



ROBOT SIMULATOR

Software Design Document

Wednesday - 11/29/2017

Syreen Bn
5191855

TABLE OF CONTENTS

1. INTRODUCTION	2
1.1. Purpose	2
1.2. Scope	2
1.3. Overview	2
1.4. Reference Material	2
1.5. Definitions and Acronyms	2
2. SYSTEM OVERVIEW	3
3. SYSTEM ARCHITECTURE	6
3.1. Architectural Design	6
3.2. Decomposition Description	7
4. COMPONENT DESIGN	11
5. HUMAN INTERFACE DESIGN	11
5.1. Overview of User Interface	11
5.2. Screen Images	11

1. Introduction

1.1. Purpose:

This design document describes the architecture and system design of a video game which is called Robot Simulator.

1.2. Scope:

This document provides a detailed information describing how the Robot Simulation is designed and built. It describes the implementation of the arena class in very detailed. Also, it talks specifically about how the entities in the arena interact with each other.

1.3. Overview:

This document provides all definitions of game items and entities to make reading the document easier and more understandable. Then, it shows a specific explanation about the instructions of the game and how the user can play this game. After that, it talks about the architectural design followed with all possible scenarios between all entities and what is the taken action.

1.4. Definitions and Acronyms:

- Player: the basic building block of the game, this entity could be controlled by the user by the arrow keys.
- Robots: There are five robots. They are mobile entities moving around and try to avoid hitting the player.

- HomeBase: It is mobile entity moving around the arena. If it hit the robot, the robot will change to be a SuperBot entity.
- RechargeStation: It is a immobile entity and centered in the arena. All mobile entities deal with this entity as an obstacle except the player. The player uses his entity to recharge his battery.
- Obstacles: There are five obstacles. They are a immobile entities and they act as a wall.
- SuperBot: once the game is started there are not any SuperBot entity. However, the first one will produce when the robot hit the HomeBase.
- Sensor Touch: this sensor works when the robot collides with any entity to take the new angle and new point of contact after the collision happened.
- Sensor Proximity: each robot has two proximity sensors, one for the right side and one is for the left side. These two sensors are to sense if is there any entity around the robot and then take the appropriate action.
- Sensor Distress Call: This sensor when the robot is frozen to send signs to the all other robots who is in the range to come and unfreeze the caller.
- Sensor Entity Type: this sensor to return the type of the entity to help the proximity sensor to take the right action.

2. SYSTEM OVERVIEW:

The robot simulator is a video game. There are thirteen entities, some of them are mobile while the others are immobile. Six of these entities are immobile and they divide into two categories which are obstacles and recharge station. The number of the obstacles are five and

they act as solid walls. The last immobile entity is the recharge station and its job recharges the player battery. The other category is the mobile entities and there are three types of them. They are five robots, a player, and a home base, and they are moving around. However, there are some difference between these three types of entities. The home base is moving around by itself and nobody effect on its move. Also, the heading angle of the home base is changing randomly. When the home base collides with any robot, the robot will change and become a Super Bot entity and it cannot be a robot again.

Super Bot is mobile entities and moving around. The user will not see this type of the entity until the home base hit any robots. If the Super Bot hit the player, the player will freeze for a specific period of time.

Also, the player is a mobile entity. It could be controlled by a user by using the arrow keys. The right is to change the heading clockwise and the left arrow is to change the heading counter-clockwise. Moreover, the up arrow is to increase speed while the down one to decrease speed. The player has a sensor touch to calculate the angle of contact and point of contact when the player touches the walls or any other entities. Moreover, the player has a battery and it is depleting as the robot move. The depletion process depends on the speed and the distance. The battery level is printed on the screen under the player name and id. If the charge approaches 4.00, the speed of the robot will be 1 and the user should head to the recharge station to recharge the player again. Furthermore, if the charge approaches 0.00, that means the user loses the game and “YOU LOST THE GAME, Restart Again” will print on the screen under the battery level, that is one way to lose the game. So, the user needs to press restart button to play the game again.

The last type of mobile entities are the autonomous robots. There are five autonomous robots and they are moving around. Each robot has five sensors, which are sensor touch, a distress call sensor, entity type sensor and two proximity sensors one to the right side and one to the left side. The proximity sensors have limited ranges and they sense to tell the robot if there is any entity around the robot or not. If the player, obstacles, recharge station or walls are close to the robot, the robot will move in the opposite side to avoid touching these types of entities. If the player touches the robot, the robot will freeze and will start sending distress calls for all other robots to come and unfreeze. If one of the robots receives these calls, it will move forward to the frozen robot and release it. However, if the home base is in the range of the robot, the robot will go forward to the home base trying to hit it. When the robot touches the home base, it will change to be a Super Bot entity. The Super Bot is a mobile entity and it is moving around. It is exactly like the robot, but it has a distinct color and it is impossible for the player to freeze the Super Bot. If the player tries to freeze the Super Bot by touching it, the player will freeze for a short period of time and act as an immobile entity during this period, that means it cannot move or be controlled by the arrow keys. If all robots touch the home base and become Super Bot entities before the player freezes them, the player will lose the game and "YOU LOST THE GAME, Restart Again" will print on the screen under the battery level. So, the player needs to restart the game, to play again and this is the other way of losing the game.

To win the game, the player should freeze all robots before being out of charge.

Furthermore, the user should avoid hitting the obstacles because if the player bumps into obstacles, the speed will slow down, and the battery will deplete. If the user wins the game, "WOW, YOU WON THE GAME" will print on the screen under the battery level of the

player and the home base and the player will stop moving. If the user wants to stop playing the game for a while, he/she could press on “Pause” button and the game will stop. And, if he/she press “Play” button again, the same game will continue.

3. SYSTEM ARCHITECTURE

3.1. Architectural Design:

The arena class has access to all entities, events, and sensors. Each event has a special situation. Firstly, the collision event is checked in CheckEntityCollision function which takes two arena entities as parameters. It regularly checks the distance between two entities if it is not greater than 0, a new EventCollision will generate and register and each entity will take the appropriate action. Secondly, the in the proximity event is checked in CheckForEntityProximity function which takes two parameters. One of these parameters is a robot and the other one is an arena entity. The function checks to see if the ArenaEntity located inside the range of the ProximitySensor of the Robot. When the function finds that the ArenaEntity is in the range, the EventProximity will generate and register. Then, the robot will take an appropriate action. If the ArenaEntity is Player, RechargeStation, Obstacle or Wall, the robot will avoid colliding with these types of entities. However, if it is HomeBase, the robot will move forward to it. The last situation is when the ArenaEntity is a robot, this situation depends on the status of the robot. If the ArenaEntity which is a robot is inDistress situation, the robot will move forward to it while no action will be taken if it is not inDistress situation. Finally, the distress call from robot is checked in the CheckForRobotvoidDistressCall function. This function takes two robots and check

they are in the distressRange of each other. If they are, it starts to check if any of them inDistress situation to go forward to it and release it. When the event is generated, the sensor pointer of type Sensor will accept this event and pass it to the entity to take the action.

3.2. Decomposition Description:

- ♦ Player
 - Player → Wall: the player will bounce off.
 - Player → Obstacle: the player will bounce off and the battery will deplete because of the collision.
 - Player → RechargeStation: the battery of the player will recharge, and the player will bounce off.
 - Player → Robots: the player will freeze the robot time at the position of the collision event and then the player will bounce off. The battery of the player will deplete because of the collision.
 - Player → HomeBase: the player will bounce off. The battery will deplete because of the collision.
 - Player → SuperBot: the player will freeze for a period at the same position of the collision event and the SuperBot will bounce off.
- ♦ Robots
 - Robot → Wall: the PoximitySensor will notify the robot to avoid hitting the walls.
 - Robot → Obstacle: the PoximitySensor will notify the robot to avoid hitting the Obstacle.

- Robot → RechargeStation: the PoximitySensor will notify the robot to avoid hitting the RechargeStation.
- Robot → another Robot: there are two situations. Firstly, if one of the robot is frozen and sending a distress call. The PoximitySensor of mobile robot will notify the robot to go forward the frozen one to unfreeze and both will bounce off. However, if two robots are moving around and hit each other, both will bounce off and continue moving.
- Robot → HomeBase: if the HomeBase in the range of the PoximitySensor of the robot, the robot will go forward to the HomeBase and touch it. When the robot touches the HomeBase, it will be a SuperBot entity. Then, the HomeBase and the SuperBot both will bounce off.
- Robot → SuperBot: there are two situations. Firstly, the robot is frozen and sending a distress call. The SuperBot will receive this distress by the PoximitySensor and go forward to unfreeze the robot and then both will bounce off. However, if the robot and the SuperBot is moving around, no action will be taken. If they hit each other, they will bounce off and continue moving.
- Robot → Player: the PoximitySensor will notify the robot to avoid hitting the player. However, because the player could be control by a user and increase its speed, the collision could happen. If the collision happened, the robot will freeze at the position of the collision event and started sending a distress call for all other robots, while the player will bounce off and continue moving.
- ♦ HomeBase

- HomeBase → Wall: the HomeBase will bounce off and continue moving around.
- HomeBase → Obstacle: the HomeBase will bounce off and continue moving around.
- HomeBase → RechargeStation: the HomeBase will bounce off and continue moving around.
- HomeBase → Robot: The robot will be a SuperBot entity and the HomeBase will bounce off and continue moving around.
- HomeBase → SuperBot: the HomeBase and SuperBot will bounce off and continue moving around.
- HomeBase → Player: The player and HomeBase will bounce off and continue moving around.
- ♦ SuperBot
 - SuperBot → Wall: the SuperBot will bounce off and continue moving around.
 - SuperBot → Obstacle: the SuperBot will bounce off and continue moving around.
 - SuperBot → RechargeStation: the SuperBot will bounce off and continue moving around.
 - SuperBot → another Robot: there are two situations. Firstly, the robot is frozen and sending a distress call. The SuperBot will unfreeze the robot and then both will bounce off. However, if the robot and the SuperBot are moving around, they will bounce off and continue moving.

- SuperBot → HomeBase: The HomeBase and the SuperBot will bounce off and continue moving around.
- SuperBot → SuperBot: both SuperBot will bounce off and continue moving around.
- SuperBot → Player: The player will freeze at the position of the collision event for a fix period of time. While the SuperBot will bounce off and continue moving around.
- ◆ RechargeStation
 - RechargeStation → Robot: the robot will bounce off and continue moving around.
 - RechargeStation → SuperBot: the SuperBot will bounce off and continue moving around.
 - RechargeStation → Player: The battery of the player will recharge again and the player will bounce off and continue moving around.
 - RechargeStation → HomeBase: the HomeBase will bounce off and continue moving around.
- ◆ Obstacles
 - Obstacle → all mobile entities: the battery of the player will deplete before bouncing off while all other mobile entities will just bounce off.
- ◆ Ending the game:
 - Losing the game: there are two situations to lose the game. Firstly, when the battery of the player approach 0. Secondly, when all robots hit the HomeBase

and became SuperBot entities. In both situation, “YOU LOST THE GAME, Restart Again” will print on the screen under the battery level of the player.

- Winning the game: the player has to freeze all robots before any of them collide with HomeBase. If the user wins the game, “WOW, YOU WON THE GAME” will print on the screen under the robot and the player will stop moving.

4. COMPONENT DESIGN:

When the Sensor class receives the event, it will check what is the id of the entity. After that, the Accept function of the entity class will be called and the event will pass. In each class of mobile entities, they are many accept function each one receives a type of the event. Inside the accept function, it will check the type of the other entity which the event happened with to take the right reaction.

5. HUMAN INTERFACE DESIGN

5.1. Overview of User Interface:

The user can play this game by using the arrow keys. The right arrow is to change the heading clockwise while the left arrow is to change the heading counter-clockwise. Moreover, the up arrow is to increase speed when the down one to decrease speed. If the user presses any other key, the window of the game will close. When the user loses the game, “YOU LOST THE GAME, Restart Again” will print under the battery level of the player entity. However, “WOW, YOU WON THE GAME” will print when the user won the game.

5.2. Screen Images:

