**CREATE EDITOR SCRIPT FOR TASKMANAGER**

**ENG**

Каждая таска может быть запущена, отменена, перезапущена или выполнена.

**ChapterTask** – это таска-глава, которая хранит в себе одиночные таски (*SingleTask*)

**SingleTask** – конкретная одиночная таска, в котором описывается конкретное задание для выполнения игроком. Например: «сбор монет».

CurrentTask – stores current task which need to manage on.

ShowOrInitializeTaskTrigger – shows if task triggers are initialized or initializes tasks on game scene.

CancelTask – cancels current task

CompleteTask – completes current task

ReloadTask – reloads current task

StartTask – starts current task

TaskBase – parent class of ChapterTaskBase and SingleTaskBase

ChapterTaskBase – parent class of concrete chapter task classes (main logic for chapter task, which contains single classes)

SingleTaskBase – parent class of concrete single task classes (main logic of concrete task)

TaskTriggerBase - parent class of TaskTrigger2D and TaskTrigger3D

TaskTrigger2D – task trigger places on 2D game scene. Offers to start the concrete task when player entered that trigger.

TaskTrigger3D - task trigger places on 3D game scene. Offers to start the concrete task when player entered that trigger.

TaskTitle – stores task titles TaskManager can recognize which task have to start

**TaskManager:**

* **CurrentTask** – хранит в себе текущую запущенную таску
* **StartTask(TaskTitle)** – проинициализировать или создать все нужные объекты; поместить их в список, через который можно будет их контроллировать (например, удалить при отмене таски) и разместить их на своих местах.
* **CancelTask(TaskTitle)** – удалить созданные объекты (те, которые создавались при запуске таски и добавлялись в список созданных объектов таски).
* **ReloadTask(TaskTitle)** – все созданные объекты сбросить к начальному состоянию
* **CompleteTask (TaskTitle)** – завершить таску, выдать награды (если таковы имеются), или показать popup окно, а потом выдать награды; удалить все созданные объекты (те, которые создавались при запуске таски и добавлялись в список созданных объектов таски).

**TaskTitle –** enumeration of task titles

**TaskBase** (main class of task):

* **StartTask ()**
* **CancelTask ()**
* **ReloadTask ()**
* **CompleteTask ()**

**ChapterTask (**derived from TaskBase**)**

**SingleTask (**derived from TaskBase)

**Task structure:**

* **TaskTrigger**
* **TaskPopup**
  + **ChapterTask\_Popup (ChapterTask Name & Description (when trigger entered))**
  + **SingleTask\_Popup (SingleTask name (can show multiple names))**
  + **AdditionTask\_Popup (Addition tasks)**
* **TaskManager:**
  + **TaskBase**
    - **ChapterTask**
    - **SingleTask**
  + **TaskTitle**

1. **Player entered TaskTrigger – PopupManager.Show(ChapterTask\_Popup)**
2. **Player exit TaskTrigger - PopupManager.Hide(ChapterTask\_Popup)**
3. **Player Shift button downed – PopupManager.Hide(ChapterTask\_Popup),**
4. **ChapterTask.Start() - ChapterTask.OnTaskStarted action**
5. **ChapterTask Started – Initialized every SingleTask in the array**
6. **ChapterTask Subscribes on every SingleTask.OnCompleted action. Whole ChapterTask completed when every SingleTask completed.**
7. **First SingleTask.Start()**
8. **SingleTask.IsStarted=true**
9. **Chapter task can start SingleTask what it wants. But checked !SingleTask.IsStarted or !SingleTask.IsCompleted before (Может запускать любую одиночную таску, но перед этим, проверяет не запущенная ли она или не завершенная)**
10. **SingleTask.Complete() – ChapterTask.SingleTaskCompletedCounter++**
11. **ChapterTask.Cancel() - every SingleTask in the array Canceled; ChapterTask.OnTaskCanceled action**
12. **ChapterTask.Reload() - every SingleTask in the array Reloaded if started or completed; ChapterTask.OnTaskReloaded action**
13. **ChapterTask.Complete() - ChapterTask.OnTaskCompleted action (PopupManager.Show(ChapterTaskCompleted\_Popup))**

**List all of taskTriggers:**

* **disable everything when TaskStarted**
* **enable everything when task canceled**
* **delete current task when it completed and enable all TaskTriggers**

**ShowTaskTriggers() – The function which shows all tasks.**

**HideTaskTriggers() – Hides all tasks**

**DeleteTaskTrigger(TaskTriggerTitle) – Deletes task trigger from group after task was complete**

* **TaskTrigger**
* **TaskManager:**
  + **ChapterTask**
* **PopupManager:**
  + **ChapterTaskCompleted\_Popup:**
  + **ChapterTaskStart\_Popup:**
    - **ChapterNumber**
    - **ChapterIcon**
    - **ChapterName**
    - **ChapterDescription**
  + **SingleTask\_Popup:**
    - **TaskIcon**
    - **TaskName**
    - **TaskDescription (could be health bar, counter, some description information**

Tasks are ScriptableObjects. Every task is common logic for specific task. For example: CollectItems. We are creating the script derived from SingleTask which is ScriptableObject script. Next step is making base task logic, collecting items. Next step: to make the ScriptableObject “CollectItems” from AssetMenu (Create-Task\_Manager-SingleTasks-CollectItems) and call it “CollectCoins”. Than make script setting needed for script logic (list of coin prefabs, location coordinates that coins and subscribe OnStart, OnCancel, OnReload, OnComplete actions to a specific methods.

And next tasks creating the same way.

Create new script. Paste line code above class



Input ChapterTaskName on menu name(specific task name which task is using for),

Input fileName on next format: ChapterTask\_Name, where Name is specific task name for example CollectItems. Than fileName will be next ChapterTask\_CollectItems. The same thing on SingleTask

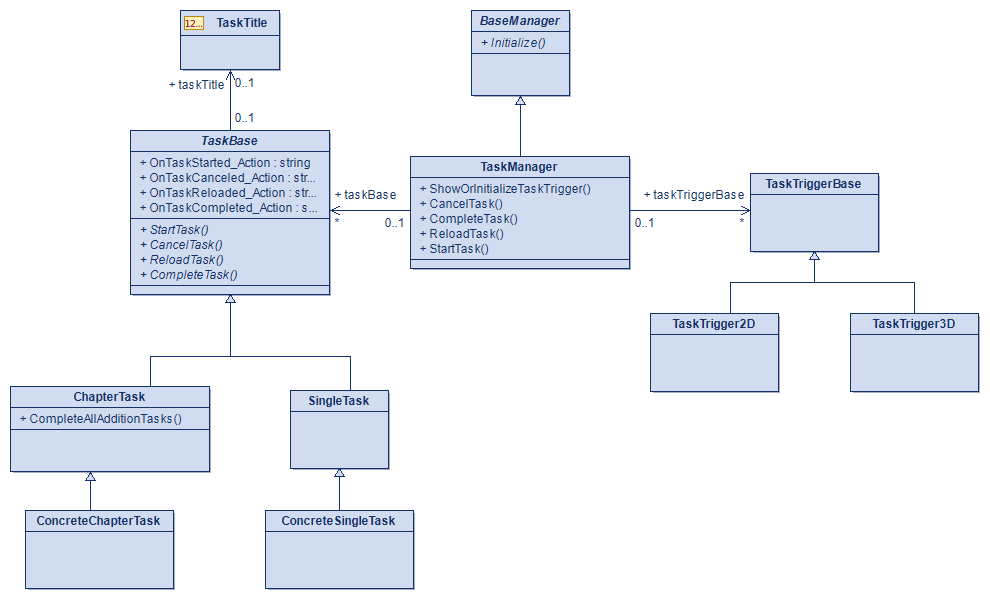


GameActor

GameTriggerActor3D

GameTriggerActor2D

**Class Diagram**

****