Bringing It Together
Robbie Robot Shop

CSE 1325 – Fall 2016 – Homework #5
Due Thursday, October 20 at 8:00 am
with Sprint Reviews due 8:00 am on October 6 and 13

One-week projects are fairly rare in the “real world”, though not extinct. “Short” projects take many weeks, while longer projects can run into months or years. We can't squeeze in a multi-year project, but multi-week – that we can do! In this homework, you'll apply a simplified Scrum process to manage the 3-week development of a simple robot shop management system.

Be advised that the next homework will build on this one, adding a graphical user interface Controller and View to this homework's Model. Also note that this project has a lot in common with Homework #4, the Library Management System. Remember, code reuse is your friend.[1]

Introduction

Robbie Robots is a fledgling start up in the exciting field of robotics. They are seeking bright young programmers to build a custom solution for defining new robot products using their growing patented line of modular components, tracking customers and the robots they buy, rewarding their sales people well enough to keep them motivated, and otherwise taking care of business.

You're job is to win this project from RRS by producing a proposal package, including a prototype that wows and other creative and persuasive artifacts that prove you know your stuff. You have exactly 3 weeks to deliver RRS Manager (Prototype) v1.0 with your proposal package, featuring a glorious text menu or command line interface (CLI).

Overview

Robbie Robot Shop assembles their robots from 5 different components – a torso, head, arm(s), locomotor, and battery(ies). A Product Manager (PM) defines new robot models from these components, assigns a product name and price, writes up a brief sales description, and approves the result. The robot then appears at their Point Of Sale (POS) systems so that their Sales Associates (SA) can arrange for their Beloved Customers (BC) to order them, as well as on their web store so that BCs can order them direct. The Pointy-haired Boss (PB) will need to track profit margins, sales, and such from the robot product lines to ensure the business is profitable.

1 You may consider the example solution that I provided for Homework #4 as in the Public Domain, and adapt it as you see fit. Or you may choose to reuse your own solution, or start from scratch. It's your choice.

Requirements

The Product Owner has identified the needs of each of the actors that will interact with the system, and has assembled these into a prioritized Product Backlog in the Scrum spreadsheet, which is included in this set of Requirements by reference.

Product Manager (PM)

The PM needs to be able to create new instances of robot components in the system. For each component, they need to specify a name, part number, type (torso, head, arm, locomotor, or battery), weight, cost, and a brief description. In addition, some types of components need additional data: each torso may have from 1 to 3 battery compartments, each locomotor should specify a maximum speed (in MPH) and power consumed when operating (in watts), each arm should specify power consumed when operating (in watts), and each battery should specify the energy it contains (in kilowatt hours).

The PM needs to be able to define new robot models by selecting one or more of each type of robot component. Only one component of each type may be added, except that up to 2 arms and as many batteries as will fit into the selected torso may be attached. For each robot model, the PM also needs to define a model name, model number, and price. They will want to know the total cost of all components when setting the price to ensure adequate profit for each model.

Beloved Customer (BC)

A BC needs to be able to browse the catalog of robots, with pictures (eventually – not expected on the command line version), prices, shipping costs, and description at the POS terminal and (eventualy) on the web store. Each customer will also need to view their orders and their outstanding bill.

Sales Associate (SA)

A Sales Associate (SA) needs to order robot models for Beloved Customers and present a bill of sale listing the order number, date of sale, customer name, robot(s) ordered, and price (subtotal, shipping, tax, and total). Each Sales Associate would like a sales report for the sales they personally completed with which to lobby for a raise.

Pointed-haired Boss (PB)

The Boss needs to know overall shop metrics, such as the profit margin on each robot and how many were sold, a complete list of orders during a specified period, and sales for each Sales Associate with which to approve or deny requests for raises.

Process

Each developer will follow a simplified Scrum process as discussed in class, and is free to negotiate changes to the Product Backlog features and priority with the Product Owner (the Professor or TA). An approved change order is an email from the Product Owner clearly stating the the change in the Product Backlog is acceptable. A verbal change order is not worth the paper it's printed on, and will not be accepted for credit.

An example (and informal) UML class diagram and sample CLI menu design is provided for your consideration. These are NOT required, and you are encouraged to adapt and improve on or to ignore them. **You must provide your own UML models and design your own menus** – that's part of this assignment. **Do your UML design before you start coding** – that's just good OO!

Every class documented in the standard C++ library is now available for your use on this project. You may also consider third party libraries, but each must be approved by the Product Owner in writing prior to use. The best code you can write is often “new JustWhatINeededWasInTheLibrary()”! **Invest a little time searching the official documentation and sources such as Stack Overflow to see what's available for reuse**, so you can work through more backlog items in 3 weeks. For example, saving the shop's data *may* be best implemented for your design using something like TinyXML (1 or 2). Or maybe not. Investigate and decide, but regardless, **comply with all license agreements!**

Students are permitted (but not required) to form teams of up to 4 students. The teams will self-organize as desired. For example, a team may choose to create a Git Hub project and collaborate through that, or they may decide to keep independent local git repositories and designate one team member as a Build Master to whom all patch sets are sent for the next build. Learn new things, and if you make the Finals, share what you learn in your Class Demo! (More on that later.)

The product baseline should be built frequently – at least once per day, or better yet after every patch set is added – and all of the automated tests run to identify any breakage that needs to be added to the Sprint Backlog of tasks, prioritized, and addressed. The resulting build should be available to all team members (if any) for interactive testing and working on their current backlog item. Remember that team members select tasks from the Sprint Backlog to work (tasks are not assigned). Be proactive!

In lieu of weekly sprint demos, students will submit “Evidence of Compliance” via Blackboard at the end of each sprint – each Thursday at 8 am (*innocent look*). The product itself will not be graded in the first two sprints, but the evidence that the student or team has (1) worked the highest priority backlog features, (2) delivered a tested, working prototype suitable for delivery if the Product Owner decided it's needed in production, and (3) is making suitable progress toward completion of a prototype by the 3rd sprint.

Questions? Concerns? Puzzlement beyond all reason? Contact your Product Owner ASAP! You have only 3 one-week sprints to complete your initial proposal package, so don't waste time waffling.

Deliverables

Each student or team is required to use the git version management system, and may optionally use GitHub. **An archive of the current, stable git repository with commit history or a GitHub link, or a link with access granted to the Product Owner, representing that sprint's deliverable-ready milestone must be delivered to Blackboard *every week*.** {10 points per sprint, 30 total}

Each student or team is required to maintain and work from the Product Backlog and (for each sprint) the Sprint Backlog, using the Scrum spreadsheet provided. Each Sprint Backlog item (task) must be implemented by the single team member with whose initials it is associated. This will enable the graders to estimate the portion of work contributed by each team member. **An updated copy of the Scrum spreadsheet must be delivered to Blackboard *every week*.** {10 points per sprint, 30 total}

Each student or team is required to maintain and work from a UML design, updated as the project progresses. These will include *at a minimum* a class diagram and use case diagram. **Each student or team will deliver an updated set of UML diagrams representing progress toward the final architecture and design *every week*.** These may be in Umbrello format or as PNG, JPG, or GIF images. Note that if you ask for help with your project, the Professor or TA will almost certainly start with your UML diagrams – and if you haven't any, that's your answer! {20 points per sprint, 60 total}

The student or team will deliver a final proposal package to Blackboard that presents their capabilities in the best possible light by the homework due date. The proposal package will include:

1. A working prototype of the system in source code form;
2. The git source code repository with the complete commit history of the project;
3. Supporting UML diagrams including a use case diagram and a class diagram that accurately reflects the code, with additional consideration given for other useful supporting diagrams);
4. A brief user manual (a few paragraphs); and
5. Any sales material deemed to increase the probability of winning the contract (e.g., PowerPoint slides, PDF brochures, YouTube videos).

The average student is expected to complete the top **4** Product Backlog items[2] in 3 sprints, including a working text interface and reasonably complete automated (non-interactive) test code. Partial work may be submitted *for Sprint 3 only* for grading consideration. Completing additional Features from the backlog will be treated as extra credit work similar to Bonus levels. Persistence is worth up to 30 bonus points, and each additional Feature is worth up to 15 bonus points. {180 points plus bonuses}

Each student or team should be prepared to provide a final Product Demo to the class, if selected and scheduled by the Professor after completion of the 3th and final sprint. The number of Product Demos, if any, is dependent on availability of class time. More info to follow. {Up to 30 points bonus}

---

2 Add 2 Features per additional student for teams – 6 for 2-person teams, 8 for 3-person teams, etc.

Grading

This project is graded on a 300 point scale (since this is a 3 week project), with opportunity for substantial additional credit.

Sprint #	1	2	3	Total
Git Repository	10	10	10	30
Scrum Spreadsheet	10	10	10	30
UML Diagrams	20	20	20	60
Final Package			180	180
Total	40	40	220	300

The Fine Print

Robot images are used under license from Graphic Stock. Students may not under any circumstances use graphics provided along with this project for any other purpose, as such use would constitute copyright infringement. Remember, *we talked about this*!