CH7_104

EXERCISE

AIWHEELEDVEHICLE: STOPPING AND RESUMING

Exercise Files

Starter – "Kit/gpgt/server/scripts/gpgt/chapter7/exercise104.cs"
Answers – "Kit/gpgt/server/scripts/gpgt/chapter7/answers/exercise104_f.cs"

Exercise Mission

Chapter 7: "104 AIWheeledVehicle: Stopping and resuming"

Synopsis

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

Prerequisites

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

Exercises

1. *Stop and Resume (pg 2)*

AIWHEELEDVEHICLE: STOPPING AND RESUMING

1 Stop or Resume

Goal: Learn how to stop and resume AIWheeledVehicle motion.

Starter Code: You are provided with a fully-defined datablock definition (stopAndResumerWheeled) a starter function (startexercise104), 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 WheeledVehicleData( stopAndResumerWheeled : DefaultCar )
{
   category = "gpgt";
  maxSteeringAngle = 0.785;  // Better than original car
   maxAISpeed = 1.0;
   moveTol    = 5.0;
};

package exercisePackage_104
{

function startexercise104()
{
   exerciseCenter.createSimplePath( "testPath" , 25 );
   %theBot = AIWheeledVehicle::spawn( exerciseCenter.getTransform() ,
stopAndResumerWheeled );
   %theBot.assignPath( testPath );
   %theBot.currentPathNodeNum = 0;
   %pathNode = %theBot.myPath.getObject( %theBot.currentPathNodeNum );
   %theBot.setMoveDestination( %pathNode.getTransform() , true );
   %pathNode.visibleMarker.setSkinName("green");
   %theBot.isMoving = true;
   %theBot.schedule( 1000 , stopOrResume );
}
function stopAndResumerWheeled::onAdd( %DB, %theBot )
{
         %callerDBName = %DB.getName();
         echo(%callerDBName @ "::onAdd( " @ %DB.getID() @" , " @ %theBot @ " )");
   Parent::onAdd( %DB , %theBot );
         %theBot.setMoveSpeed( %DB.maxAISpeed );
         %theBot.setMoveTolerance( %DB.moveTol );
}
```

(more code on next page)

AIWheeledVehicle: Stopping and Resuming

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

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 AIWheeledVehicle::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 );
   }
}
```

AIWHEELEDVEHICLE: 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 bot's 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 moving towards it (please fill this code in).

Output Goal:

If you run the mission after editing this code, the AIWheeledVehicle will move around, randomly navigating the path, but it will stop moving and then resume moving on alternating seconds. When it starts moving 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?