CO19-320212 Software Engineering
Spring 2019

Sprint 5 Write-up

Technical details

Released

Wednesday, 23.04.2019 at 21:00

Deadline

Wednesday, 8.05.2019 at 21:00

Duration

14 days

Objective

This final sprint is the most fun one yet. You get to apply what you learned in the lecture yesterday — usable User Interface (UI) design!

The objectives for this sprint are as follows:

- Consider best UI practices for your program
- Implement best UI practices for your program
- Implement functionalities specified in the following pages

Deliverable

- Directory containing all the source code required for building the program
- Documentation for the project as specified in the Project Write-up
- README.md file describing all the improvements you made to the code base
- Report which includes annotated screenshots of your program assessing your UI decisions against Pressman's Golden Rules and what you learned in the lecture so far.

Implementation specification for Sprint 5 objective

Testing

We cannot stress enough how important testing is. In this sprint, improve your tests such that you achieve at least 50% code coverage as a thumb rule (i.e. you should have had a unit test for every function in your code anyways). Your TAs will use gcov and static analysis after the interviews to assess this.

Code improvements

Make improvements to the repository you received and list them in the README file.

Meaningful error messages

Implement a functionality which allows you to display an error message when an error has occurred (e.g. user clicking on a button when they are not allowed to or when a file is missing). Depending on how you implement your UI, you might or might not need this in some functions (e.g. if you hide unusable buttons you obviously won't encounter the error described previously).

Debugging tools

Upgrade the debugging tools to include the following functionalities (split them into different buttons):

- Increase intelligence currency by 10
    - Intelligence is capped at 200
- Increase marketing by 3
    - Marketing is capped at 30
- Increase pencil in inventory by 1,000
    - This will naturally increase the total number of pencils produced so far by 1,000 too (!)

Game closure

Please set your game such that it has a "game over" view once you have produced a total of 25,000 (25 thousand) pencils so far. You can include whatever you want in this view and the next possible actions after this view should also be determined by you!

Usable user-interface

Consider the best UI practices taught in the lecture and apply them to your program.

You are free to completely change the entire look of the program. This includes:

- The number of views (you can add on / remove views)
- Placement of things
- Literally everything else

However, your program must still retain all functions specified in the previous sprints:

- Sprint 2
    - Pencil production
        - Inventory
        - Make pencil button
        - Total number of pencils produced
    - Wallet system
        - Bank balance
    - Supply-demand based sales
        - Price of pencil
        - Increase price of pencil by $0.05. It is a clicker game afterall ;-)
        - Decrease price of pencil by $0.05 (!)
        - Public demand
        - Pencil sale
        - Crediting pencil sale revenue to your bank balance
    - Production material
        - Wood inventory
        - Graphite inventory
        - Buy wood (adds 100.00m of wood)
        - Buy graphite (adds 100.00m of graphite)
        - Price of wood
        - Price of graphite
    - Automatic Pencil Machine
        - Pencil production using APM 2000

        - Buy more of APM 2000
        - APM 2000 inventory
        - Price of APM 2000
- Sprint 3
    - Save and load
        - You need to adapt this to the new specification in Sprint 5
    - Intelligence currency
    - Upgrading the APM 2000
    - Marketing
    - Debugging tool
        - Split it into all functionalities (you must be able to debug each point individually, i.e. different buttons):
            - Grant players $10,000.00
            - Increase number of pencils in inventory by 1,000 (!)
                - This will naturally increase the total number of pencils produced so far by 1,000 (!)
            - Grant player 1,000.00m (!) of wood and granite
        - Don't foget to adapt the tool to the new specification in Sprint 5
- Sprint 4
    - High score board works
        - Grading note: this criterion is split into three sub-criterias:
            - POST score
            - GET score
            - Score board display
    - It is up to you how you want to implement the scoreboard (i.e. you don't need to have two views as specified in Sprint 4). You can merge everything into one view or split your program up into even more views!

Expectations for UI Implementation

You are expected to come up with a different UI from Sprint 4 where you base the design on our specification. This design was obviously not optimized for usability.

You are not expected to come up with fancy animations / illustrations as part of your UI implementation, but rather you are expected to demonstrate understanding of good UI principles. This exercise is supposed to get you to think about UI more in a logical way than an artistic way. There is no excuse for you not being an art student! Nevertheless, an aesthetically pleasing program is always appreciated, although will not necessarily be rewarded.

Some hints:

- Usable program flow
- Intuitive placement of items

Limits to creativity

There are no boundaries as to how much you can create/modify as long as you adhere to the functionality's requirement specified in this write-up. As a quick example, you can change the labels (most of you have been doing this anyways), font, add pictures, etc. You can play with the size of the window (e.g. fullscreen or not, scrollable or not) or decide whether to use pop-up or inline error messages. You might also want to implement a funnel as taught in class. Time to be creative!

If you are unsure about something, ask the TA by sending an email to:

Jin s.bien@jacobs-university.de

Mat m.chairani@jacobs-university.de

Report

In addition to a working program, you are also expected to produce a curated report with annotated screenshots explaining your design choices, based on ideas presented during the lecture (e.g.: Pressman's Golden Rule). Present your report as a PDF file.

Use these resources to base your arguments for the report and in your design considerations.

Link to UI slide: http://www.faculty.jacobs-university.de/pbaumann/iu-bremen.de_pbaumann//Courses/SoftwareEngineering/Slides/34_ui-design.pdf

Link to Design Pattern slide: http://www.faculty.jacobs-university.de/pbaumann/iu-bremen.de_pbaumann//Courses/SoftwareEngineering/Slides/36_web-design-patterns.pdf

Grading

To help you score the most points for this sprint, we include the grading scheme for Sprint 5 as a guide. You will be graded based on the following criteria:

- Game must build on TA's PC and will be graded as complete or not evident.
    - Additionally, if game does not build on TA's PC a penalty of -5% on the sprint grade will be incurred
- Testing as specified above with 5 discrete levels of score from complete (>= 70% coverage), above average (>=60% coverage), average (>= 50% coverage), subaverage (>=40% coverage), incomplete (<40% coverage).
    - Additionally, your tests must be meaningful. Try to break your code! This is an additional criterion with 5 discrete levels of grading from complete to not evident.
- Documentation (Doxygen) will have 5 discrete levels of grading from complete to not evident.
- Inline comments will be graded as complete, partial, or not evident.
- Each game functionality will be graded as complete or not evident individually.
- UI will be graded comparatively against other student's after all interviews have elapsed. The minimum acceptable UI (no evident changes) will receive a score of 60% for the criterion and others will be sorted accordingly. Score will be extrapolated from then on.
- Report will be graded and will be graded as either outstanding, complete, partial, or not evident.
- Improvements outside of the sprint objectives will be assessed by the TAs and will have 5 discrete levels of grading. List down all improvements ouside of the sprint objectives on your README file to make it easier for your TAs to give you credit. Please note that you should always make it a point to improve your code as much as possible — you will be handsomely rewarded for it.
- references.json file will be graded as complete, partial, or not evident.
- Late submissions will receive a penalty of -5% of sprint grade for every hour past the deadline. Maximum of -100%.
- If you are uncooperative during the sprint, you will incur a penalty between -50% to -100% for the sprint grade and your group partner will be graded more leniently.

Each criterion carries a different weight. As before, substantial weight will be given to improvements and unit testing. Grading scheme may be adjusted as deemed necessary by TAs at a later date.

BONUS: Multiple player sign-in (TBD)

Implement a functionality which allows you to:

- Create new player
- Type in new player's name
    - Your program should disallow duplicate names from your being used in your machine
- Delete player
- Save and load game associated to that new player

Note: Of course, after doing this, you will no longer extract the game-username from the environment variable to update the highscore board when you quit/save the game. You will still extract your jacobs-id and se-token from the environment variable.