

CH7_005

EXERCISE

AIPLAYER: STOPPING AND RESUMING

Exercise Files

Starter – "Kit/gpgt/server/scripts/gpgt/chapter7/exercise005.cs"

Answers – "Kit/gpgt/server/scripts/gpgt/chapter7/answers/exercise005_f.cs"

Exercise Mission

Chapter 7: "005 AIPlayer: Stopping and resuming"

Synopsis

In this exercise, we will learn how to make a bot stop moving and then to make it resume motion towards its last destination.

Prerequisites

1. *ch1_001.pdf "Using The Kit"*

Exercises

1. *Stop and Resume (pg 2)*

AIPLAYER: STOPPING AND RESUMING

1 Stop and Resume

Goal: Learn how to stop and resume AIPlayer motion.

Starter Code: You are provided with a fully defined datablock definition (stopAndResumer) a starter function (startexercise005), and a method (onReachdestination).

As can be seen, most of the code you need to start this exercise and to navigate (randomly) from point-to-point is already in place. However, please notice that the starter code does not select a node on the path and it doesn't set the bot into motion. Instead, it schedules a call to a method named stopOrResume.

```
datablock PlayerData( stopAndResumer : BlueGuy )
{
    category = "gpgt";
};

package exercisePackage_005
{
    function startexercise005()
    {
        exerciseCenter.createSimplePath( "testPath" , 15 );
        %theBot = AIPlayer::spawn( exerciseCenter.getTransform() , stopAndResumer );
        %theBot.assignPath( testPath );

        %theBot.isMoving = true;
        %theBot.schedule( 1000 , stopOrResume );
    }

    function stopAndResumer::onReachDestination( %DB , %theBot )
    {
        %pathNode = %theBot.myPath.getObject( %theBot.currentPathNodeNum );
        %pathNode.visibleMarker.setSkinName("red");
        %theBot.currentPathNodeNum = getRandom( 0 , 7 );
        %pathNode = %theBot.myPath.getObject( %theBot.currentPathNodeNum );
        %theBot.setMoveDestination( %pathNode.getTransform() , true );
        %pathNode.visibleMarker.setSkinName("green");
    }
}
```

AIPLAYER: STOPPING AND RESUMING

The next piece of starter code is the method I just mentioned (stopOrResume). This method has the duty of stopping or resuming the bot's motion, depending on whether the bot is moving, or stopped respectively.

```
function AIPlayer::stopOrResume( %theBot )
{
    // 1
    if ( %theBot.isMoving )
    {
        %theBot.isMoving = false;
        // ?????
        %theBot.schedule( 1000 , stopOrResume );
        return;
    }
    // 2
    else
    {
        %theBot.isMoving = true;

        // ?????

        // 3
        if( !%oldDestination || %oldDestination $= "0 0 0" )
        {
            // ?????
        }
        // 4
        else
        {
            // ?????
        }

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

AIPLAYER: STOPPING AND RESUMING

Steps:

1. If the bot is moving,
 - Clear the moving flag (done).
 - Stop the bot (please fill this code in).
 - Schedule another call to stopOrResume in one second (done).
 - Return (done).
2. If the bot is not moving,
 - Set the moving flag (done).
 - Get the bots old destination (please fill in this code and assign the value to %oldDestination).
 - Start moving again (steps 3 and 4 below).
 - Schedule another call to stopOrResume in one second (done).
3. If there was no old move destination, call the onReachDestination callback directly to start the bot in motion (please fill this code in).
4. If there was an old move destination, please resume walking towards it (please fill this code in).

Output Goal:

If you run the mission after editing this code, the AIPlayer will walk around, randomly navigating the path, but it will stop walking and then resume walking on alternating seconds. When it starts walking again, it should always move towards the node it was last seeking.

Questions:

1. In the method stopOrResume, why couldn't we assume that the old destination was valid?