Sports Games

CS134

Chris Pollett

Oct 27, 2004.

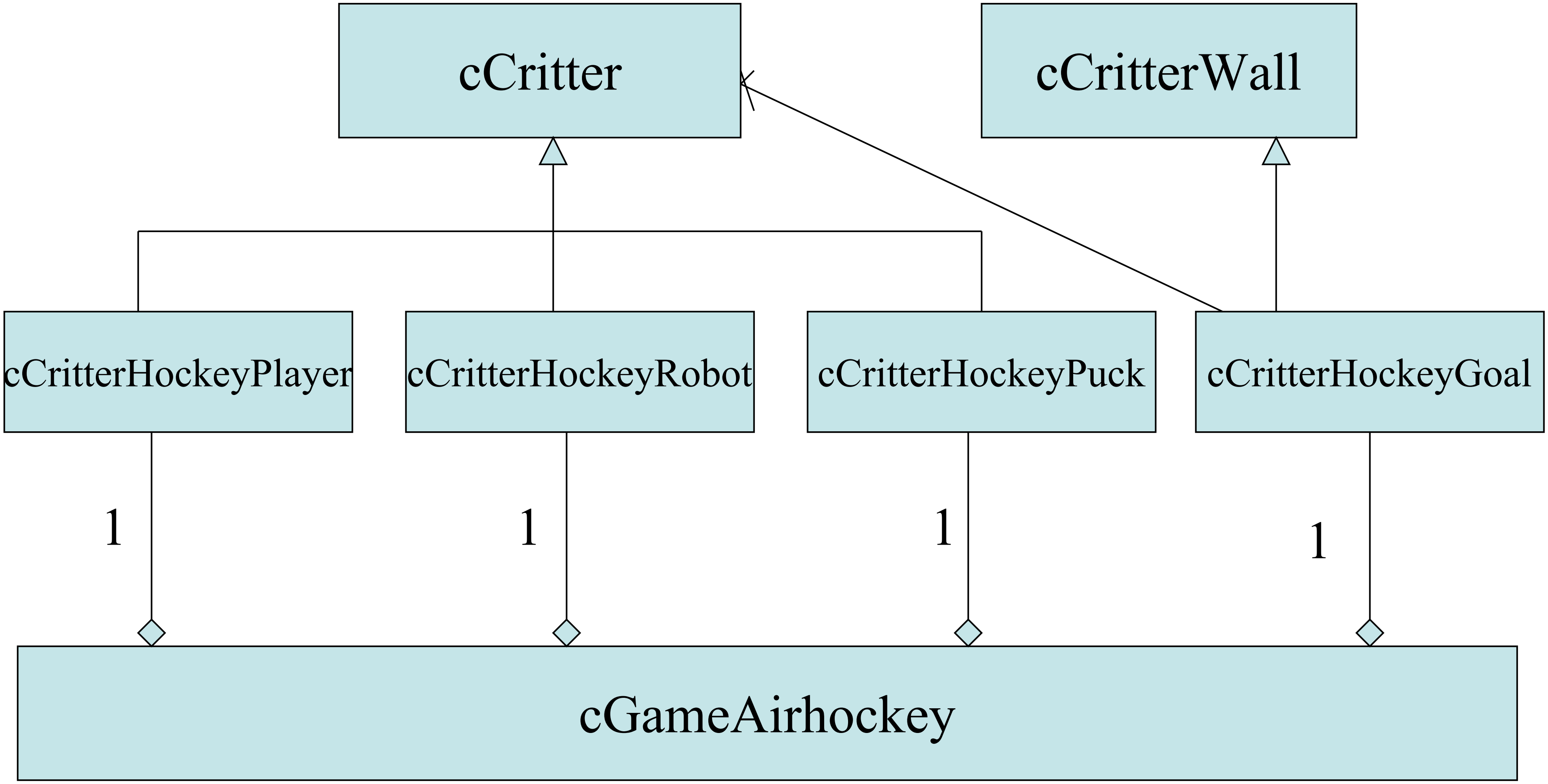
Outline

- Airhockey game
  - Specification
  - Design
  - Code
- Help files
  - How to make an effective file
  - Two kinds of help file

Airhockey Specification

- Concept
  - game is inspired by table game of same name. Players slide their pieces around and try to knock the puck into the opposing player's goal. Like real airhockey players are not allowed to cross the center divide.
- Appearance
  - Some simple diagrams of what a screenshot might look like
- Controls
  - pieces are moved with the mouse. These pieces can bump the puck. The hardness of the bump is determined by the puck speed.
- Behavior
  - Robot is aware of the puck and accelerates towards it. User or opponent gets one point for each goal they get. First to 7 points wins.

UML Design


cCritter
cCritterWall
cCritterHockeyPlayer
cCritterHockeyRobot
cCritterHockeyPuck
cCritterHockeyGoal
1
1
1
1
cGameAirhockey

Code

- Unlike some of the other games we've seen all of the critters are set up in the constructor. Order in which initialized is important as it affects the drawing order.
- Relevant code looks like:

```
setPlayer(new cCritterHockeyPlayer(this));
_ppuck = new cCritterHockeyPuck(this);
_phockeyrobot - new cCritterHockeyRobot(this);
_pmygoal = new cCritterHockeyGoal(cVector(_border.lox()+GOALOFFSET,
     -GOALRADIUS), cVector(_border.lox()+GOALOFFSET,
     GOALRADIUS), GOALTHICKNESS, this);
_pmygoal->setOpenside(BOX_LOY);
_pmygoal->setTarget(_phockeyrobot);
_probotgoal = new cCritterHockeyGoal(cVector(_border.hix()-GOALOFFSET,
     -GOALRADIUS), cVector(_border.hix()-GOALOFFSET,
     GOALRADIUS), GOALTHICKNESS, this);
_probotgoal->setOpenside(BOX_HIY);
_probotgoal->setTarget(pplayer());
```

Robot

- To get the robot to go after the puck the line:

  ```
  addForce(new cForceObjectSeek(ppuck,
     ROBOTACCELERATION));
  ```

  has been added. This by itself wouldn’t let the robot accurately hit the puck. To do this update also needs to be modified.

update

```
void cCritterHockeyRobot::update(CPopview *pactiveview)
{
        cCritter::update(pactiveview);
        cGameAirhockey *phgame = (cGameAirhockey*)(pgame());
        cCritter *ppuck = phgame->ppuck();
        cCritterHockeyGoal *probotgoal = phgame->probotgoal();
        Real puckx = ppuck.position().x();
        Real robotgoalx = probot.position().x();
        Real pucky = ppuck.position().y();
        Real robotgoaly = probot.position().y();
        if(_position.x() < puckx && puckx <robotgoalx &&
        (_position.y() >pucky && pucky >robotgoaly ||
        _position.y() <pucky && pucky < robotgoaly) )
        {
             _acceleration =ROBOTACCELERATION * togoal;
             if(togoal%topuck > COSINESMALLANGLE && tangent() % topuck >
                COSINESMALLANGLE)
             _velocity.turn(1&_personality)?PI/2:-PI/2);
        }
}
```

Help files

- This is a file where users can read within your program if they can’t figure things out. i.e., a User’s guide.
- All versions of your program should have such a guide
- Should keep the actual help file as simple as possible so that you maintain it.
- In real world would split into topics but don’t have to for the programs we are writing.
- Like commenting should avoid the obvious (this line sets x to 5) and say the nonobvious.

How to make an effective file

- Keep it single-spaced. It is going to be read online and you don’t want the user to have to scroll a lot.
- Do not put a new paragraph mark at the end of each line since will appear in a resizable window.
- You can have bitmaps but try to keep small.
- Using Word’s, Heading1, Heading2, etc.

Two kinds of help file

- Windows nows has two incompatible formats for help files: those for Version 6. Those for .NET flavors (based on HTML).
- One can also get the latter to sorta work under 6.0
- To create help files you should download HTML Help Workshop from Microsoft.

More on Help files

- Until 6.0, VS shipped with hcw.exe which was known as Help workshop. It allowed one to convert between .rtf files and .hlp files
- In VS 7, the program hhw.exe you can download from Microsoft converts .htm files into .chm files

Help Steps

- Create an .htm or .rtf file.
- Use the previously mentioned programs to convert it into a help format. For HTML help workshop
  - Look under Start| Programs | HTML Help Workshop
  - Go file File|new| Help Project File, give a name and location
  - DO Add/Remove Topic Files to add your .htm file to the project
  - Click Save and Compile
  - Can view file .chm file in Internet Explorer

More Help steps

- Include in your program htmlhelp.h and htmlhelp.lib
- Add a Menu selection so user can tell your app to open the help file. ID_HELP_USER_GUIDE is a already added in Pop together with the handler CMainFrame::OnHelpUsersGuide
- Edit your handler:

```
void CMainFrame::OnHelpUsersguide()
{
    ::HtmlHelp(GetSafeHwnd(), “Pop.chm”,
    HH_DISPLAY_TOPIC, 0);
}
```

VS 6 Help

- Similar to VS7 case except Help Workshop can be found under Start|Programs|Micrsosoft VS| MS VS Tools
- Create a project like in VS 7 case. Now add .rtf files.
- This time when edit the handler do:

```
void CMainFrame::OnHelpUsersguide()
{
    ::WinHelp(GetSafeHwnd(), “Pop.hlp”,
    HH_CONTENTS, 0);
}
```