SA Presentations

By Carson, Kevin, Logan, Matthew Neel, Tracy Rountree

Systems Analysis Presentations

- What are they?
  - A demonstration of your plan for your project's game
- When are they?
  - February 14th

Requirements

- RFP (group)
- Champion Document (individual)
- Presentation (group)

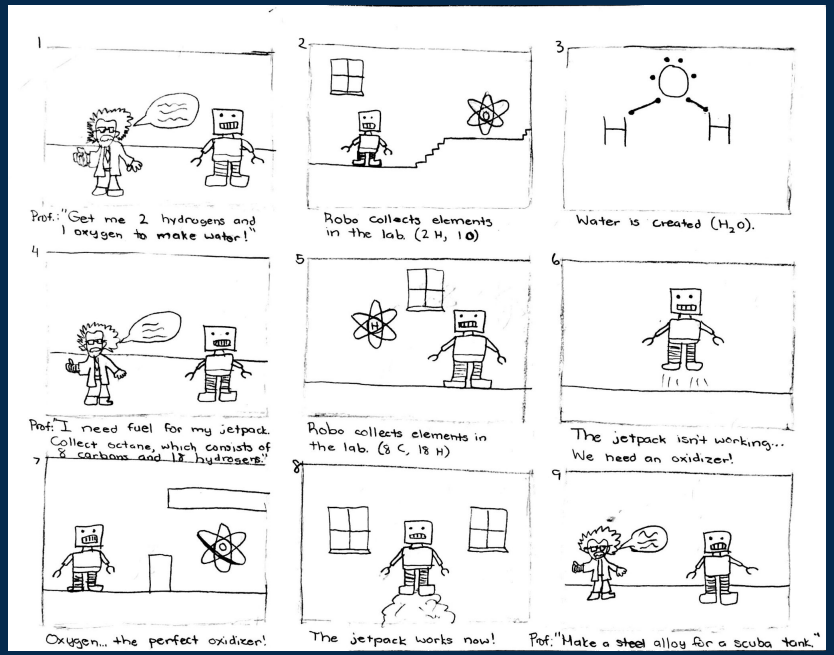
What is Storyboarding?

- A storyboard consists of an array of image panels with accompanying text
- A way of visualizing key points in your game
- Images/scenes are visualized in sequential order to convey the flow of the game
- A storyboard should simply convey the following:
    - What characters are in the frame and how are they moving?
    - What is the dialogue of the scene?
    - The position of the camera
    - Any other critical information that is necessary to the flow

Why Storyboard?

- By storyboarding, you are outlining a direction for your project
- You will be able to spot issues and brainstorm fixes before you've invested real time
- Since little technical knowledge is required, it's easy to get opinions and collaborate with others

Example Storyboard



https://sites.google.com/site/videogamedesigntsanh/home/storyboard

Super Mario Example

- Dialogue:
- Action:
  - User can select between 1 and 2 player mode
- Notes:
  - No music playing

Super Mario Example

- Dialogue:
- Action:
- Notes:
  - Display important information to the player

Super Mario Example

- Dialogue:
- Action:
  - Player is able to move the character to the right
- Notes:
  - (SFX) Overworld music begins to play
  - Time begins to count down

Super Mario Example

- Dialogue:
- Action:
  - Player jumps to interact with the block and avoid the goomba
- Notes:
  - (SFX) Block-hit sound plays

Get Good at Git

By Carson, Kevin, Logan, Matthew, Tracy Rountree

Why You Need Version Control

- Have you ever broken a project and had to spend hours reading through code to find where things went wrong?
- Have you ever wanted to work collaboratively with others on a project without screwing around?

What is Git?

- Popular Revision Control Software.
- Created by Linus Torvald in 2005.
- Allows for multiple people to work on the same project without worry.
- Nearly every operation is local.
    - Offline work is much easier with Git than other RCS systems.

What is Github?

- Git is not Github
- Github is a cloud-based hosting platform for Git repositories
- Github is owned by Microsoft
- Popular alternatives to Github include:
  - Bitbucket
  - GitLab
  - Self-hosted
- We have chosen to use Github in our demonstration later on in this presentation

How To Use Git

- Four popular methods:
  - Command line
  - Web client
  - Desktop application
  - Text-Editor/IDE integration
- We will later demonstrate how to use Git through VSCode

Basic Git Terminology

- Version
  - Shows the current version of Git installed.
- Init:
  - In order to use Git, you first need to initialize Git within your directory
  - Creates a “.git” folder which stores data about your project
- Commit:
  - Commits message to repository
  - Includes brief description for other users
- Clone:
  - To clone a repository is to copy it from a remote address to your local machine
- Status:
  - Shows all commits on the current branch.

Git Terminology Cont.

- Add:
  - Git will not automatically track changes to files. You need to directly add files to the Staging Area whenever you create or modify files
- Staging Area:
  - A preview of your next commit. Staged
- Revert
  - git revert <commit>
  - Go back (revert) to a previous version
  - “Undo” a commit, in this class, used primarily by TL 1
- Remote
  - git remote add origin <url>
  - Used to help set up git with a server
  - Popular examples: GitHub, GitLab, etc.

Git Terminology Cont.

- Stash
  - git stash
  - Nifty feature to save half-done work when you aren't ready to commit
  - Useful if you need to go to do quick work on a different branch, or quickly revert to a working state
- Branches
  - git branch <branch-name>
  - Create a new line of development
  - Do work on a different branch, and merge when you're ready
  - In this class, mainly used by TL 6
- Merge
  - git merge <commit>
  - Join two or more development histories together
  - git rebase works similarly but the way the commit history is kept is different
  - The way to squish two (or more) branches together into one
  - What happens if we work on both branches at the same time?

Git Terminology Cont.

- Divergent Branches
  - A feature of git that allows you to do work on multiple branches at once
  - Sometimes very easy to merge, “fast-forward”
  - Sometimes not so much
- Merge Conflicts
  - The same part of file has been modified in different ways on two branches
  - Generally have to simply choose what version you want to keep

Don’t Ignore .gitignore

- Some files don’t need to be included in your Git repo
  - Build files (*.o, etc…)
  - Cache files
- .gitignore templates can be found online
- Don’t be afraid to add to your .gitignore if the template doesn’t cover your project’s needs

Live Demo

Git Best Practices

- You should ideally never git push --force
- Each commit should have changes that are related
  - A function that prints a name is not related to a function that reads a file into memory (probably) and shouldn't be included in the same commit
- Make small incremental commits
  - Reduces the opportunity for merge conflicts to arise
  - If something breaks, it's much easier to identify where things went awry
- Write descriptive commit messages
  - Poor example: "Fixed bug"
  - Good example: "Fixed issue where camera didn't reset on player respawn"
- Have a consistent commit message style e.g.
  - Capitalize the first letter of the message
  - Start with a verb