CO19-320212 Software Engineering
Spring 2019

Sprint 2 Write-up

Technical details

Released

Tuesday, 5.03.2019 at 21:00

Deadline

Tuesday, 19.03.2019 at 21:00

Duration

14 days

Objective

Now that you have a functional (albeit basic) user interface, you can start adding some game elements.

- Add functionality to a working UI

Program specifications for sprint 2

Please read through this section very carefully. You are expected to implement the entirety of this section for this sprint.

Assumptions

Before reading past this point, please keep the following assumptions in mind:

- *Resources are finite and must always be greater than 0* — i.e. your program must not allow production of pencils when there are not enough materials or allow purchase of items when there is insufficient fund in your bank balance.
- *For this sprint, your program starts from default settings every time it is invoked* — i.e. you do not need to store any data in the computer.
- *Round to the nearest two decimal places in most cases. If a value should not be in fraction form, floor the value.*

Pencil production

Pencil inventory

Your program should keep track of the number of pencils currently in the inventory. The amount of pencils in the inventory increases by 1 (one) with each pencil produced and decreases by 1 (one) with each pencil sold. You start with 0 (zero) pencils at the beginning of the game (default).

Button: Make pencil

- Adds 1 (one) pencil to the inventory
- Consumes 0.21 m of wood
- Consumes 0.20 m of graphite
- This button should be disabled when there are not enough materials to produce a pencil

Total number of pencils produced

Your program should keep track of the number of pencils produced since the start of the game.

Wallet system

Bank balance

Your program should maintain a wallet system where you can credit (add) or debit (deduct) money off your bank balance based on the actions that occur throughout the game. You start with $145.00 at the beginning of the game (default).

Supply-demand based sales

Price of pencil

Your program should display the current price of 1 (one) pencil. The price of pencil is $1.00 by default. However, your program should allow the user to manually increase/decrease the price. Note that changing the price affects the rate of public demand.

Button: Increase price (of pencil)

- Increases the price of 1 (one) pencil by an increment of $0.05

Button: Decrease price (of pencil)

- Decreases the price of 1 (one) pencil by an increment of $0.05

Public demand

In the context of this game, imagine you have an invisible crowd purchasing pencils from you (i.e. consumes pencils from your inventory) automatically. The rate of public demand is defined by the following formula:

$$rate = \frac{1}{price\ of\ Pencil}$$

The rate determines the number of pencils purchased by the crowd every 0.20 seconds.

Example scenario: When $priceOfPencil = \$0.85$, $rate = 1.17$, thus the system is supposed to sell 5.85 pencils per second. However, the number of pencils cannot be a fraction. Therefore, your program should floor the number, such that the program would sell 5 pencils per second.

Pencil sale

Each sale of pencil consumes 1 (one) pencil from your pencil inventory.

Crediting pencil sale revenue to your bank balance

For each pencil sold, your program should credit the current price of 1 (one) pencil at the time the pencil was sold.

Production material

Wood inventory

Your program should keep track of the amount of wood currently in the inventory. The amount of wood in the inventory increases by 100.00 m with every purchase of wood and decreases by 0.21 m with each pencil produced. You start with 1,000.00 m of wood at the beginning of the game (default).

Graphite inventory

Your program should keep track of the amount of graphite currently in the inventory. The amount of graphite in the inventory increases by 100.00 m with every purchase of graphite and decreases by 0.20 m with each pencil produced. You start with 1,000.00 m of graphite at the beginning of the game (default).

Button: Buy wood

- Adds 100.00 m of wood to the inventory
- Debits the current price of wood fom the bank balance (more detail below)
- This button should be disabled when there is insufficient fund in the bank balance

Button: Buy graphite

- Adds 100.00 m of graphite to the inventory
- Debits the current price of graphite fom the bank balance (more detail below)
- This button should be disabled when there is insufficient fund in the bank balance

Price of wood

Your program should vary the price of wood every 5 seconds in a stochastic manner. Use a random number generator to achieve this. The price range for 100.00 m of graphite is $1,000.00 – $2,000.00.

Price of graphite

Your program should vary the price of graphite every 5 seconds in a stochastic manner. Use a random number generator to achieve this. The price range for 100.00 m graphite is $1,500.00 – $2,500.00.

Automatic Pencil Machine (APM)

Pencil production using the APM

Name the APM in this sprint using a "2000" as a suffix (e.g. AutoPencil2000). Every 2000-series APM produces 2 pencils per second, which are automatically added into the inventory. Naturally, producing pencils using the APM would also consume your production material. The APM should automatically stop production when there is insufficient material, and automatically resume production when material is available again. You are free to decide on how you implement the APMs (and any subsequent APMs) and its interaction with regard to the use of material, etc. Please note that you are **not allowed to use multithreading** in this project, to keep things simple!

Button: Buy more 2000-series APM

- Increases the 2000-series APM in your inventory by 1 (one)
- Debits the current price of 1 (one) 2000-series APM from the bank balance
- This button should be disabled when there is insufficient fund in the bank balance

2000-series APM inventory

Your program should keep track of the number of 2000-series APM in your inventory. Please cap the number of 2000-series APM in your inventory to ten for this sprint.

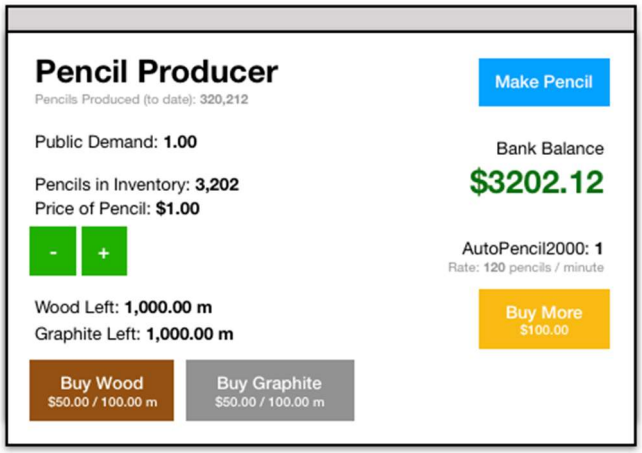
Price of 2000-series APM inventory

The price for the first 2000-series APM in your inventory is $150. The price increases by 10% for every subsequent 2000-series APM.

Example scenario: 1st APM: $150, 2nd APM: $165, 3rd APM: $181.5, and so on.

Deliverable

By the end of this sprint you are expected to have built a very basic game with a working production and sales functionality as described in the program specifications section previous. As the saying goes, a picture is worth a thousand words (or in this case 914 words) — here is a sample mockup of the deliverable for Sprint 2.



Your implementation need not look the same (in fact it should look somewhat different) to the mockup provided above.

Buttons Checklist

Please make sure your program contains the following buttons and that their intended functionalities are properly implemented:

- Make pencil
- Increase price of pencil
- Decrease price of pencil
- Buy wood
- Buy graphite
- Buy 2000-series APM

Additional tips

- Your code should demonstrate good style and discipline. Keep in mind that every time you code, someone else will read it. As a goal, make your code readable, self-explanatory, and standardized. This will also help avoid many potential bugs.
- Spend time to sufficiently comment your code and make use of Doxygen. Explain the rationale behind decisions you made (e.g. choice of class, functions, objects). You would appreciate finding a well-documented code-base in your repository during the subsequent sprints.
- Do not make anything multi-threaded.