時刻の扱い方

目次

１．Unity上における時刻の扱い方について
２．代表的なメソッド
３．アプリを落としても時間をカウントする

１．Unity上における時刻の扱い方について

ゲームを作成するにあたり、現在の時刻を取得して判定に使用するケースがいくつかある。

・スタミナゲー

・1回広告を見たら3日間広告を出さない

・リワード(報酬型)の時間を開ける場合　　などなど

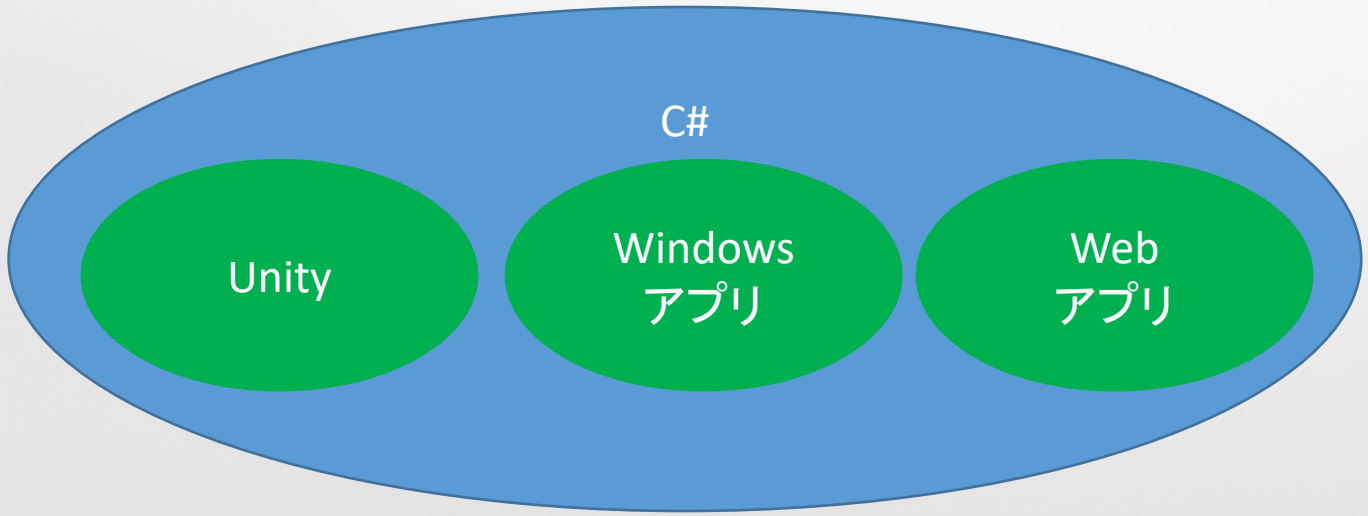
この場合、もちろんゲームを落としてもその時間はカウントする必要がある。
色々やり方はあるが、今回はその中でスタンドアローンで出来る方法について紹介する。

１．Unity上における時刻の扱い方について

ただし、Unityのリファレンスを見ても時刻に関するクラスやメソッドは存在しない。
その場合、どうするか？

→Unityのリファレンスを見てもない場合、C#のリファレンスを確認する！

イメージとしては以下の感じ。UnityはC#の中に包括される形なので、
C#の機能を確認すること。



１．Unity上における時刻の扱い方について

C#の機能として用意されているDateTime型を使用して取得する。

[Microsoft リファレンス]

https://learn.microsoft.com/ja-jp/dotnet/api/system.datetime?view=net-7.0

Unityを使用する上で時刻の取得の他、ファイルの読み書きなどC#の機能を使用する場合も多々あるので、C#としてどのような機能があるのか知っておくと良い。

また、C#標準機能を使用する場合、Systemをusingする場合が多いので忘れないように気をつけてください！

(Cでいうstdio.hをインクルードするみたいな感じ)

２．代表的なメソッド

DateTime型の代表的なメソッドやメンバ変数を紹介する。

メソッド名	内容
Now	現在の時刻をDateTimeクラスに戻り値として返す Now.Years：現在の年のみ返す Now. Month：現在の月のみ返す Now.Day：現在の日のみ返す
AddYears(n)	設定されている時刻にn年、追加する
AddMonths(n)	設定されている時刻にn月、追加する
AddDays(n)	設定されている時刻にn日、追加する
AddHours(n)	設定されている時刻にn時間、追加する
AddMinutes(n)	設定されている時刻にn分、追加する
AddSeconds(n)	設定されている時刻にn秒、追加する
ToString(“表示したい形式”)	形式を指定し、string型に変換する 例：("yyyy/MM/dd HH:mm:ss")

２．代表的なメソッド

使用例：実際にどのような出力となるか確認してみましょう！

内容	記述
今の時間	DateTime dt = DateTime.Now; Debug.Log(dt);
今月の1日を指定	DateTime dt2 = new DateTime(DateTime.Now.Year,DateTime.Now.Month,1); Debug.Log(dt2);
今の時間に1日加算する	DateTime dt3 = DateTime.Now; dt3 = dt3.AddDays(1); Debug.Log(dt3);
時:分:秒 年/月/日 の順で表示する	DateTime dt4 = DateTime.Now; Debug.Log(dt4.ToString("HH:mm:ss yyyy/MM/dd"))

また、2つの時間を比較する場合は単純にIf文で比較することもできます！

(より時間が経過している方が大きい扱い)

２．代表的なメソッド

Practice1.csを作成し、以下の問題をやってみてください！

１：今の日時をDebug.Logに「2023/1/1 00:00:00」の形式で表示させる

２：今の時刻から、以下の日時を足したものを表示させる

(1)2年後、(2)2ケ月後、(3)2日前、
(4)2時間後、(5)2分後、(6)2秒後

３：今月の最終日を表示させる(月が変わっても対応できるように)

４：2030年4月の営業日数(土曜と日曜を除いた曜日)を求めてください

５：今日から過去4日間に、土曜日が存在するか判定してください

６：今日から次の自分の誕生日までの日数を表示してください

３．アプリを落としても時間をカウントする

DateTimeとPlayerPrefsを使用することで、日時の比較を実装することが出来ます、PlayerPrefsについて軽くおさらいしますね！

＜string型を保存する＞

PlayerPrefs.SetString(“KEY”,保存する値);

＜string型を読み込む＞

string str = PlayerPrefs.GetString(“KEY”,初期値);

(初期値：KEYが存在しない場合に取得できる値)

＜練習問題＞

intで要素数が3つの配列を用意(値は任意)、ボタンを2つ配置してください。

1つ目のボタンを押した際に、上記の配列を保存してください。

2つ目のボタンを押した際に、Debug.Logで読みだしてください。

３．アプリを落としても時間をカウントする

＜答え＞

```
public void WriteData()
{
    int[] test = new int[3] { 1, 2, 3 };

    PlayerPrefs.SetInt("HP_0",test[0]);
    PlayerPrefs.SetInt("HP_1",test[1]);
    PlayerPrefs.SetInt("HP_2",test[2]);
}
```

```
public void ReadData()
{
    int read;

    read = PlayerPrefs.GetInt("HP_0",0);
    print(read);
    read = PlayerPrefs.GetInt("HP_1",0);
    print(read);
    read = PlayerPrefs.GetInt("HP_2",0);
    print(read);
}
```

３．アプリを落としても時間をカウントする

思い出したところで、では実際に時間を保存してみましょう！
時刻は直接PlayerPrefsで保存出来ないので、バイナリに変換しさらに
文字列に変換して保存することで安心です！

保存：

例：DateTime dt = DateTime.Now;
string str = dt.ToBinary().ToString();
PlayerPrefs.SetString(“KEY”,str);

バイナリ形式→コンピュータが読み取れる形式の数値のこと。
(対比としてテキスト形式がある)

３．アプリを落としても時間をカウントする

読み出し：

例：string str = PlayerPrefs.GetString(“KEY”,”none”);

long temp = Convert.ToInt64(str);

DateTime dt = DateTime.FromBinary(temp);

３．アプリを落としても時間をカウントする

では、まずは「あるアクション」を実施してから1分間経過したかどうかを判定してみましょう！今回、「あるアクション」は特定のボタンを押した時、にします！

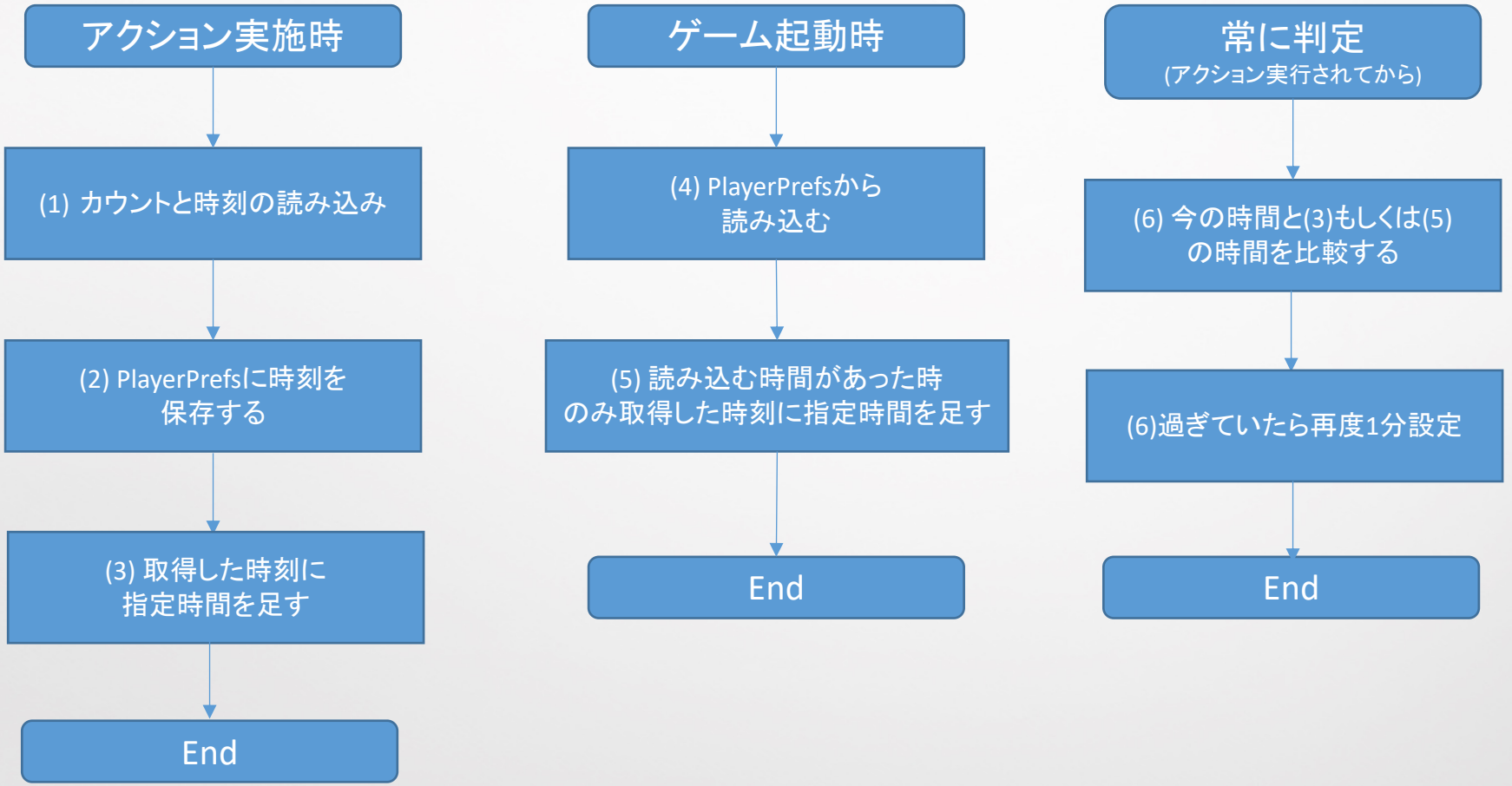
＜問題＞

ボタンを押す →ボタンを押してから1分後にDebug.Logへ出力する
ボタンを押してからアプリ終了し、アプリ再起動後でもボタンを押してから1分経過しているか判定すること。

Practice2.csを作成して、実施してみてください。
よく分からない人は次ページにヒントの載せているので見ながら考えて見てください。

３．アプリを落としても時間をカウントする

＜あるアクション実施から指定時間経過したかチェックする＞



３．アプリを落としても時間をカウントする

次に放置ゲームにありそうな設定を実装してみましょう、30秒ごとにカウントを1プラスしていきます！

自力でやっても良いですし、見ても構わないので実際にやってみてください。

＜仕様＞

- 30秒ごとにカウントを1プラスする
- カウントを1プラスカウントするタイミングで時刻を保存する
- アプリ終了→再開時、時刻を読み出し30秒ごとにカウントを1プラスする
  (30秒以内のカウントは切り捨てとする)
- もちろん、カウントも保存する

３．アプリを落としても時間をカウントする

大まかな流れは以下の通りです。