* インスタンス化

クラスで書いた空想を、物として創造すること。

実際に使うときは　○○○ = new ×××;　や

GameObject 〇 = Instantiate(×);　などで使ったりする。

* カプセル化

機能の制限をすることによって、利用者が不用意に内部機能にアクセスしないようにすることを防ぐための機能。

例として、弄られたくない変数はpublicにするのではなく、privateでget,setすることで不用意な変更を防ぐことができる。

* ゲームをビルド

ゲームをコンパイルして.exeファイルを作ること。

ゲープロのゲームをビルドしたこと無かったので記入しました。

* Var

変数の型が決まっているときに使う

コンパイル時に、varでint型とchar型など、別の変数型を宣言しているとコンパイルエラーになる。var型の基本的な使い方としてはfor文などで処理を回すとき使うと便利になる

* Ray

カメラか、もしくは指定した場所から光線をとばして、光線に当たったオブジェクトの情報を取得する機能

例として、VRゲームで可視化したRayを飛ばし、当たったオブジェクトを破壊したりできる。

* Mathf

数学の計算関数がまとめられているクラス(Unityなどのゲームエンジン専用)

Mathfは日本語で言うと数学

Floor,Ceil,Round,は左から切り捨て、切り上げ、四捨五入となっており、これを使うとゲーム中のスコアの計算などが便利になる。

他には、累乗や平方根、三角関数など様々な計算関数があり、1から計算して答えをはじき出さなくてもできるので、自分で計算しない分、時間の短縮になる

* ストリング(toストリング)

文字列を扱うためのクラスが入っている。

例として、Lengthを使うと、文字数を取得ことができることやReplaceを使って

文字を置き換えることもできる。

(ToStringは数値を文字列に変換でき、前にplayerが入力した数値を記録し、その入力した数値を文字列に変換して画面に出力することもできる。)

* メソッド

オブジェクトの操作の事

キューブのオブジェクトがあったとして、その動きを定義して動かすことがメソッド

キューブのオブジェクトに転がるを定義した。←これもメソッド

* Findメソッド

C#ではListから条件に合った要素を取り出すもの

ゲームでは、分岐ルートを作るのに使ったりすることもある