UF UNIVERSITY of FLORIDA
Herbert Wertheim College of Engineering

Computer Engineering
CEN 4908C

M5: Release Candidate

Overview

Preceding the final release milestone, each team will complete and deliver a ***release candidate*** – the build of a project following beta testing, usually used to catch last-minute issues and refine the build before release. This build will be used to identify and repair any remaining bugs in the system before its wide distribution. The release candidate is polished and ready for distribution but may contain hard-to-find bugs. The release candidate should represent approximately 130 hours of work per team member, or about 390-650 person-hours for the entire team (not including design documentation).

Specification

The release candidate is intended as a polished and repaired version of the beta build with all features integrated and other refinements based on lessons learned from testing with the beta build. All major elements should be complete and usable, including the **external interface**, **persistent state**, and **internal systems**. The release candidate should be, to the team's good faith belief, bug-free or nearly bug-free. Any bugs must be clearly identified in the documentation accompanying the build. There should be absolutely no "showstopper" bugs (those resulting in critical failure, loss of information, or non-functional elements) whatsoever this build.

Usability

The user feedback system, including any UI and sensory response elements, should be polished and complete. All elements of user control should be refined based on past experiences from alpha and beta testing. As with previous builds, standard expectations of usability apply:

Interface – accessible, usable, intuitive, and consistent; accessible persistent state

Navigation – functional, tested, discoverable, and predictable; meets user expectations

Perception – pleasant to use; usability-tested; provides sensory feedback; establishes persistent state

Responsiveness – responsive; yields computing resources; task completion / failure evidence to user

Build Quality

This build should have few, in any bugs; all content should be well-integrated. As with previous builds, standard expectations of build quality apply:

Robustness – no crashes; edge cases tested; glitches rare or non-existent; gracefully handles failures

Consistency – acts predictably; same input yields same result; unpredictable behavior only when by design

Aesthetic Rigor – No cosmetic issues; artifact tested / refined; issues do not prevent use; assets well-integrated

Submissions

Your submission will be text or markdown README file with the following elements:

- Complete and detailed description of the project
- Video demonstration of all project features, shared as an unlisted internet resource (<10 minutes)
- Link to repository holding all sources / schematics (shard 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.