

CH7_002

EXERCISE

AIPLAYER: POINT-TO-POINT

Exercise Files

Starter – "Kit/gpgt/server/scripts/gpgt/chapter7/exercise002.cs"

Answers – "Kit/gpgt/server/scripts/gpgt/chapter7/answers/exercise002_f.cs"

Exercise Mission

Chapter 7: "002 AIPlayer: Point-to-point navigation"

Synopsis

In this exercise, we will refresh our memories regarding AIPlayers' datablocks and then we will learn how to make an AIPlayer move back-and-forth between two-points.

Prerequisites

1. *ch1_001.pdf "Using The Kit"*
2. *ch7_001.pdf "AIPlayer: Basic Creation"*

Exercises

1. *Define Datablock (pg 2)*
2. *Bot Creation (pg 3)*
3. *Navigating (pg 5)*

AIPLAYER: POINT-TO-POINT

1 Define Datablock

Goal: Refresh our memories regarding AIPlayer and datablocks.

Starter Code: You are provided with a partially defined datablock definitions.

```
//datablock ?????( pointToPointGuy : BlueGuy )  
//{  
//    category = "gpgt";  
//};
```

Steps:

1. Uncomment this code and fill in the appropriate information to define a datablock suitable for creating (initializing) an AIPlayer.

Output Goal:

None.

Hints:

None.

AIPLAYER: POINT-TO-POINT

2 Bot Creation

Goal: Learn how create a bot using the datablock you created (above,) and then prepare this bot for its back-and-forth movement.

Starter Code: You are provided with a partially defined function (startexercise002.) This is the starter function for this exercise (called automatically when the lesson is loaded). When you have finished the exercise, this function will test your code.

```
function startexercise002()
{
    // 1
    // %theBot = ?????

    // 2
    // %theBot.point0 = ?????
    // %theBot.point1 = ?????

    %theBot.currentPoint = 0;

    %theBot.schedule( 1000 , moveToPoint );
}
```

Steps:

1. Fill in appropriate code to create a bot at the center of the exercise area.
2. Save the following data (in the bot) using these specified dynamic fields.
 - point0 – Bot's current position plus "-10 0 0".
 - point1 – Bot's current position plus "10 0 0".

Output Goal:

After you finish editing, save your work, and run the mission. If you have written your code properly, you will see a blue bot standing in front of you.

AIPLAYER: POINT-TO-POINT

Hints:

1. You should use the utility code that has been supplied for creating bots and paths.
2. To add vector information, use the `addVector()` function.

```
%newVector = vectorAdd( "1 2 3" , "4 5 6" );  
echo(%newVector); //prints: 5 7 9
```

3. To add get a GameBase object's position, use the `getPosition()` method.

```
%position = %obj.getPosition();
```

AIPLAYER: POINT-TO-POINT

3 Navigating

Goal: Learn how to cause a bot to move from one point to another and back again in an endless cycle.

Starter Code: You are provided with a partially defined method (moveToPoint.) This method will instruct the AIPlayer to move from one point to another, wait one second, and repeat, ad infinitum.

```
function AIPlayer::moveToPoint( %this )
{
    // 1
    //?????

    %this.currentPoint = !%this.currentPoint;

    %this.schedule( 1000 , moveToPoint );
}
```

Steps:

1. Fill in the necessary code to cause the bot to move to the position stored in the current target point (currentPoint field stores the index).

Output Goal:

After you finish editing, save your work, and run the mission. If you have written your code properly, you will see a blue bot running back and forth between two positions.

Hints:

1. You can access the point (position) information in your bot like this.

```
%position = %this.point[%this.currentPoint];
```