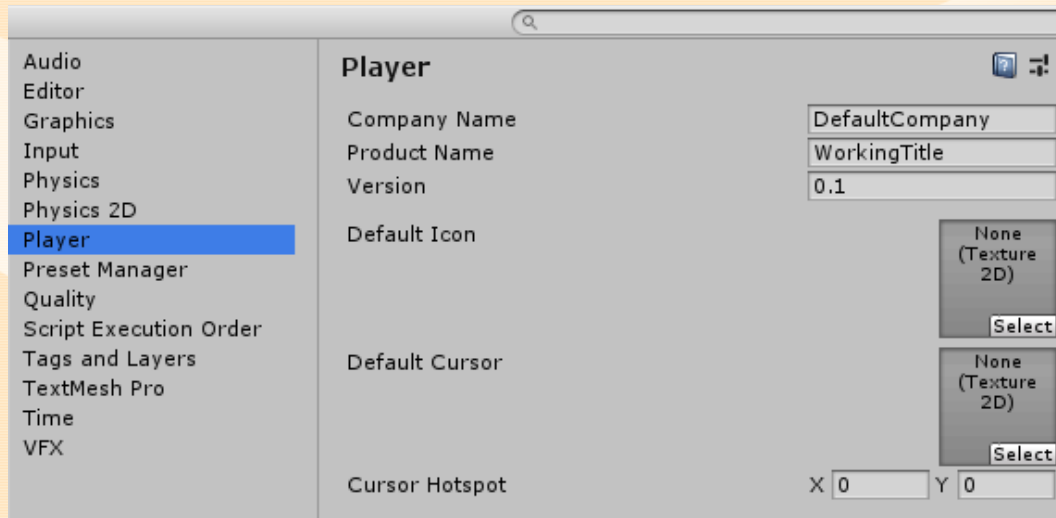

Target Platforms

Project Settings

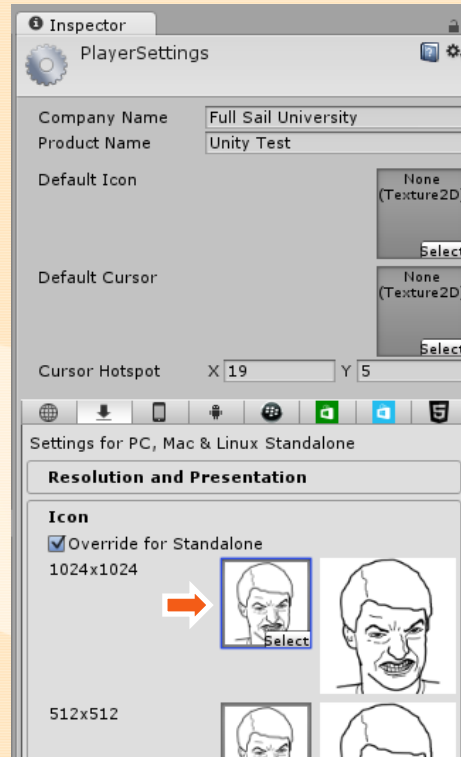
Set the projects settings that may have been ignored up until now like

- Company name
- Product name
- Icon



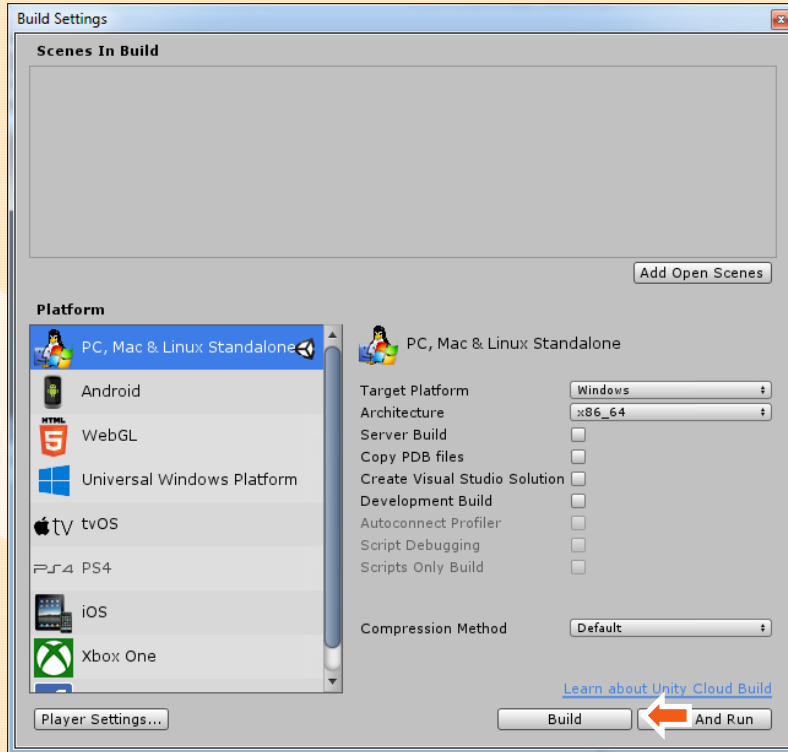
Icons

- Select to override icon for standalone build



Icons

- The next time you build you will have an icon



Installer

Installer

- There are lots of options for creating an installer
- Manually creating one inc#/c++
 - Not worth the time investment most of the time
 - Only needed for custom install processes
- Using a installer creator application
 - Fairly standard
 - Many are free to use even commercially

Installer

Installer creators are free and easy to use

- Click team installer creator
 - http://www.clickteam.com/install_creator-2
 - Wizard based
 - *The free version of Install Creator is fully functional for both commercial and non commercial products. Installers built with this version contain an extra page when the install is complete, linking back to Clickteam to spread the word about Install Creator.
- Inno Setup
 - <http://www.jrsoftware.org/isdl.php>
 - Script based
 - Completely free of charge, even when deploying commercial applications.

Installer

Make a clean build of the game for the installer

- Build into an empty folder
- Only include the files are needed to be installed
 - No code, no project files, no debug files...
- Create an installer for only the build of the game and the resources it loads

Installer

Test the installer before the milestone

- Make sure everything runs as intended
- Ensure it creates desktop and start menu entries for the game
 - Must not have to go to program files to find the game post install

WebGL

Platform support

- Making the build is the easy part
- Hosting adds complications
 - Read their limits and work within them

Itch.io

- 1GB total project
- 100MB per file

Newgrounds

- 1000MB
- Submit a maximum of two uploads per day

Simmer.io

- 512MB total project
- Does not accept compressed files
- Must be public on free accounts

Your own host option

- Completely DIY
- No limits

Platform support

- Make the links private/unlisted while we continue to develop
 - Don't want people playing and reviewing our games while we are still in QA

Prepping for Sprint Review

(end of lecture portion)

Sprint 4: Beta

Sprint Goals

- All assets are in their final, nonplaceholder form
 - (sprites, particles, animations, sounds, music...)
- Final game progress
 - (all levels, challenges, modes...)
- All target platforms functioning
 - (PC, web player, tablet)
- Various finalizing
 - Final Credits
 - Installer created
 - Verifying TRC compliance

Code Freeze

Code Freeze

In order to ensure the quality and integrity of the build we must go through a code freeze process

1. The entire team meets for final integration
 - General product development is halted
2. All work has been confirmed to have been integrated
 - Each team member ensures their master branch is fully updated
 - All work committed and merged, and each local copy fully updated
 - Peer checks on user stories have been completed
3. A build is exported and shared with the team
 - In the milestone build folder
4. Each team member completes a play through of the build and signs off that the build is ready to present

At that point, the master branch should be frozen

- No further code changes or development until unfrozen by the next sprint or release

Hold each other accountable

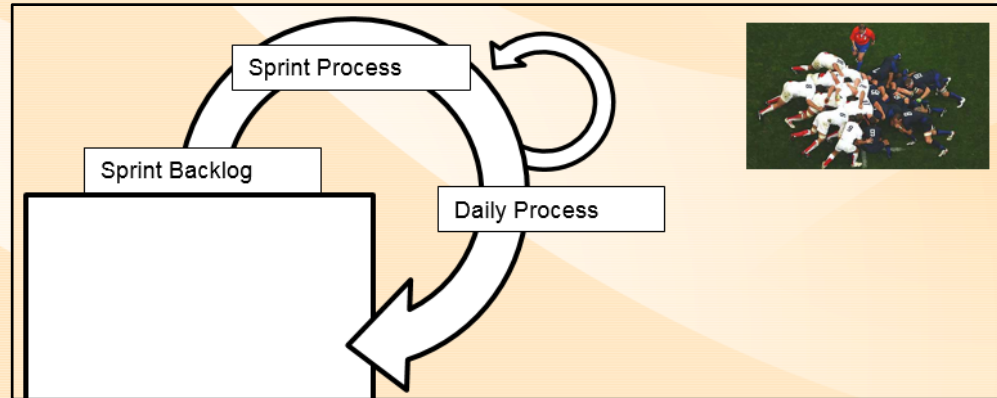
- Start the code freeze process no later than 5pm Tuesday.
- If someone is not done with their work, we need to make the build without that work.
- We need the build in the milestone build folder
 - Installer
 - Link to WebGL or APK file
- Everyone needs to be able to come into class at 9am and be productive.
 - Milestone review
 - Sprint planning

Daily Stand Up meeting

Stand up meeting

Keep everyone informed

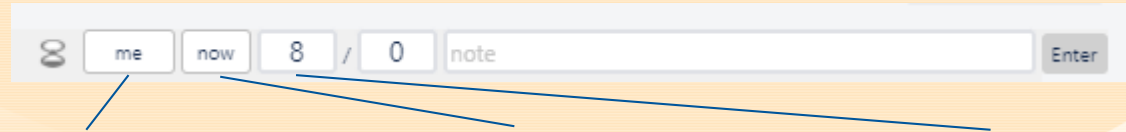
- What did you do?
- What are you about to do?
- What stands in your way?



Keep everything up to date

Update the task board (trello) while working

- Enter hours spent on any userstory worked on



A screenshot of a time entry form. It features a light blue header bar with a stylized '8' icon on the left. Below the icon are three input fields: 'me' (with a blue line pointing to it from the text 'Who did the work?'), 'now' (with a blue line pointing to it from the text 'When was the work done?'), and '8 / 0' (with a blue line pointing to it from the text 'How much work was done?'). To the right of these fields is a larger text input field labeled 'note' and an 'Enter' button.

Who did the work? When was the work done? How much work was done?

- The burn down chart will be updated automatically as hours spent get entered

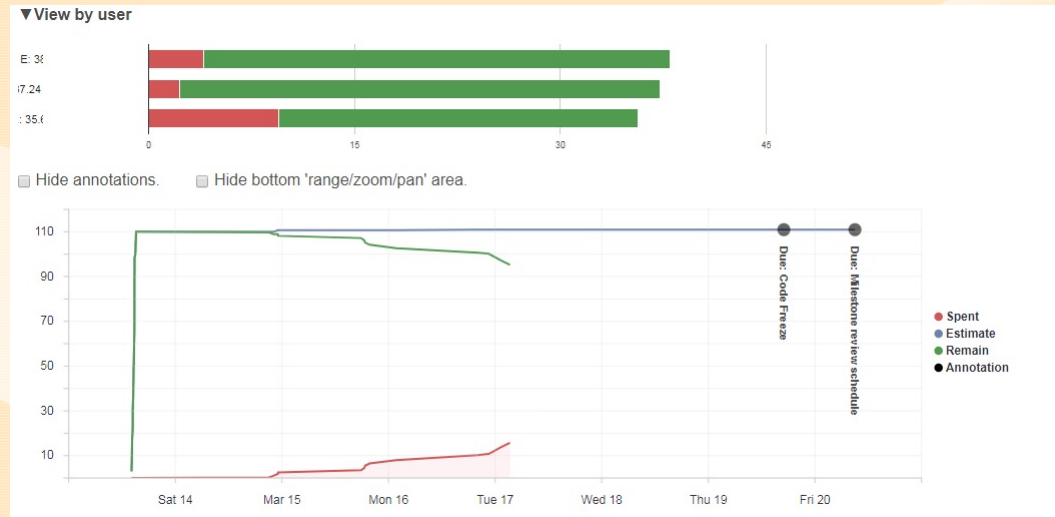


Track Progress:

Burn down/Burn Up charts

- Check your “velocity” versus the time remaining
- Fix issues when they only require small changes in work patterns
- Avoid needing the crunch at the end

This team is behind, needs to correct their work habits

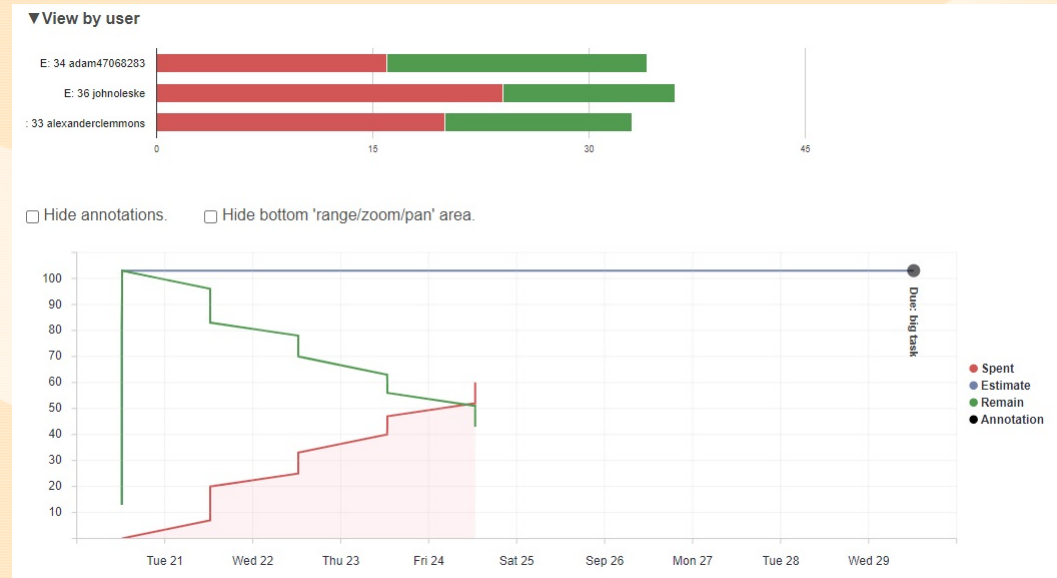


Track Progress:

Burn down/Burn Up charts

- Check your “velocity” versus the time remaining
- Fix issues when they only require small changes in work patterns
- Avoid needing the crunch at the end

This team is on track and should finish the sprint fine



Sprint 4: Beta

By the end of the day

- At least 21 hours logged per team member
- Burn down charts 3/4 complete

