Technical Requirements Checklist

Summary

This workshop introduces the concept of compliance testing through the use of a Technical Requirements Checklist (TRC). A TRC is a list of tests that a game must pass before the platform licence holder will allow the game to be released on their platform. It is a vital process toward the end of a game's development lifecycle. We also discuss how to finish a game or a piece of work, so that passing TRC becomes more likely.

Introduction

At the end of a game's development cycle, the completed game goes through what is known as a *submission process* with the console manufacturer, or other platform holder. In order to ensure that the software is up to the expected standard, each platform provider has a detailed set of technical requirements which the software must meet to pass the submission process.

The name of this set of requirements varies from one console manufacturer to the next, but the term most often used within development studios is the *Technical Requirements Checklist* or TRC. This particular wording comes from Sony's submission process, while Microsoft swap the initials around a bit for their *Technical Certification Requirements* (TCR), whereas Nintendo enjoy being different with their *LotCheck*. There are also compliance checklists for pretty much every platform for games, such as Steam, Android Market and Facebook.

The TRCs are not focused on finding bugs in the game, they are focused on ensuring that the software works correctly within the constraints of the particular platform – examples would be how a game reacts when a controller loses power, how the console's name is presented on-screen, or how an app addresses privacy issues.

In general, a set of TRCs must be adhered to 100% - any failure to comply will result in the game failing the submission process, which can be a very expensive outcome for the developer and publisher.

Why do TRC's exist?

The reasons for platform holders imposing a strict TRC are two-fold:

- Protect the brand legally. Some TRCs are in-place for reasons of legality, privacy, or health and safety.
- Promote the brand. If a purchaser knows that a certain quality level is guaranteed, then a further purchase is more likely.

Some Examples

Typically, there will be close to two hundred TRCs which must be met in order to meet submission requirements on a home console. We won't look at a full list, but just pull a few examples to give an idea of the range, and detail, which must be covered.

Example 1: System Setting for language

> *If the title implements a method to obtain the language setting from the System Software and uses that language in game. If the System Software language setting is set to a language unsupported by the title, the title selects a language that it supports. Obtain the language used in the System Software with the cellSysutilGetSystemParamInt() function and carry out the appropriate operation as follows:*
>
> - *If a language unsupported by the title is set, have the user select one of the languages supported by the title, or set one of the languages as the default language.*
> - *If a language supported by the title is set, it is recommended for the title to use the same language also, unless there is a special reason to do otherwise.*

This first example, is a fairly straightforward one, which would usually be handled toward the end of a project, by ensuring that the necessary functionality, and menu screens, are included.

Example 2: Non-widescreen render resolutions

> *The title supports render resolution 480 (4:3) if it targets NTSC region, and render resolution 576 (4:3) if it targets PAL regions.*
>
> - *Wide-screen (squeezed) televisions 480(16:9) and 576(16:0) can also be supported*
> - *Both NTSC and PAL can be supported. However, consoles other than those targeting PAL regions do not support PAL output.*

The second example has wider consequences, so needs to be accounted for from the outset of the project, as it influences the development of the rendering technology, the frustum culling, and even the amount of the environment to be modelled.

Example 3: Disc read speeds

> *The title does not implement synchronisation processes that assume certain read speeds from the Blu-ray disc or HDD. Take into account that the read speed may change depending upon the condition of the disk drive and media. When streaming moves, always synchronise the video and audio using a time code, for example, instead of relying on the timing the data is read.*

Not taking this TRC into account early in development, could result in a very risky and time-consuming fix if the submission is failed, especially for open-world games which tend to access the storage medium almost permanently.

So what?

As a developer you need to be aware that the submission process is coming, and to prepare for compliance testing. Become familiar with the TRCs, especially the sections which are relevant to your work (e.g. if you are an online coder, or a HUD coder, there are significant numbers of TRC's directly related to your area of development).

Larger publishers increasingly assign a "closing team" to a major project toward the end of development, which will typically contain one or more TRC expert. However, as we have seen in the examples, not all TRC's are easily addressed at the end of the project, so relying on a closing team to fix all TRC issues is a dangerous plan.

During the Beta stage of a project, a coder's main focus is the bug database; TRC's are issues which absolutely must be addressed, so dealing with them earlier in the project means there is more time to concentrate on bug-fixing and balancing during Beta.

The console licence-holders often employ a "three strikes and you're out" policy, whereby all testing on a title will cease if three minor TRC failures, or one major failure, are found. This means that failing a submission may mean that large sections of the game have yet to be tested by the console license holder, making passing a further submission less predictable.

As a developer, the greatest thing you can do to increase the chance of passing the submission process, is to actually finish your piece of work. A good rule of thumb is that the first 90% of the work takes 50% of your time, and the final 10% of the work takes another 50% of your time. Ensure that all eventualities of player interaction are addressed, make sure that you have resolved any relevant TRCs, and don't expect the magic pixies to come in and finish your work for you.

Age Rating

We should also briefly mention age-ratings of games. The age-rating is a separate issue to the TRC. Most territories have a set of possible age-ratings that can be applied to a game (eg U, 12, 15, 18). The age-rating is awarded by a body, often government-funded, which reviews the finished game and awards a certificate depending on the content of the game. There is seldom a set of requirements that must be ticked off, to then ensure a specific age-rating – more often there is a set of guidelines which are used as the basis for the board's review and certification process.

Games must be submitted to the age-rating board for each country that the game is to be released in. There are unlikely to be any technical requirements attached to the age-rating, as the focus is on the game content, story, dialogue, artwork, etc.