

C++の授業

C++講義計画（C言語からの移行・ゲームオブジェクト構築まで）

☒ 目標

C言語を半年学んだ学生が、C++の基礎文法からオブジェクト指向（継承・ポリモーフィズム）を学び、最終的に `GameObject` クラスを継承してプレイヤーや敵を実装できるようにする。

☒ 全体構成（全15回想定）

回	テーマ	技術目標	演習例
1	C++導入・Cとの違い	<code>iostream</code> / 名前空間 / <code>new</code> / <code>delete</code>	Cのprintf版→C++出力へ書き換え
2	関数と参照・オーバーロード	参照渡し・デフォルト引数・関数多重定義	距離計算関数のオーバーロード
3	クラス・コンストラクタ入門	<code>class</code> / <code>public</code> / <code>private</code> / <code>this</code>	Vector2構造体→Vector2クラス化
4	クラスの設計とヘッダ分離	<code>.h</code> / <code>.cpp</code> 分離と include 管理	Vector2, Rect クラス実装
5	動的メモリとコンテナ基礎	<code>new/delete</code> , <code>std::vector</code>	<code>std::vector</code> で敵リストを作る
6	コンストラクタ/デストラクタ/コピー	RAIIの考え方	ファイル読み込み管理クラス
7	継承入門	<code>Base</code> / <code>Derived</code> クラス / <code>protected</code>	<code>Enemy</code> ← <code>Slime</code> , <code>Bat</code> など作る
8	仮想関数とポリモーフィズム	<code>virtual</code> , <code>override</code> , <code>GameObject*</code> 管理	<code>GameObject</code> 基底で Draw/Updateを仮想化
9	純粋仮想関数・抽象クラス	<code>=0</code> の概念、基底クラス設計	<code>GameObject</code> を抽象化して管理リストに追加
10	スマートポインタ・メモリ安全化	<code>std::unique_ptr</code> , <code>std::shared_ptr</code>	<code>std::vector<unique_ptr<GameObject>></code>
11	継承+多態性の活用	<code>virtual</code> 呼び出しと <code>override</code> 確認	<code>Enemy</code> / <code>Player</code> / <code>Item</code> を一括更新
12	クラス間関係・静的メンバ	<code>static</code> , <code>const</code> , クラス内定数	<code>ObjectManager</code> 実装
13	名前空間・クラス分割整理	<code>namespace</code> , include guard	<code>game/</code> , <code>engine/</code> に分離
14	総合演習：ミニゲーム制作	ゲームループ構築	コンソール or DxLibで動作確認
15	発表・振り返り	設計比較・自己評価	ソース+UML提出・口頭説明

☒ ステップ詳細

STEP 1~4：C++基礎をCとの違いで理解

- Cの書き方との比較で、スムーズに導入。
- `printf`→`std::cout`、`struct`→`class`、`malloc`→`new` など差分理解。

STEP 5~6：クラスとメモリ管理

- C構造体+関数をクラスに統合。
- `vector` と RAII の基本を体験。

STEP 7~9：継承とポリモーフィズム

- `GameObject` に `virtual void Update()` を持たせて多態の体験。
- `Player` , `Enemy` , `Item` の派生クラスを一括更新。

```
for (auto obj : objects) obj->Update();
```

STEP 10~11：メモリ安全化

- `unique_ptr` 導入でC言語的delete漏れを防止。
- `ObjectManager` による一元管理へ。

STEP 12~15：まとめとゲーム制作

- 名前空間整理とクラス分離で設計の見通しをよくする。
- 最終的にプレイヤー・敵・アイテムなどを `GameObject` 継承で統合。

☒ 最終課題例

タイプ	内容
コンソールRPG	<code>Character</code> 継承で <code>Player</code> / <code>Enemy</code> を共通管理
2Dアクション (DxLib)	<code>GameObject</code> 継承で <code>Ball</code> / <code>Block</code> / <code>Paddle</code> 管理
スネーク風ゲーム	<code>GameObject</code> 多態性で <code>SnakeHead</code> / <code>Fruit</code> 管理

☒ 授業設計のポイント

- 毎回「CからC++へどう変わるか」を比較して示す。
- すぐ動かせるコード例を提示。
- 最終回で「OOPでコードがどう整理されたか」を可視化する。

☒ まとめ

“15回で「C構文→クラス→継承→仮想関数→安全なゲーム構造」まで自然に到達できる構成。最終的には、ポリモーフィズムを使って `GameObject` を扱う“C++らしいゲーム設計”を完成させる。

🕒Revision #1

★Created 8 October 2025 01:53:26 by youe2

🔧Updated 9 June 2026 00:17:51 by youe2