カウントダウンタイマークラスを作ってみようｚ

CDTimer.h

```
#pragma once
#include "Engine/GameObject.h"

class CDTimer :
    public GameObject
{
private:
 double CountDownTimer_;  //現在の残り時間
 double TIMER_INIT_TIME_; //初期時間
 bool isTimerRun_;//タイマーが動いているかどうか？
 DWORD oldTime_;
public:
 //コンストラクタ
 //引数：parent  親オブジェクト（SceneManager）
 CDTimer(GameObject *parent);
 CDTimer(GameObject *parent, double _initTime);


 //初期化
 void Initialize() override;

 //更新
 void Update() override;

 //描画
 void Draw() override;

 //開放
 void Release() override;

public:

 bool IsTimeOver();//タイマーは０になりましたか？ＹＥＳ？ＮＯ？
 void ResetTimer();//タイマーを初期時間に戻す
 void StartTimer();//タイマーをスタートします
 void StopTimer();//タイマーをストップします
 void SetInitTime(double cdTime) { TIMER_INIT_TIME_ = cdTime; ResetTimer(); }
 double GetTime() { return(CountDownTimer_); }
};
```

CDTimer.cpp

```
#include "CDTimer.h"


const int DEF_TIMER_TIME{ 5 };

CDTimer::CDTimer(GameObject* parent)
 :TIMER_INIT_TIME_(DEF_TIMER_TIME),
 CountDownTimer_(DEF_TIMER_TIME),
 isTimerRun_(true)
{
 oldTime_ = timeGetTime();
};
```

```
CDTimer::CDTimer(GameObject* parent, double _initTime)
 :TIMER_INIT_TIME_(_initTime),
 CountDownTimer_(_initTime),
 isTimerRun_(true)
{
 oldTime_ = timeGetTime();
};

bool CDTimer::IsTimeOver()
{
 return(CountDownTimer_ <= 0);
}

void CDTimer::ResetTimer()
{
 CountDownTimer_ = TIMER_INIT_TIME_;
 StartTimer();
}

void CDTimer::StartTimer()
{
 isTimerRun_ = true;
}

void CDTimer::StopTimer()
{
 isTimerRun_ = false;
}

void CDTimer::Initialize()
{

}

void CDTimer::Update()
{
 DWORD nowTime = timeGetTime();
 //下staticは要らんかった。。。
 //static int deltaTime = nowTime - oldTime_;
 int deltaTime = nowTime - oldTime_;
 if (isTimerRun_)
  CountDownTimer_ = CountDownTimer_ - (float)deltaTime/1000.0;
 oldTime_ = nowTime;
}

void CDTimer::Draw()
{
}

void CDTimer::Release()
{
}
```

Revision #8
★Created 12 May 2025 01:32:24 by youe2
Updated 8 June 2026 03:23:00 by youe2