<https://classroom.github.com/a/POCmC_Rn>

### Project Examples

1. **ReNew**

Our company wants to design a system that is aimed at helping people do renovation projects.

The ReNew system is required to provide it’s users with a list of various products for home improvement and handles the renovation process. So once a user is signed in he is presented with the options to create a new project and choose the products he wants to use. The end goal is to obtain a list of products to be purchased (a shopping list) and a summary of the renovation process (installation summary).

Every product has a price, color and unit of measure and is tied to a specific install method. An install method describes how the product is installed and what additional items you need (glue, nails etc.) it has a name and description and a price for the work required to be performed.

The shopping list should contain a list of products and associated items based on install method for a given project. It should display the price computed for every line item and a total price.

The installation summary should contain a list of install methods needed for a given project. It should display the price computed for every line item and a total price.

1. **Styx’s Treasure Hunt**

Our company wants to design a game called “Styx’s Treasure Hunt” that simulates treasure hunting within goblin culture.

The game should allow players to create multiple goblin characters of varying features: height no greater than 1.5m, skin color, eye color, number of gold teeth etc. There are three types of goblins: goblins (+2 Stealth bonus), hobgoblins (+1 Treasure bonus) and bugbears (10% chance to cause guards to flee if spotted). These characters then compete in Styx’s Treasure Hunt Challenge for the glory of earning the title of Most Shiny of Goblins.

The challenge consists of a series of 10 random events of varying difficulty and size. Event difficulty is defined by the number of people a goblin has to sneak past to get to the treasure (no more than 5).

Success is determined by an opposed check between a goblin’s Stealth value (randomized between 1 and 10 when creating a character) + random number between 1 and 20 and the guard’s Perception value (randomized between 1 and 10 when generating the event) + random number between 1 and 20.

Each event grants treasure items at the end. The number of items granted and their value is determined by the difficulty of the event. At the end of the event items go in the goblin’s inventory. A goblin’s inventory is limited to 10 slots, however a goblin may find additional bags at the end of events that take an inventory slot but grant extra slots (no more than 8 per bag). If a goblin’s inventory is full he cannot take the treasure with him at the end of the event unless he also found an additional bag.

At the end of the challenge goblins are ranked based on the value of the items in their inventory and the winner gets fame throughout the goblin kingdoms.

1. **Online Whiteboard**

We would like to design and implement an online whiteboard to be used as a means of connecting remote teams within our company. This application has already been broken down in the following modules.

*Collaboration Module*

This part of the application handles user accounts and sessions. Once signed into the app a user has the ability to create a shared whiteboard and be presented with a link to this whiteboard. The content of the whiteboard is stored on the server. Once a person uses the link their account is associated with the whiteboard for future use. Users get notified when people access or leave a shared whiteboard and changes a user makes to the whiteboard are visible to all associated users.

*Approval Module*

Accounts can have multiple roles – architect, technical lead, developer. Once a design for a project is considered acceptable any user can initiate a review process. The review process consists of the following workflow: first technical leads are asked to review the design. If they are unable to decide then architects are asked to review the design. Any further changes to the whiteboard marks it as not reviewed and the process is resumed.

*Drawing module*

The application should support various UML shapes and features: class, interface, dependency, aggregation, composition etc. Users should be able to undo, redo changes to the online whiteboard as well move and resize multiple shapes at once and also save the contents to a file.

1. **EDI**

Enhanced Defense Intelligence (EDI) System is a system that functions as the electronic warfare defense for the starship Normandy SR-2. The Commander of the Normandy, once voice authenticated, should be able to have streamlined access to various ship subsystems through interactions with EDI.

The subsystems monitored by EDI are the following:

* life support: responsible for managing oxygen levels throughout the starship
* weapons array: responsible for monitoring weapon heat levels and the damage they might sustain
* shield matrix: responsible for managing the energy levels of the ship’s forward and rear shields

Each subsystem should notify EDI if the following threats occur:

* oxygen levels fall under 10%
* a weapon is overheating
* a weapon is damaged or jammed due to overheating
* a shield’s energy level falls under 25%

If a threat occurs EDI should be able to generate a report for the starship’s Commander and act based on his or her input.

The following commands can be given to EDI:

* evacuate ship(in case of loss in oxygen levels)
* take weapon offline(prevents overheating)
* repair weapon(nanobots are dispatched to fix damage or jamming)
* boost shields(increases shield levels by 20% while taking all weapons offline)