UF UNIVERSITY of FLORIDA
Herbert Wertheim College of Engineering

Computer Engineering
CEN 4908C

M4: Beta Build

Overview

As a team, you will complete and deliver a ***beta build*** – the build of a project used for extended formal user testing, and which generally precedes a release candidate or preview release. This build will be used for fine-grained usability, stress, and/or performance testing. Typically, the beta incorporates all features, though some may be less polished than others. The beta build should represent approximately 105 hours of work per team member, or about 315-525 person-hours for the entire team (not including design documentation).

Specification

The beta build iterates on the alpha build, with the added expectation that all features should be integrated, and that lessons learned from testing with the alpha build have been used to iterate and refine the artifact.

Usability

The user feedback system, including any UI and sensory response elements, should be in place by this milestone. All elements of user control should be present, and they should be refined based on testing with the alpha build.

Interface

All interface elements (e.g., UI or API) should be accessible and usable. Options should be intuitive and consistent. Additionally, any persistent state must be easily accessible (e.g., highlighting of selected menu options, illumination of LEDs, and/or state API calls).

Navigation

All user mechanics should be functional. Any controls should be thoroughly tested. Bugs should be minimal, if present, and must not detract significantly from the overall experience. Any controls should not require significant time for acclimation; they should be predictable and easy to use. Features should be discoverable and well-explained.

Perception

The artifact must be intuitive to users. The artifact should help users attain stated goals with minimal frustration. Teams should have solicited user testing from non-team members to ensure general usability. Users should report an enjoyable experience. Any control interactions must be indicated by the sensory experience (visually and/or audibly). Changes in application state should be indicated on or near the relevant interface elements.

Responsiveness

The artifact must be responsive to user input. For any operation that requires significant I/O operations, the call should yield the CPU while awaiting response – i.e., there should be no busy wait loops. For APIs, if any call may have a delayed return, there should be a non-blocking option within the API call. Completion or failure of tasks must be clearly indicated to the user.

Build Quality

This build should endeavor to avoid bugs. All content should be well-integrated.

Robustness

By this milestone, crashes should not occur within the context of regular (unexceptional) use. Additionally, all edge cases should be tested, and bugs within edge cases should be rare. There should also be no major or noticeable glitches within regular or exceptional use.

Consistency

Except where explicitly otherwise by design, the system should act predictably, i.e., for the same input and use case, it should yield the same result. When and if behavior is unpredictable, it must be for a clear and compelling reason.

Aesthetic Rigor

No cosmetic software issues should be present. An aesthetic design for any physical artifacts should have been tested with users to refine its design. Aesthetic issues should be rare, but if present, should not cause the artifact to be difficult to use or unusable. All assets should be well-integrated and functional.

Features

All major elements should be complete and usable.

External Interface

All use cases must be implemented within the interface. This must connect to the *persistent state*.

Persistent State

All use cases must have use of data store functioning. This must connect to the *external interface* and the *internal systems*.

Internal Systems

All use cases must have data processing / handling implemented. This must connect to the *persistent state*.

Submissions

Your submission will include the following elements:

- Complete and detailed description of the project via a README and other documents
- Video demonstration of all project features, shared as an unlisted internet resource (<10 minutes)
- Link to repository holding all sources / schematics (shared with instructors)
- Any instructions necessary to navigate, understand, and build components of the system
- Outline of the work that was ***specifically completed in this milestone***
- Clear, unambiguous list of all known bugs

Note that failure to document bugs (by intent, mistake, or negligence) is grounds for grade reduction.