CH7_103

EXERCISE

AIWHEELEDVEHICLE: RANDOM PATH

Exercise Files

Starter – "Kit/gpgt/server/scripts/gpgt/chapter7/exercise103.cs"
Answers – "Kit/gpgt/server/scripts/gpgt/chapter7/answers/exercise103_f.cs"

Exercise Mission

Chapter 7: "103 AIWheeledVehicle: Random path following"

Synopsis

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

Prerequisites

1. *ch1_001.pdf "Using The Kit"*
2. *ch7_101.pdf "AIWheeledVehicle: Basic Creation"*

Exercises

AIWHEELEDVEHICLE: RANDOM PATH

1 Reinforcing Skills and Random Node Selection

Goal: Reinforce your skills regarding path and bot creation, and bot movement. Add a small twist to the node selection logic.

Starter Code: You are provided with a fully-defined datablock definition (wheeledRandomPathFollower) and a starter function (startexercise103).

```
datablock WheeledVehicleData( wheeledRandomPathFollower : DefaultCar )
{
   category = "gpgt";
};

package exercisePackage_103
{


function startexercise103()
{
   // 1
   // ?????

   // 2
   // ?????

   // 3
   // ?????

   // 4
   // ?????

   // 5
   // ?????
   // ?????

   %pathNode.visibleMarker.setSkinName("green");
}
```

AIWHEELEDVEHICLE: RANDOM PATH

Steps:

1. Create a path for our AIWheeledVehicle to follow.
2. Create an AIWheeledVehicle using the supplied datablock.
3. Assign the path to the AIWheeledVehicle.
4. Assign a random first node to the bot.
5. Start the AIWheeledVehicle moving towards the initial node.

Hints:

1. Torque supplies a function called getRandom(min , max) for getting random numbers, where the returned random number is in the range [min,max].

Output Goal:

If you run the mission after editing this code, the AIWheeledVehicle will drive to a random node (in the path) and then stop moving.

2 Random Navigation

Goal: Learn how to make a bot follow a path in random order.

Starter Code: You are provided with a partially defined method (onReachDestination).

```
function wheeledRandomPathFollower::onReachDestination( %DB , %theBot )
{
   %pathNode = %theBot.myPath.getObject( %theBot.currentPathNodeNum );
   %pathNode.visibleMarker.setSkinName("red");

   // 1
   // ?????

   // 2
   // ?????
   // ?????

   %pathNode.visibleMarker.setSkinName("green");
}
```

Steps:

1. Select a new node from the path as the bot's next navigation point. (Remember, it should be randomly selected.)
2. Tell the bot to move to this new navigation point.

Output Goal:

If you run the mission after editing this code, the AIWheeledVehicle will drive to a random node, select a new random node, drive to it, and so on and so forth.

Questions:

1. What other namespaces would have worked for the onReachDestination callback?

Hints:

1. Remember, the path only has eight nodes.

AIWHEELEDVEHICLE: RANDOM PATH

3 Bonus

Goal: Learn how to write AIWheeledVehicle creation code on your own.

Steps:

1. Go back and remove the spawn code and then replace it with your own AIWheeledVehicle creation code.

Hints:

1. There is no answer for this in the answer key, but you can verify your work against the spawning code. However, please try to get it working without referring to the utility code first.