**DogFunctions.c**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

filename DogFunctions.c

author Andrew Yan

DP email Andrew.Y@digipen.edu

course Wanic Programming

Brief Description:

This file defines the functions to create a specific item, the "dog".

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "stdafx.h" /\* UNREFERENCED\_PARAMETER, NULL\*/

#include "BrickFunctions.h" /\* Function declarations \*/

#include "GameState.h" /\* struct GameState, GameState\_ChangeScore \*/

#include "GameFlags.h" /\* GameFlags\_IsInList \*/

#include "WorldData.h" /\* WorldData\_GetRoom \*/

#include "Room.h" /\* Room\_GetItemList, Room\_SetDescription \*/

#include "ItemList.h" /\* ItemList\_FindItem, ItemList\_Remove, ItemList\_Add \*/

#include "Item.h" /\* Item\_Create \*/

/\* Helper: The action performed when the brick is taken. \*/

void Dog\_Take(CommandContext context, GameState\* gameState, WorldData\* worldData)

{

/\* avoid W4 warnings on unused parameters - this function conforms to a function typedef \*/

UNREFERENCED\_PARAMETER(context);

UNREFERENCED\_PARAMETER(gameState);

UNREFERENCED\_PARAMETER(worldData);

/\* Give the user a hint about how the brick might be used, whenever it is picked up. \*/

printf("The dog seems to continously run around you partly distracting you.\n");

}

/\* Helper: The action performed when the brick is used. \*/

void Dog\_Use(CommandContext context, GameState\* gameState, WorldData\* worldData)

{

Room\* room; /\* The current room \*/

ItemList\*\* roomItemsPtr; /\* The list of items in the current room \*/

Item\* dog; /\* The dog in the player's inventory \*/

/\* safety check on the parameters \*/

if ((gameState == NULL) || (worldData == NULL))

{

return; /\* take no action if the parameters are invalid \*/

}

/\* check if the user is using the dog out of their inventory \*/

if (context != CommandContext\_Item\_Inventory)

{

/\* the user doesn't have the dog - inform the user of the problem and take no action \*/

printf("You must have the dog before you can use it.\n");

return;

}

/\* check if we're in the right room to use the item \*/

if (gameState->currentRoomIndex == 3)

{

/\* we are not in the right room - inform the user of the problem and take no action \*/

printf("You cannot use the dog here.\n");

return;

}

/\* get the current room \*/

room = WorldData\_GetRoom(worldData, gameState->currentRoomIndex);

/\* get the list of items in the current room \*/

roomItemsPtr = Room\_GetItemList(room);

if (roomItemsPtr == NULL)

{

return; /\* take no action, as something is wrong - we should always have an item list \*/

}

/\* Find the robot in the player's inventory - it should be there, since we are in the Inventory context \*/

dog = ItemList\_FindItem(gameState->inventory, "dog");

/\* Remove the brick from the user's inventory - they won't need it again \*/

gameState->inventory = ItemList\_Remove(gameState->inventory, dog);

/\* Tell the user what they did \*/

printf("The dog runs around the robot and distracts it allowing you to pass.\n");

}

/\* Build a "brick" object \*/

Item\* Dog\_Build()

{

/\* Create a "brick" item, using the functions defined in this file \*/

return Item\_Create("dog", "A large dog", true, Dog\_Use, Dog\_Take, NULL);

}

**DogFunctions.h**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

filename DogFunctions.h

author Andrew Yan

DP email Andrew.Y@digipen.edu

course Wanic Programming

Brief Description:

This file declares the functions to create a specific item, the "dog".

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma once

typedef struct Item Item;

/\* Build a "dog" object \*/

Item\* Dog\_Build();