

Dokumentation des TaskManager

Es gibt jetzt einen TaskManager, der dazu dient Anfragen/Aufgaben an andere Controller zu schicken.
zB. um

- Grafische Effekte bei Ereignisse zu erzeugen
- GUI Elemente zu zeigen/verstecken/etc
- Logische Ereignisse zu starten/beenden/etc

Wie funktioniert?

Alle Controller sollten in ihrer update() Methode in etwa folgendes tun:

```
// Process tasks from other controllers:
```

```
    foreach (Task task in TaskManager.Instance.PhysicsTasks)
    {
        if (task is ExampleTask)
        {
            Logger.Instance.log(Sender.Physics, "Ok, hello world!", PriorityLevel.Priority_5);
        }
    }
```

```
// MAKE SURE TO CLEAR:
```

```
TaskManager.Instance.PhysicsTasks.Clear();
```

Wie kann man Tasks hinzufügen?

Einfach die Klasse ExampleTasks in Misc/Tasks kopieren und in MeineKlasseTasks umbenennen (Vererbung beachten!). Warum hat jeder Sender eine eigene Task-Klasse? Weil man so beliebige Informationen weitergeben kann. Im PowerUpManagerTask gebe ich z.B. das betroffene Powerup und ein enum PowerUpState {Finishing, Starting } mit. Der Empfänger dieses Tasks braucht dann nur in der Klasse PowerUpManagerTask nachschauen welche Informationen im zur Verfügung stehen und kann dann den Task implementieren wie er mag.

Wenn man die Klasse erstellt hat und alle nötigen Member hinzugefügt hat, kann man beispielsweise so neue Tasks erstellen:

```
TaskManager.Instance.addTask(new PowerUpManagerTask(Sender.Graphics, "Please visualize that this PowerUp started.", 0, powerUp, PowerUpState.Starting));
```