SECTION 1 Defining Localizations

Introduction

The computer and video game industry continues to grow each year, and much of this growth can be attributed to the availability of international versions of the games. These versions include games localized for distribution into a variety of regions, including the United States, Europe, Asia, and the Middle East.

In order for developers and publishers to capitalize on these international markets, they need to develop a global mindset toward game development. If they lack the basic understanding of what it means to localize games for specific markets, publishers may lose sales. Or worse, publishers can spend time and money on a poorly planned localization that never gets released.

This section provides a general overview of what localization entails and how to think with an international mindset. Topics include:

- General Overview of the Localization Process
- Understanding Cultural Differences and Creating International Content
- Software Age Ratings

1 General Overview of Localization

In this chapter:

- Internationalization
- Localization
- Overview of the Localization Process

Introduction

Since international markets contribute heavily to a publisher's bottom line, publishers are paying more attention to ways they can maximize their sales abroad. One way they are doing this is by publishing international versions of their games that are translated into several languages. Most publishers automatically plan for French, German, Italian, and Spanish localizations, since these languages cover most of the European and Latin American territories, however, it is becoming increasingly popular for developers to translate games into Russian, Japanese, Korean, and Hebrew. While creating these international versions can be complicated and time consuming, publishers and developers are learning how to make this process more efficient by applying past experiences to their current processes.

International versions are often perceived as lackluster among gamers. An Italian gamer commented that he prefers to buy the original language versions of games (usually English), even if an Italian version is available because the source version often has a better quality of overall gameplay experience. He feels that all the elements of the source version—UI, story, voice acting, gameplay—mesh, while the Italian version often seems like an afterthought. He says that the Italian versions he has tried have typos, incorrect translations, text that is not translated, voice acting that is out of context, and a variety of other things that make the quality of the Italian version less than the original source version. This attitude demonstrates how important quality versions of international games are to the intended audience. Even though the international versions of the game have the same functionality and features, the gamer can be easily pulled out of the gameplaying experience if the quality of the localization is not good. Once developers are willing to think with a global mindset, quality localizations will be within reach.

Internationalization and localization are the two basic areas to consider when creating international versions of game software. *Internationalization* means creating a product that can be easily adapted for release in other countries without having to change the design of the product. This means the user interface (UI), control scheme, game content, and other areas of the game are designed to accommodate international versions of the game. For example, the UI screens are designed to accommodate traditional European text that is read from left to right, and Hebrew text, which is read from right to left. It also means that the various formats for dates, times, and currencies can all be accommodated in the game.

Localization is the actual process of translating the language assets in a game into other languages. For example, the text and voiceover assets are translated into French and German for release in the appropriate country, while other assets remain the same, such as UI, content, and characters. In comparison to internationalization, localization is straightforward because it involves altering just the language assets and not other aspects of the game. If these areas are fully understood, publishers and developers will be one step closer to creating quality international products and conquering the international market successfully.

Internationalization

The overall goal of internationalization is to create a project that can be easily localized with a minimum amount of work on the developer's part—the same game features, functionality, and gameplay experiences are present in all international versions of the game. International users should feel that the product was made specifically for them, and that they are getting the same gaming experience as the source language users. If internationalization is planned for properly during pre-production, these goals can be easily accomplished.

The main goals for internationalization of a game are designing a code base, core feature set, and UI that are generic enough to accommodate translations of any language. The code should support accented characters, international keyboard layouts, and international date and currency formats. For example, if a stock-market game is going to be released in the United States and Europe, the game should support international currency formats and properly balance currency conversions. If the game has a feature that allows the player to customize a character, the character editor should include a choice of different ethnicities and nationalities. Details like this go far with gamers and give companies an enhanced reputation for its quality of localizations.

Game Code

At minimum, the game's code base should support the ability to enter Latin-based characters and diacritics. Latin-based characters consist of the 26 characters from A to Z that are used to write languages for Europe, the Americas, and other parts of the world. Diacritics are accents placed under, over, or through Latin-based characters. Some examples of diacritic characters are Ş, ů, and ǽ.

The engine should not limit itself to ASCII text, which can only display 256 unique characters, but should instead support *Unicode*, which can display more than 65,000 unique characters, including Asian characters. To display Unicode, the engine must be double-byte enabled, since Unicode uses two bytes to display each character. The engine should also support the ability to display and accept input for bidirectional text, such as Hebrew, which is read from right to left, instead of left to right.

User Interface

The visual look and feel of the game gives users tangible proof that this game was designed with them in mind. The first sign to the user that the game has not been built specifically for them is a cluttered UI with overlapping and truncated text. Traditionally, translated text is about 20% to 30% larger than source language text, so if the UI is designed strictly for a specific language, the translated text in the UI will either be cut off or overlap in areas.

One way to keep the UI simplified and streamlined across all languages is to use icons. Icons cut down on the number of words that need to be translated and integrated into the UI and are generally universal in meaning. For example, an arrow pointing through a door to indicate "EXIT" is just as effective as translating "exit" into French, German, Spanish, or other languages. Additionally, this is often easier than finding a way to fit "AUSGANG," the German word for "exit," onto a button that barely has enough room for the word "exit." Be wary of which icons are used, since they might have double meanings or might not be meaningful to an international player.

Displaying Text

Early planning helps avoid common text display problems that are encountered when integrating translated text into the game. If the UI buttons, boxes, or columns are not scaled larger to accommodate translated words, the text will be truncated or overlap the area.

Truncated text can be misleading, since the user will not necessarily understand the context. An example of truncated text can be found in a beta version of *Tom Clancy's Ghost Recon®: Island Thunder™*. The UI screen on the English Xbox® version of *Tom Clancy's Ghost Recon: Island Thunder* has three fields on the right-hand side labeled "Shots Fired," "Total Hits," and "Total Hits %." These three fields display completely different player statistics.

When they were initially translated into French, all three fields were translated as "Total Tirs." If left as is, these terms would confuse and frustrate the player, who may wonder why this same field is displayed three times in the UI but has three different values. The problem with this translation is that all the information after "Total Tirs" was truncated. Once the translator recognized how the translations were displayed in the game and had a better understanding of how the statistics related to each other, he was able to come up with a creative and workable solution.

After seeing the translations implemented in the game, the translator saw that these three fields would always appear in the same order. Because of this, he labeled the first line "Total Tirs" and then indented the text in the second and third lines, which said "– Reussis" and "– Reussis %." With the way these columns were positioned on the UI screen, the player would read the field labeled "Total Tirs" first and see that the second and third fields were subcategories of the first field. With these changes in place, the player now understands the information presented on the screen.

Overlapping text is also something that should be corrected because it looks unprofessional and thus lowers the quality of the international version. The text can be difficult to read since it will run together with other text in the UI.

Many of these text issues can be avoided by having enough UI real estate to allow for the longer text. A general rule of thumb is that translated text takes up to 30% more space than source text. If the UI screens are designed with this rule in mind, many instances where the text is truncated or overlaps can be avoided. The buttons and boxes can also be programmed so that they scale up or down according to the size of the word.

During pre-production, a second set of UI screens can be mocked up with translated text. This will help identify worst-case scenarios of the text display so that solutions for problem areas can be identified and addressed before production begins.

Displaying Accented Characters

Another common text display problem is how accented characters are displayed. As discussed earlier in this chapter, game code limited to using the ASCII character set will not display accented characters correctly, or, in some cases, will not display the characters at all. When a character cannot be displayed in the code, a placeholder symbol will usually be seen instead. Including Unicode support in the code will prevent many of these character display errors. It is important to include support for both upper- and lowercase versions of accented characters. If not, words that should be displayed with all uppercase letters will instead be displayed with the accented letters in lowercase and the other characters in uppercase. For example, the German word "Bogenschütze" might be displayed as "BOGENSCHüTZE" in the game.

Consoles

If console games are being developed, it is important that the game engine supports both the NTSC and PAL video standards. NTSC is the video display standard for the United States and Japan. In this format, the video image delivers 525 lines of resolution at 60 half-frames per second. PAL is the video display standard for Europe, and the video image delivers 625 lines at 50 half-frames per second.

If the game does not support PAL standards, it will display incorrectly on PAL video monitors. If an NTSC image is displayed on a PAL video monitor, the image will appear to have black bars at the top and bottom of the screen because NTSC has 100 less lines of resolution. In addition, the image will flicker because a game running at 60 half-frames per second is being displayed on a monitor that can only support a refresh rate of 50 half-frames per second.

Additionally, the console developers Sony®, Microsoft®, and Nintendo® each have specific technical requirements, since their game standards differ in Asia, Europe, and the United States. These requirements must be addressed in all international versions.

Cultural Context

Another internationalization issue to address in pre-production is the cultural context of the game. Since many games are developed with English as the source language, developers sometimes gear all text displays and cultural choices to English. For example, developers will display dates in the American format of month, day, year and will not include support for the European format of day, month, year. International display formats also differ in regard to time, numbers, and currencies.

In addition, when designing the game, culturally specific references such as the name of a popular movie star or well-known TV show should be limited, unless it is essential to the story or gameplay. For example, the game can refer to "a famous star" instead of "Tina Fey," since she is not as well known outside the United States. If the game remains as culturally neutral as possible, it is less noticeable to an international gamer that the game was developed primarily for a specific demographic.

If end users are convinced that the international versions were planned for them from the beginning, they will be satisfied that they are getting the same game experience as someone who plays the source version of the game. If the game code is developed with an eye toward international versions, the actual localization process where the assets are translated, integrated, and tested will go smoothly.

Avoid Retrofitting

Retrofitting a game to be localization-friendly after it is developed involves more time, money, and resources. If the publisher decides to do a Spanish translation, it is better if the game already supports the necessary accented characters in both upper- and lowercase. Adding Spanish character support after the game code is complete will likely entail additional engineering and testing time. Depending on the state of the code, this can be time-consuming and expensive, especially if the fonts, text display support, and keyboard inputs are scattered throughout the code. Additionally, if Spanish language support is not added at the outset, it would be difficult to allow for game compatibility between the source language and Spanish versions.

Localization

Localization is the process of translating the game into other languages. If the product has been properly internationalized, the game will not need to be redesigned or have additional features added to accommodate the translations. This makes the actual localization process fairly painless. Creating a Japanese or Russian version is much easier if the game already displays accented characters, accommodates international keyboard layouts, and does not have culturally-specific references to correct.

The extent to which game assets are localized can vary from project to project, depending on how many resources are available to invest in the localization and the likely return on the investment. More gamers are likely to buy a game that is localized specifically for their native language. If the game is only available in one language, that same gamers might not purchase it if it is not in their native tongue, resulting in a direct sales loss. However, the extra work required for localizing a game means additional risk for the publisher since more money is required. Other localization risks include not selling enough copies to break even on the localization development costs, missing a key ship date (like Christmas) because the amount of work was underestimated, or finding a critical bug that cannot be fixed without adding additional time and resources. One way to minimize these risks is to scale the localization process according to the needs and expectations of the game.

No Localization

Not localizing the game at all requires the least amount of time and resources. The publisher ships the original language version in the original packaging directly to the international markets. The main drawback of this is that the game has not been personalized for any international markets.

Games with small development budgets and schedules—sometimes called "budget" titles—are usually shipped directly into international markets without being localized. The international market is often seen by the publisher as an opportunity to sell a few extra copies of a game with little extra investment. This level of localization is the lowest risk for the developer since no additional work is needed.

Packaging and Manual Localization

Localizing the game's packaging and manual, commonly referred to as "box and docs," is another level of localization. The game code and language are unchanged from the original version, but the manual, packaging, and other supporting documentation are localized into the target language. "Box and docs" localization is typically done for a game that is not expected to sell more than a few thousand copies in other countries. Some games that are big sellers in France and Germany sometimes receive this type of treatment if they are being sold in smaller secondary European markets like Sweden or Denmark because they are not expected to sell a large number of copies.

"Box and docs" localization is low risk for the developer since no game code is altered. However, the developer might have to assist the translator with understanding the game-specific terms to be translated for the packaging. Additionally, on PC versions, the developer may have to double-check the functionality of international keyboards to ensure the source language keyboard commands are carried over correctly to international keyboards.

The disadvantage of this method is that the end user is playing a game that is not translated. A localized manual does help because the user is able to fully understand how to play the game, but the experience may not be as immersive for the player. A big advantage of a "box and docs" localization is that all language versions can be shipped simultaneously with the source language version.

Partial Localization

A partial localization means that only the in-game text is translated, not the voiceover. This method is cost effective, since time and money are not spent to translate the extra voiceover text, set up recording sessions, hire actors, process the sound files, and complete other tasks necessary for localizing voiceovers. In some cases, the voiceover files can be subtitled, but only if the code supports this feature.

Additionally, properly integrating localized voiceover files into the game can be challenging, especially if lip-syncing is used for the in-game characters. When lip-syncing is used, facial animations might have to be redone to get the highest quality product.

A text-only localization is riskier because it involves altering the actual game code, which means more development and testing time. The text assets also need to be closely tracked to make sure any text changes in the source language version are transferred over to the localized versions. Since more people are involved and the game code has to be altered, this type of localization costs more money and has a greater chance of not being completed on time. The benefit is that the international gamer gets a more personalized gaming experience. Partial localizations are usually created for quality games released in secondary markets such as Holland and Italy.

Full Localization

A full localization includes translating the text, voiceover, manual, and packaging. This is the most expensive and risky localization and is usually reserved for big-budget games. Often, a smaller team within the main development team will be charged with the completion of the localizations, since full localizations are time-consuming. This team works closely with the main team to organize the assets for translation, integrate the assets into the game, and coordinate the localization testing. Every aspect of the game has to be thoroughly reviewed to make sure that all the text and voiceover files are localized. This can be costly and challenging if the game code is not localization-friendly and the assets are not well organized within the code. In cases like this, the developer will spend a lot of time hunting down the assets and formatting them for translation.

Since a full localization is dependent on the progress of the original version, the localized versions will be late if the original version is delayed. This can interfere with the original language version because the luxury of making text or voiceover changes at the last minute is removed. Ideally, the localizations should begin early in the development process to make sure all the assets are properly translated.

The biggest advantage of doing a full localization is that the player can buy a game that is fully tailored to his language-specific needs. Full localization shows the player that the publisher is committed to providing the best quality gaming experience for its international customers.

■ Overview of the Localization Process

The localization process can go smoothly if preparations are made and followed every step of the way. The three main phases of localization, as presented in this book, are planning, producing, and concluding localizations. This book gives a detailed explanation of each phase in its respective section. A brief overview of what is included in each phase is presented here.

Overview of the Planning Phase

During the first phase of localization, when the game is still in the planning and pre-production stages, the developer and publisher stand to reap the most benefits by careful planning. Take time to assess the expectations of the game engine, gameplay design, UI design, and asset organization within the code. Additionally, work with the sales and marketing departments to help determine which level of localization is appropriate for each language. This is often based on historical sales data for past international releases. If sales and marketing can determine their language needs up front, the developer can plan more effectively during pre-production and production. Some decisions to make at this phase include:

- Will the engine support Unicode?
- Will formatting for international currency, date, time, and number formats be supported?
- How will the assets be organized in the game code?
- Does the game design contain culturally-specific references or slang? If so, are they necessary to gameplay?
- Will subtitling functionality be necessary?
- How will lip-syncing be handled for localized versions?
- Will PAL support be necessary for console versions?
- Will the UI be scalable to accommodate longer translations?
- How will the asset-tracking pipeline be organized?
- What level of localization is necessary for this game?
- How will version control be handled?
- What resources are needed to complete the localization?
- Will any proprietary tools need to be created for the localized versions?
- Will localizations be produced by the development team or by external vendors?
- Which localizations will be shipped simultaneously with the source language version?

Overview of the Production Phase

In the second phase of localization, the developer is in the process of producing the localized versions. This entails having the assets translated, integrated, and tested as efficiently as possible. If the planning phase of localization is completed successfully, the producing phase is fairly straightforward. This phase focuses on the specific tasks of organizing assets for translation, integrating them into the game, and testing the localized versions. Some decisions to make in this phase include:

- When will final assets be ready for translation?
- What methods will be used to organize these assets for the translators?
- What game documentation will be necessary for translators to understand the context of the assets they are translating?
- What methods will be used to get the assets integrated into the game?
- Who will be responsible for integrating these assets into the game?
- How will functionality and linguistic testing be handled?
- What communication pipelines need to be established to make sure all participants fully understand the localization process and the status of the localization?
- Who is responsible for determining when the localized versions are ready for code release?
- What is the code-release deadline?

Overview of the Post-Production Phase

In this phase, the game code is localized and released and it is time to concentrate on finalizing the other localization items such as manuals, packaging, and demos. This is also the point where a localization kit is created to ease the production of future localizations.

At this point, marketing and sales will likely start asking for additional deliverables such as high-resolution artwork or localized screenshots. If this phase of localization is completed without any major problems, a well-thought-out and efficient localization has been achieved. Some decisions to make in this phase include:

- Will localized demos need to be produced?
- When do the localized console demos need to be submitted to third parties for approval?
- How will changes for the manual be tracked and communicated?
- What other sales and marketing assets will need to be produced?
- What is included in the localization kit and when is it due?
- Are there any last-minute issues that need to be addressed before the game ships?
- If the game needs patches, how will the localized patches be handled?

Localization Overview Checklist

Figure 1.1 is a localization overview checklist—a helpful reminder of which items should be focused on during each localization phase. The checklist presents key tasks to complete for each localization, but does not break them down in detail. The details will be worked out as the localization process is established for each project.

PLANNING PHASE CHECKLIST	Y/N	NOTES
TECHNICAL CONSIDERATIONS		
Does game support Unicode?		
Are all language assets in an easily accessible directory in the game?		
Will subtitling functionality be needed?		
Are localized keyboards supported for player input?		
Will several languages ship on a single disc?		
Will localized versions be multiplayer compatible?		
Do boxes in the UI scale to accommodate different size text strings?		
Is any additional software needed to aid in localization?		
Are international currencies and date/time formats supported?		
Has a version control system been decided on for the localizations?		
Has the localization pipeline been decided on?		
OTHER CONSIDERATIONS		
Will the localized versions ship simultaneously with the source language version?		
Has the asset overview form been filled out and sent to the translator?		
Have the languages been determined?		
Will external vendors be producing the localizations?		
If so, are the bid packages prepared?		
Has the budget been completed and approved?		
Has the level of localization been determined for each language?		
Has the overall schedule been completed and finalized?		
Are there development resources available for the localizations?		
Has a method for integrating text assets been determined?		

FIGURE 1.1 Localization overview checklist. *(continues)*

PLANNING PHASE CHECKLIST	**Y/N**	**NOTES**
Has a method for integrating voiceover assets been determined?		
Has a pipeline been determined for fixing bugs?		
Have the appropriate measures been taken to comply with all of the international ratings boards?		
Have the third-party publishers been contacted about the localized versions?		
Will PAL support be necessary for console versions?		
Is there enough hardware for functionality and linguistic testing?		
CREATING PHASE CHECKLIST		
Has a detailed schedule been completed and communicated to the team?		
Has the localization overview document been sent to the localization coordinator or translators?		
Has all the pre-production game documentation been sent to the localization coordinator or translators?		
Has the latest build of the game been sent to the translators?		
Have the text assets been organized for translation and sent to the localization coordinator?		
Have the voiceover script and character casting notes been sent to the localization coordinator?		
Have the final source language voiceover files been sent to the localization coordinator?		
Have all the art assets to be localized been sent to the localization coordinator?		
Have all the cinematic assets and time codes been organized and sent to the translator?		
Are the translations for the text assets completed?		
Have the localized voiceover files been recorded and processed?		
Have the text and voiceover files been integrated?		
Have the cinematics been localized?		
Have the localized versions been sent to the appropriate ratings board for approval?		
Does the master contain demos from other games that were requested by marketing?		
Is functionality testing completed?		
Are all functionality bugs fixed and has the game been code released?		
Is linguistic testing completed?		

FIGURE 1.1 Localization overview checklist. *(continued)*

PLANNING PHASE CHECKLIST	Y/N	NOTES
Are all linguistic bugs fixed and has final linguistics approval been given?		
Have the localized versions been sent to the replicator (PC) or submitted to the third-party publisher (consoles and cell phones)?		
POST-PRODUCTION PHASE CHECKLIST		
Have the manual and box text been sent for translation?		
Does a localized demo need to be produced?		
Have localized screenshots been taken for the manual and box?		
Has a closing kit been created for all the localized versions?		
If necessary, have all patches been localized and made available?		

FIGURE 1.1 Localization overview checklist. *(continued)*

DEVELOPER INTERVIEW

Tom Sloper, President, Sloperama Productions

Mechwarrior®, Thunderbirds®, Blast Chamber®

I've localized literally dozens of titles on practically every computer and console platform. The most interesting ones were Thunderbirds (NES®), Mechwarrior (SNES®), Blast Chamber (PlayStation®), Shanghai Dynasty® (Windows®/Mac®), and Shanghai Pocket® (Game Boy Color™). On the above-mentioned console titles, the localization was managed by one producer in the United States but performed in Japan or Australia by the original developer of the game. On the PC titles, producers in the Japanese and American offices cooperated to make the localization, which was tested in both countries.

None of the titles mentioned shipped simultaneously. The console titles were shipped first in Japan by their Japanese publishers, then were licensed for localization and released in the United States. The PC title was shipped first in the United States, then localized for other markets. Some of the games had minor content changes before they were released in some international markets. For Mechwarrior, special mechs had to be created for legal reasons. For Blast Chamber, the title had to be changed.

The thing that's so interesting about the Shanghai Dynasty localization is the extent of the extra functionality that was added for the Japanese market. Shanghai Dynasty wouldn't have released in time for Christmas if the full feature set was implemented before release, so an executive greenlight committee made the decision to release the U.S. version first, then create the additional Japan-specific features, and make the new features available to U.S. players as a patch. Shanghai Dynasty included the classic table game of mah-jongg, which is played differently in various countries. The Japanese release needed to include the Japanese variant of mah-jongg. That was a challenge to design, since at the time there were no English-language resources on Japanese mah-jongg. A local expert was brought into the project.

Since Shanghai Dynasty was going to be localized into Japanese, as well as European languages, all the game text was converted to Unicode format in order to properly display Japanese text. Converting to Unicode also permitted online players of all languages to view properly displayed text of all other languages. To test multiplayer compatibility between the various languages, localized versions of the game were installed in the test lab and players went online and typed messages in the game.

The Japanese voiceover recording for Shanghai Dynasty was handled by the Activision® office in Japan. They translated the script from English, hired actors in Tokyo, and recorded and processed the voiceover files. That process was fairly transparent to the American production team. Since no faces or mouth movements were visible in the game, it was a simple matter of straightforwardly substituting the voice assets. Shanghai Dynasty was very localization-friendly because we planned for international versions in the original design document.

I have run across some interesting situations in creating localized versions of games. I can mention a story that happened with Shanghai Pocket. The Japanese licensee, Sunsoft®, had created a game for sale in Japan. Sunsoft's American office would sell the game in the United States. My job was to approve the design and the final product since Shanghai was an Activision intellectual property (IP). I don't read Japanese, so when it came to approving the story for the Japanese version, I asked Sunsoft for a translation. The translation wasn't objectionable, so I approved it. Then when it came time to approve the English version, I asked for the English text—they told me, "you already approved it." I said I had only approved the translation of the Japanese text—the text that would work well for American players in the context of the game was another matter. They furthermore told me that I could see the text working in the game by entering a special cheat code at the title screen. This was something they'd neglected to tell me before. In playing the game, then, I found that the complete translation (which was actually quite wordy) existed verbatim in the game's tiny little text display area. Many presses of the A button were required to read even sentences of average length!

In order to make the text more palatable, I found it easier to rewrite it myself than to tell them to do it. I wrote the text in very short couplets (possibly in iambic pentameter, but what do I know of poetry terminology?) that could be read with the fewest possible A button presses per story segment.

Here's another story. The project was Thunderbirds NES. I knew that the game was based on a TV show involving puppets or something. I hadn't followed the show, so I turned out to be the main problem in that localization. The Japanese company, Pack-In Video, had given me a translation of the game text, which included translated character names. One character's name had been translated from English into Japanese and then back into English as "Fud." Since I didn't know the TV show, I didn't know what this name was supposed to be. I figured "Fahd" to be a likely choice, but it turned out that "Hood" is a recurring villain in the series! If my localized text showing "Fahd" had been implemented in the game, that would have been a problem for all the Thunderbirds fans who bought the game!

Chapter Summary

Since publishers rely on international versions of their games to increase worldwide sales, they look to developers to create games that can be quickly and easily localized. The developer can improve his track record for creating international versions by outlining a plan for internationalization and localization during pre-production. This process must include solutions for designing the UI, displaying text, and implementing gameplay features that can be reused without any changes in the international versions. If these items are done with the international versions in mind, the developer can focus all efforts on getting the game assets translated, integrated, and tested.

This chapter provided a good overview on what items to consider when creating international versions. Information on whether a full or partial localization is appropriate for the game was presented, along with key decisions to make about the overall localization process. The next chapter builds on this base by discussing issues to be aware of when developing games for international markets, including the culturalization of game content.