

M3: Alpha Build

Overview

As a team, you will complete and deliver an **alpha build** – the build of a project used for the first round of formal testing. This build will be used for usability, stress, and/or performance testing in the next phase of the project. The features demonstrated will encompass a **vertical slice** – i.e., an iteration of your project that has all core features integrated in at least one dimension, though not necessarily polished. The alpha build will be iterated upon in the next milestone to produce the beta build. The alpha build should represent approximately 80 hours of work per team member, or about 240-400 person-hours for the entire team (not including design documentation).

Specification

The alpha build iterates on the Design Prototype, with the added expectation that a full set of vertical features should be integrated to facilitate testing before the beta build.

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.

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. 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. 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

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 major bugs. All content, including art and code systems, should be present.

Robustness

By this milestone, crashes should not occur within the context of regular (unexceptional) use. There should also be no major or noticeable glitches within regular 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 major cosmetic software issues should be present. An aesthetic design for any physical artifacts should be prepared and ready for testing. Any aesthetic issues that are present should not cause the artifact to be unusable. All assets (e.g., images and sounds) should be present and functional.

Vertical Features

All major architectural elements should be complete with at least one demonstrative implementation.

External Interface

For each major use-case, at least one variant must be implemented within the interface. This must connect to the *persistent state*.

Persistent State

For each major use case, there must be a demonstration of use of a data store in the system. This must connect to the *external interface* and the *internal systems*.

Internal Systems

For each major use case, at least one variant of the data processing / handling step must be implemented. This must connect to the *persistent state*.

Submissions

Your submission will be a PDF report with the following elements:

- Complete and detailed description of the project, with references to the above specification criteria
- Link to repository holding all design materials: source code, schematic/PCB files, and documentation
- Any instructions necessary to navigate, understand, and build components of the system
- Outline of effort, and report on progress 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.