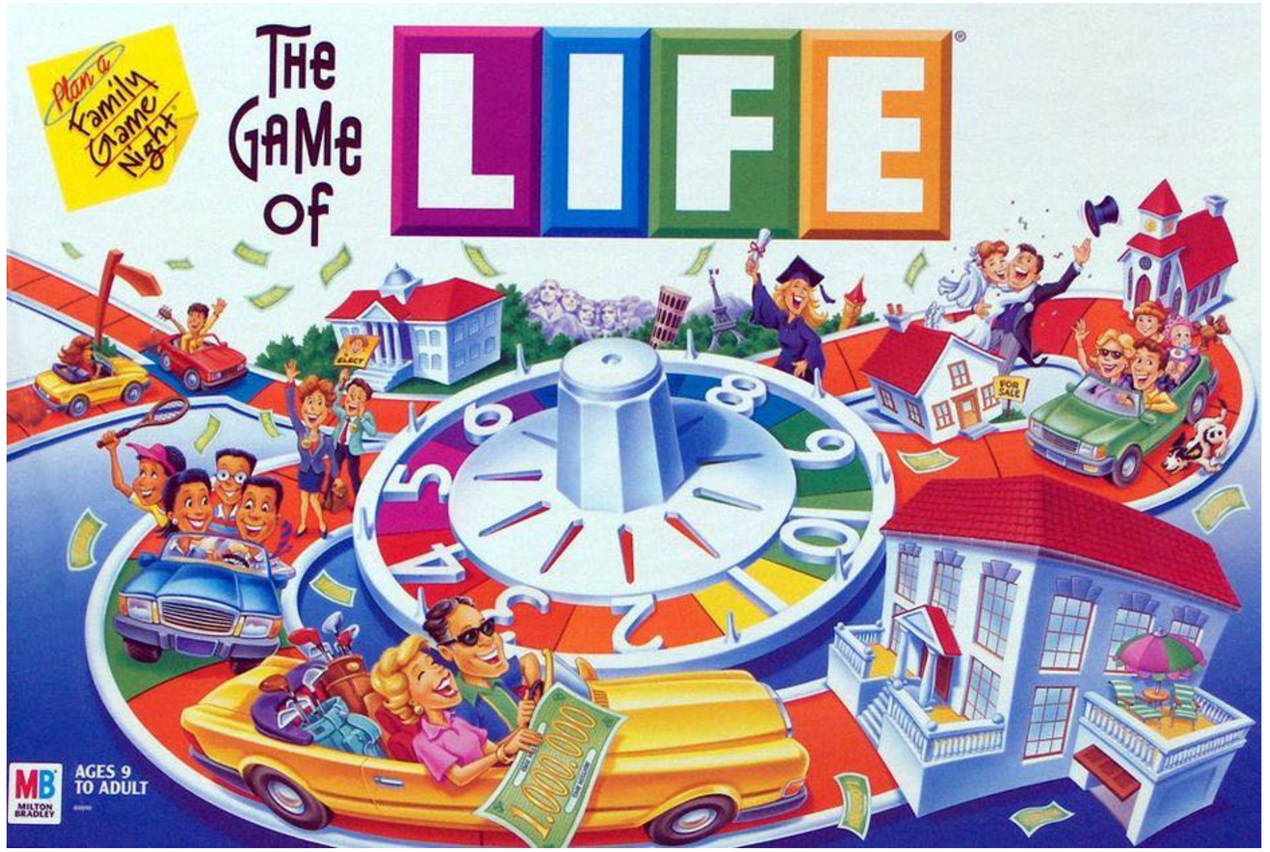
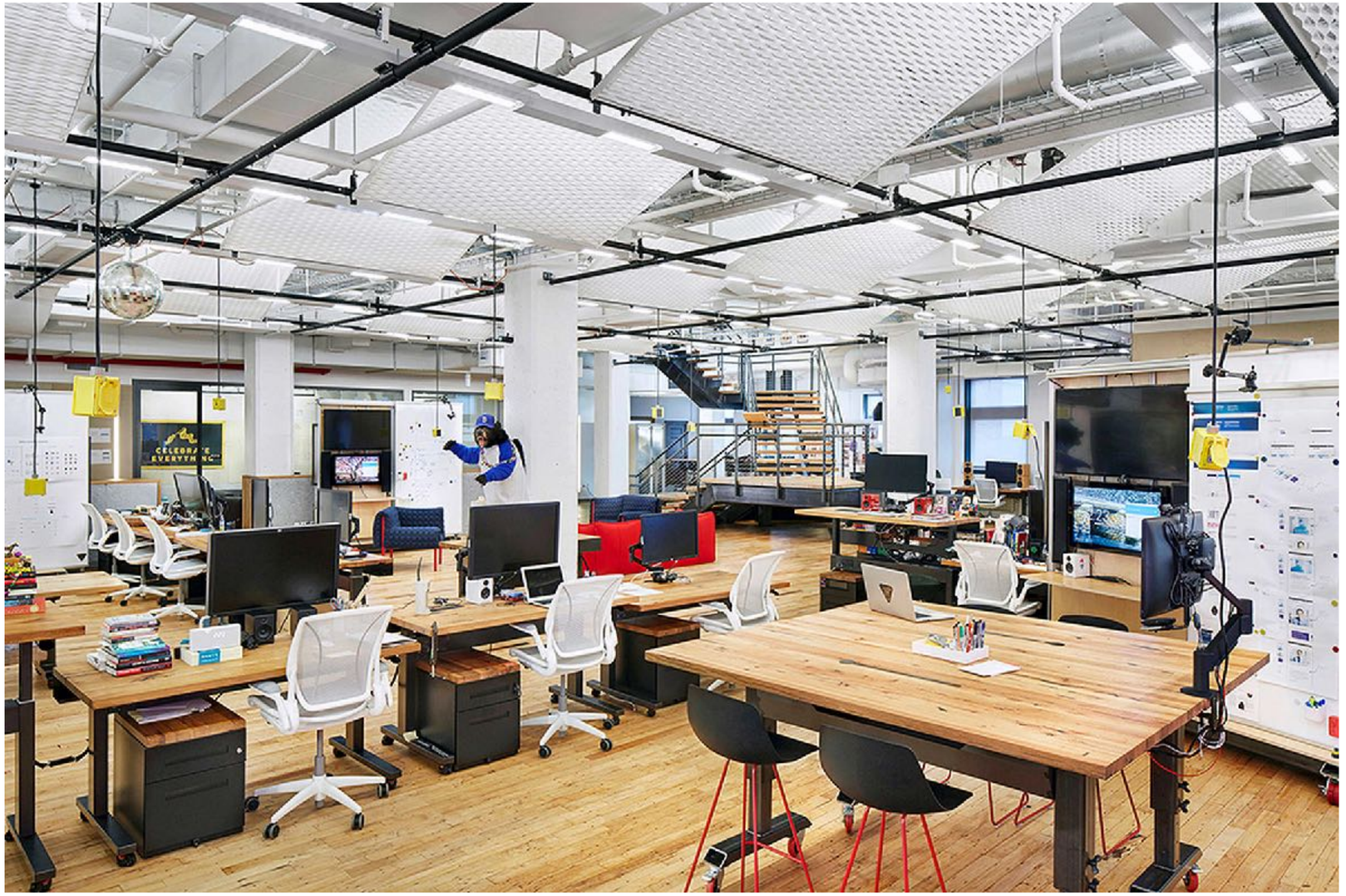
GRACE SHOPPER


Plan a Family Game Night
The GAME of LIFE
AGES 9 TO ADULT
MB
MILTON BRADLEY
FOR SALE



VICTORY CONDITION

Users can visit a deployed production site, browse a list of products, add them to a cart, check out, and make a payment with a credit card using Stripe, either as guests or as logged-in users

LOSS CONDITIONS

LOSS CONDITION #1

LOSS CONDITION #1

- sensitive data exposed or vulnerable to simple attacks

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:
    - emails

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:
    - emails
    - passwords

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:
    - emails
    - passwords
    - payment info

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:
    - emails
    - passwords
    - payment info
    - order history

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:
    - emails
    - passwords
    - payment info
    - order history
    - API keys

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:
    - emails
    - passwords
    - payment info
    - order history
    - API keys
  - simple attacks:

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:
    - emails
    - passwords
    - payment info
    - order history
    - API keys
  - simple attacks:
    - navigating to specific API routes in the browser

LOSS CONDITION #1

- **sensitive data exposed or vulnerable to simple attacks**
  - data:
    - emails
    - passwords
    - payment info
    - order history
    - API keys
  - simple attacks:
    - navigating to specific API routes in the browser
    - requests sent directly to the server w/ Postman or curl

LOSS CONDITION #2

LOSS CONDITION #2

- lack of teamwork

LOSS CONDITION #2

- **lack of teamwork**
  - unreviewed PRs merged to master branch

LOSS CONDITION #2

- **lack of teamwork**
  - unreviewed PRs merged to master branch
  - not all team members committed code to production site at every layer of the stack

LOSS CONDITION #2

- **lack of teamwork**
  - unreviewed PRs merged to master branch
  - not all team members committed code to production site at every layer of the stack
  - skipping daily standup

LOSS CONDITION #3

- no tests

BONUS POINTS

BONUS POINTS

- nice design / UI / UX

BONUS POINTS

- nice design / UI / UX
- extra features

BONUS POINTS

- nice design / UI / UX
- extra features
- continuous integration

BONUS POINTS

- nice design / UI / UX
- extra features
- continuous integration
- clean, readable code

BONUS POINTS

- nice design / UI / UX
- extra features
- continuous integration
- clean, readable code
- reusable components

BONUS POINTS FOR TEAMWORK

BONUS POINTS FOR TEAMWORK

- pair program

BONUS POINTS FOR TEAMWORK

- pair program
- use and maintain Github projects board

BONUS POINTS FOR TEAMWORK

- **pair program**
- **use and maintain Github projects board**
  - lots of small, isolated tickets connected to branches and PRs

BONUS POINTS FOR TEAMWORK

- **pair program**
- **use and maintain Github projects board**
  - lots of small, isolated tickets connected to branches and PRs
  - cleary-defined swim lanes

BONUS POINTS FOR TEAMWORK

- **pair program**
- **use and maintain Github projects board**
  - lots of small, isolated tickets connected to branches and PRs
  - cleary-defined swim lanes
  - kept up to date (you might get audited!)

BONUS POINTS FOR TEAMWORK

- **pair program**
- **use and maintain Github projects board**
  - lots of small, isolated tickets connected to branches and PRs
  - cleary-defined swim lanes
  - kept up to date (you might get audited!)
- **dedicated Slack channel integrated w/ Github repo**

BONUS POINTS FOR TEAMWORK

- **pair program**
- **use and maintain Github projects board**
  - lots of small, isolated tickets connected to branches and PRs
  - cleary-defined swim lanes
  - kept up to date (you might get audited!)
- **dedicated Slack channel integrated w/ Github repo**
- **clean commit history**

BONUS POINTS FOR TEAMWORK

- **pair program**
- **use and maintain Github projects board**
  - lots of small, isolated tickets connected to branches and PRs
  - cleary-defined swim lanes
  - kept up to date (you might get audited!)
- **dedicated Slack channel integrated w/ Github repo**
- **clean commit history**
  - lots of small commits focused on a single problem

BONUS POINTS FOR TEAMWORK

- **pair program**
- **use and maintain Github projects board**
  - lots of small, isolated tickets connected to branches and PRs
  - cleary-defined swim lanes
  - kept up to date (you might get audited!)
- **dedicated Slack channel integrated w/ Github repo**
- **clean commit history**
  - lots of small commits focused on a single problem
  - semantic commit messages

WHAT KIND OF PRODUCT?

WHAT KIND OF PRODUCT?

- your choice

WHAT KIND OF PRODUCT?

- your choice
  - (keep it clean)

Grace Shopper Takeaways

- Education (synthesis)
- Collaboration
- Planning / Agile
- Fun!

Logistics

- Teams of ~4
- Starts today
- Code reviews (2)
- “Presentation to management” (*us!*) next Wednesday
- Rotating team “roles”

STAND UP

A stand up a day
keeps the merge conflicts away

- **Goal:**
  - stay updated on each other's progress
  - organize efficiently (no double-work!)
  - remove blockers
- **Short (10 minutes or less)**
- **Each person in round robin style:**
  - What did I do yesterday?
  - What am I doing today?
  - Do I have any *blockers?*
- **Blocker: something *outside of your control* that is preventing progress on the thing you're doing today**

- **Alice:** “Yesterday, I refactored most of the product and orders routes, but there’s still a bit left to do. Today, I’m going to finish those routes. I don’t have any blockers.

- **Bob:** “Yesterday, I finished scaffolding the Orders page with dummy data. Today, I want to make the UI interact with the live data, but my blocker is that I need Alice’s backend changes to be merged before I do that.

- **Alice:** “Good to know - the orders changes are ready, it’s just the product routes that need fixing. I can make a pull request right after this, if you can review it.

ROLES

Taskmaster

- **Keep track of todos / issues**
  - Bite-size
  - In project management tool (Github projects)
- **Assign pairs / tasks each day**
- **Facilitate communication and minimize double work**
  - Make sure everybody knows what everybody else is doing

Gitmaster

- **Make sure everybody's `commit`-ing responsibly**
- **Make sure people are making pull requests**
- **Make sure people are reviewing pull requests**

Testmaster

- Make sure tests are being written
- YOU DO NOT WRITE ALL THE TESTS

Fellow === Project Manager

- **A simulation of “agile”**
- **Lead standup every day**
  - Discuss yesterday’s roles, assign today’s roles
- **Customized support for your specific team**

GIT'ING STARTED

- **Create a Github organization**
- **Add team, “project manager,” and instructors**
  - Give everyone **owner** privileges
- **Create a repo**
- **Push an initial commit**
- **Protect the master branch (Settings > Branches)**
- **Create a project**
  - Add at least four columns: *To Do, In Progress, Needs Review*, and *Done*

WRITING ISSUES

User Stories

- *As a <persona>, I want to <do something> so that <reason>*
- **Software should be designed based on what the users of that software want**
- **Stories !== implementation details**

Implementation Detail

- **How you will actually fulfill a user story**
- **Split each story into specific, bite-size tasks (implementation details)**
  - Should always serve a user story
- **Write a ticket for each implementation detail**
  - User story !== implementation detail
  - Associate each ticket with a GitHub issue

Bad Issue

```
We need a route to serve up all the products
```

Good user story (not an issue)

```
As a visitor, I want to see all of the products,
so that I can choose one to get more details or
add one to my cart right away
```

Good implementation detail (not tied to a user story)

An Express route on the backend (GET `/api/products`) should serve up the name, price, quantity, and imageUrl of all the products from our Postgres database. No special access control is needed, since any visitor should be able to receive this information.

Good Issue!

Story: As a visitor, I want to see all of the products, so that I can choose one to get more details or add one to my cart right away

Implementation: an Express route on the backend (GET `/api/products`) should serve up the name, price, quantity, and imageUrl of all the products from our Postgres database. No special access control is needed, since any visitor should be able to receive this information.

DOING WORK

Feature Branches

- **Never work directly on master!**
- **Always work from another branch (“feature” branch)**
- **Connect the branch to Github projects issues with issue number**
  - `git checkout -b <branchName>-#<issueNumber>`
  - `make sure to pull master first!`

MERGING TOGETHER

Pull Requests

- **Push your branch up to the remote**
  - `` `git push origin my-feature-#issue` ``
- **Go to your remote repo on Github**
  - Make a pull request (it should prompt you)
  - if you have conflicts when you make a PR you will need to merge master locally and fix the conflicts locally before pushing again
- **Someone else reviews your pull request**
- **If they request changes, make them locally and then push remotely again**

Merging

- Complete requested changes
- Merge the PR on Github
- Delete the branch

PRODUCTION

- **Deploy your app to Heroku**
- **Practice Continuous Integration**
  - Integrate code into master several times **EACH DAY**
- **…we will provide more guidance in a couple days**

TIPS

- Aim for uninterrupted focus
- Do not specialize (i.e. split on feature NOT tech)
- Rotate: pair with each member of your team
- Small, frequent PRs
- Protect master (e.g. review PRs)
- Don’t just give your opinion, ASK for other thoughts
- Challenge yourself!

SENIOR PHASE HELP TICKETS

- Can take longer and be more complicated
- Keep using the queue
- Do your research FIRST
- Beware when using technologies that aren’t part of FSA’s core curriculum