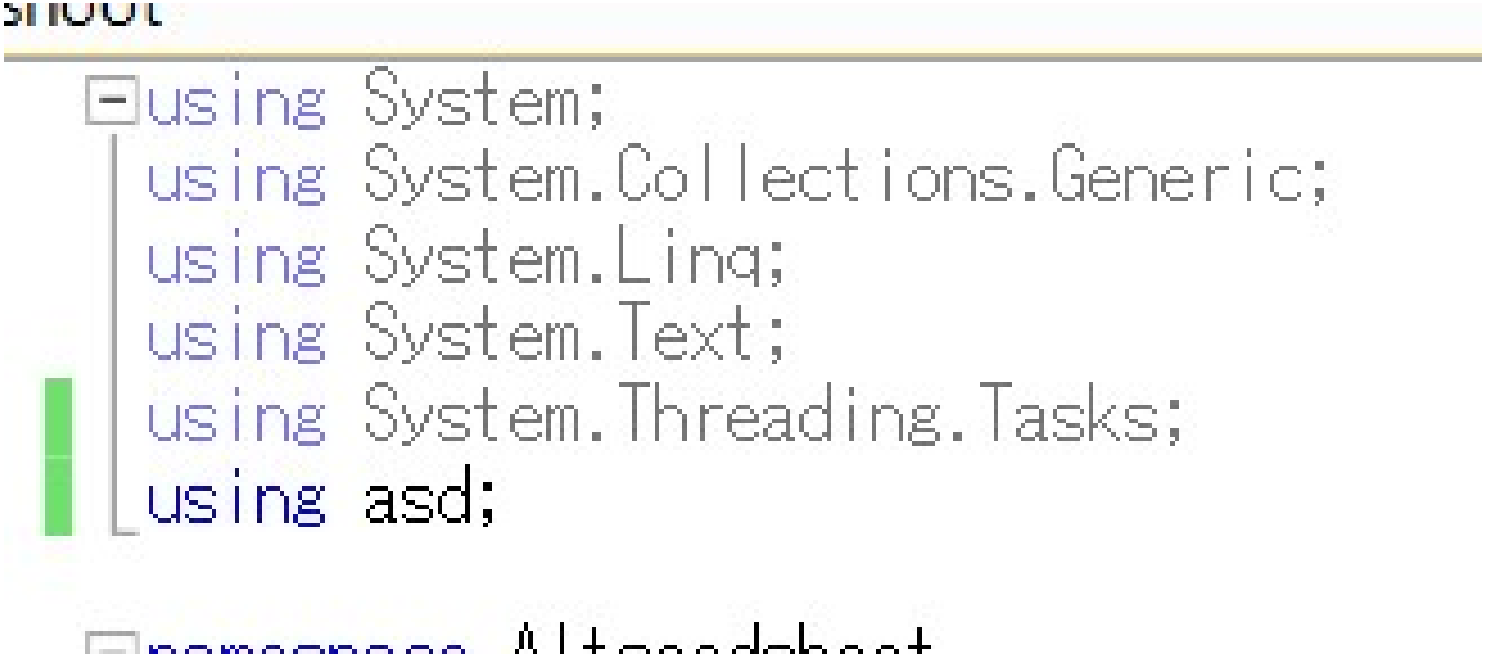
Altseedの導入

- Altseed.dllをさっきの「Runtime」から拾ってくる
- Altseed.dllにチェックしてOKする



とりあえず実行してみる

- 上のusingに「using asd;」を追加
- おまじない。



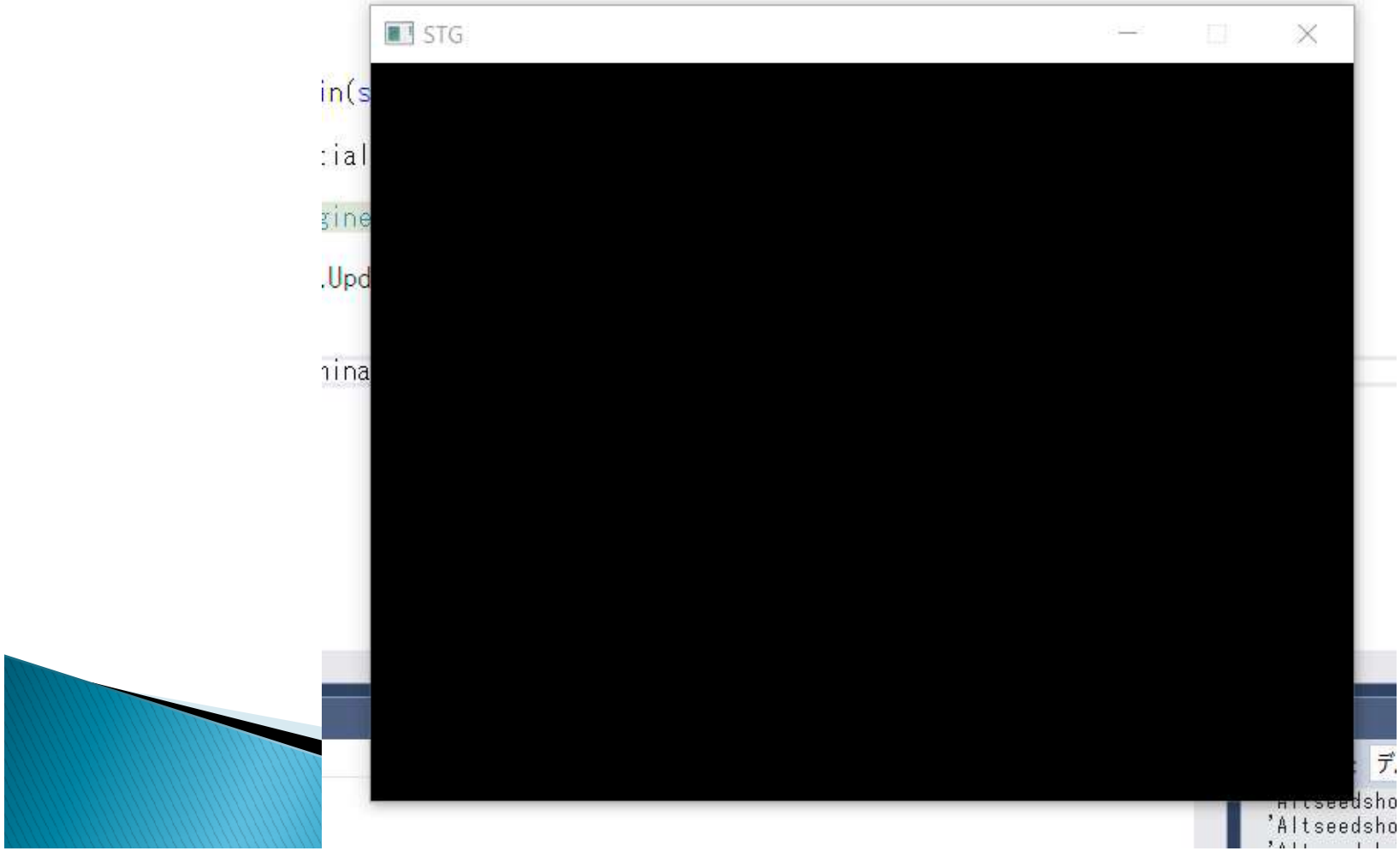
とりあえず実行してみる

- Main関数内にこう書く
- 「template.cs」

```
class Program
{
    static void Main(string[] args)
    {
        Engine.Initialize("STG", 640, 480, new EngineOption());

         while (Engine.DoEvents())
         {
            Engine.Update();
         }

           Engine.Terminate();
    }
}
```

黒い画面

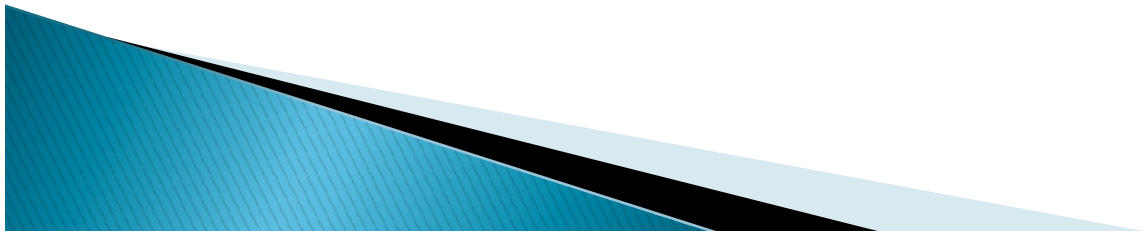
- これがでたら成功



これからやること

- 何週間かかけて簡単なシューティングを作る

- 自機が動いて弾を撃つ
- ボスが弾を撃つ、死ぬ
- HPとかボムの数とかを表示
- タイトル画面、ゲームオーバー画面

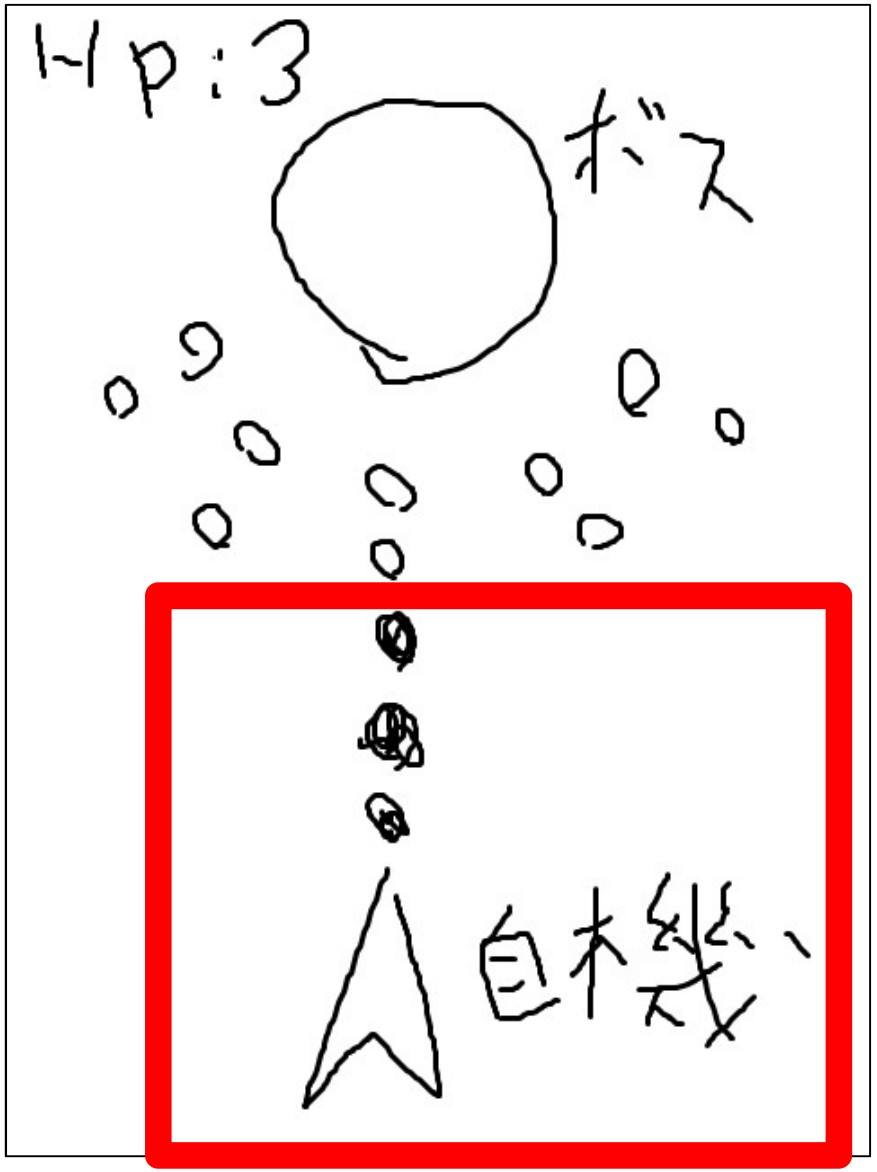


こんな感じのやつ

- 伝われ。

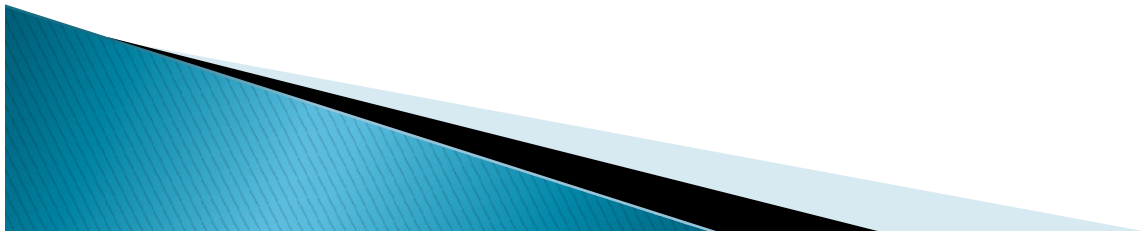
自機を作る

- 自機を作ります



自機

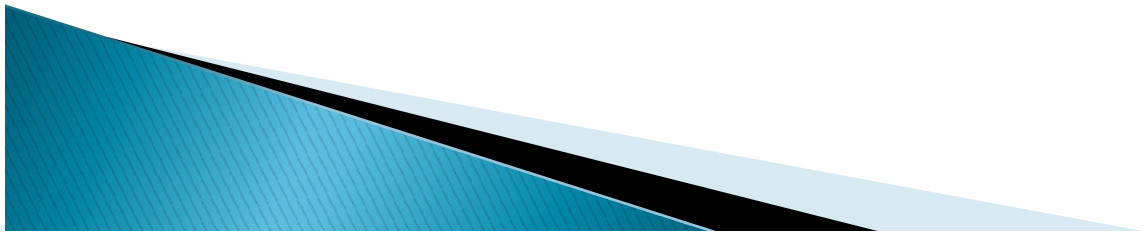
- 自機の画像を表示
- 上下左右に動く
- 弾を発射する



画像表示

- まず自機の画像を用意します

- いらすとや
- http://www.irasutoya.com/

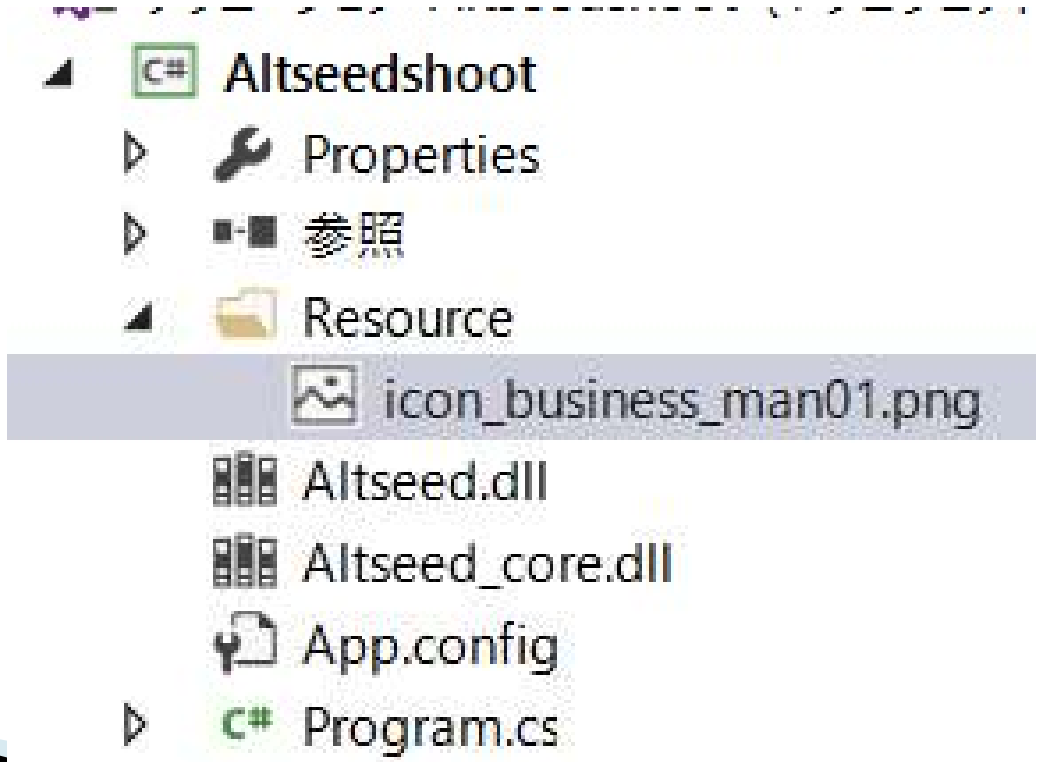


画像表示

- 今回は例としてこの画像を自機にします

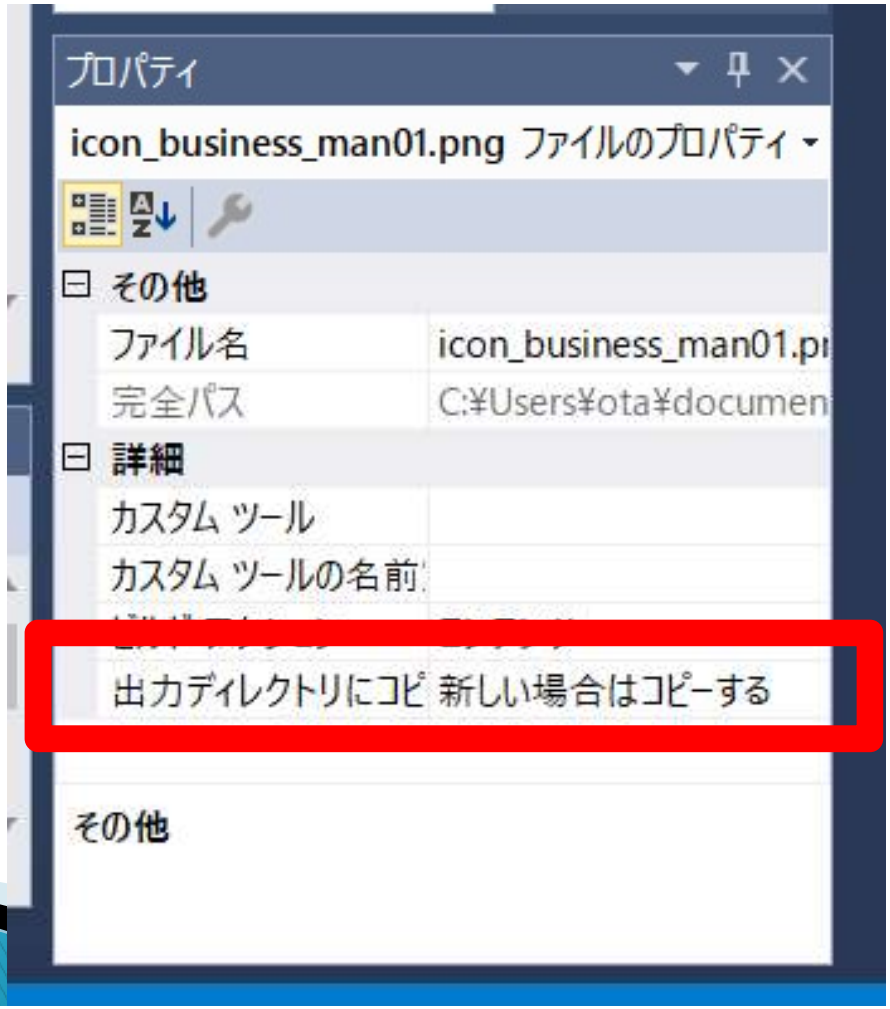
画像表示

- 「Resource」みたいなディレクトリを作る
- 画像を入れる



画像表示

- プロパティを変える！！！！！！！！！



画像表示

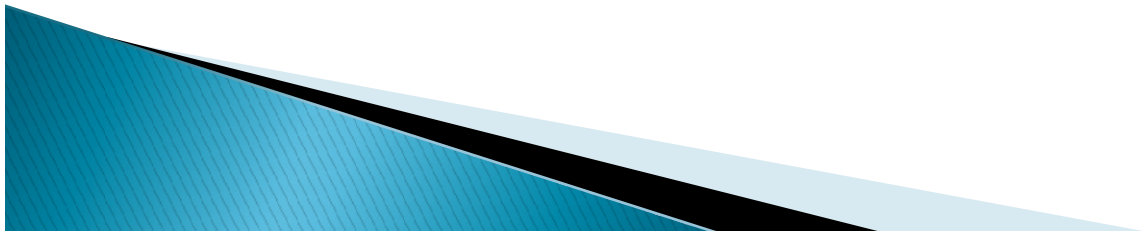
- Main関数内をこのようにしてください

```
Engine.Initialize("STG", 480, 640, new EngineOption());

//追加
TextureObject2D player = new TextureObject2D();

player.Texture = Engine.Graphics.CreateTexture2D("Resource/icon_business_man01.png");

Engine.AddObject2D(player);
//ここまで

 while (Engine.DoEvents())
 {
    Engine.Update();
 }

   Engine.Terminate();
```

画像表示

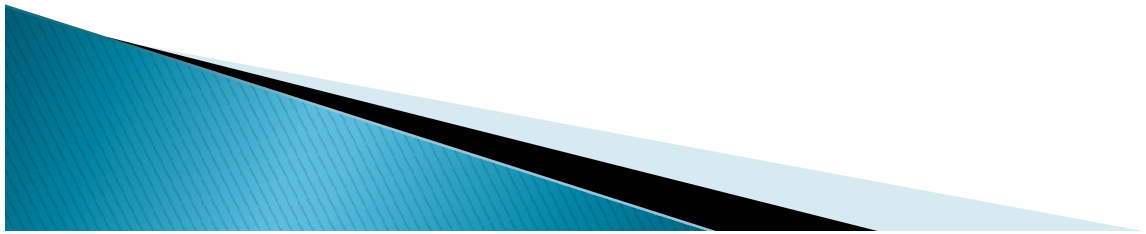
ちょっと解説

- TextureObject2D
  - Altseedで用意されている画像表示用クラス
  - playerはこのクラスのインスタンス
- player.Texture
  - TextureObject2Dのプロパティ
    - メンバ変数？
- Engine.AddObject2D(player)
  - 実際にplayerを表示している



画像表示

- できた
- けどでかい
- あと場所の調整もしたい

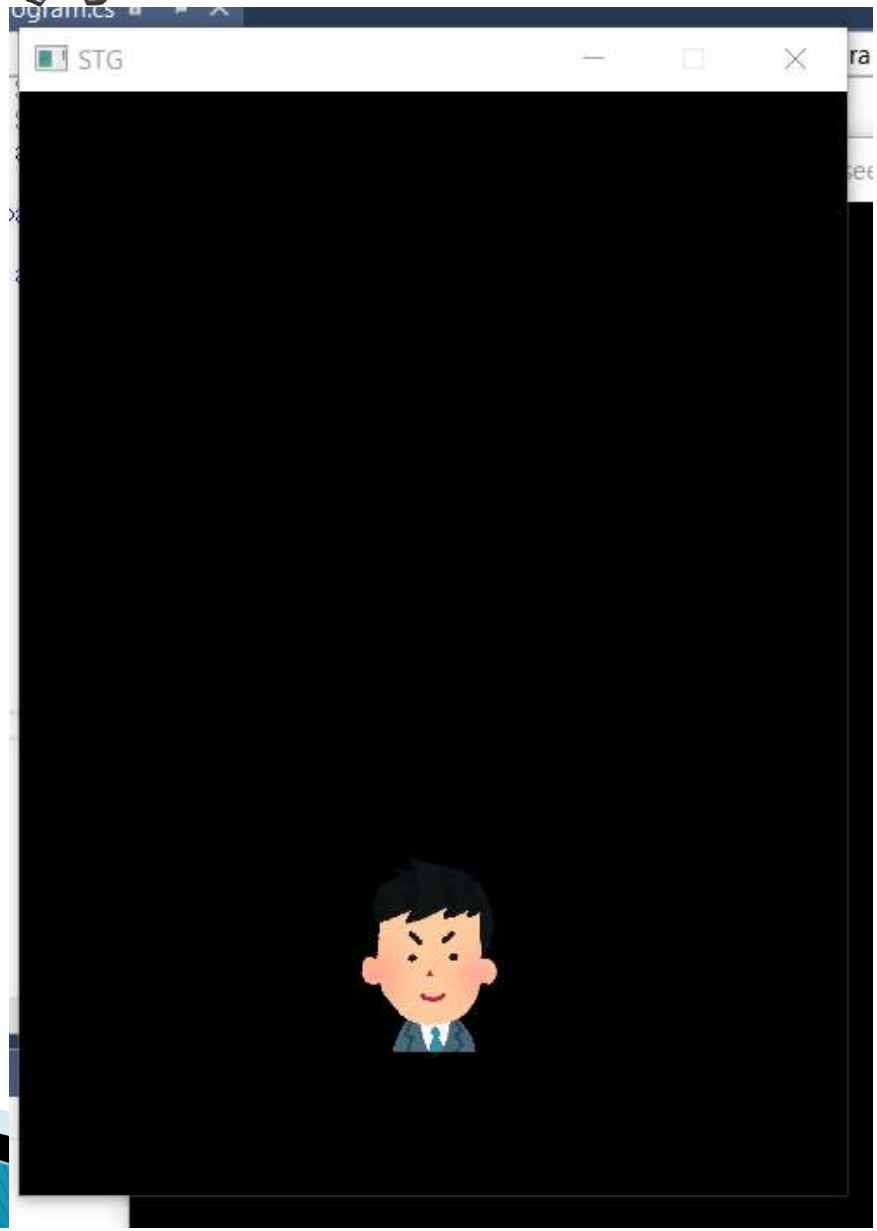


画像表示

- このへんを追加

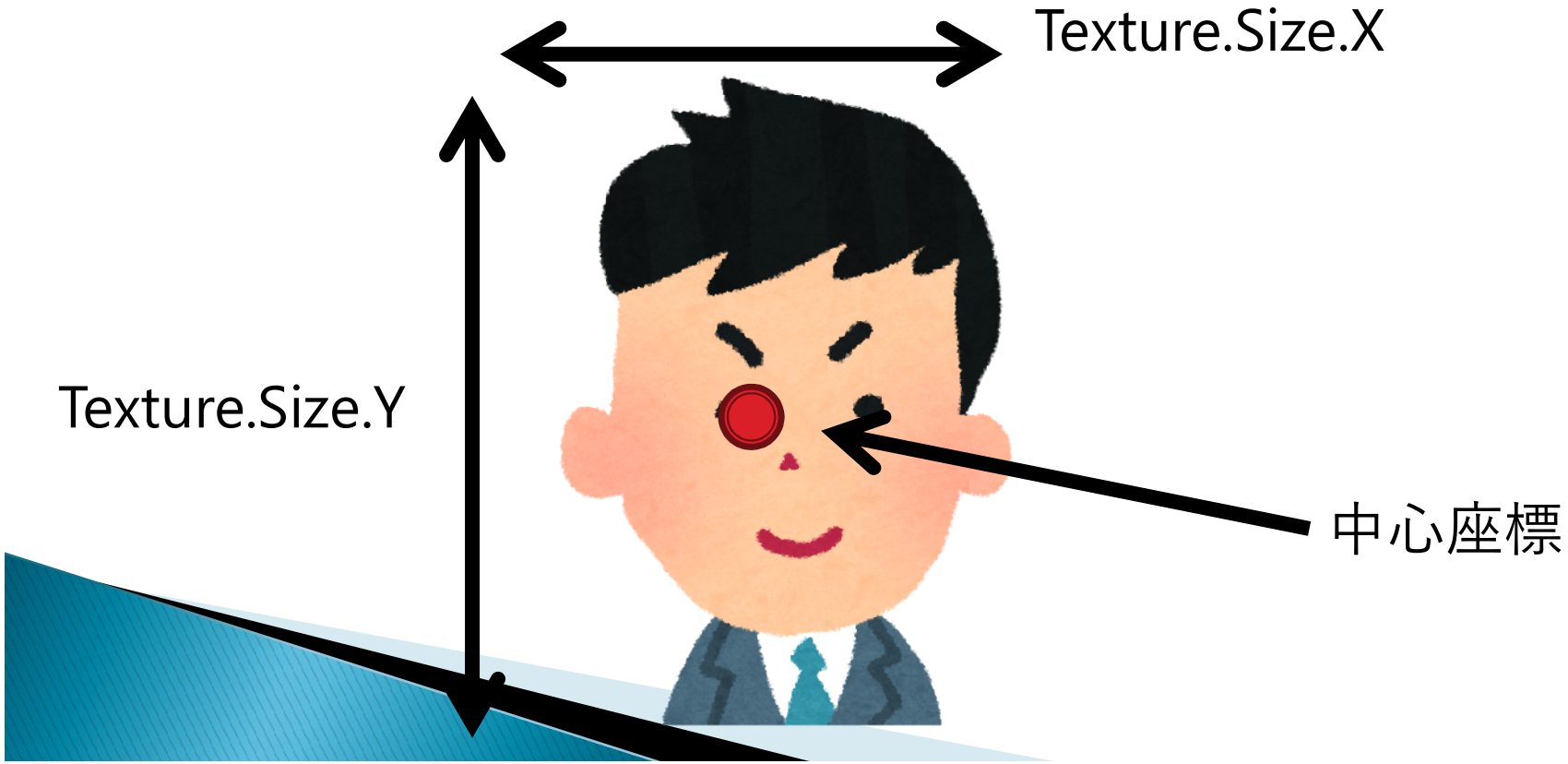
```
player.Texture = Engine.Graphics.CreateTexture2D("Resource/icon_business_man01.png");

Engine.AddObject2D(player);

//追加
player.CenterPosition = new Vector2DF(player.Texture.Size.X / 2.0f, player.Texture.Size.Y / 2.0f);

player.Position = new Vector2DF(240, 500);

player.Scale = new Vector2DF(0.2f, 0.2f) ;
//ここまで

while (Engine.DoEvents())
{
    Engine.Update();
}
```

画像を表示



ちょっと解説

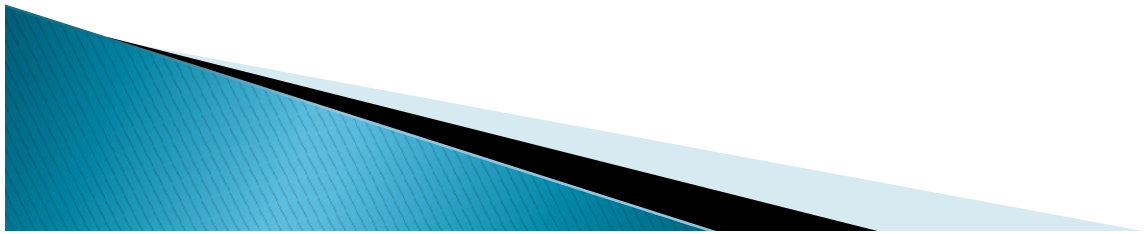
- player.CenterPosition
  - プレイヤーの中心座標
  - 画像の表示や拡大縮小などがここを中心に行われる



ちょっと解説

- Position
  - プレイヤーの座標

- Scale
  - 画像の拡大率
  - 1で100%
    - float型なので、小数なら0.2fなどと書く



キーの取得

```
while (Engine.DoEvents())
{
    float speed = 1;

    if (Engine.Keyboard.GetKeyState(Keys.Up) == KeyState.Hold)
    {
        player.Position = player.Position + new Vector2DF(0, -speed);
    }

    if (Engine.Keyboard.GetKeyState(Keys.Down) == KeyState.Hold)
    {
        player.Position = player.Position + new Vector2DF(0, speed);
    }

    if (Engine.Keyboard.GetKeyState(Keys.Left) == KeyState.Hold)
    {
        player.Position = player.Position + new Vector2DF(-speed, 0);
    }

    if (Engine.Keyboard.GetKeyState(Keys.Right) == KeyState.Hold)
    {
        player.Position = player.Position + new Vector2DF(speed, 0);
    }

    Engine.Update();
}
```

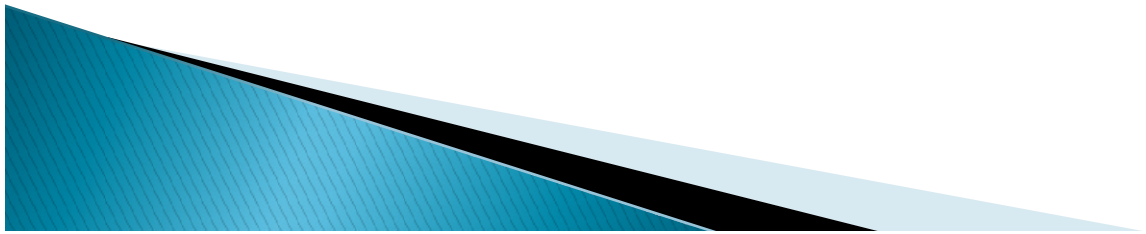
キーの取得

- プレイヤーが動く！！！！！！！！！！！！

解説

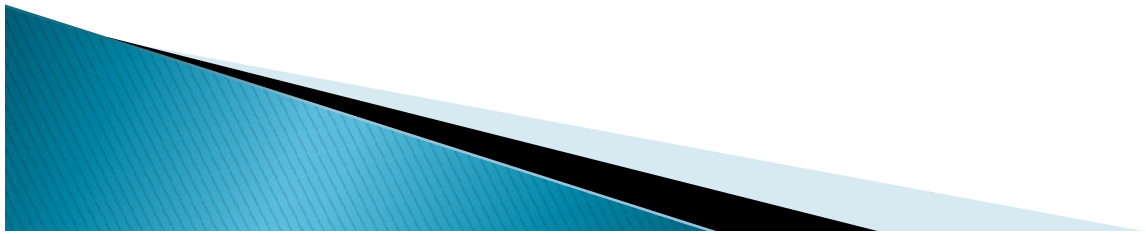
- while (Engine.DoEvents()){}
  - この中にプログラムを書くと、毎フレーム実行してくれる

- Engine.Keyboard.GetKeyState(Keys.Up)
  - キーの状態を取得
  - この場合、上キーの状態を取得している



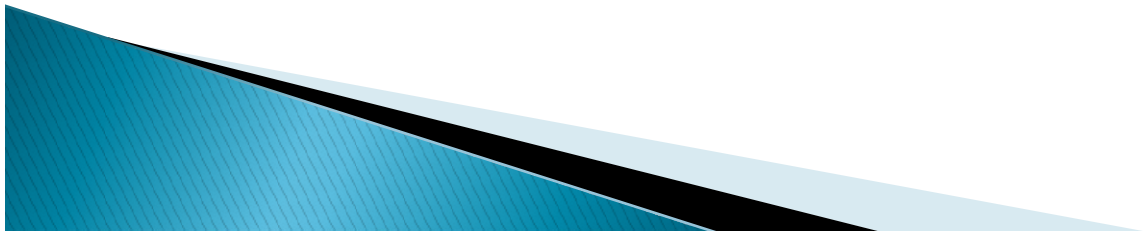
解説

- KeyState.Free　キーが押されていない
- KeyState.Hold　キーが押されている
- KeyState.Push　キーが押された
- KeyState.Release　キーが離された



弾を撃つ

- 時間があればやります



お疲れさまでした

- 次回は5/20(土)
- 新歓コンパもあるよ