

Übung 4 PrgT: Container und EventHandler

Aufbauend auf der letzten Übung, bauen wir weitere Teile des Pong-Spieles.

1. Klonen sie die Musterlösung

Falls Sie in der letzten Übung nicht fertig geworden sind, können sie die Mustervorlage von https://github.com/tarTG/uebungen_prgt_1718.git klonen. Die Lösung befindet sich im Ordner uebung_3.

2. Event Typen

Erstellen Sie einen Header mit dem Namen EventType. Diese Datei enthält eine enum-Klasse, welche alle auftretenden Events auflistet.

Für diese Übung erstellen sie für Paddle 1 und 2 eine Definition jeweils für die Bewegung nach oben und nach unten.

3. Observer Interface

Schreiben Sie eine Klasse IObserver. Diese hat eine „pure virtual“ Funktion catchEvent, welche einen EventType als Übergabeparameter hat.

Leiten Sie die Klassen Paddle und Ball vom Interface ab und ergänzen sie die Funktion dort.

Testen Sie mittels Compilieren, ob alles syntaktisch stimmt.

Setzen sie nun den Richtungsvektor in der Paddle variable auf private und ergänzen sie die catchEvent Funktion so, dass entsprechend dem Event der Richtungsvektor ändert.

4. Observed-Klasse

Um die Events zu bearbeiten, brauchen wir eine Observed Klasse. Erstellen sie diese.

Damit die der Umgang mit den Containern einfacher wird. Kopieren sie diese typedefs:

```
typedef std::vector<IObserver*> ObserverList;  
typedef std::unordered_map<EventType, ObserverList> observersTable;
```

Erstellen sie eine Member-Variable :

- observersTable : Hier speichern wir die EventType → Observer Zuordnung
- Vector<EventType>: Hier speichern wir die aktuellen Events

Nun brauchen wir noch 3 public Funktionen:

addObserver(IObserver* observer, EventType eType) fügt einen Observer hinzu

sendEvent(EventType event) speichert ein Event in der Eventliste

update() ruft alle aktuellen Events auf und löscht am Schluss die Eventliste

5. Einbinden in die main

1. Erstellen sie ein Objekt der Observed Klasse.
2. Fügen sie für alle vier Events die Observer hinzu. Hierfür müssen sie mittels dynamic_cast den Pointer des unique_ptr zu IObservers* casten.
3. In dem Abschnitt, in dem bisher die Tastendrucke abgehandelt wurden, ersetzen Sie die direkten Richtungsänderungen durch einen entsprechenden Eventaufruf.
4. rufen sie vor den draw aufrufen, die observer update Methode auf.