DIGITAL GOOD – EVENT MANAGER

Primary macro is **__EVENT_MANAGER__** on server source and **ENABLE_EVENT_MANAGER** on client source. This macro does not affect **AniImageBox SetScale** or **Client-Quest trigger**, those two are independent. Each event has a prepared event quest. You can extend the auto spawn events by adding more cycles and configuring their duration by editing the function.

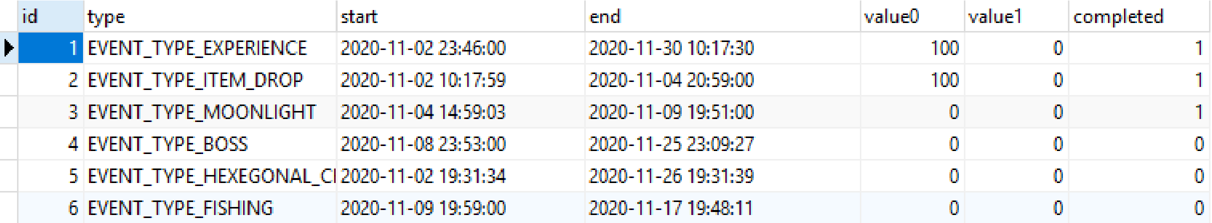
Event Manager is fully active on one core, configured to channel 99. This was done to avoid event overflow (event starting on multiple cores at once). Others are partially active, serving information to players when they request it.

SQL:

You can schedule events in an SQL table.

Structure:

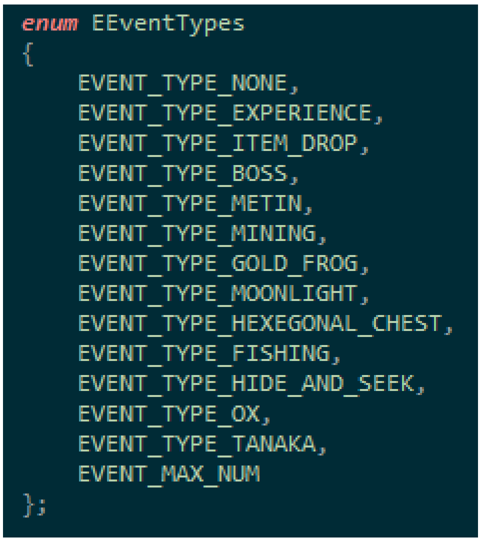
- “id” – holds unique id of the event.
- “type” – type of the event you would like to schedule.
- “start” – start date and time of the event
- “end” – end date and time of the event
- “value0, value1” – these columns have no special meaning unless they are used as variables in your events.
- “completed” – will be marked as true (1) when an event is finished.



id	type	start	end	value0	value1	completed
1	EVENT_TYPE_EXPERIENCE	2020-11-02 23:46:00	2020-11-30 10:17:30	100	0	1
2	EVENT_TYPE_ITEM_DROP	2020-11-02 10:17:59	2020-11-04 20:59:00	100	0	1
3	EVENT_TYPE_MOONLIGHT	2020-11-04 14:59:03	2020-11-09 19:51:00	0	0	1
4	EVENT_TYPE_BOSS	2020-11-08 23:53:00	2020-11-25 23:09:27	0	0	0
5	EVENT_TYPE_HEXEGONAL_CI	2020-11-02 19:31:34	2020-11-26 19:31:39	0	0	0
6	EVENT_TYPE_FISHING	2020-11-09 19:59:00	2020-11-17 19:48:11	0	0	0

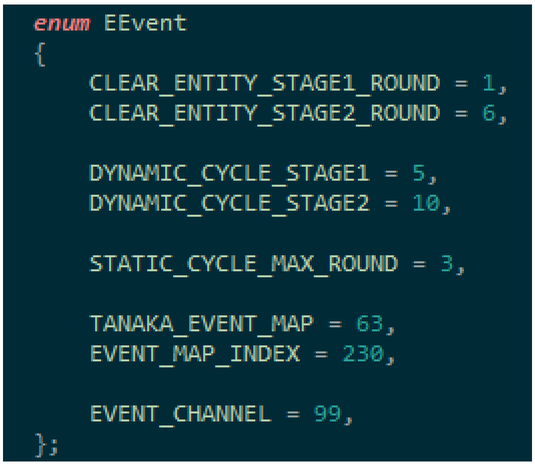
Order of the event types in the enum is not important as they are recognized by string and not numerical order.

Server Source:



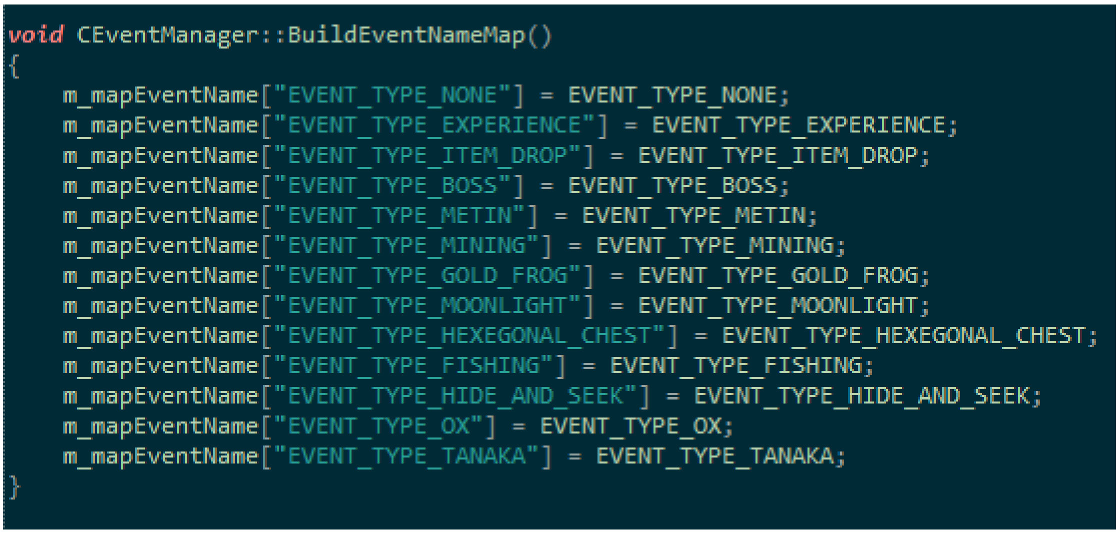
```
enum EEventTypes
{
    EVENT_TYPE_NONE,
    EVENT_TYPE_EXPERIENCE,
    EVENT_TYPE_ITEM_DROP,
    EVENT_TYPE_BOSS,
    EVENT_TYPE_METIN,
    EVENT_TYPE_MINING,
    EVENT_TYPE_GOLD_FROG,
    EVENT_TYPE_MOONLIGHT,
    EVENT_TYPE_HEXEGONAL_CHEST,
    EVENT_TYPE_FISHING,
    EVENT_TYPE_HIDE_AND_SEEK,
    EVENT_TYPE_OX,
    EVENT_TYPE_TANAKA,
    EVENT_MAX_NUM
};
```

Enum that holds numerical value of each event type.



```
enum EEvent
{
    CLEAR_ENTITY_STAGE1_ROUND = 1,
    CLEAR_ENTITY_STAGE2_ROUND = 6,

    DYNAMIC_CYCLE_STAGE1 = 5,
    DYNAMIC_CYCLE_STAGE2 = 10,

    STATIC_CYCLE_MAX_ROUND = 3,

    TANAKA_EVENT_MAP = 63,
    EVENT_MAP_INDEX = 230,

    EVENT_CHANNEL = 99,
};
```

- Clear entity round holds the round at which entities (monsters/metins) will be purged from the map for new monsters/metins to appear.
- Dynamic spawns have two stages (can be extended) at which they execute different regen files (different level ranges).
- Static spawns means it has one specific regen file which is executed multiple rounds.
- Tanaka event map is index of the map where Tanakas are spawned.
- Event map index is index of the event map.
- Event channel is the channel of event manager.



```
void CEventManager::BuildEventNameMap()
{
    m_mapEventName["EVENT_TYPE_NONE"] = EVENT_TYPE_NONE;
    m_mapEventName["EVENT_TYPE_EXPERIENCE"] = EVENT_TYPE_EXPERIENCE;
    m_mapEventName["EVENT_TYPE_ITEM_DROP"] = EVENT_TYPE_ITEM_DROP;
    m_mapEventName["EVENT_TYPE_BOSS"] = EVENT_TYPE_BOSS;
    m_mapEventName["EVENT_TYPE_METIN"] = EVENT_TYPE_METIN;
    m_mapEventName["EVENT_TYPE_MINING"] = EVENT_TYPE_MINING;
    m_mapEventName["EVENT_TYPE_GOLD_FROG"] = EVENT_TYPE_GOLD_FROG;
    m_mapEventName["EVENT_TYPE_MOONLIGHT"] = EVENT_TYPE_MOONLIGHT;
    m_mapEventName["EVENT_TYPE_HEXEGONAL_CHEST"] = EVENT_TYPE_HEXEGONAL_CHEST;
    m_mapEventName["EVENT_TYPE_FISHING"] = EVENT_TYPE_FISHING;
    m_mapEventName["EVENT_TYPE_HIDE_AND_SEEK"] = EVENT_TYPE_HIDE_AND_SEEK;
    m_mapEventName["EVENT_TYPE_OX"] = EVENT_TYPE_OX;
    m_mapEventName["EVENT_TYPE_TANAKA"] = EVENT_TYPE_TANAKA;
}
```

These strings must be identical to SQL enum as those then hold numerical value of the event type.

QUEST

When a new event starts/ends its quest will immediately appear/disappear with the help of new quest triggers **event_begin/event_end**.

Manual event start through quest: **start_event(***eventIndex***)**

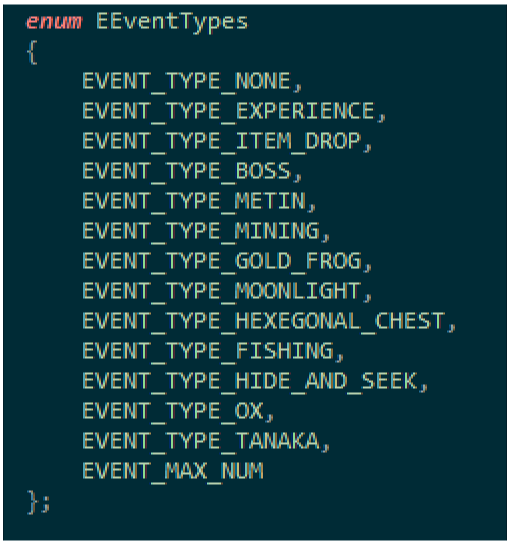
Manual event end through quest: **end_event(***eventIndex***)**

Quest strings are stored in translate = {} table. Adapt the name to your own if needed.

Enums/Functions that must be extended when adding a new event type:

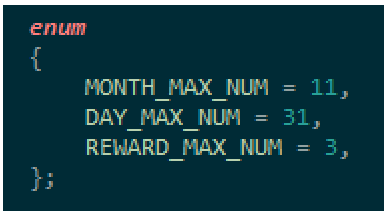
- **SQL Enum,**
- **enum EEventTypes**
- **void CEventManager::BuildEventNameMap(),**
- **bool CEventManager::GetEventState(int iEventType),**
- **void CEventManager::SetEventState(TEventTable* table, bool bState),**
- ***A new event function with start/end concepts.**
- **event_panel.quest and questlib.lua**

CLIENT



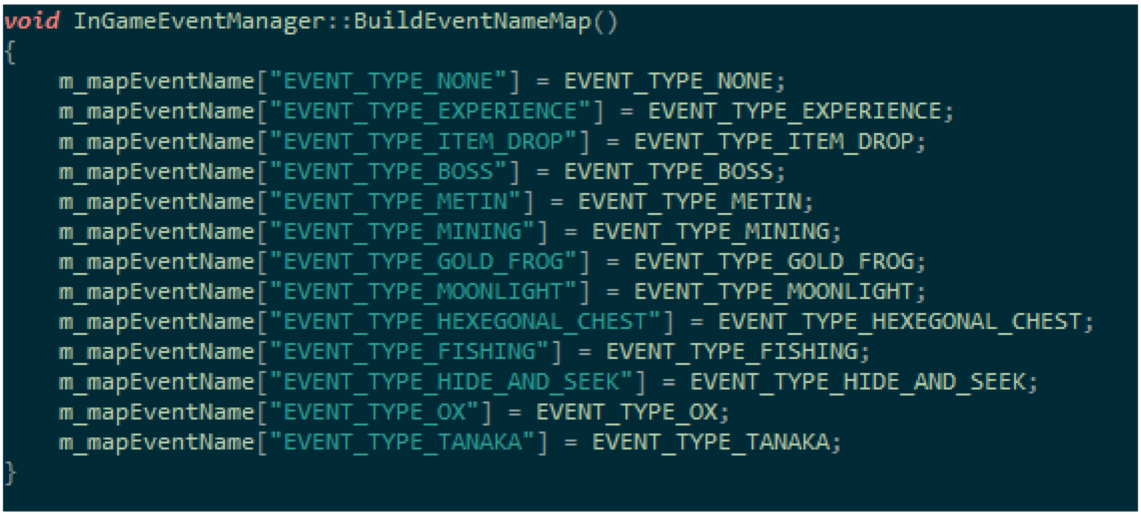
```
enum EEventTypes
{
    EVENT_TYPE_NONE,
    EVENT_TYPE_EXPERIENCE,
    EVENT_TYPE_ITEM_DROP,
    EVENT_TYPE_BOSS,
    EVENT_TYPE_METIN,
    EVENT_TYPE_MINING,
    EVENT_TYPE_GOLD_FROG,
    EVENT_TYPE_MOONLIGHT,
    EVENT_TYPE_HEXEGONAL_CHEST,
    EVENT_TYPE_FISHING,
    EVENT_TYPE_HIDE_AND_SEEK,
    EVENT_TYPE_OX,
    EVENT_TYPE_TANAKA,
    EVENT_MAX_NUM
};
```

- Identical to server source enum.



```
enum
{
    MONTH_MAX_NUM = 11,
    DAY_MAX_NUM = 31,
    REWARD_MAX_NUM = 3,
};
```

- Maximum number of months [0-11]
- Maximum number of day in a month [1-31]
- Maximum number of rewards in reward window [1-3]



```
void InGameEventManager::BuildEventNameMap()
{
    m_mapEventName["EVENT_TYPE_NONE"] = EVENT_TYPE_NONE;
    m_mapEventName["EVENT_TYPE_EXPERIENCE"] = EVENT_TYPE_EXPERIENCE;
    m_mapEventName["EVENT_TYPE_ITEM_DROP"] = EVENT_TYPE_ITEM_DROP;
    m_mapEventName["EVENT_TYPE_BOSS"] = EVENT_TYPE_BOSS;
    m_mapEventName["EVENT_TYPE_METIN"] = EVENT_TYPE_METIN;
    m_mapEventName["EVENT_TYPE_MINING"] = EVENT_TYPE_MINING;
    m_mapEventName["EVENT_TYPE_GOLD_FROG"] = EVENT_TYPE_GOLD_FROG;
    m_mapEventName["EVENT_TYPE_MOONLIGHT"] = EVENT_TYPE_MOONLIGHT;
    m_mapEventName["EVENT_TYPE_HEXEGONAL_CHEST"] = EVENT_TYPE_HEXEGONAL_CHEST;
    m_mapEventName["EVENT_TYPE_FISHING"] = EVENT_TYPE_FISHING;
    m_mapEventName["EVENT_TYPE_HIDE_AND_SEEK"] = EVENT_TYPE_HIDE_AND_SEEK;
    m_mapEventName["EVENT_TYPE_OX"] = EVENT_TYPE_OX;
    m_mapEventName["EVENT_TYPE_TANAKA"] = EVENT_TYPE_TANAKA;
}
```

- Identical to server source event name map.

Enums/Functions that must be extended when adding a new event type:

- **enum EEventTypes**
- **void InGameEventManager::BuildEventNameMap()**
- **eventMgr module constants (PythonGameEventManagerModule.cpp)**
- **EVENT_NAME_DICT (uiEvent.py)**
- **def OpenEventDescription(self):**

EVENTS

- Experience Event
- Item Drop Event
- Boss Hunter Event (Bosses are spawned in Event Map)
- Metin Shower Event (Metins are spawned in Event Map)
- Mining Event (Veins are spawned in Event Map)
- Golden Frog Event (Gold Frogs are spawned in Event Map)
- Moonlight Treasure Event
- Hexagonal Chest Event
- Fishing Event (Player must bring Minnows to the Fisherman to receive the reward)
- Hide & Seek Event (Led by the administration)
- OX Event (Led by the administration)
- Tanaka Event (Pirate Tanakas are spawned through desert map carrying precious items)