

整合性確認

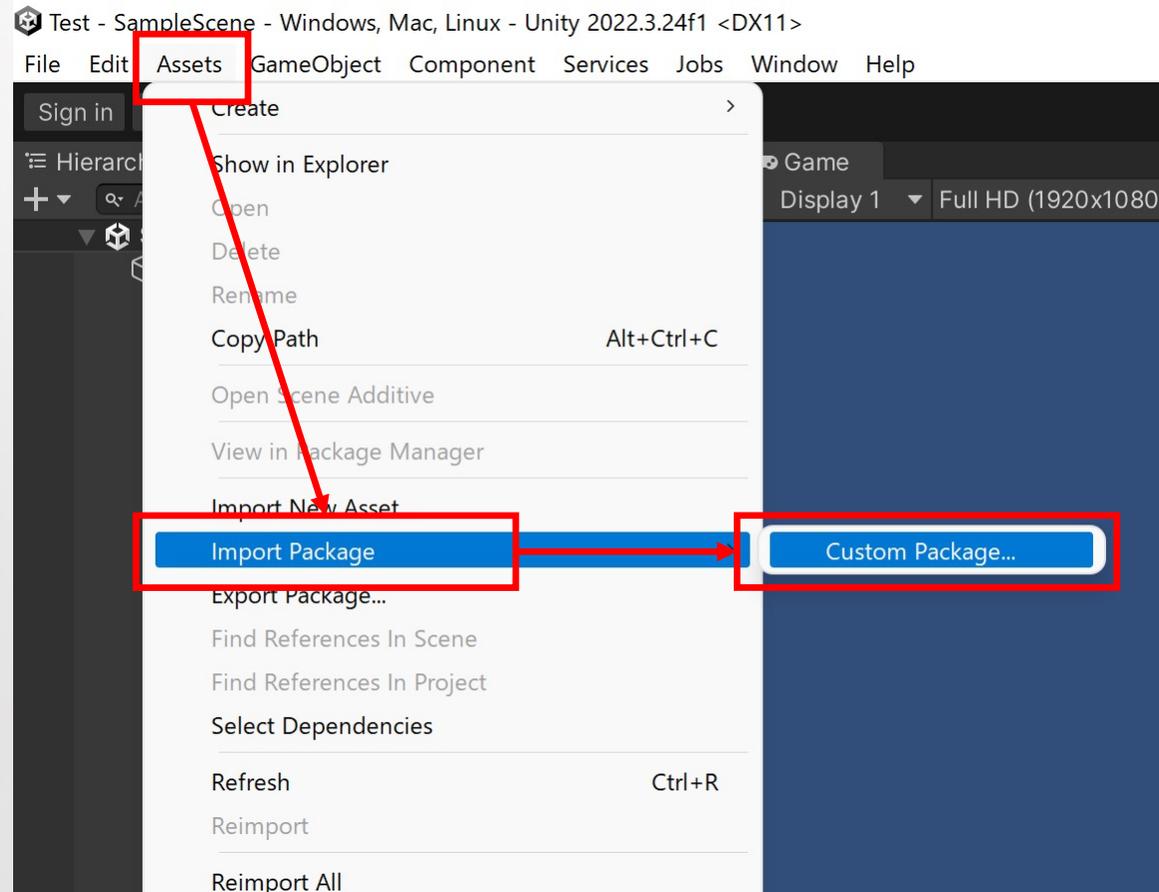


目次

1. 整合性確認とは
2. 方法① Keywordの読み出し
3. 方法② 二重チェック
4. 方法③ チェックサム方式
5. 練習問題

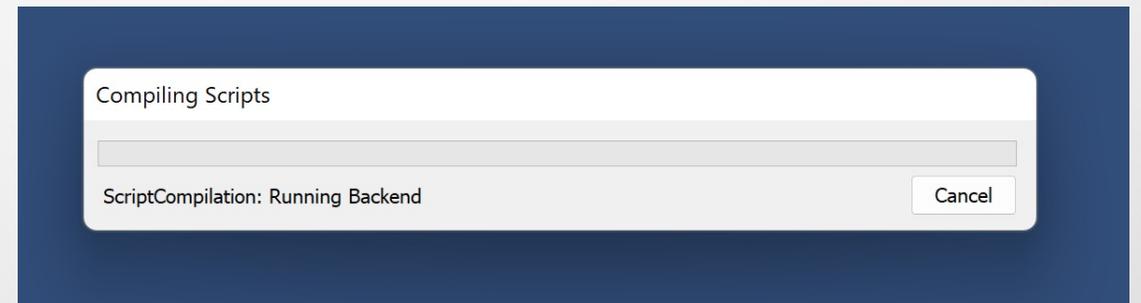
0. UnityPackageを使用しての準備

次に、unitypackageというパッケージファイル(データが詰め込まれたファイル)を読み込みます。



0. UnityPackageを使用しての準備

ファイルを指定し、以下の画面が出たらImportを押すと、ファイルが展開されます。



1. 整合性確認とは

PlayerPrefsの読み出しやサーバーなど、自身のプログラムで任意で決まらない値を読み込んだり、外部からデータを受け取る場合、そのデータが正しいデータかチェックする必要がある。

(プログラムは自身のデータ以外のものは、基本的に信用しない)

チェックの方法としては様々であるが、ここでは3つの方法について紹介、実装する。

方法① Keywordの読み出し

方法② 二重チェック

方法③ チェックサム方式(発展系としてCRC方式)

2. 方法① Keywordの読み出し

以下のように、データとデータの間で固定のデータ値を挟み込む方法。

整合性の確認方法としてはかなり緩い。。

ただ、整合性確認自体の処理負荷は高く無いため、たまーに使われる。

(整合性確認自体はゲームの見た目などに影響するものではないので、出来るだけ最小限にする必要がある)

Keywordが異なっていた場合の

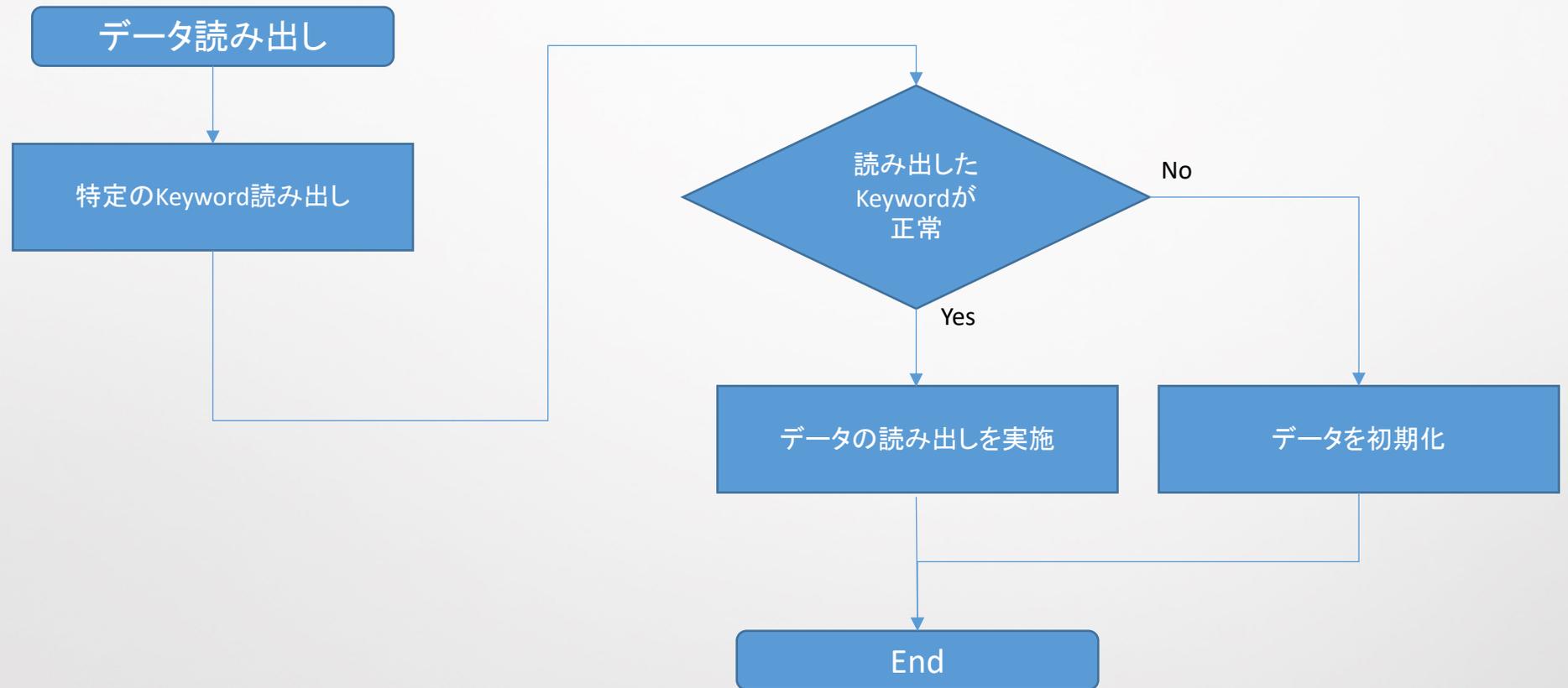
仕様は様々。

全て初期化する or 特定の部分

| | |
|---------|-----------|
| HP | 100 |
| MP | 10 |
| Attack | 20 |
| Defence | 15 |
| Keyword | 123456789 |

2. 方法① Keywordの読み出し

<読み出し時の処理内容>



3. 方法② 二重チェック

以下のように、同じデータを2つ持ち2つのデータが同じ値となっているか確認をする。異なっていた場合はデータが破損したとみなし、初期値を取るようにロジックを組む。

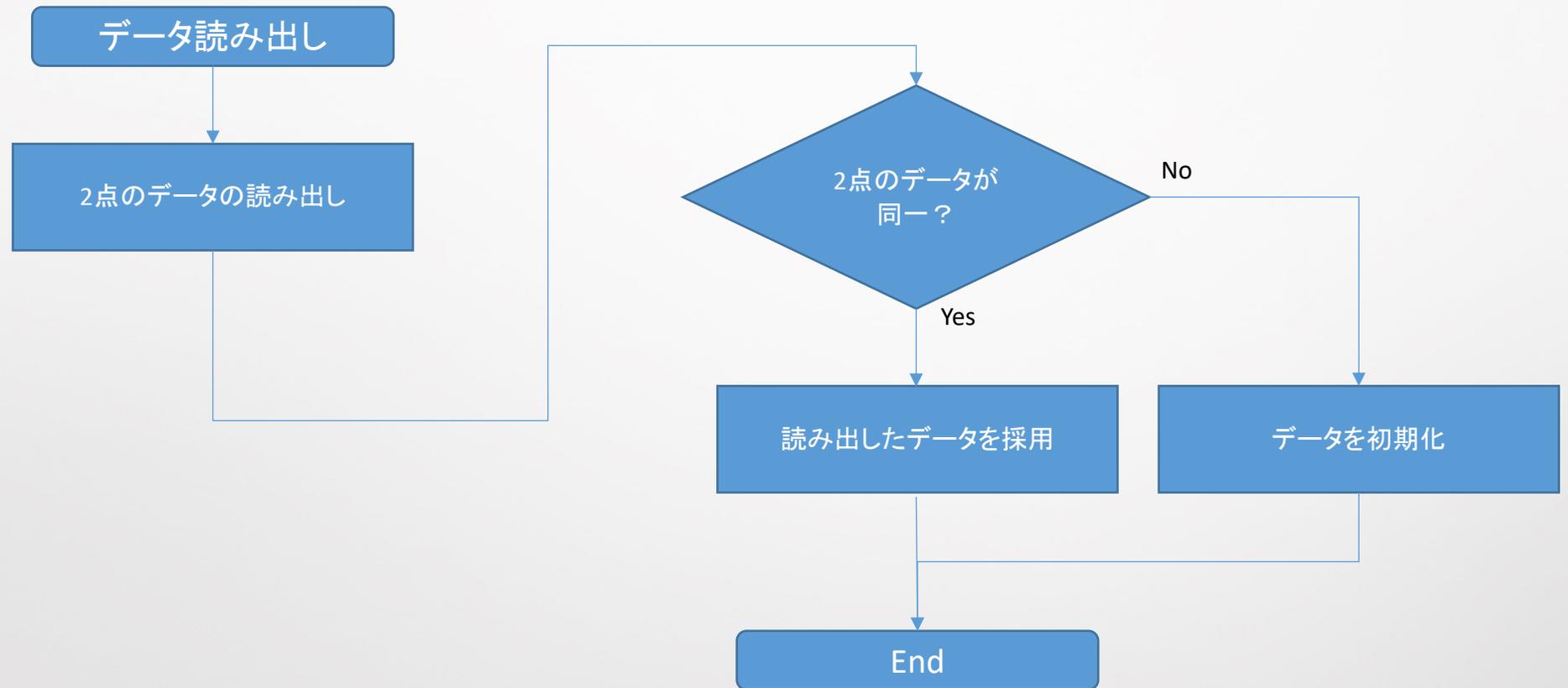
今回は二重だが、三重チェックとかもまれにある。

AブロックとBブロックがあり、それぞれの項目が同一であることをチェックする。

| | |
|-----------|-----|
| HP-A | 100 |
| MP-A | 10 |
| Attack-A | 20 |
| Defence-A | 15 |
| HP-B | 100 |
| MP-B | 10 |
| Attack-B | 20 |
| Defence-B | 15 |

3. 方法② 二重チェック

<読み出し時の処理内容>



4. 方法③ チェックサム方式

一般的によく実装される方式。保存するデータをある一定の計算式に当てはめ、データが正しいことを証明する方式。

サム方式は簡単過ぎてバレルので、CRC方式がよく使われる。

ただ、CRCはちょっとロジックが難しい為、気になった人は調べてみてください。

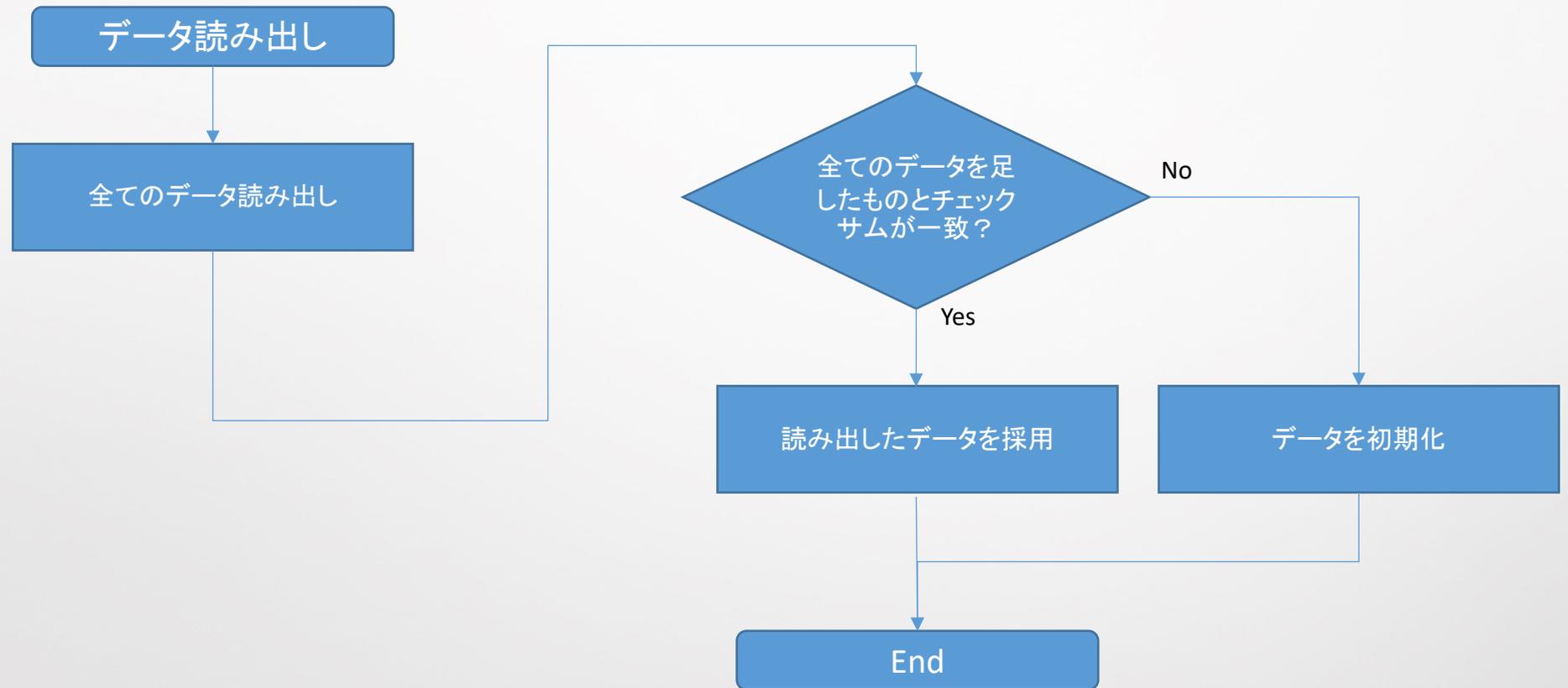
サム方式：全ての項目を足したものをデータとして持つ。

$$\text{HP}(100) + \text{MP}(10) + \text{Attack}(20) + \text{Defence}(15) = \underline{145}$$

| | |
|----------|-----|
| HP | 100 |
| MP | 10 |
| Attack | 20 |
| Defence | 15 |
| Checksum | 145 |

4. 方法③ チェックサム方式

＜読み出し時の処理内容＞



5. 練習問題

実際に、読み出し処理を作成してみよう！

Keywordが間違っていたら、全てのデータを1に設定し、一致していたらデータの読み出しを行う。

書き込み1ボタン：

Keyword含めてデータを書き込む。

デバッグ1ボタン：

Keywordの値を壊すボタン。

読み込み1ボタン：

データの整合性を行うボタン。

| | |
|---------|-----------|
| HP | 100 |
| MP | 10 |
| Attack | 20 |
| Defence | 15 |
| Keyword | 123456789 |

5. 練習問題

実際に、読み出し処理を作成してみよう！
二重チェックをして異なっていたら、全てのデータを1に設定し、一致していたらデータの読み出しを行う。

書き込み2ボタン：

二重チェックの書き込み

デバッグ2ボタン：

二重チェックの片方の値を壊すボタン。

読み込み2ボタン：

データの整合性を行うボタン。

| | |
|-----------|-----|
| HP-A | 100 |
| MP-A | 10 |
| Attack-A | 20 |
| Defence-A | 15 |
| HP-B | 100 |
| MP-B | 10 |
| Attack-B | 20 |
| Defence-B | 15 |

5. 練習問題

実際に、読み出し処理を作成してみよう！

サム値が間違っていたら、全てのデータを1に設定し、一致していたらデータの読み出しを行う。

書き込み3ボタン：

チェックサム値を含めて書き込む

デバッグ3ボタン：

サム値を壊すボタン。

読み込み3ボタン：

データの整合性を行うボタン。

| | |
|----------|-----|
| HP | 100 |
| MP | 10 |
| Attack | 20 |
| Defence | 15 |
| Checksum | 145 |