UX Goals:

- Define framework and behavior of loading.
- Create a global framework that supports all loading scenarios.

Design Requirements:

- Acts as a visual representation of loading status.
- Communicates relevant information about level that is being loaded, to prep user.

UX Concerns:

- We want to keep layout consistent within each base loading framework. [Titles, Info, Animations, Loading..,etc should appear in the same spot]
- We want to know when the user has seen “training” and enhance their experience by giving them other relevant information regarding the level they are about to play (i.e. local leader boards, goals, tips, etc)
- We do not want to just drop the user into play if they are not ready. Since our experience has a lot of variables (it’s active, there is information to be gathered and it will be gathered at various rates depending on the user). Instead we will devise a method to let the user know the next level is ready and allow the user to choose when they are ready to proceed thru handles. Note: We can do decide not to use the handle as well. We also need to know how long to force the user to sit on the loading screen.
- We need a 4:3 Solution for the loading framework

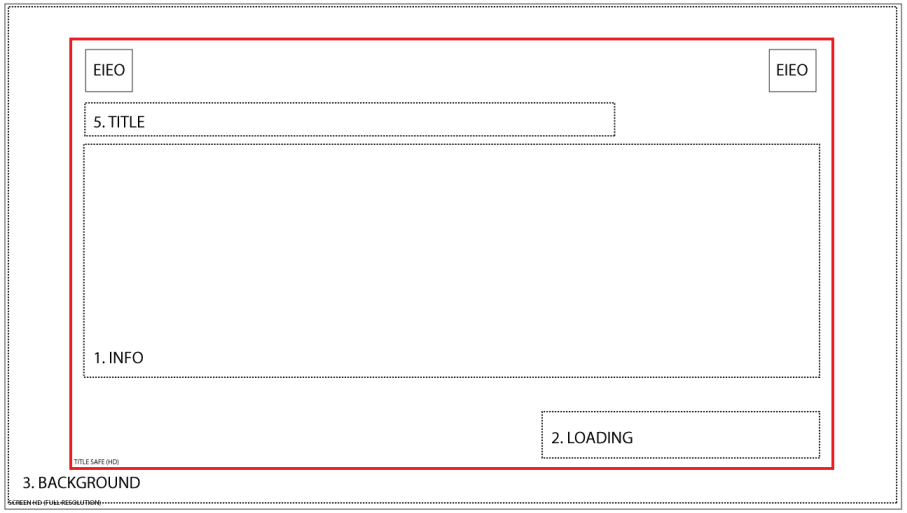
Base Framework Design:

We will have two base frameworks. In the beginning each framework will pretty much be setup the same. We wanted to leave some freedom for adventure loading to have a different layout, and possible different inputs, hence the separation.

- **Activity Based**
- **Adventure Based**

The following depicts the base framework design with all components laid out.

Note: This layout is subject to change based on art, the idea is to understand all the components that must be visible on screen and which components need to be within title safe. The Red outline indicates Title Safe. All items that have information that cannot be missed must reside within title safe. The only component that can reside outside is the background component. Components must remain in the same position within a framework.



We also support an animation layer which can have custom animation and art per level or activity or training video. This layer sits on top of the other layers and supports transparency. For example the signs that currently pop up in loading, would reside on this layer.

Understanding Each Component:

As pictured above you can see the Loading Framework is made up over multiple components. This allows us to use one framework that covers many scenarios. Each component has it's own inner workings and states and get's called into the main framework.

1. Informational Component: Displays relevant information to the user about the next level and support the following modes. We can determine what mode to load based on user status (i.e. has the user seen training already, if so show tips, stats], and game status [i.e. what is loading…show training related to that activity or show adventure instructions related to that adventure]

- **Training:** animated flash movies, with localized text.
- **Stats/Tips:** local leader boards, global tips [localized]
- **Adventure Instructions:** localized instruction pertaining to next part of adv.

2. Loading Component: Displays Loading Text and supports two modes. We can use either mode as we learn more from UR.

- **Text Only:** Loading Text [localized] Animated fades off [user dumped into game]
- **Handle Support:** Loading Text[localized] Animated turns into Ready Handle

3. **Background Component:** Displays static background image that takes up entire screen. [no modes needed]

4. Animation Layer Component: Displays animated graphics/text that needs to layer on top of all other components. It supports custom animation and art per level or activity or training video.

5. Title Component: Displays Level, or Adventure Title of currently loading activity.

- Display Level Title
- Display Adventure Title
- Display Adventure Title + Level Title