ETSA03-SRS-DLH

SOFTWARE REQUIREMENTS SPECIFICATION for the Dullahan

Version 0.9

Prepared by Henrik Abrahamsson

Group 5

May 8, 2020

Revision History

Name	Date	Reason For Changes	Version
Henrik Abrahamsson	2020-03-26	Initial draft.	0.1
Henrik Abrahamsson	2020-04-05	Alpha release: Features described.	0.5
Henrik Abrahamsson	2020-04-10	Specified detailed requirements.	0.6
Henrik Abrahamsson	2020-04-25	External interface draft and removal of one feature.	0.7
Henrik Abrahamsson	2020-05-08	Beta release	0.9

Contents

1 Introduction

1.1 Purpose

The purpose of this Software Requirement Specification (SRS) is to describe the software requirements for the Dullahan, a droid developed by group 5. The Dullahan will be a droid purchasable on the open market before the LU Rumble. The key feature of the Dullahan will be its ability to communicate and detect enemies without an active team-leader broadcasting. This is something which gives a team containing multiples of this droid an edge compared to a team consisting of other droids.

1.2 Document conventions

In the remainder of this document, the Dullahan will be referred to as ”the droid” or ”the robot”. The terms ”teammate” and ”ally” will be used interchangeably. The term ”heading” will be referring to the direction that the droids body will be facing, while the term ”gun heading” will be used to refer to the direction of the gun.

When the term ”win rate” is used, it refers to the percentage of which the team with the droid lands on 1st place. For example, if the Dullahan team lands on 1st place 93 out of 100 rounds, the win rate would be 93%. Note that the term **does not** refer to the score percentage.

This document is organized according to a template for a SRS that respects IEEE standards by Jean-Philippe Eisenbarth. It is also heavily inspired by the SRS for the Basic Melee Bot developed by the course administration.

For requirements, the following formats will be used. It has been copied from the SRS for the Basic Melee Bot due to it being a convenient numbering convention, with the addition of the last format for the system level quality requirements.

- FREQ-<Feature Number>-<Running Number> for functional requirements.
- QREQ-<Feature Number>-<Running Number> for feature specific quality requirements.
- QREQ-S-<Running Number> for system level quality requirements.

1.3 Intended Audience and Reading Suggestions

The main readers of this document will be other groups who are looking to purchase the droid, but also the course administration.

The general structure of the document is as follows, with a chapter for each bullet point:

- The droid is introduced and document conventions and references are explained and given.
- The droid is briefly described on a higher level.
- The requirements for the external interface are described, such as the events that the droid intercepts from Robocode.
- The system features are described in-depth with a numbered list of the software requirements necessary in order to implement the feature. The features are also given a priority.
- The quality requirements of the droid are described, more specifically the requirements of the performance of the droid, e.g. requirements for win rate against other robots and similar requirements.
- Appendixes with e.g. explanations of abbreviations.

The other groups are mostly going to be reading this document in order to understand what the droid is capable of and are therefore advised to focus on chapter 2 for the basic understanding of the droid and chapter 4 for a more in depth understanding of the features of the droid. Specifically for quality engineers, the specific requirements listed under each feature in chapter 4 are going to be of interest.

1.4 Project scope

Even though the droid will be compatible with team leaders using the ETSA03 RoboTalk protocol v1_5, the idea of the droid also working without a team leader gives the project a unique touch. The idea is unique to the point that it would be highly unlikely for every droid on the market to implement it, which would give further incentives to other teams to buy our droid.

The group currently has the following two business goals: 1) having at least one group purchasing five copies of our droid, and 2) selling at least nineteen copies of our droid. Basically, the development aims to awe at least one other group to the point of them being completely bought in on our niche market while also spreading the sales enough that at least three to four groups buy copies of the droid.

The following types of student groups will be targeted as main costumers:

- Groups who need extra firepower but have already purchased the maximum amount of normal robots.
- Groups who do not have enough cooperation between their robots and feel like this would put them at a disadvantage in the LU Rumble.
- Groups who feel like having a large amount of robots (or droids) of the same type would be an advantage in the LU Rumble.

1.5 References

RoboWiki, available at `http://robowiki.net/wiki/`

The Robocode API, available at `https://robocode.sourceforge.io/docs/robocode/`

Software Requirements Specification that respects IEEE standards by Jean-Philippe Eisenbarth, available at `https://github.com/jpeisenbarth/SRS-Tex`

Software Requirement Specification for the Basic Melee Bot by the course administration, available at `https://github.com/lunduniversity/introsofteng/blob/master/project-rumble/basic-bots/requirements-specifications/ETSA02-SRS-BMB.pdf`

The ETSA03 RoboTalk protocol v1_5, for communication between robots in the same team, developed by the course administration, availably at `https://github.com/lunduniversity/introsofteng/blob/master/project-rumble/document-templates/ETSA03%20RoboTalk%20v1_5.pdf`

2 Overall description

2.1 Product Perspective

The Dullahan is a droid with the unique trait of being able to communicate with each other and coordinate fire, without the use of a radar, even if the team leader should be killed. The idea is that this gives the droid an advantage when compared to other droids who are completely team leader-dependant. Thus, it can give teams an advantage in the LU Rumble because the team isn't required to alter their strategy to be compatible with this droid. The droid is made to be able to run entirely without the use of a leader while still being compatible with leaders who also use the ETSA03 Robotalk protocol.

2.2 Product Functions

For a more detailed description, with more specific system requirements for each feature, refer to chapter 4. The droid shall have the following functions:

- Be able to communicate their own positions and positions of enemies discovered via bulletHit-events to each other without a team leader present.
- Avoid driving into walls and allies using an anti-gravity movement system that calculates a ”gravity force” causing the robot to gravitate away from allies and walls.
- Detect enemies by shooting semi-random low power bullets around the battlefield and get enemy positions from the bullet that hit an enemy. Will also fire back when fired upon and fire when colliding with a robot that isn't a teammate, to further increase the chance of finding enemies.
- Use the shared information about enemy locations to coordinate attacks where some or most of the bullets are probable to hit the enemy and therefore trigger another coordinated attack on the new enemy location. The idea is that this will trigger yet another hit, which will trigger yet another coordinated attack, causing a chain reaction.

3 External Interface Requirements

3.1 Robocode Interface

The Dullahan intercepts a number of events from Robocode in order to determine how to act. In table 1 below, all events that the bot intercepts are listed along with a description of when the event is triggered and why the robot intercepts it.

Table 1: Robocode events intercepted by the Dullahan

Event	**Description and purpose**
onStatus()	Triggers every turn. Cue for the robot to broadcast its position.
onMessageReceived()	Triggers when the bot receives a message which happens multiple times per turn. The message is decoded and color is changed, teammate positions are updated and potential enemy positions are added.
onBulletHit()	Triggers when a bullet fired by the bot hits another robot. The robot broadcasts the found position and whether or not it belongs to our team. The position is received from the fired bullets' position when the event triggered.
onHitByBullet()	Triggers when the bot is hit by a bullet. The event is used to determine if the robot who fired as a teammate or not and is also a cue for the bot to fire back, if necessary.
onRobotDeath()	Triggers when a robot has died. Used to trigger updates in internal lists of enemies and teammates. Dead robots are instantly removed from all lists.
onHitRobot()	Triggers when a collision happens between the robot and another robot. The isMyFault() method is used to determine who caused the collision. If the Dullahan caused it, it will attempt to retaliate.

3.2 Communications Interfaces

The droid uses the ETSA03 RoboTalk protocol provided by the course administration for broadcasting messages. The protocol will be used mainly for communicating the positions of the own team and potential found enemy positions but also for setting the robots colors at the beginning of the battle.

4 System requirements

The droid has the following primary features: a) Leaderless communication, b) Anti-Grav Movement, c) Team-Coordinated Firing Solutions, d) Bullet Saturation for Enemy Identification.

4.1 Feature 1 - Leaderless Communication

4.1.1 Description and Priority

The droid shall be able to communicate with other droids of the same type by sending messages between them even if there is no leader present. The messages will contain the location of an enemy, if the droid has found one by hitting it with a bullet (see feature "Bullet Saturation for Enemy Identification"). The other droids must be able to understand that message and act upon that information in a meaningful way.

This feature is the basis of the entire droid and a requirement for it to be able to operate at all.

Business priority: **very high**

4.1.2 Functional requirements

FREQ-1-1 If the droid finds an enemy position by hitting an enemy, it will broadcast that position to the team.

FREQ-1-2 The droid will share its own position to the team each turn.

FREQ-1-3 The droid will delete enemy positions when 50 turns has passed in order to minimize obsolete information.

4.2 Feature 2 - Anti-Grav Movement

4.2.1 Description and Priority

The droid will use a movement system that the developers have named "Anti-Grav Movement". It works by assigning a pushing force from teammates and walls, causing the droid to steer away from these and naturally gravitate towards areas with lower density of allies and walls.
Note that the QREQ-2-1 includes that two bots who collide trigger **two** collision events.
Business priority: **high**

4.2.2 Functional requirements

FREQ-2-1 The droid will use an anti-gravity movement system to steer away from and minimize the amount of collision with walls and teammates.

4.2.3 Quality requirements

QREQ-2-1 Five Dullahans in a team will only trigger collision events with walls and teammates a maximum of 110 times per 1000 turns when 5 Dullahans battle against 5 spinbots on a 1200x1200 battlefield.

4.3 Feature 3 - Team-Coordinated Firing Solutions

4.3.1 Description and Priority

The droid shall be able to, when paired up with other droids of the same type, coordinate their attacks and determine which enemy to fire at, if one has been found.

Business priority: **medium**

4.3.2 Functional requirements

FREQ-3-1 The droid will fire upon the first position in its internal list of enemy positions, which will be the same for every droid in the same team - causing a swarm of bullets towards that target.

FREQ-3-2 The firing will continue until either the target has been eliminated or no bullet fired by the droid has hit the target in 50 turns.

4.4 Feature 4 - Bullet Saturation for Enemy Identification

4.4.1 Description and Priority

"The droid will use low-energy bullets shot mostly at random around the battlefield in order to get the coordinates of an enemy, using the BulletHitEvent. Along with the random bullets, the robot will also try to detect enemies by firing when it collides with an enemy and firing back when it is hit by a bullet. These bullets will be referred to as either "detection bullets" or "bullet saturation bullets".

This feature is also a requirement for the droid to operate, since it will not be able to detect enemies in any other way (without the use of a compatible team leader).

Business priority: **very high**

4.4.2 Functional requirements

FREQ-4-1 When the droid has no known enemy positions, it will fire low power bullets at semi-random positions around the battlefield.

- If the droid is closer than 300 distance units to a wall, the droid will not fire a detection bullet towards that wall.

5 Other quality requirements

This chapter describes quality requirements not directly linked to a specific feature. These describe the requirements of the working system (i.e. droid) as a whole.

5.1 Team Battle Performance

The droid should perform well against sample robots provided by Robocode. The win rates are calculated on the battlefield size that will be used in the LU Rumble, 1200x1200. Note the explanation of the term ”win rate” and what it refers to in chapter 1. Also note that the win rate is calculated over a high number of rounds in order to get an accurate representation. The following win rates are only guaranteed when **calculating over 1000 rounds or more**.

QREQ-S-1 A team of 5 Dullahan droids (no specified team leader) will have a win rate of at least 90% against a team of 5 Spinbots (no specified team leader).

QREQ-S-2 A team of 5 Dullahan droids (no specified team leader) will have a win rate of at least 90% against the sample team MyFirstTeam 1.0.

QREQ-S-3 A team of 5 Dullahan droids and the BasicLeaderBot will have a win rate of at least 95% against a team of 5 Spinbots (no specified team leader).

QREQ-S-4 A team of 5 Dullahan droids and the BasicLeaderBot will have a win rate of at least 95% against the sample team MyFirstTeam 1.0.

Appendixes

Appendix A: Glossary

SRS - Software Requirement Specification.

Robocode - the platform on which the developed droid will compete.

LU Rumble - the droid battle in which the developed droid will compete in a team against other teams.

TBD - To Be Determined, parts that have been postponed until later. Refer to Appendix B for a list of such items.

Appendix B: To Be Determined List