

GG1003

Windows编程

胡学萱

胡学萱

实验6-6 Framework的使用

余庆祥

2220631136

BDP

2022/12/16

**内页写作格式**

**一、实验名称：**要用最简练的语言反映实验的内容。

**二、实验日期：**写明做实验的具体年、月、日及组別。

**三、实验目的：**使用简洁的文字或关键字来敘述，是以怎样的目的作此实验的。

**四、实验环境：**实验的操作系统和软件等。

**五、实验的步骤和方法：**这是实验报告极其重要的内容。这部分要写明经过哪几个步骤。

**六、数据记录和计算：**指从实验中测到的数据以及计算结果。

**七、实验结果或结论：**即根据实验过程中所见到的现象和测得的数据，得出结论。

**八、备注或说明**：可写上实验成功或失败的原因，实验后的心得体会、建议等。

|  |  |
| --- | --- |
| **1、实验名称** | 实验6-5 游戏主控类的实现及使用 |
| **2、实验日期** | **2022/12/16** |
| **3、实验目的** | |
| 1.理解整个框架的设计，理解其中各个类的作用和意义，理解框架的扩展。  2.理解CGame的虚基类设计，能自己写出CGame的子类，重写Init函数，从而实现具体的游戏类。能在main中创建游戏类对象，调用其Run函数进入游戏循环，从而使用框架完成简单实验。 | |
| **4、实验环境** | |
| 1、vs2019 | |
| **5、实验的步骤和方法** | |
| 1、参照大课的讲解，自己实现游戏类CMyGame（派生自CGame），重写Init函数完成初始化工作，重写Update函数完成齿轮旋转动作的更新操作；在main中使用CMyGame类，完成大课效果。  大课课件中，Update是根据流逝时间来改变齿轮精灵的旋转角度；  可以更改为每次Update旋转角度增加一个常量，例如0.02，即每帧旋转角度增加0.02弧度。  改变帧频，分别运行以上两种实现效果，解释其区别。  2、在1题基础上继续完成，增加一种新的精灵，它可以沿窗口顺时针巡逻。增加Monster，可以四方向移动，若碰到新的精灵，则新精灵消失。效果如Demo2.exe  分析：  基于CAnimationSprite派生子类CMySprite，增加属性方法，实现沿窗口矩形顺时针方向移动的移动逻辑。  基于CSpriteManager派生新的CMySpriteManager，在其中增加对CMySprite的管理。  在CMyDemo游戏类中，精灵管理类由CMySpriteManager对象来负责，创建多种精灵，实现移动、碰撞等游戏逻辑。  更多练习：  改写作业一代码，使用框架实现简易推箱子。其中线、文本都可以做成精灵；其他功能要求不变。  改写作业二，设计一种新的类CMySprite，它有特殊的移动逻辑，例如按照直线、沿某矩形范围移动等。需要考虑如何扩展框架，从而在游戏种体现这种精灵。 | |
| **6、数据记录和计算** | |
| **（代码编写思路及关键代码）**  第一题关键代码：  关键代码：  Main.cpp:  #pragma comment(lib,"msimg32.lib")  #include "CMyGame.h"  CGame\* pGame;  int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE hPrevInst, LPSTR lpszCmdLine, int nCmdShow)  {  pGame = new CMyGame(60);  if (pGame)  {  pGame->Run(hInstance);  delete pGame;  }  return 0;  }  CMyGame.h:  #pragma once  #include "Game.h"  class CMyGame : public CGame  {  public:  CMyGame(int fps = 0);  ~CMyGame(void);  void Init();  void Update(float fDeltaTime);  private:  CSprite\* pLeftGear;  CSprite\* pRightGear;  CSprite\* pBackground;  bool bPlaying;  const int STEP\_walker = 10;  };  CMyGame.cpp:  #include "CMyGame.h"  CMyGame::CMyGame(int fps) :CGame(fps)  {  }  CMyGame::~CMyGame(void)  {  }  void CMyGame::Init()  {  pResManager->LoadPictureFromFile(L"Background", L"background.bmp");  pResManager->LoadPictureFromFile(L"Gear", L"gear.bmp");  pBackground = pSpriteManager->CreateSprite(L"Background");  pBackground->SetDestRect(0, 0, rectClient.right, rectClient.bottom);  pLeftGear = pSpriteManager->CreateSprite(L"Gear", 400, 200);  pLeftGear->SetTransparent(true);  pRightGear = pSpriteManager->CreateSprite(L"Gear", 515, 200);  pRightGear->SetTransparent(true);  POINT pt = { pLeftGear->GetWidth() / 2, pLeftGear->GetHeight() / 2 };  pLeftGear->SetHotSpot(pt);  pRightGear->SetHotSpot(pt);  bPlaying = true;  }  void CMyGame::Update(float fDeltaTime)  {  CGame::Update(fDeltaTime);  if (!bPlaying)  {  return;  }  static float fTime = 0;  fTime += fDeltaTime;  pLeftGear->SetRotationAngle(fTime);  pRightGear->SetRotationAngle(-fTime - ((360 / 8 / 2) \* 3.1415 / 180));  }  第二题关键代码：  CMyGame.h:  ...  CAnimationSprite\* pWalker;  BITMAP bmWalker;  CAnimationSprite\* pMonster;  BITMAP bmMonster;  enum {  right,  down,  left,  up  } dir;  const int STEP\_walker = 10;  const int STEP\_Monster = 20;  void HandleKeyDown(WPARAM, LPARAM);  CMyGame.cpp:  Init(){  ...  pResManager->LoadPictureFromFile(L"Walker", L"walker.bmp");  pResManager->LoadPictureFromFile(L"Monster", L"monster.bmp");  PResDesc resWalker = pResManager->GetResource(RES\_BMP, L"Walker");  GetObject((HBITMAP)resWalker->dwHandle2, sizeof(BITMAP), &bmWalker);  pWalker = pSpriteManager->CreateAnimationSprite(L"Walker", 4, 5, 0, 0, bmWalker.bmWidth / 4, bmWalker.bmHeight / 4);  pWalker->SetTransparent(true);  pWalker->Play();  pWalker->SetTextureStartPos(0, pWalker->GetHeight() \* 2);  dir = right;  PResDesc resMonster = pResManager->GetResource(RES\_BMP, L"Monster");  GetObject((HBITMAP)resMonster->dwHandle2, sizeof(BITMAP), &bmMonster);  pMonster = pSpriteManager->CreateAnimationSprite(L"Monster", 4, 5, 100, 100, bmMonster.bmWidth / 4, bmMonster.bmHeight / 4);  pMonster->SetTransparent(true);  pMonster->Play();  }  Update(){  ...  switch (dir)  {  case right:  if (pWalker->GetXPos() < rectClient.right - bmWalker.bmWidth / 4)  {  pWalker->SetPos(pWalker->GetXPos() + STEP\_walker, pWalker->GetYPos());  }  else {  dir = down;  pWalker->SetTextureStartPos(0, 0);  }  break;  case down:  if (pWalker->GetYPos() < rectClient.bottom - bmWalker.bmHeight / 4)  {  pWalker->SetPos(pWalker->GetXPos(), pWalker->GetYPos() + STEP\_walker);  }  else {  dir = left;  pWalker->SetTextureStartPos(0, pWalker->GetHeight());  }  break;  case left:  if (pWalker->GetXPos() > 0)  {  pWalker->SetPos(pWalker->GetXPos() - STEP\_walker, pWalker->GetYPos());  }  else {  dir = up;  pWalker->SetTextureStartPos(0, pWalker->GetHeight() \* 3);  }  break;  case up:  if (pWalker->GetYPos() > 0)  {  pWalker->SetPos(pWalker->GetXPos(), pWalker->GetYPos() - STEP\_walker);  }  else {  dir = right;  pWalker->SetTextureStartPos(0, pWalker->GetHeight() \* 2);  }  break;  }  // 检测碰撞  // 获取精灵的包围盒  RECT rtWalker = \*(pWalker->GetBoundingBox());  if (pMonster->CollideWith(rtWalker))  {  pSpriteManager->DeleteSprite(pWalker, true);  }  }  void CMyGame::HandleKeyDown(WPARAM wParam, LPARAM lParam)  {  switch (wParam)  {  case VK\_LEFT:  pMonster->SetTextureStartPos(0, pMonster->GetHeight());  pMonster->SetPos(pMonster->GetXPos() - STEP\_Monster, pMonster->GetYPos());  break;  case VK\_RIGHT:  pMonster->SetTextureStartPos(0, pMonster->GetHeight() \* 2);  pMonster->SetPos(pMonster->GetXPos() + STEP\_Monster, pMonster->GetYPos());  break;  case VK\_UP:  pMonster->SetTextureStartPos(0, pMonster->GetHeight() \* 3);  pMonster->SetPos(pMonster->GetXPos(), pMonster->GetYPos() - STEP\_Monster);  break;  case VK\_DOWN:  pMonster->SetTextureStartPos(0, 0);  pMonster->SetPos(pMonster->GetXPos(), pMonster->GetYPos() + STEP\_Monster);  break;  return;  }  } | |
| **7、实验结果或结论（**实验结果怎么样？你从这个实验你学会了什么？得出了什么结论？） | |
| **（实验结果截图+文字说明）**  两道题结果如录屏 01.mp4 所示 | |
| **8、备注或说明**  **）** | |
| **旋转角度利用公式从弧度制转化为角度值可以更直观更灵活地修改角度。** | |