

Práticas recomendadas para localização de jogos

Versão inicial por Richard Honeywood, vice-presidente

Grupo de localização de jogos (SIG)

Associação internacional de desenvolvedores de jogos (IGDA)

1º de fevereiro de 2011

Segunda versão por Jon Fung

1º de fevereiro de 2012

Esta é a segunda versão do guia de "práticas recomendadas" para a tradução e culturalização do conteúdo de videogames. É uma compilação de sugestões de pessoas com anos de experiência na área, todos membros do grupo de localização de jogos da IGDA. O objetivo é ajudar os iniciantes a aprender, assim como oferecer dicas e truques que até os mais experientes possam adaptar em projetos futuros.



Best Practices for Game Localization by Richard Honeywood, Jon Fung, Ana Rodrigues, Bella Cardoso de Carvalho, Carolina de Góes, Debora Weirich, Diego Hernandez, Paulo Noriega, Sheila Gomes and Thomas Melo is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

ÍNDICE

CULTURALIZAÇÃO	4
O que é "culturalização" de um jogo?	4
Níveis de culturalização de um jogo	4
Quatro variáveis culturais principais	5
Práticas recomendadas de culturalização	6
INTERNACIONALIZAÇÃO	9
Interface de usuário	9
Arquitetura	11
Programação	14
Conselhos para equipes de desenvolvimento	26
LOCALIZAÇÃO	27
Familiarização	27
Criação de glossários e guias de estilo	28
Tradução	29
Gravação de <i>voice over</i>	30
Controle de qualidade linguístico	33
Masterização e finalização	34
PLANEJAMENTO DE PROJETO	35
Pré-produção	35
Produção	35
Estágio alfa	36
Estágio beta / finalização da localização / cópia final	36
APÊNDICE 1 - LISTA DE PRÁTICAS RECOMENDADAS	37
APÊNDICE 2 - EXEMPLOS DE CULTURALIZAÇÃO	41
APÊNDICE 3 - EXEMPLOS DE <i>TOKENS</i> GRAMATICAIIS	43
APÊNDICE 4 - EXEMPLO DE CRONOGRAMA DE LOCALIZAÇÃO	51
COLABORADORES	52

CULTURALIZAÇÃO

[Nota: colaboração de Kate Edwards; esses tópicos são trechos de seu livro ainda não lançado sobre culturalização de jogos]

O que é "culturalização" de jogos?

Culturalização é um passo além da localização. É uma avaliação fundamental das hipóteses e escolhas de um jogo para então estudar a viabilidade dessas escolhas criativas num mercado global e multicultural, assim como em locais específicos. Enquanto a localização apenas ajuda os jogadores a compreender o conteúdo do jogo através da tradução, a culturalização permite um engajamento maior com o conteúdo, em um nível mais significativo. Além disso, garante que os jogadores não serão afastados por um conteúdo considerado impróprio ou até ofensivo no ambiente do jogo.

Equívocos culturais custam muito aos desenvolvedores e às distribuidoras. Não apenas em relação à perda da renda potencial, mas também pelo grande impacto negativo em relações públicas, pelos danos à imagem da empresa e pelas relações tensas com governos locais. Na pior das hipóteses, um governo local pode proibir o jogo ou mesmo tomar medidas mais drásticas contra a empresa, incluindo detenção de funcionários locais para interrogatório e até mesmo sua prisão.

Níveis de culturalização de jogos

A necessidade de localização de jogos é bem conhecida na indústria dos videogames, mas a necessidade de culturalização ainda é relativamente ignorada. A culturalização não é apenas uma tarefa específica; ela também tem como objetivo maior a adaptação internacional de conteúdo. Em sua forma mais básica, a culturalização de conteúdo consiste de três etapas:

1. **Culturalização reativa:** Tornar o conteúdo viável, ou seja, evitar questões problemáticas para que o jogo permaneça no mercado local.
2. **Localização e internacionalização:** Tornar o conteúdo claro, ou seja, realizar uma localização "típica" para que o jogo seja compreendido.
3. **Culturalização proativa:** Tornar o conteúdo significativo, ou seja, adaptar e fornecer opções locais específicas para que o jogo seja relevante ao local.

Alguns esclarecimentos sobre essas três etapas:

- ◆ A localização é crítica, mas o processo de alcançar a clareza através da tradução não é o único passo necessário na preparação do conteúdo para outras culturas. Isso é válido tanto para videogames quanto para qualquer tipo de conteúdo.

- ◆ Pode-se debater que um jogo deva ser "claro" antes de ser "viável". Mas um governo terá restrições a um jogo baseado em questões delicadas, esteja ele localizado ou não.
- ◆ As três etapas não são uma hierarquia. Assim como a localização, a culturalização acontece em várias etapas em um ciclo de desenvolvimento típico. É uma coordenação de várias tarefas e prioridades orquestradas durante todo o processo.

Quatro variáveis culturais principais

O esforço de pensar além de nossa própria visão cultural muitas vezes é um fator que explica o porquê de um designer de jogos de um determinado local não estar ciente dos problemas que pode causar em outro local. Mas é possível reduzir o surgimento de problemas, levando em conta ao menos as seguintes variáveis culturais que geram a maior parte dos conflitos entre o contexto dos jogos e culturas locais:

1. História: passado e presente

A questão da precisão histórica é uma das mais delicadas para os mercados locais. Muitas culturas são extremamente protetoras em relação a suas origens e seu legado, qualquer alteração ou imprecisão histórica pode gerar uma reação emocional negativa. História é um tema atraente, mas é muito raro fornecer o contexto completo de um evento histórico em um jogo. Mas não é apenas a história distante que se mostra problemática. A história recente pode ser uma questão muito delicada, pois a lembrança dos eventos e seus resultados ainda são vívidos nas mentes das pessoas.

2. Religião e sistemas de crenças

Criadores de jogos devem ter sensibilidade em relação aos mecanismos subjacentes das culturas em que seus títulos serão lançados. Em geral, uma sociedade baseada em regras sagradas tende a ser menos flexível em relação ao contexto no qual as informações aparecem, porque segue o que considera ser um padrão mais elevado que o julgamento humano. Ou seja, basta o conteúdo problemático surgir, não importa o contexto, para dar margem a uma reação negativa.

3. Etnia e atritos culturais

Além das questões históricas e religiosas, mais voláteis, há muitos problemas que se encaixam em uma categoria mais ampla. Ela engloba vários desacordos, equívocos, atitudes e atritos contínuos entre grupos de culturas diferentes. O principal entre eles é o uso de estereótipos culturais e/ou étnicos, além da percepção de inclusão e exclusão com um viés negativo em relação a um grupo específico.

4. Imaginações geopolíticas

É comum os governos reforçarem sua visão de mundo local e a extensão de sua soberania geográfica através de mídias digitais, incluindo mapas on-line e jogos. Isso envolve uma situação em que o governo reclama certos territórios e espera que tais territórios sejam mostrados integrados à nação, seja num mapa funcional, seja no mundo de um jogo (por isso o termo "imaginação geopolítica", visto que o retrato exigido não reflete a realidade). Alguns governos, como China e Índia, não deixam margem para erros nessa questão, pois mantêm leis que ditam como mapas territoriais devem aparecer ou como sua situação política deve ser mostrada.

Veja no Apêndice 2 exemplos de jogos que contêm essas quatro variáveis de culturalização.

Práticas recomendadas para culturalização

O princípio subjacente da culturalização é que um pequeno investimento de tempo e esforço durante o desenvolvimento de um jogo compensa uma grande perda de tempo, dinheiro e desgaste de relações públicas na resolução de problemas pós-lançamento. Felizmente, há alguns passos fundamentais que desenvolvedores podem seguir para serem mais proativos em relação a sua estratégia de culturalização.

Informe-se

1. **Busque informações básicas:** um passo fundamental é estar consciente dos potenciais problemas culturais; gerenciadores e criadores de conteúdo precisam entender que problemas culturais podem acontecer e saber quais mercados serão afetados por qual tipo de conteúdo. Por exemplo, a maior parte das pessoas está ciente de que China, Índia, Coréia e Oriente Médio podem ser mercados sensíveis. Muitas pessoas também sabem que certos tipos de conteúdo podem tornar-se um foco de reações negativas, tais como mapas, bandeiras e informações históricas.
2. **Questione:** O objetivo é fazer as perguntas adequadas durante o desenvolvimento, não tornar-se um especialista no tema. Por exemplo, o jogo *Kakuto Chojin* (2002) continha uma pequena faixa de áudio em que uma parte do Alcorão islâmico é entoada. Isso resultou numa reação negativa generalizada e a comercialização do produto foi interrompida. O problema poderia ter sido evitado se alguém tivesse perguntado: "De onde vem a letra dessa música e qual seu significado?". Se algo não parece certo – mesmo que não se saiba o porquê – levante a questão imediatamente.
3. **Estabeleça responsabilidades:** para a culturalização ser bem sucedida, deve ser tratada como um componente padrão do processo de desenvolvimento. Isso significa que a responsabilidade pelo projeto deve ser atribuída a uma pessoa ou

equipe específica, geralmente os coordenadores de conteúdo e/ou editores. Além disso, um novo tipo de *bug*, "cultural", "geopolítico" ou o que for mais adequado, deve ser criado no sistema de rastreamento de *bugs* para certificar-se de que os problemas sejam identificados e resolvidos.

Identifique os problemas

Como mencionado anteriormente, quanto antes a culturalização for aplicada ao conteúdo do jogo, mais eficiente será. Ela engaja a equipe em discussões sobre significado, intenção e propósito dos personagens, enredo, ambiente e objetos, entre outros, o que permite a identificação da maior parte dos problemas em potencial durante os estágios conceituais. Aqui estão os pontos básicos para identificar problemas potenciais:

1. **Proximidade contextual:** De forma simples, proximidade contextual é o conceito de que quanto mais um elemento do conteúdo se aproxima do contexto original, pessoa, lugar, tempo ou forma, maior é o potencial de sensibilidade cultural. Os desenvolvedores devem buscar conteúdos que imitam locais, construções, pessoas, eventos, religiões, nacionalidades, etnias, etc. do mundo real, para então avaliar o grau em que esses conteúdos se assemelham à sua inspiração real.
2. **Aproveitamento de recursos externos:**
 - (a) Referências textuais: muitas obras de referência são úteis na pesquisa básica, tais como estudos culturais, guias específicos sobre o país, dicionários de simbologia, enciclopédias sobre religiões e divindades, entre outros.
 - (b) Pesquisa on-line: Wikipédia, sites de organizações governamentais, de organizações não governamentais (ONGs), organizações religiosas, entre outros.
 - (c) Opiniões locais: avaliar o conhecimento das pessoas de um determinado local ou cultura pode ser especialmente útil. Se você trabalha em uma grande empresa multinacional, aproveite a diversidade interna e peça a opinião de seus colegas. Uma alternativa é solicitar opiniões on-line em fóruns (por exemplo, Yahoo Answers). Esse conjunto de opiniões pontuais pode conter pontos de vista subjetivos, mas uma amostra grande o bastante pode revelar um padrão claro.
 - (d) Especialistas no assunto: se as pesquisas mencionadas não produzirem resultados claros, procure pessoas de diferentes áreas como história, estudos interculturais ou geografia.

Avalie a gravidade

Nem todos os problemas identificados na pesquisa precisam ser corrigidos.

Após identificar potenciais problemas culturais, é crucial saber determinar com eficácia quais devem ser corrigidos na próxima etapa.

1. **Faça uma triagem dos problemas encontrados:** separe as "ofensas claras" – coisas que você tem certeza de que serão um problema – dos "riscos razoáveis" – coisas que podem criar preocupações, mas provavelmente não impedirão que um jogo seja comercializado no local planejado.
2. **Documente suas escolhas:** todas as distribuidoras de jogos têm a escolha de mudar ou não conteúdos potencialmente problemáticos. A maior parte das empresas acabam fazendo, mas há momentos em que não há sentido fazer mudanças, mesmo que pequenas, porque o problema está no limite do aceitável. Nesses casos, é fundamental documentar a cadeia de decisões em uma explicação defensiva, caso seja necessário explicar a um governo ou consumidores que levantem a questão.

Implemente com precisão

Muitos designers de jogos têm uma noção preconcebida de que culturalização envolve mudanças enormes e repensar totalmente a ideia do jogo. Essa é uma percepção equivocada e é o principal motivo por trás da decisão de muitos deles de não confrontar aspectos culturais e geopolíticos, pois acreditam que é muito trabalhoso. E na verdade um dos princípios mais importantes da culturalização é:

1. **Seja preciso:** Faça o mínimo de mudanças na menor quantidade de conteúdo possível. Mude apenas o que realmente deve ser mudado para assegurar a distribuição do jogo no mercado alvo. Na maior parte dos casos de problemas culturais, a solução é um reparo pequeno e preciso de um símbolo, palavra ou design de personagem específicos. Não costuma ser um problema de maior dimensão, como a premissa do jogo (embora isso possa ocorrer).

Conclusão

Crie o jogo que você quer criar, mas não se esqueça do público global e multicultural que participará de sua visão e, se tudo der certo, aproveitará o jogo sem nenhum estranhamento cultural. A culturalização bem executada no ciclo de desenvolvimento não é uma fórmula pronta, e sua implementação leva um certo tempo. Entretanto, os benefícios à qualidade do conteúdo de uma empresa, às relações com governos e à sua imagem pública entre os jogadores locais se mostrarão um investimento inestimável a longo prazo.

INTERNACIONALIZAÇÃO

Internacionalização, abreviada como *I18N* em inglês, é o processo que permite que a localização aconteça. Antes da internacionalização, um produto somente pode exibir o conteúdo do jogo em um único idioma. Depois da internacionalização, o código base do produto, sua arquitetura e interface de usuário são capazes de processar e exibir o conteúdo em múltiplos idiomas. Um produto internacionalizado não contém nenhum elemento que dependa de um determinado local.

Interface de usuário

Fontes

Idiomas asiáticos como japonês, coreano e chinês usam, predominantemente, fontes de largura fixa. Se os programadores da versão original asiática não estiverem dispostos a reprogramar o sistema para lidar com fontes proporcionais das versões de idiomas europeus, essas versões europeias terão um espaço indesejado entre as letras. Isso afeta negativamente o visual do jogo, assim como a qualidade da tradução, devido à redução dos limites de largura de linha.

Apesar de não ser mais um problema na maior parte dos jogos de computador e consoles, ainda parece ser um problema em jogos para portáteis.

Para maximizar o espaço na tela, além de uma fonte proporcional, é bom utilizar um modo de exibição que permita o *kerning* (espaçamentos diferentes entre combinações de letras que otimiza o espaço em branco sob letras sobrepostas, entre outros fatores).

Além de ser uma dor de cabeça para os programadores acostumados com idiomas de fonte monoespaçada, é mais complicado para o tradutor conferir o limite da largura da linha. É preciso criar ferramentas simples de conferência de largura de linha ou macros no Excel que somem a largura de pixels de cada letra e que informem se elas estão dentro do limite de espaço reservado para a *string* na tela. Mas a melhoria na qualidade da tradução e no visual do jogo certamente compensam o trabalho extra.

É importante que a equipe de tradução e a de desenvolvimento trabalhem juntas desde o início do ciclo de desenvolvimento para escolher as fontes. Os tradutores precisam ter a oportunidade de conferir se a fonte escolhida disponibiliza todas as letras usadas em seu idioma. Além disso, é preciso fazer escolhas quanto ao tamanho e tipo da fonte, não só para garantir a inteligibilidade do texto mostrado na tela, mas também para determinar a largura em pixels de cada letra para o uso em testes de limite de largura (isto é, para garantir que a tradução vai caber em cada espaço alocado na tela). Apenas um falante nativo de um idioma sabe dizer se a “impressão” dada por uma fonte parece natural, seja o aspecto de um acento gráfico em um idioma europeu, ou se os caracteres chineses podem ser lidos em outros idiomas asiáticos, então não pule o passo de

aprovação da fonte antes do início da tradução.

Interface de usuário e janelas de mensagens

Ao lidar com vários idiomas, você logo percebe que o *layout* do texto, quando mostrado na tela do jogo, inevitavelmente vai precisar de ajustes entre idiomas. Para melhor lidar com isso, sugerimos:

- ◆ Que você permita que “balões de fala” sejam redimensionados de forma automática nas janelas de mensagens ou áreas da tela mais flexíveis. (Um único investimento inicial para programar um sistema assim vai economizar muito tempo mais tarde, em que a sua equipe de desenvolvimento teria que fazer tudo manualmente). Se um sistema totalmente automatizado for demais, pelo menos permita que os tamanhos sejam redimensionados com a especificação das coordenadas x, y, w, h nos arquivos de texto. Dessa forma, a equipe de tradução poderá ajustá-los em cada idioma, em vez de sobrecarregar a equipe de desenvolvimento com tais ajustes.
- Se todas as caixas de texto precisam ser idênticas em todos os idiomas, outra alternativa é dimensioná-las pensando na largura das traduções previstas, em vez da largura do texto em inglês. A regra é estimar que as traduções sejam 30-40% mais compridas que o inglês.
- ◆ Que você implemente um sistema automático de quebra de texto, isto é, em vez de pedir aos tradutores que codifiquem cada quebra de linha manualmente (e depois ter que ajustar o texto constantemente para fazer com que a quebra apareça no local certo da tela), automatize o sistema do jogo para fazer isso. (Também é possível permitir ajustes para quando você quiser uma separação de palavras específica de uma linha para outra... falaremos sobre isso depois).
- ◆ Isso é praticamente uma obrigação para jogos de PC, já que estes permitem a mudança de resolução e tamanho da tela pelo usuário.
- ◆ Se for adequada, outra opção é implementar um sistema de expansão e contração de fonte, de forma que o texto encaixe perfeitamente na área alocada. Por exemplo, em um título de menu, se o texto exceder um pouco o espaço alocado, o sistema diminui a fonte para que ele caiba (é melhor do que cortar a tradução no meio da palavra).
- ◆ E também, se for adequada, outra opção é permitir a rolagem do texto ou a expansão em uma tela de *pop-up*. Assim, se a interface de usuário do jogo tiver um espaço limitado, o jogador ainda poderá ver o texto completo em destaque, em vez de ter que decifrar abreviações ou palavras quebradas.

Legendas em filmes pré-renderizados

Para economizar memória, assim como para evitar que a sua equipe de desenvolvimento tenha que renderizar de novo várias versões do mesmo filme, tente implementar um sistema em que as legendas sejam mostradas por cima do filme, em tempo real, em vez de “queimá-las” no arquivo do filme. Se, por um lado, criar um sistema dá muito trabalho, pense no custo de corrigir um bug nas legendas cada vez que ele aparecer em cada idioma. (Além disso, são raras as vezes em que as legendas ficam bonitas e perfeitamente cronometradas logo de primeira, então é altamente recomendado um sistema mais flexível do que renderizar tudo de novo, várias vezes).

Estas são as práticas recomendadas para legendagem:

- ◆ Incorpore uma fonte sem serifa em cor clara, com contorno escuro, para que ela seja facilmente visível tanto em fundos claros quanto escuros (veja o *Red Dead Redemption*, por exemplo).
- ◆ Posicione as legendas na mesma área da tela durante todo o jogo; de preferência, no terço inferior da tela, exceto no caso de interferência com a interface.
- ◆ Mostre cada bloco de texto durante 3 segundos, alinhado à esquerda, permitindo 96 caracteres por linha, e limitado a 3 linhas de texto por bloco. Linhas a mais dificultam a leitura pelo usuário. [Nota do tradutor: no Brasil, a tradução audiovisual de modo geral usa no máximo duas linhas de texto por bloco]
- ◆ Use texto estático, evite a rolagem de texto sempre que possível.
- ◆ Quando duas pessoas estiverem conversando, indique a mudança do interlocutor com um hífen (-) no começo da linha da fala, especialmente se o interlocutor não estiver visível (isso também deveria acontecer nas legendas originais em inglês).

Arquitetura

O conteúdo para localização deverá ser armazenado em um sistema de gerenciamento de conteúdo que tenha uma estrutura de pastas acessíveis à equipe de localização. Para facilitar a importação / exportação, sugere-se armazenar o conteúdo categorizado em nível, local do jogo ou idioma.

Idiomas e códigos regionais

É tentador simplesmente adicionar uma letra (como “E, F, I, G, S, J, K ou C”) para representar os idiomas ou versões regionais ao final dos nomes de arquivos, nomes de diretórios ou identificadores de programas. No entanto, é comum deparar-se com questões como inglês americano versus britânico, espanhol e português europeus versus espanhol latino-americano e português brasileiro, e até mesmo mandarim padrão simplificado versus mandarim tradicional e cantonês tradicional de Hong Kong. Além disso, pode haver confusão sobre qual código representa qual idioma (por

exemplo, se alemão deve ser “Ge” de German ou “De” de Deutsch). Para evitar confusões relativas a idiomas e códigos regionais, é uma boa ideia ater-se aos padrões ISO, reconhecidos por todos os setores.

http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

http://en.wikipedia.org/wiki/ISO_3166-1

(ou http://www.loc.gov/standards/iso639-2/php/code_list.php)

Por exemplo, se você juntar o código do idioma de duas letras com o código do país de duas letras, você terá uma forma curta e facilmente reconhecível de diferenciar versões linguísticas e regionais. (Quando um idioma é utilizado em vários países, como na América Latina, você pode simplesmente escolher o país mais representativo, como o México). Use um hífen entre o código do país e do idioma (PT-BR) ou um sublinhado (PT_BR) ou, para ser ainda mais conciso, caso seu sistema permita, escreva um em caixa baixa e o outro em caixa alta (ptBR).

Assim você terá:

enUS	inglês americano
enGB	inglês britânico
deDE	alemão
esES	castelhano
esMX	espanhol latino-americano
frFR	francês (europeu)
itIT	italiano
jaJP	japonês
koKR	coreano
ptBR	português brasileiro
ptPT	português ibérico
plPL	polonês
ruRU	russo
zhCN	chinês simplificado (mandarim)
zhTW	chinês tradicional (mandarim)

Pode demorar um pouco até todos se acostumarem a usar esses códigos, mas eles

ajudam muito a esclarecer as coisas.

Ferramentas de localização e formatação de arquivos

Da mesma forma que as equipes de desenvolvimento utilizam várias ferramentas diferentes para administrar seus recursos (Visual Source Safe, AlienBrain, etc.) e facilitar a programação (Visual Programming Environment etc.), as equipes de tradução também precisam de ferramentas adequadas para auxiliar seu trabalho. Ao mesmo tempo em que muito do trabalho pode ocorrer em arquivos simples de texto ou de Excel, existem ferramentas de gerenciamento de arquivos, de memória de tradução, dicionários eletrônicos e ferramentas de comparação de arquivos (BeyondCompare ou ferramentas no estilo "Diff") que otimizam o processo de tradução, especialmente quando o trabalho envolve a alteração de arquivos de recursos ou grandes equipes. Não entraremos em detalhes sobre tais ferramentas, mas sugerimos que você encontre a melhor solução para as necessidades de suas equipes de desenvolvimento e tradução.

Ainda no assunto de formatos de arquivos, se você estiver usando arquivos simples de texto, lembre-se das diferenças em formato de arquivo, especialmente quando for trabalhar com idiomas asiáticos. Em vez de lidar com problemas de arquivos corrompidos entre, por exemplo, códigos ASCII estendidos e JIS, talvez seja melhor trabalhar com um código que permita que todos seus idiomas de destino coexistam desde o início, como o UTF-8 ou similar.

Uma armadilha comum em jogos é a questão do texto nos gráficos. Sejam texturas em 2D ou modelos poligonais em 3D, a equipe de desenvolvimento precisa demarcar onde todos os "textos" aparecem no jogo, para que eles sejam localizados corretamente. Prepare-se para redesenhar todos os textos gráficos para cada versão linguística (ou encontre uma solução que permita usar ícones e símbolos gráficos no lugar de texto, para reduzir a carga de trabalho dos artistas gráficos). Esses textos precisam ser fornecidos ao tradutor e ter suas alterações controladas da mesma forma que todos os outros recursos de texto.

Arquivos de recursos e codificação de caracteres

Recomenda-se não programar permanentemente o texto da interface de usuário no código-fonte (isso vale para qualquer *script* do jogo que possa gerar problemas se for alterado acidentalmente pelos tradutores). Além disso, é muito importante que os arquivos sejam formatados de forma lógica. Se os "eventos" ou os enredos do jogo estiverem separados em diferentes arquivos, ou se não estiverem na ordem, não se esqueça de incluir comentários ou um sistema de identificação que ajude os tradutores a entender a ordem do texto e onde ele aparece no jogo.

Indique a qual plataforma as *strings* pertencem, especialmente para salvar, carregar ou outras funções que afetem a conformidade da plataforma. A equipe de localização vai precisar tomar um cuidado especial ao traduzir essas *strings*, de forma a usar os termos

sugeridos pelos fabricantes do console.

Tudo fica mais organizado e descomplicado se todos os arquivos tiverem formatos parecidos. Especialmente se as equipes de desenvolvimento ou de tradução estiverem usando ferramentas para conversão, quanto menos alterações na formatação, menos trabalho (e possibilidade de erros) a conversão e reconversão de arquivos dará.

Além disso, dependendo do código do arquivo, podem ocorrer problemas de memória. Se os idiomas asiáticos usarem conjuntos de caracteres de dois bytes – o que também pode ser esperado do inglês e outros idiomas europeus –, logo se vê que a largura das *strings* poderão ocupar mais memória do que o esperado. (O inglês usa, em média, 1,5 caracteres a mais que o japonês, e o alemão ou espanhol, por exemplo, podem usar mais que o dobro de letras que o japonês). Para evitar o desperdício de memória ou que a capacidade da memória seja excedida, use um código que mantenha as letras europeias em um único byte (como o UTF-8 ou outros códigos de largura de byte variável).

Por outro lado, no caso do chinês, coreano e japonês, se os desenvolvedores ocidentais tiverem problemas para colocar todo o conjunto de fontes na memória (por exemplo, em jogos portáteis), uma solução é carregar em tempo real os caracteres ou glifos que apareçam naquela parte específica do jogo. Ou então, se você precisa carregar a fonte inteira, converse com os tradutores sobre quais segmentos da fonte eles não vão precisar e remova-os do arquivo da fonte. No japonês, por exemplo, a maioria das frases podem ser traduzidas utilizando um conjunto de 2000 caracteres (o número de caracteres que ensinam às crianças na escola).

Programação

Dicas Gerais de Programação

Você deve evitar textos fixos no código, já que isso significa que um programador vai ter que ajustar essas *strings* manualmente a cada versão, em vez de simplesmente deixar que o programa o faça automaticamente na hora da compilação. Além do mais, quando o texto está fixo no código, o programador também vai precisar separar da fonte do programa o texto a ser traduzido e recolocá-lo manualmente, ou ainda entregar o código-fonte aos tradutores para que eles encontrem a tradução (e correr o risco deles acidentalmente alterarem o código do programa). A prática recomendada é isolar todas as *strings* de texto usadas no jogo em um arquivo de recursos para cada idioma.

Como mencionado na seção de Interface de usuário e janelas de mensagens, quando você quiser centralizar o texto ou posicioná-lo em locais específicos, em vez de colocar as coordenadas fixas no código, é melhor permitir que o sistema realize a centralização ou posicionamento do texto para você, baseado na largura das *strings* de chegada.

(Senão, novamente, um programador terá que corrigir tudo manualmente a cada versão. Uma opção para quando o texto precisa ser posicionado ou centralizado, mas o sistema não suporta isso automaticamente, é permitir que as coordenadas apareçam nos arquivos de texto ou em arquivos de recursos separados, de forma que os tradutores ou pessoal do CQ (controle de qualidade) possam realizar as mudanças de posicionamento para cada idioma entre compilações, em vez de tomar o tempo precioso dos programadores).

Os arquivos de recursos devem ser separados por idioma. Mesmo que possa parecer organizado colocar todos os idiomas em páginas ou colunas separadas em um arquivo de Excel, na prática isso significa que os tradutores dos diferentes idiomas não conseguirão trabalhar no mesmo arquivo ao mesmo tempo, e alguém vai acabar tendo que copiar e colar os textos de cada idioma de volta em um arquivo, sujeitando o processo ao erro humano e desperdiçando tempo. Recomenda-se que os arquivos de cada idioma tenham seus próprios diretórios e que de lá sejam copiados na hora da compilação. Para auxiliar a equipe de localização, inclua uma lista das alterações de texto no idioma de partida para todas as versões. Isso vai diminuir a probabilidade da equipe de localização perder uma *string* nova ou editada.

Usando o formato XML para a tradução de textos

Algumas empresas consideram que o XML (*Extensible Markup Language*) é uma boa alternativa em relação a arquivos de texto simples ou tabelas de Excel para se fazer a tradução de projetos. Por exemplo, se você sabe que certos idiomas precisarão de somente uma versão de *string*, enquanto outros necessitarão de até seis (talvez porque o segundo conte com marcas de singular/plural ou masculino/feminino/neutro na sua gramática, ao passo que o primeiro não as tem), então poderá usar tags de XML para tratar de ambos os casos. Logo, ao invés de fazer com que 6 versões da mesma *string* sejam traduzidas de maneira idêntica, os tradutores poderão traduzi-la apenas uma vez.

Ao passo que os idiomas que necessitarem de versões adicionais poderão incluí-las

(na memória do programa você poderá escolher mostrar a quantidade máxima de combinações em cada linha, mesmo que a maior parte delas apareça como *strings* NULL ao final. Como o arquivo XML será validado e compilado, as *strings* que de fato existem no XML serão inseridas no código-fonte e o sistema será indicado para alternar entre estas no tempo de execução).

Assim, nos idiomas japonês, chinês e inglês poderá haver atendentes que o saúdam com um simples "Olá!", enquanto que em italiano teríamos o equivalente a "Bem-vindo!", "Bem-vinda!", "Bem-vindos!" e "Bem-vindas!", cada um destes com tags de XML definidos de forma que o sistema do jogo registre a existência de variações entre singular/plural e masculino/feminino e processe-as em tempo real de acordo com o contexto interno do jogo.

XLIFF (XML Localization Interchange File Format)

Ainda que muitas plataformas, engines de jogos e outras tecnologias armazenem os textos em diferentes formatos de arquivos, trabalhar com eles pode ser problemático para as equipes de localização. Isso complica, primeiramente, o ciclo de solicitar e implementar traduções, uma vez que cada formato exige um fluxo operacional independente. Também prejudica a capacidade da sua equipe de desenvolver ferramentas de tradução econômicas e de qualidade.

Em 2002, o formato XLIFF foi estabelecido para resolver tal problema e oferecer uma maneira padronizada para todos os desenvolvedores trocarem textos com equipes de localização. Implementar essa padronização requer converter os textos de seu formato nativo para o XLIFF ao solicitar traduções. A implementação da tradução necessita pós-processar o XLIFF de volta ao formato de arquivo de texto nativo.

Para mais informações sobre o XLIFF, e ferramentas auxiliares, consulte os seguintes links:

<http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html>

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff

<http://en.wikipedia.org/wiki/XLIFF>

Ordem de variáveis em um texto

Ao utilizar o `printf` do C ou C++ ou funções de exportação de texto semelhantes, lembre-se de que a ordem das variáveis `%s` e `%d` em *strings* de texto provavelmente será diferente entre os idiomas. Por exemplo:

```
msg="%s comprou %d %s por %d reais.
```

terá sua ordem modificada em japonês e coreano para...

```
msg="%s が %s を %d 個買って、 %d ドルを支払いました。 ;
```

(ou seja, a ordem: nome_personagem, número, nome_item, preço se tornará: nome_personagem, nome_item, número, preço.)

Observação: como falamos dos *tokens* de singular/plural acima, mostraremos agora de que forma a implementação do sistema citado pode fazer o texto resultante parecer muito mais natural e correto.

```
ex: "John comprou 6 poção por 1 reais."
```

... está errado, mas se implementarmos o sistema de *tokens*, o texto ficará assim...

```
msg="<Cap><Personagem1>          comprou          <SE_SING_Número1>
<ART_INDEF_SING_ITEM_2>  <SENÃO_NÃO_SING_Número1>  <Número 1>
<PLR_ITEM_2>    <INDEF_SING_ITEM_2>    <FIM_SE_SING_Número1>  por
<Número2> <SE_SING_Número2> real <SENÃO_NÃO_SING_Número2> reais
<FIM_SE_SING_Número2>;
```

... então, se implementado corretamente, o sistema acima será capaz de oferecer todos os seguintes resultados...

John comprou 1 maçã por 1 real.

John comprou 2 maçãs por 2 reais.

Maria comprou uma pera por 3 reais.

e assim por diante...

Data, hora, moeda e formatos numéricos

Cada região tem sua maneira de representar datas e números. Conversar com sua equipe de localização antes de iniciar a produção do jogo permitirá que as equipes de desenvolvimento determinem todas as áreas afetadas e quais formatos devem ser implementados para cada idioma.

Abaixo estão listados alguns dos formatos padronizados nos idiomas mais comuns.

	Inglês (EUA)	Francês	Alemão	Espanhol	Italiano	Japonês
Data	mm/dd/aaaa	dd-mm-aaaa	aaaa-mm-dd	dd/mm/aaaa	dd/mm/aaaa	aaaa 年 mm 月 dd 日
Hora	hh:mm:ss am/pm (relógio de 12 horas)	hh:mm:ss (relógio de 24 horas)				
Separador decimal	ponto (.)	vírgula (,)	vírgula (,)	vírgula (,)	vírgula (,)	ponto (.)
Separador de milhares	vírgula (,)	espaço ()	espaço () ou ponto (.)	espaço ()	espaço () ou ponto (.)	vírgula (,)
Exemplo de algarismo	15,631.87	15 631,87	15 631,87 ou 15,631.87	15 631,87	15 631,87 ou 15,631.87	15,631.87
Moeda	\$12,345.67	12 345,67	€ 12 345,67	€ 12 345,67	€ 12.345,67	¥ 12,345
Ordinais	1st 2nd 3rd ...	1er 2e 3e ... ou 1re 2e 3e ...	1. 2. 3. ...	1º 2º 3º ... ou 1ª 2ª 3ª ...	1º 2º 3º ... ou 1ª 2ª 3ª ...	1 日 2 日 3 日...

Uma das melhores de formas de implementar esses diversos formatos é incluir uma *string* nos arquivos de recurso para cada tipo de formato numérico que é traduzido.

Dessa forma, caberá à equipe de tradução identificar corretamente a ordem do ano, mês e dia, além dos diferentes caracteres usados na separação dos algarismos.

Quando se localiza moedas correntes no jogo, deve-se, além de se atentar para os símbolos únicos de cada moeda, aplicar taxas de câmbio aproximadas. Por exemplo, nos Estados Unidos é aceitável que o jogo venda um x-burguer a \$2,00. Contudo, não seria realista fixar o preço a ¥ 2 na versão japonesa. Provavelmente o preço seria ¥160 (considerando uma taxa de câmbio de 80 yen por dólar americano).

Quebra automática de linha e ajuste do texto

Implementar um sistema de quebra automática de linha ou de ajuste de textos em jogos com muitos textos diminuirá bastante a probabilidade de bugs por conta de transbordamento de caracteres. Também tornará o trabalho de produção e de CQ muito mais rápido do que se os escritores originais e os tradutores tivessem de inserir todas as quebras de linha manualmente.

Além disso, caso o mesmo texto seja exibido em diversos locais diferentes e com larguras variadas, não será preciso preparar múltiplas versões dele (ou, pior ainda, forçar que os tradutores preparem o texto pensando na largura máxima e deixem o restante do espaço nas demais telas sem uso).

Ainda assim, haverá momentos em que você precisará anular o algoritmo de quebra automática de linha. Por exemplo, é comum que empresas de consoles imponham uma regra determinando que certas terminologias não sejam quebradas entre as linhas. Nesse caso, pode ser útil usar um símbolo de "espaço em branco inquebrável". Exemplo: um código especial de caractere oculto no conjunto de fontes Unicode (0xA0), ou um *token*, como o "<nbsp>" ou o "_" (sublinhado) que é convertido no tempo de execução para se parecer com um espaço vazio, mas que constitui uma exceção ao algoritmo de quebra de linha. (Ademais, se o seu algoritmo de quebra de linhas também quebra hífen, pode ser necessário empregar um símbolo de hífen especial, ou um "<nbhy>", de forma que as palavras que não podem ser quebradas no hífen não o sejam, embora tais casos sejam raros).

Outro exemplo, o jogo pode conter circunstâncias especiais como a seguinte:

"Brigadoon ->

<- Xanadu"

Se depender apenas do sistema de quebra automática de linha, o texto dessa placa na estrada apareceria assim:

"Brigadoon -> <-

Xanadu"

Mais uma vez, um caractere de quebra de linha ou *token* como "<lb>" resolve esses problemas sem que os programadores precisem colocar exceções no código cada vez que surgirem problemas.

Exceções de pontuação em inglês, alemão, espanhol e italiano

O algoritmo básico de quebra automática de linha funciona da seguinte maneira: se a linha superar a largura da janela, ele procurará a última letra que se encaixa na tela e, a partir daí, voltará no texto para encontrar um espaço " " ou hífen "-". Em seguida, o algoritmo deslocará todo o texto após o espaço ou hífen para a linha seguinte e assim por diante. Se não for encontrado qualquer espaço ou hífen ao voltar no texto, então a única opção será quebrar a linha na última letra que estiver dentro dos limites da tela (em termos gramaticais, seria mais correto, ao quebrar uma palavra ao meio, fazê-lo em uma das sílabas e acrescentar um hífen, mas tal algoritmo é avançado demais para alguns jogos. Situações onde a linha é quebrada no meio de uma palavra são consideradas raras exceções e podem ser resolvidas manualmente). Pode-se fazer uma pequena otimização programando o algoritmo para procurar o espaço ou hífen no máximo 15 letras para trás, em vez da linha inteira (principalmente caso implemente o sistema de "*tokens* ocultos de quebra de linha" descrito abaixo).

Trate vírgulas ",", pontos finais ".", pontos de exclamação "!", pontos de interrogação "?", ponto e vírgula ";", e dois pontos ":" como se fizessem parte da palavra anterior a que estiverem conectados, de forma a evitar que apareçam sozinhos na linha seguinte.

Tenha cuidado, também, com o "travessão" e a "meia-risca" ("hífens" longos que têm o comprimento de um m e n, respectivamente), além do "raya" em espanhol (um hífen com quase o triplo do tamanho usado como aspas, com código de caractere 0x97). Eles podem ser deslocados para a próxima linha. No entanto, muitas vezes quando a fonte ou o sistema de exibição de texto não aceitam tais caracteres, os tradutores podem usar dois hífen como alternativa. Assim, o sistema terá de fazer exceções e quebrar a linha antes do primeiro de dois hífen consecutivos, ou após o segundo.

Observação: embora o alemão use aspas em formato de 99 e o espanhol tenha um ponto de exclamação invertido ao início da frase, ambos os idiomas seguem as mesmas regras de pontuação citadas acima quanto à quebra automática de linha (as aspas no alemão têm o formato „ ... " e o código 0x84 e 0x93).

Falando em pontuação, é bom certificar-se de que os tradutores tenham a opção de usar tanto aspas simples " " quanto duplas """. O inglês britânico emprega aspas simples para citar textos, enquanto o inglês americano emprega aspas duplas, mas ambas são usadas alternadamente ao citar textos inseridos em outras citações. Pode-se também empregar apóstrofos retos ao invés de reutilizar o caractere de fechar aspas simples (ou seja, não usar as aspas em formato de 6 ou 9, mas sim um apóstrofo vertical com o código 0xB4). Isso ajuda a diferenciar as aspas simples dos apóstrofos em textos com

abundância de citações, embora poucas empresas procurem facilitar a leitura dessa maneira.

Caso o sistema do jogo ou as ferramentas de tradução, por quaisquer que sejam os motivos, não consigam trabalhar com pontuações tais como "''", então pode-se considerar, como medida alternativa, implementar *tokens* como <66>, <99>, <11>, <6>, <9> e <1> a serem substituídos no tempo de execução ou compilação por seus respectivos símbolos. (são *tokens* fáceis de serem visualizados e inseridos por tradutores quando necessário).

Exceções de pontuação em francês

O francês tem algumas regras de pontuação diferentes que devem ser levadas em conta ao implementar o sistema de quebra automática. Os pontos de exclamação "!" e interrogação "?", o ponto e vírgula ";", os dois pontos ":" e as aspas duplas "''" e "''" (em formato de 66 ou 99) normalmente sucedem um espaço em branco. Para lidar com isso, sugerimos as seguintes medidas:

1. Instrua os tradutores a empregar um espaço que não quebre linhas (0xA0), ou então um *token* como "<nbsp>", que não é tratado como espaço em branco, mas que será exibido como tal no tempo de execução. Antes de entregar, os tradutores deverão verificar o texto à procura de pontuações precedidas por espaços que quebrem a linha e substituí-los. Como parte das práticas recomendadas, os engenheiros de localização responsáveis por implementar o texto deverão verificar se não estão faltando espaços que não quebrem a linha e apresentá-los à equipe de tradução.
2. Configure o sistema de quebra de linhas para tratar a pontuação de maneira correta ao exibir a versão francesa do texto no jogo. Ou seja, tratar o trecho "Bonjour !" como uma única unidade de "palavra + espaço + pontuação especial" (alternativa preferível).
3. Outra alternativa, mais deselegante, é ocupar com pixels o espaço que antecede os seguintes caracteres: "!", "?", ";", ":", "''" e "''". (Pode-se fazer isso acessando os dados de largura de fonte e inserindo, à frente dos citados caracteres, pixels que preencham o tamanho ocupado por uma tecla espaço.) Os tradutores precisarão ser avisados para não inserir espaços antes destes símbolos, mas sim confiar que aparecerão corretamente na tela sem que seja necessário reprogramar o algoritmo de quebra de linhas. Porém, pode ser necessário inserir *tokens* (como <!>, <?>, <:>, <:>, <'>, e <'>) ou caracteres alternativos que aparentem ser iguais às versões de largura comum dessas fontes em casos especiais como "Que p... !?!", onde a pontuação aparece logo ao lado.

Pontuação japonesa, coreana e chinesa

Embora tais idiomas não sejam tão rígidos quanto aos pontos em que se pode quebrar o texto, há sim algumas normas gerais de pontuação que devem ser seguidas, mesmo que você adote uma fonte com largura fixa ou algoritmos de quebra de linhas. (Observe que os idiomas asiáticos não empregam espaços com tanta frequência, então na maior parte das vezes o texto será quebrado quando as letras chegarem ao fim da linha). As exceções se assemelham aos idiomas europeus: pontos finais “。”, vírgulas “、” e “、”, fechar aspas “」” e “』”, e o ponto equivalente ao hífen “・” não devem aparecer ao começo de uma nova linha, e o abrir aspas “「” e “『” não pode aparecer ao final da linha sem que haja texto subsequente. Basicamente, deixe um pouco de espaço ao final da área de exibição do texto para que se possa encaixar pontuações. Do contrário, você terá que deslocar o caractere anterior para a linha seguinte junto à pontuação para que esta não apareça ao início de uma linha nova. Por outro lado, não quebre a linha em um caractere de aspas ou semelhante: deixe que este seja deslocado para a linha seguinte com os caracteres que o seguem, de forma a cumprir com as corretas regras de pontuação.

Consulte este link para obter mais informações:

http://en.wikipedia.org/wiki/Line_breaking_rules_in_East_Asian_language

Opção avançada para inglês e idiomas europeus: *tokens* ocultos de quebra de linha

Especialmente quando um texto é gerado pelo tempo real do sistema de jogo, ou quando o texto é mostrado em áreas de exibição diferentes (ou redimensionáveis), palavras longas podem começar a distorcer o equilíbrio do comprimento das linhas expostas na tela, reduzindo o aspecto das janelas de texto ou da IU (interface de usuário). Podemos usar um truque das mídias impressas e implementar um sistema onde as palavras longas são separadas adequadamente, na sílaba mais próxima ou mais lógica, e um hífen é acrescentado no fim da linha, indicando que a palavra será transportada à linha seguinte.

Se oferecer aos tradutores (e escritores no idioma de origem) “*tokens* ocultos de quebra de linha”, eles poderão colocar *tokens* em palavras longas ou problemáticas, como um guia para o sistema de quebra de linha automática saber onde quebrar o texto. Em vez de um *token* longo como <hlb>, se o texto de chegada não usar os símbolos do til “~” ou de intercalação “^” na tela, você pode usá-los como *tokens* e inseri-los entre as sílabas de palavras longas. O seu sistema de exibição de texto nunca mostraria de fato estes *tokens* na tela, mas a sua rotina de quebra de linha automática também pode fazer a quebra nestes *tokens* buscando o hífen mais próximo ou espaço dentro das últimas 15 letras da *string* de texto.

p. ex. Se estivesse escrevendo à mão ou exibindo o texto em mídias impressas, você separaria a palavra “**espada**” em “**es-**” (ou “**espa-**”) e levaria “**pada**” (ou “**da**”) para a

linha seguinte. Então os tradutores poderiam escrever “**es^pa^da**” e o sistema faria exatamente isso se a palavra passasse do fim da linha.

Da mesma forma, se houvesse uma palavra como “**otorrinolaringologista**” causando problemas de quebra de linha no jogo, recomendamos acrescentá-las em algumas quebras de linha ocultas nas sílabas principais: “**otorrino^larin^golo^gista**”, mesmo que você queira ser mais preciso e colocá-los em cada sílaba, como “**o^tor^ri^no^la^rin^go^lo^gis^ta**”. Esta liberdade melhorará o visual e a legibilidade do texto na tela. (Ainda que possa parecer um pouco extravagante em inglês, para idiomas com palavras mais longas, como o alemão, isso pode fazer uma grande diferença na otimização do espaço da tela).

Mas o resultado é que os tradutores precisarão acrescentar estes símbolos na sua tradução, ocupando tempo e causando dores de cabeça com autocorreção e funções do verificador ortográfico, mas uma boa ferramenta de pré-visualização ou de verificação de linha ajudará muito a melhorar o trabalho.

Uma forma ainda mais avançada: Uma alternativa para adicionar *tokens* ocultos de quebra de linha diretamente ao texto da tradução é preparar um dicionário ou tabela de referência para cada idioma, indicando como devem ser feitas as quebras de linha em palavras longas. (isto é, uma tabela que diga ao sistema na hora da compilação para substituir todas as ocorrências de “**espada**” por “**es^pa^da**” etc.) Assim os tradutores não precisam colocar as palavras separadas no texto, mas quando os arquivos forem pré-processados para a integração ao sistema do jogo, qualquer palavra na tabela de dicionário/referência será substituída e funcionará como se os próprios tradutores tivessem feito isso no texto. Assim a carga sobre os tradutores será menor, pois não terão que digitar esses símbolos por conta própria em todas as partes da tradução (além de não terem que lidar com alertas desnecessários dos verificadores ortográficos e programas com autocorreção).

Permitir o uso de símbolos especiais

Mencionamos uma alternativa para quando sinais de pontuação como “””” causam conflito com o sistema do jogo ou com ferramentas de localização, que é a implementação de *tokens* como <66>, <99>, <11>, <6>, <9> e <1>. Eles podem ser substituídos na compilação ou execução, para representar os sinais de pontuação. Da mesma forma, é possível controlar o uso do hífen, meia-risca e travessão, usando <->, <-->, e <---> como alternativas também. Aqui usamos <xxx> para simbolizar *tokens*, mas eles também poderiam ser {xxx} ou/xxx ou qualquer outra representação que funcione no seu sistema.

Outra opção, caso a sua fonte tenha símbolos especiais que não sejam de acesso fácil por digitação em processadores de texto, editores de texto ou no Excel, é permitir o acesso dos tradutores por algum tipo de sistema de *tokens*. Aqui estão alguns exemplos

de glifos encontrado em idiomas asiáticos a que os tradutores de idiomas europeus podem querer ter acesso também (se compartilharem a mesma fonte ou estes símbolos de byte duplo podem ser disponibilizados na sua fonte de byte único também)...

← <l_arrow>, → <r_arrow>, ↑ <u_arrow>, ↓ <d_arrow>, ⇔ <2way_arrow>, ⇒ <doub_arrow>, ☆ <star>, ★ <blackstar>, ※ <snow>, ~ <swungdash>, ♪ <music> (ou <.|>), △ <triangle>, ▲ <blktriangle>, ▽ <invtriangle>, ▼ <invblktriangle>, □ <square>, ■ <blksquare>, ○ <circle>, ● <blkcircle>, °C <Celsius>, ° <degree>, ① <circleone>, ② <circletwo>, ③ <circlethree>, ④ <circlefour>, ⑤ <circlefive>, ⑥ <circlesix>, ⑦ <circleseven>, ⑧ <circleeight>, ⑨ <circlenine>, ⑩ <circleten>, ♂ <male>, ♀ <female>, + <plus>, = <equal> e / <slash> etc.

Esta é uma extensão fácil de implementar, que oferece a todos os idiomas acesso a símbolos úteis ou divertidos nas suas traduções também.

Tokens gramaticais

Para tornar o seu texto gramaticalmente correto e mais natural nos idiomas da tradução, é uma boa ideia implementar um sistema como o dos seguintes *tokens*. Separe e escolha o que precisa para atender às exigências do seu título. Para títulos com pouco texto, pode até ser razoável agregar o texto ao código e preparar *strings* exclusivas para cada caso. Mas para RPG ou MMORPGs grandes, onde as combinações de texto variáveis são praticamente ilimitadas, os sistemas de *tokens* são praticamente obrigatórios para se ter legibilidade e compreensão.

AVISO: A SQUARE ENIX tem os direitos de propriedade intelectual de algumas destas implementações (pedido de patentes japonês 2003-178063), então você pode até tentar entender os conceitos, mas terá que encontrar uma implementação diferente que se ajuste ao seu sistema. Por questão de segurança, não colocaremos nenhum tipo de código-fonte aqui, para prevenir infrações acidentais de direitos autorais.

(Dito isso, a SEGA tem a patente das telas de entrada de nome e de substituição do nome do jogador no texto de jogo, mas quase todos os jogos usam tais sistemas sem medo de ações legais. Então fica a critério dos desenvolvedores decidir qual implementação fazer e julgar os riscos associados).

Como antes, estamos usando <xxx> para indicar um *token*, mas você pode usar igualmente {xxx} ou /xxx ou qualquer coisa que funcione no seu sistema.

- ◆ O *token* <Cap> para transformar a primeira letra da variável seguinte em maiúscula.
- ◆ O *token* <CAP> para transformar em maiúsculas todas as letras do texto que o segue.
- ◆ *Tokens* <negrito> e <itálico> para acesso às versões em negrito e itálico da

fonte, se o seu sistema tiver suporte a elas.

- ◆ *Tokens* de cores, tal como <vermelho> ou <branco>, caso o seu sistema suporte texto colorido
- ◆ O *token* <esperar x>, para fazer uma pausa durante a exibição do texto que segue, por um certo número de quadros. É possível acrescentar um nível extra de "desempenho" ao texto, fazendo uma pausa para ênfase, que também pode ajudar a ajustar o timing das legendas com o diálogo falado.
- ◆ *Tokens* de artigo (por exemplo, a substituição do número 1 por *the, a, an* em inglês ou até mais combinações em português)
- ◆ *Tokens* de itens no singular/plural e coletivos. (Se tiver muitos nomes, provavelmente é uma boa ideia fazer um sistema com uma tabela de referência em vez de tentar apenas acrescentar um "s" automaticamente. Até mesmo em inglês há muitas exceções como *goose -> geese; genius -> geniuses; fish -> fish*. Não se esqueça também de tratar das exceções em casos de se contar um par de objetos como um.)
 - Observação: Idiomas asiáticos como o japonês e o chinês não usam muito singular/plural, então será necessário que os programadores para estes idiomas entendam conceitos não existem em seus próprios idiomas.
 - Observação: Tenha cuidado com o 0... na maior parte dos Idiomas europeus ele é tratado como um número plural, mas em francês, como singular. Não se esqueça de fazer o seu sistema lidar com esta exceção se o francês ou idiomas similares forem parte do jogo.
- ◆ *Tokens* gramaticais masculinos/femininos/neutros para lidar com coisas como "dela/dele/seu/sua" - como variações de gramática em idiomas europeus.
- ◆ *Tokens* que trazem textos diferentes dependendo se a primeira letra é uma vogal ou não.
 - Um teste simples com a primeira letra pode funcionar na maioria dos casos. Mas pode ser preciso estendê-lo no caso do francês para incluir o h's, y's silenciosos e outros casos especiais. Para grandes jogos, geralmente é melhor ter uma tabela de referência para todos os nomes que aparecem, indicando se cada nome deve ser tratado como um vogal ou consoante em vez de ensinar código para lidar com cada exceção em cada idioma.
- ◆ *Tokens* que trazem textos diferentes conforme a última letra seja uma vogal ou uma consoante. Isto é especialmente útil no idioma coreano, onde os artigos vêm depois do sujeito/objeto nas frases e têm que ser diferentes, dependendo de a última letra ser uma vogal ou não.

- ◆ *Tokens* que trazem textos diferentes conforme a última letra do texto seja um “s” ou um “S”. Estenda isto para o alemão para tratar também do z, Z, x, X, e ß (*shaffen-S*, o duplo s alemão). Muitos idiomas como o inglês têm regras de possessivos sobre o acréscimo do apóstrofe s ('s) à maior parte dos substantivos, exceto na maioria dos casos onde a palavra já termina em s, em que basta acrescentar o apóstrofe.
- ◆ *Tokens* que verificam se o texto substituído por uma variável é um substantivo próprio ou não.
- ◆ *Tokens* que lidam com os casos do alemão (nominativo / genitivo / dativo / acusativo) e declinações russas (singular / plural / reflexivo x 1o / 2o / 3o (Masculino / Feminino / Neutro) x nominativo / genitivo / dativo / acusativo / instrumental / preposicional).
- ◆ *Tokens* que exibem os contadores corretos depois de números em japonês, coreano e chinês. (Isto é, normalmente é preciso por um caractere depois de cada número para representar que tipo de coisa está sendo contada, portanto os pequenos animais peludos têm um tipo de contador, os animais maiores têm outro, os objetos planos têm um símbolo, pequenos itens têm outro ... um conceito que não existe em idiomas europeus). Uma marcação simples depois de cada nome na lista de nomes deve ser suficiente. Então você só tem a referência do *token* que marca para decidir qual contador usar.

Sugerimos que você fale com os seus tradutores sobre os requisitos para cada título e veja o que a capacidade, tempo e o orçamento de equipe de desenvolvimento permite.

Ainda que defendamos a gramática correta em todos os idiomas, também entendemos que um texto incorreto em outros idiomas pode ser uma decisão comercial.

Observação: mesmo com o acréscimo da beleza e da correção gramatical destes *tokens*, há o dilema da maior complexidade para o programador e os tradutores e o nível mais elevado de exigência para o QA -- que testa o jogo para ter certeza de que um sistema de texto mais flexível lida adequadamente com mais combinações de exibição de texto. Mas para MMORPGs, RPGs e outros jogos centrados em texto, isso faz uma diferença gigantesca para a compreensão e a qualidade do jogo. (qualquer situação que implique interação entre seres humanos, terá que ser clara sobre o que é singular/plural e quem está fazendo o quê).

TV PAL vs NTSC

Uma dica à parte para equipes de desenvolvimento que estão fazendo jogos para consoles e que podem ser exibidos em televisões antigas: não se esqueçam de levar em conta os padrões de televisão regionais na sua programação. Por exemplo, se desenvolverem jogos nos EUA ou no Japão, e depois decidirem lançar uma versão

européia, haverá problemas com taxas de quadros (60fps contra 50fps) e tamanhos de telas. Se você não programar de forma que o código seja executado independentemente da taxa de quadros da TV, sua versão PAL pode apresentar uma diminuição de 1/5 na velocidade do jogo, animações granulares e reprodução de vídeo pré-renderizadas e provavelmente haverá grandes faixas pretas nas partes de baixo e de cima da tela. Como o novo hardware dos consoles e das televisões HD digitais estão aliviando este problema, não entraremos em detalhes sobre como contornar este problema na programação ou como re-renderizar filmes etc. neste momento. Queremos apenas chamar a sua atenção para isso, pois é uma armadilha da localização em que muitas equipes de desenvolvimento caíram quando o ciclo de desenvolvimento já estava avançado demais.

Conselhos para equipes de desenvolvimento

Trabalhe com a sua equipe de localização o mais cedo que puder no seu projeto.

1. Se levar as necessidades de localização em conta durante o desenvolvimento, você reduzirá o volume de trabalho repetido para cada versão de idioma que será feita.
2. Ao tornar seus recursos no idioma de partida mais sistemáticos, você reduzirá o tempo necessário para traduzir o conteúdo, reduzindo erros de tradução e limites de linhas excedidos. Isso aumentará a qualidade e reduzirá o número de bugs que a sua equipe de desenvolvimento precisará dedicar tempo a corrigir, ao invés de poder trabalhar no próximo jogo.
3. Ao se trabalhar em formatos e processos padronizados, o tempo gasto pelas partes na localização é reduzido, pois os envolvidos não precisam se preocupar em reinventar a roda ou passar tempo adaptando-se às diferenças entre equipes de desenvolvimento.
4. Comunicação clara é essencial para habilitar que o seu produto seja aprovado pelo seu distribuidor regional, fabricantes de consoles e conselhos de avaliação regional. (p. ex., não se esqueça de passar aos tradutores orientações atualizadas da interface ou outros documentos dos fabricantes de consoles, conforme seja necessário, pois pode haver alterações a serem feitas na compilação e cópias finais que retardem a entrada no mercado.)

Lembre-se, a localização é o caminho para um público maior. Se você quiser que as pessoas de outras culturas gostem do seu jogo no idioma delas tanto quanto gostariam da versão original (e continuem comprando os seus produtos), tente dar o máximo de informações possível à sua equipe de tradução e esteja disposto a fazer alterações, se for necessário.

LOCALIZAÇÃO

A localização, abreviada como *L10N* em inglês, é o processo de criar um produto adaptado a uma localidade específica. Isto inclui a produção de recursos ou elementos localizados, implementando os elementos em uma compilação internacionalizada, testes, edição e correção de erros. Os elementos para localização podem incluir arquivos de recursos traduzidos, arquivos de *voice over* localizados, materiais de embalagem traduzidos e metadados traduzidos, entre outros.

O processo de localização de um videogame pode ser dividido nas seguintes etapas:

- ◆ Familiarização
- ◆ Criação do glossário e do guia de estilo
- ◆ Tradução
- ◆ Produção de *voice over*
- ◆ Controle de qualidade linguístico
- ◆ Masterização e finalização

A tradução, a edição, a gravação e CQ podem se sobrepor e acontecer paralelamente para otimizar o planejamento. Consulte o cronograma de amostra no Apêndice 4 para ajudar a visualizar como cada etapa pode ficar sobreposta.

Familiarização

Seja qual for o tamanho do projeto, os tradutores precisam ter lido ou visto o conteúdo antes de iniciarem a tradução, de modo a se familiarizarem com o assunto. Espera-se que um tradutor de romances ou de filmes tenha lido/visto a obra antes da tradução, então por que não esperar que os tradutores de um jogo recebam o contexto? Essa questão se torna muito mais complicada com a presença de múltiplas ramificações de história e arquivos de texto sem nenhuma ordem lógica. O tempo disponível para alocação e para quem o trabalho será alocado depende do tamanho do título e das limitações de orçamento. Jogar e ler a documentação de referência por no mínimo três dias é um bom ponto de partida para títulos pequenos, além de outros dois dias para familiarização com os títulos da franquia localizados anteriormente, quando apropriado. Para MMORPGs grandes, talvez seja necessário reservar pelo menos um mês para jogar todo o conteúdo (mesmo utilizando comandos de depuração ou “modo deus”). Se o título ainda estiver em produção, fornecer versões iniciais e documentos de concepção terá de bastar até que estejam disponíveis versões jogáveis.

Os tradutores devem registrar nomes e ideias à medida que avançam no conteúdo do jogo. Para acompanhar o ritmo deles, pode ser sensato pedir relatórios de

acompanhamento diários ou semanais. Se eles utilizarem comandos de depuração, também devem jogar um pouco sem utilizá-los para saberem quais elementos são importantes para a jogabilidade. Eles também devem jogar todos os modos e explorar todas as áreas do jogo, inclusive o modo multijogador e componentes on-line. Quanto mais jogarem durante esse estágio inicial, menos terão de interromper a tradução para perguntar sobre o jogo mais tarde. A lista de termos deve ser confrontada com o glossário debatido na seção seguinte para garantir sua abrangência e consistência.

Criação de glossários e guias de estilo

Quando os tradutores estiverem familiarizados com o conteúdo, a equipe de desenvolvimento deve fornecer o guia de estilo do idioma de partida e uma lista de termos comuns, inclusive as pronúncias, para que os tradutores possam usar como base na criação dos seus guias de estilo e dos seus glossários no idioma de chegada. Se for uma sequência de outro projeto, os materiais do título anterior devem ser fornecidos como referência para manter a continuidade da tradução. Se não houver tais documentos, os tradutores devem ser capazes de criá-los a partir dos arquivos-fonte e das anotações feitas durante a familiarização.

Para um jogo pequeno, pode ser preciso uma semana para nomear todos os elementos, criar um guia de estilo e caracterizar todos os personagens ou classes principais. Grandes MMORPGs podem levar até seis semanas e ainda não terem todos os nomes finalizados, mas se todas as políticas gerais e os trechos maiores de cada categoria estiverem nomeados e com estilo definido, o restante da tradução deve fluir rapidamente.

Os tradutores devem fazer uma troca conjunta de ideias, se dividindo em grupos especializados para títulos maiores, quando necessário, e levando os resultados de volta ao grupo. Dessa forma, há a adesão de toda a equipe e ela consegue ver o rumo do projeto como um todo. Mesmo as ideias que não entrarem na versão final dessa etapa devem ser guardadas como ideias secundárias, caso a equipe precise alterar os nomes mais tarde. Afinal, é mais rápido resgatar essas ideias secundárias que pensar em novos nomes. Além disso, as razões para as escolhas de nome devem ser registradas, especialmente quando a equipe estiver renomeando algo ou divergindo de forma deliberada sobre as caracterizações e os nomes originais. Esse processo poupará a equipe de ter que justificar constantemente suas posturas no futuro e, caso outros idiomas estejam traduzindo a partir de outra tradução (ou seja, uma tradução no estilo “telefone sem fio”), os tradutores desses idiomas poderão tomar decisões inteligentes sobre se manterem fiéis ao material original ou seguirem as versões provisórias de seus idiomas.

Sem um glossário e um guia de estilo de trabalho, será difícil para os tradutores entregarem um trabalho consistente. Para quaisquer novos nomes que surjam após a criação do glossário, sugerimos que o primeiro tradutor que se depare com um nome escolha uma versão temporária/funcional até que o grupo ou líder possa aprová-lo.

Depois os novos nomes devem ser acrescentados a um glossário atualizado.

As equipes de desenvolvimento devem estar disponíveis para responderem perguntas sobre o conteúdo. Quanto mais respostas e contexto forem dados pelos criadores originais, melhor será a execução do trabalho dos tradutores em seus idiomas de chegada para um público mais amplo.

O guia de estilo deve listar todas as regras de ortografia, gramática e pontuação aplicáveis ao título. Se alguma gramática ou dicionário estiver sendo usada como base para o estilo e para a ortografia, isso deve ser anotado, assim como deve haver um exemplar disponível para todos os tradutores tomarem como referência durante a tradução. Para as caracterizações, faça uma lista com explicações e amostras de como deve ser a voz de um personagem, e listas de escolhas de palavras e de sotaques ou problemas na fala, etc. (ex: não diga apenas “britânico” ou “americano padrão”, mas sim liste o período histórico exato, a região, os níveis linguísticos, etc., juntamente com amostras reais e escolhas de palavras. Seja específico. Não diga “Ele tem a voz parecida com a do Brad Pitt”, mas sim qual o personagem de cinema a que você se refere.)

Um glossário, guia de estilo e referências (gramáticas e dicionários) atualizados também devem ser fornecidos para o CQ como norma para comparação durante as etapas de teste.

Tradução

Quando os guias de estilo e os glossários estiverem prontos, os tradutores podem prosseguir para a tradução. Os tradutores devem conferir as traduções uns dos outros antes de finalizá-las. Se possível, recomenda-se ter editores ou revisores especializados que façam o controle para garantir a consistência. Mais que isso, um linguista deve sempre ter a palavra final ou ser a referência para quaisquer decisões que precisem ser tomadas.

Como regra geral fundamentada em experiência, há uma proporção de um editor para cada três tradutores de idiomas orientais e um editor para até quatro tradutores de idiomas europeus. Os editores não precisam ser tão fluentes no idioma de partida quanto os tradutores, já que devem focar principalmente no idioma de chegada. Entretanto, quanto mais conhecimentos tiverem no idioma de chegada, maiores também são as chances de encontrarem erros de tradução e problemas de consistência antes que o CQ os encontre.

Cada tradutor terá o seu próprio ritmo e os roteiros de áudio levarão mais tempo do que os textos da IU. Para fins de planejamento, considere que cada tradutor traduz 2.000 palavras em inglês (4.000 caracteres japoneses) por dia para textos padrões até ajustar o atual desempenho da sua equipe. Se houver algum áudio, considere que a velocidade dos tradutores para os textos com seções de voz caia pela metade, dependendo da quantidade de texto com sincronismo labial e com restrição de tempo

envolvidos. Mais ainda, as equipes começarão mais devagar antes de atingirem a sua verdadeira velocidade. Por isso, verifique o ritmo constantemente para ver o andamento do cronograma.

Se possível, tente fazer um teste de integração do texto ao jogo e ofereça aos tradutores a oportunidade de ver o texto traduzido na tela antes de o projeto ir para o controle de qualidade. Ver o texto com roteiro de áudio legendado na tela antes do início da gravação das vozes permitirá aos tradutores corrigir o texto antes das despesas da etapa de gravação. Isso também fornece alertas prévios de problemas envolvendo o tamanho do texto ou a sincronização do áudio.

Gravação de *voice over*

Se o título tiver áudio, geralmente será necessário traduzir primeiro o roteiro de gravação, assim como também permitir a realização de uma sessão extra no final do projeto a fim de gravar quaisquer novas falas ou solucionar problemas no áudio que surjam antes da finalização da localização. A direção de voz também deve ser traduzida (ou substituída por instruções localizadas quando necessário). Seja específico quanto às falas que precisam ter sincronismo temporal ou labial para se encaixarem no vídeo.

Conforme os jogos ficam maiores e mais complexos, cada vez torna-se mais difícil lidar com a quantidade de áudio que precisa ser gravada e monitorada. A equipe de desenvolvimento precisa garantir que estará apta a gerenciar grandes quantidades de arquivos em seus sistemas e que entregará de forma organizada todos os arquivos finais (com as cenas inutilizadas removidas) para a equipe de tradução.

Para a escalação de vozes, não hesite em localizar as vozes dos personagens de forma fidedigna. Por exemplo, os idiomas orientais podem dar preferência às vozes femininas agudas, ao passo que os idiomas ocidentais podem preferir vozes mais graves. Dessa forma, permita que cada equipe de tradução decida o que for melhor para seus respectivos idiomas em vez de forçá-los a soar de forma parecida. É raro as pessoas compararem as vozes de um idioma com as de outro, então é mais importante que essas vozes funcionem para o público-alvo de cada país.

Um tradutor deve estar sempre presente no momento da gravação para ajudar na explicação do material para o diretor de dublagem e para os dubladores. Além disso, como podem ser necessárias mudanças imediatas por conta do sincronismo e de outras questões, um tradutor precisa estar disponível para realizar tais mudanças. (Lembre-se de atualizar quaisquer alterações de improviso nas legendas.)

Para poupar erros caros, recomenda-se o seguinte:

Peça um orçamento

- ◆ Peça orçamentos com base em um escopo detalhado de áudio (contagem de

falas e palavras, número de dubladores, restrições de tempo por linha) para permitir que os fornecedores deem estimativas precisas. Orçar com base em outras informações, como o tamanho das gravações no idioma de partida, acrescentará variações significativas à estimativa.

Tradução e preparação de roteiro

- ◆ Certifique-se de que a direção de voz seja clara e reserve tempo/orçamento para que seja traduzida. (Talvez nem todos os estúdios de gravação sejam capazes de entender o idioma de partida ou os tradutores podem mudar a caracterização das vozes, então é uma boa ideia levar isso em conta.)
- ◆ Indique quais falas exigem sincronismo temporal ou labial antes da tradução e envie os arquivos de áudio em inglês com o roteiro. As falas com sincronismo temporal ocorrem durante eventos roteirizados dentro do jogo ou em outros momentos em que os diálogos e as ações do jogo devam se encaixar de forma aproximada. As falas com sincronismo labial ocorrem durante as cenas de transição ou em outros momentos em que os lábios dos personagens estão visíveis. É preferível que os movimentos labiais sejam ajustados para cada idioma (usando sistemas como o Face FX). No entanto, não é incomum um jogo utilizar as cenas com os lábios animados para falas em inglês em todos os idiomas. Repare que a gravação das vozes pode ser até quatro vezes mais longa por conta do sincronismo temporal e labial.
- ◆ Para falas com sincronismo labial, mande vídeos de referência para que seja feita a sincronia pela equipe de tradução e pelas equipes de gravação. Se não houver nenhum vídeo de referência disponível, a equipe de localização pode tentar encaixar sílabas que se assemelhem ao sincronismo labial.
- ◆ Indique quais falas podem ser gravadas de forma livre para garantir que todas as falas sejam entregues com a restrição de gravação apropriada (temporal, labial ou livre).
- ◆ Além disso, esclareça a discrepância existente entre o tamanho do idioma de chegada e do original. Se for necessário que todas as falas encaixem nos movimentos labiais ou temporais do original, pode-se tornar custoso e levar um bom tempo para gravar todo o áudio. Contudo, se o personagem do áudio não aparecer e o tempo das falas for flexível, essa liberdade permitirá que a tradução e a gravação dessas falas sejam mais rápidas.
- ◆ Tome cuidado para que a convenção de nomeação dos arquivos seja lógica, de forma que os estúdios de gravação ao redor do mundo não fiquem procurando por arquivos nomeados aleatoriamente quando os estiverem reordenando entre os personagens e as cenas.
- ◆ Escolha todas as cenas que serão usadas no jogo o mais cedo possível e

providencie para que o roteiro final e as cenas finais coincidam! (Geralmente, as alterações de improviso são implementadas apenas na programação do jogo e não são passadas para os roteiros originais, dando dor de cabeça para tentar descobrir o que é o certo.)

- ◆ Evite reutilizar arquivos de voz entre as cenas. (Mesmo que o reaproveitamento do mesmo arquivo funcione em diversas situações no idioma de partida, para os idiomas de chegada talvez seja preciso alterar as traduções para explicar as situações que envolvam singular/plural ou masculino/feminino/neutro.)
- ◆ Formate os roteiros para deixar claro com quem o dublador está falando ou interagindo. Isso permite que os estúdios tenham a oportunidade de reutilizar dubladores em múltiplas cenas ou arquivos, permitindo-lhes evitar sequências em que um dublador faça a voz de dois personagens que estão conversando um com o outro.
- ◆ Não suponha que os esforços do idioma de partida possam ser reutilizados em outros idiomas quando não contiverem palavras distintas. Os sons das reações variam enormemente entre os idiomas e as culturas. Acrescente esses arquivos ao roteiro e peça para que a equipe de localização considere sua reutilização em todos os idiomas.

Escalação de vozes internacionais

- ◆ Crie um pacote de escalação de vozes para a equipe de localização que contenha as descrições dos personagens (idade, sexo, tipo de voz, histórico do personagem, voz parecida, etc.), imagens dos personagens e amostras de voz.
- ◆ Se utilizarem dubladores famosos no idioma de partida, decida quais versões localizadas precisarão de voice talents famosos. Essa situação surge geralmente em jogos ligados a séries de televisão ou filmes localizados em mercados estrangeiros. Frequentemente, paga-se mais caro aos voice talents famosos e eles terão uma disponibilidade reduzida, afetando potencialmente o orçamento e o cronograma do projeto.
- ◆ Peça um banco de dados ou faça a escalação de vozes ao vivo para os personagens principais do jogo. Utilizar um banco de dados com vozes é muito mais rápido e barato e, normalmente, os estúdios enviam até três amostras de voz para que a escolha seja feita. Ponha um falante nativo ou especialista em escalação de vozes para revisar as amostras de voz e para haver uma coerência com os personagens.
- ◆ Permita que os estúdios de gravação tenham uma certa liberdade para fazer a escalação dos personagens menos importantes do jogo. Devido à complexidade dos custos e cronogramas de gravação, isso os permitirá reutilizar dubladores para papéis secundários, poupando tempo e dinheiro. Essa é uma das razões pela

qual é importante especificar o público-alvo no roteiro – de modo a evitar uma situação em que um ator que esteja interpretando dois personagens fale com ele mesmo quando não for a intenção.

Especificações para arquivos de áudio

- ◆ Forneça arquivo(s) em inglês com volume balanceado para que os estúdios internacionais normalizem o volume usando-os como referência.
- ◆ Entregue os arquivos em inglês na estrutura que você deseja receber os arquivos internacionais. Você pode pedir para que os estúdios acrescentem códigos de idioma ao final do nome de cada arquivo.
- ◆ Determine o formato, fluxo de transferência de bits, frequência e número de canais em que você deseja que os estúdios de gravação internacionais devolvam o áudio.
- ◆ Liste todos os efeitos de pós-processamento e requisitos para os arquivos de áudio. Alguns personagens podem precisar de reverberação ou mudança de tom; outros podem precisar de efeitos especiais diferentes, criados por um designer de áudio.
- ◆ Permita que o volume dos arquivos de voz seja alterado entre os idiomas. (Mesmo que você possa determinar o nível do volume geral, quando começar a lidar com grandes quantidades de áudio, é inevitável que alguns arquivos precisem de ajuste de volume. O melhor é deixar que o seu sistema de áudio o faça, em vez de pedir que os estúdios de gravação ou o seu departamento de som corrijam o volume de cada arquivo manualmente.)

Controle de qualidade linguístico

A equipe de *testers* deve ser composta por uma soma de bons linguistas em cada idioma local aliados a bons jogadores. O tamanho da equipe varia de acordo com o tamanho do projeto e com o cronograma. Enquanto algumas empresas permitem que os *testers* corrijam os bugs encontrados, se o orçamento/cronograma permitir, é melhor que os tradutores originais “assumam a tradução”, para que apenas tradutores façam alterações. É comum que haja alterações ou caracterizações deliberadas que possam ser confundidas com bugs. Todavia, o tradutor precisa justificar tais alterações deliberadas ao abrir mão de bugs. Permita que os *testers* também contribuam com sugestões gerais (mas não as misture com os bugs, priorizando assim os problemas mais importantes). “Regressar” ou confirmar que o bug foi, de fato, corrigido (sem nenhum efeito colateral inesperado!) deve ser feito mais de uma vez em versões diferentes do jogo. É comum que, acidentalmente, algum texto antigo seja implementado numa versão nova do jogo, então é sempre importante verificar de novo!

Também é importante ter uma política clara: por exemplo, quando o CQ e os tradutores

não concordam entre si, a última palavra cabe ao tradutor principal.

Masterização e finalização

Recomenda-se que um tradutor esteja presente em caso de mudanças de última hora durante a etapa de masterização. Além do CQ, um ou mais tradutores também devem jogar o jogo todo para confirmar se a tradução foi integrada corretamente e se está tudo pronto para a produção e lançamento.

Durante essa fase, além da tradução de documentos e da embalagem, haverá uma grande quantidade de traduções promocionais e checagem do guia. Portanto, tenha um linguista de plantão após o jogo ser finalizado.

Dicas para MMORPGS...

Considere cada atualização de conteúdo como um ciclo de localização único e siga as etapas listadas acima em menor escala. Isso significa que os tradutores precisam se familiarizar e trocar ideias a respeito do novo conteúdo.

Garanta que a equipe de desenvolvimento notifique os tradutores sobre quais seções devem ser traduzidas a fim de evitar traduções desnecessárias. (Você pode, por exemplo, usar um esquema de semáforo: Verde = texto no idioma de origem está consistente, portanto, pronto para ser traduzido; Laranja = ainda pode haver algumas alterações mínimas, mas pode ser traduzido caso não haja mais textos em Verde disponíveis. Vermelho = traduzir por sua conta e risco, já que o texto de origem ainda não está consistente e pode ser reescrito.) Monitore a razão entre o progresso dos tradutores e dos autores do texto original e garanta que os autores parem de acrescentar conteúdo para que tradutores e CQ tenham tempo de acompanhar.

Os tradutores tendem a se especializar em determinadas áreas em grandes títulos (um trabalhando nos textos de missões ou num sotaque regional específico; outro, nos nomes de itens; outro cuidando da interface). Tenha em mente que, em projetos de longa duração, o cansaço é algo comum e, para manter a criatividade e a motivação, permita que eles alternem papéis ou até mesmo projetos. É nesse ponto que bons guias de estilo, glossários e períodos de treinamento/transição ajudarão a manter a consistência e a qualidade.

PLANEJAMENTO DE PROJETO

Esta seção mapeará as práticas recomendadas discutidas anteriormente para o processo de desenvolvimento típico de um jogo. Como os processos e expectativas variam bastante de empresa para empresa e as metodologias aplicadas em diferentes plataformas de distribuição estão em constante mudança, isso deve ser encarado apenas como um guia.

Pré-produção

- ◆ Esta é a fase do projeto na qual o plano de lançamento é decidido – quais as plataformas, os idiomas, o orçamento inicial do projeto e a data de lançamento.
- ◆ Quando existir um documento de tratamento inicial, documento de design e arte conceitual, você pode começar a revisar possíveis problemas culturais no produto.
- ◆ Selecione fornecedores de localização, teste, revisão de culturalização, firme contratos e monte sua equipe.

Produção

- ◆ Dê início ao projeto de localização – chegue a um acordo sobre um processo e um cronograma inicial, incluindo uma atualização de localização para a equipe de desenvolvimento, caso necessário. O cronograma inicial deve incluir a conclusão do roteiro e a entrega do áudio em inglês (ambos devem ocorrer nesta etapa). O cronograma também deve incluir ciclos de teste linguísticos e um cronograma de uma versão provisória.
- ◆ Inicie a familiarização e a preparação de um glossário e um guia de estilo o quanto antes.
- ◆ Após as biografias dos personagens estarem disponíveis e os áudios em inglês gravados, você pode iniciar o processo de escalação de vozes. Trabalhe com seu fornecedor de localização para determinar a quantidade de linhas, contagem de palavras aproximada para cada personagem e para distinguir os personagens principais dos secundários. Conceda pelo menos 2 semanas após a escalação de vozes e antes da gravação internacional para que o seu fornecedor reserve o dublador.
- ◆ Execute os processos de tradução de roteiro e de gravação de vozes internacionais; conceda tempo suficiente para o balanceamento de volume, pós-processamento e verificação de nome de arquivo antes da implementação na primeira versão internacional.
- ◆ Comece traduzindo a interface e o texto do jogo. Lembre-se de que muitos tradutores trabalham num ritmo de 2.000 palavras por dia e, quanto mais tempo

you concede para que o inglês seja editado, menos mudanças serão necessárias na próxima fase para cada idioma.

- ◆ Trabalhe com a equipe de desenvolvimento para determinar o que é necessário para uma versão internacional, incluindo tecnologia para importar e exportar os textos com facilidade. Faça um teste de uma atualização de texto e de criação de uma nova versão.
- ◆ No final da fase de produção, você deve ter uma versão consistente com todo ou com a maior parte do texto localizado e áudio implementado.

Estágio alfa

- ◆ Execute o processo de teste de internacionalização e garanta que todos os recursos localizados estão funcionando, todas as fontes funcionando em cada idioma e que a interface de usuário permite traduções longas. Algumas equipes de desenvolvimento criam uma versão monolíngue, usando as maiores *strings* dos idiomas para gerar “o pior cenário” de teste na interface de usuário.
- ◆ Assim que tiver uma versão funcionando e com todos os bugs de internacionalização resolvidos, você pode iniciar o controle de qualidade da localização. Esta será provavelmente a etapa mais ocupada do projeto, com correções de bugs ao lado de novas solicitações de tradução. Manter atualizado um cronograma do trabalho a ser feito e comunicar-se com todos os parceiros diminuirá o atrito que pode ser gerado nessa fase.
- ◆ Faça um acordo com a equipe de desenvolvimento sobre uma data para concluir todas as alterações no inglês. Já que a localização depende do texto original, é possível que esta pare caso o inglês continue sofrendo alterações.
- ◆ Caso necessário, execute uma segunda rodada de gravação do *voice over* internacional para identificar os principais bugs ou novas falas adicionadas ao roteiro.

Beta / Finalização da localização / Etapa da cópia final

- ◆ Assim que todas as solicitações de tradução estiverem concluídas e todos os problemas de localização resolvidos, o produto já está pronto para ser lançado internacionalmente. Parabéns!
- ◆ Antes que passe muito tempo, realize uma breve inspeção e tome nota de ideias úteis para projetos futuros.

APÊNDICE 1 - LISTA DE PRÁTICAS RECOMENDADAS

Culturalização

1. Defina a profundidade da culturalização desejada para o produto, tornando-o viável, claro e/ou significativo.
2. Tenha em mente as quatro variáveis culturais principais que se aplicam aos seus mercados-alvo: história, religião, etnias e imaginário geopolítico.
3. Identifique possíveis problemas de culturalização, avalie sua gravidade e implemente as alterações no conteúdo o mais cedo possível no ciclo de desenvolvimento.
4. Seja preciso em suas alterações de conteúdo. Faça apenas alterações mínimas que garantam a distribuição nos mercados-alvo e que evitem problemas após o lançamento.

Internacionalização

5. Seleção de fonte e dimensionamento é um esforço conjunto entre as equipes de localização e de desenvolvimento.
6. Use fontes proporcionais aos idiomas norte-americanos e europeus. Idiomas asiáticos exigem largura fixa.
7. Crie uma interface que permita traduções longas ou ofereça um sistema para reposição de texto em cada idioma.
8. Para áreas em que o texto tenha várias linhas, ative a quebra de texto e permita a opção de rolagem, caso as traduções sejam muito longas.
9. Para áreas com texto em única linha, ative o ajuste de espaço para comprimir traduções longas.
10. Para legendas, use fontes sem serifa com cor clara e contorno escuro. Posicione as legendas no terço inferior da tela, exiba-as por três segundos e não use mais de duas linhas por vez. Indique alterações de interlocutor com um hífen no início da fala.
11. Armazene os recursos de localização em uma pasta de fácil acesso para permitir a importação e exportação direta.
12. Use códigos de idioma e região ISO 639-1 e 3166-1 para identificar recursos

destinados a um local específico.

13. Destine tempo ou orçamento para desenvolver ferramentas que automatizem as tarefas de localização.
14. Para recursos de texto, escolha uma codificação que suporte todos os idiomas de chegada.
15. Prepare-se para trocar recursos de arte na interface de usuário ou no mundo do jogo caso contenham texto legível.
16. Arquivos de recursos devem armazenar todo o texto usado no jogo; não utilize *strings* de texto fixas no código fonte.
17. Processe os arquivos de recursos em arquivos de idiomas separados para que os tradutores trabalhem em paralelo. Indique as *strings* que são específicas da plataforma para garantir que as traduções obedeam aos termos do fabricante de consoles.
18. Para cada versão localizada, forneça uma lista de alterações no idioma de origem.
19. XML, ou um de seus formatos derivados, é preferível para tradução em vez de arquivos de texto simples ou de Excel.
20. Permita que a ordem de variáveis em *strings* localizadas seja alterada de acordo com o idioma e implemente um sistema de *tokens* para garantir que substantivos, verbos, artigos e outros sejam colocados na ordem adequada para cada idioma.
21. Permita que data, hora, moeda e números sejam exibidos com diferentes unidades monetárias e separadores de números por idioma.
22. Permita que as tabelas sejam ordenadas por idioma, já que traduções podem alterar a ordem alfabética.
23. Implemente um recurso automático de quebra de linha e de texto para caixas de texto com várias linhas. As regras que determinam onde a quebra de linha deve ocorrer variam de acordo com o idioma. Permita que o sistema de quebra de linhas seja anulado pelo uso manual de caracteres como um espaço sem quebra (0xA0).
24. Forneça suporte para modo PAL 60Hz caso deseje distribuir seu jogo na Europa.

Localização

25. Garanta que as equipes de tradução tenham um tempo de familiarização com o jogo antes da tradução. Materiais a que elas devem ter acesso incluem: versões, roteiros, arte conceitual, documentos de design, tratamentos ou versões anteriores, caso

existam.

26. Os tradutores devem criar um registro de nomes e ideias enquanto se familiarizam com o conteúdo do jogo.
27. Forneça o guia de estilo do idioma de origem, glossário e pronúncia dos termos importantes à equipe de tradução. As equipes de tradução desenvolverão guias de estilo e glossários para cada idioma de destino.
28. Os textos no idioma de origem devem passar por uma verificação de ortografia, gramática e estilo antes de enviar um pedido de tradução.
29. Planeje um ritmo de tradução de pelo menos 2.000 palavras por dia e reserve um dia para que os tradutores solicitem informações adicionais, quando necessário.
30. Realize um teste de integração das traduções em uma versão anterior ao CQ linguístico. Quando possível, permita que os tradutores visualizem as legendas antes da gravação do áudio.
31. Tente gravar o áudio localizado em uma única sessão para garantir que o mesmo diretor de áudio esteja disponível para cada idioma e para minimizar os custos de aluguel do estúdio. Planeje uma sessão extra de emergência no final do projeto para corrigir bugs graves ou verificar novas linhas no texto original adicionadas ao jogo.
32. Forneça informações de escopo detalhadas aos fornecedores de localização com o intuito de obter orçamentos de gravação precisos. Eles precisarão, no mínimo, da contagem aproximada de falas, quantidade de atores e restrições quanto ao tipo de gravação.
33. Roteiros de áudio devem incluir: nomes dos arquivos, interlocutor, texto em inglês, restrições de gravação (tempo de sincronia, sincronia dos lábios ou livre). Deixe o roteiro em ordem cronológica para que os diálogos possam ser entendidos no contexto e os participantes possam ser identificados (indispensável para atores acostumados a interpretar múltiplos personagens).
34. Pacotes de elenco para fornecedores de localização devem incluir: descrições dos personagens, imagens e exemplos de vozes no idioma de origem.
35. Permita que os fornecedores selecionem o elenco de personagens sem importância para que o cronograma de gravação seja mais fácil de gerir.
36. Planeje pelo menos 2 semanas para reservar os atores após o elenco internacional ser escolhido.
37. Especifique os formatos dos arquivos de áudio, nomes dos arquivos, estrutura das pastas e efeitos pós-processamento que você deseja receber dos fornecedores. Forneça também um arquivo de origem com volume balanceado para que os estúdios normalizem o volume usando-os como referência.

38. Selecione uma equipe de controle de qualidade composta por um conjunto de bons linguistas em cada idioma de destino aliados a bons jogadores.
39. Bugs escritos nos recursos de localização devem ser direcionados ao fornecedor de localização, e bugs escritos na internacionalização devem ser direcionados à equipe de desenvolvimento.
40. A equipe de controle de qualidade linguístico deve voltar e confirmar se todos os bugs foram corrigidos.
41. Durante as etapas finais de um projeto, mantenha a equipe de plantão para garantir uma solução rápida para quaisquer bugs de última hora ou alterações de conteúdo.
42. Quando todo o conteúdo localizado estiver funcionando de forma adequada no jogo e todos os bugs de localização tenham sido resolvidos, solicite à equipe de controle de qualidade linguístico para dar aval à qualidade do jogo.

APÊNDICE 2 - EXEMPLOS DE CULTURALIZAÇÃO

1. História: passado e presente

1. **Age of Empires (1997):** Em Age of Empires, criou-se um cenário no qual o exército japonês do período Yamato invadiu a península da Coreia e derrubou o regime Choson. Os game designers replicaram minuciosamente essa realidade, confirmada por historiadores. No entanto, o governo da Coreia do Sul tinha uma visão diferente da história, contestou a veracidade do cenário e não permitiu a comercialização do jogo. Para mudar sutilmente o cenário, foi desenvolvido um patch para download que mostrava a invasão Yamato como algo menos absoluto, isto é, os exércitos Choson ganharam uma chance de lutar no mundo do jogo.
2. **Six Days in Fallujah (2009):** Este título reproduziu os eventos da Segunda Batalha de Fallujah, no Iraque, pelo governo dos Estados Unidos no final de 2004, na qual tropas de Estados Unidos, Inglaterra e Iraque tentaram derrotar os insurgentes usando Fallujah como base. A luta foi controversa e teve um alto custo para ambos os lados. As forças da coalizão contaram mais de 100 baixas e 600 feridos, enquanto os insurgentes ultrapassaram a marca de 1.300 perdas. Se diversas décadas tivessem se passado desde o evento, sem dúvida alguma este seria um ótimo enredo para um jogo. Porém, a natureza recente dos fatos criou tal controvérsia que a Konami desistiu de lançar o título.

2. Religião e sistemas de crenças

1. **Kakuto Chojin (2002):** Este jogo de luta foi visto como uma importante adição à biblioteca original de jogos do Xbox. Infelizmente, foi incorporada ao jogo uma breve faixa de áudio que continha um trecho recitado do Alcorão. O erro foi consertado, mas algumas cópias não corrigidas chegaram às prateleiras, e o problema veio a público em algumas semanas. O jogo foi banido na Arábia Saudita e em outros países muçulmanos. A reação negativa foi tamanha, que levou ao cancelamento mundial do produto. Todo o trabalho árduo e as boas intenções por parte da equipe de desenvolvimento foram jogados fora, em resumo, por causa de um só elemento.
2. **Resistance: Fall of Man (2006):** Quando este jogo foi lançado no Reino Unido, a Igreja da Inglaterra ficou chocada ao descobrir que a Catedral de Manchester foi recriada em seus mínimos detalhes no mundo do jogo. No entanto, ela não foi apenas recriada, mas a violência foi retratada na própria catedral, algo considerado extremamente perturbador pela Igreja. Por fim, a Sony emitiu um pedido de desculpas e a Igreja da Inglaterra publicou uma nova edição das "Sacred Digital Guidelines" (Diretrizes Digitais Sagradas) para ajudar os desenvolvedores de videogames e outros a respeitarem suas estruturas religiosas.

3. Etnia e atritos culturais

1. **Resident Evil 5 (2009):** Até mesmo antes de seu lançamento, este título gerou publicidade negativa devido ao racismo velado. No jogo, o protagonista branco e certinho é visto percorrendo um vilarejo na África subsaariana, atirando contra habitantes desarmados e obviamente pobres. Embora a produtora Capcom foi rápida em informar que os aldeões africanos eram zumbis infectados, a forte imagem de um homem branco matando os habitantes evocou imagens negativas marcantes. Os conceitos "grande caçador branco" e "África, o continente negro", entre outros, logo vieram à mente de muitas pessoas. Embora os desenvolvedores tivessem uma lógica clara para o conflito no contexto do jogo, a reação negativa foi razão suficiente para a produtora parar e refletir se simular esse tipo de imagens negativas seria apropriado.
2. **Pocket God (2009):** Neste jogo para iPhone, o jogador é o "deus" de uma pequena ilha fictícia e pode atormentar pequenos nativos por meio de atividades como usá-los de comida para tubarões ou formigas assassinas, deixá-los cair de grandes alturas, fazer um vulcão expelir lava quente neles e assim por diante. O desenvolvedor do jogo, Bolt Creative, deixou claro que a intenção do jogo não era a de retratar uma nacionalidade específica. No entanto, os diversos artefatos na ilha (incluindo as estátuas moai de cabeças da Ilha de Páscoa), as vestimentas dos nativos e seu tom de pele bronzeado foram o suficiente para que defensores dos habitantes das ilhas do Pacífico fizessem reclamações e protestos contra o jogo pelo uso descarado de estereótipos étnicos "primitivos".

4. Imaginações geopolíticas

1. **Hearts of Iron (2004):** No jogo Hearts of Iron e suas sequências, o mapa estava dividido em setores arbitrários, semelhante ao clássico jogo de tabuleiro Risco. Ambientado durante a Segunda Guerra Mundial, Hearts of Iron retrata uma China dividida em diversas partes distintas, incluindo o Tibete e Taiwan como regiões separadas. Por causa disso, o governo da China banuiu o jogo, mesmo que o contexto do jogo fosse a Segunda Guerra Mundial, quando o Tibete e Taiwan ainda não faziam parte da República Popular da China (que só se formou em 1949, sendo que Taiwan não faz parte da China continental no momento). Os fatos históricos e geopolíticos ficaram em segundo plano diante da necessidade de o governo atual reforçar sua própria percepção do território em cada contexto possível.
2. **Ghost Recon 2 (2004):** Na Coreia do Sul, o Korea Media Rating Board (KM RB, Conselho de Classificação de Mídia da Coreia), agora Game Rating Board (GRB, Conselho de Classificação de Jogos), banuiu este título porque o jogo retratava um general norte-coreano hostil. O governo sul-coreano se opõe com firmeza ao fato de seu vizinho ser retratado como um agressor devido à tensão contínua

entre ambos os países, além do longo interesse na reunificação com o norte por parte da Coreia do Sul. O KMRB também rejeitou a venda de Mercenaries em 2005 pelos mesmos motivos. Em 2007, o GRB mudou de opinião devido à pressão de gamers locais sobre a liberdade de expressão e agora permite títulos como Ghost Recon 2.

APÊNDICE 3 - EXEMPLO DE *TOKENS* GRAMATICAIS

Veja aqui uma explicação mais detalhada com nomes de *token* reais (mas sem o código-fonte, embora não seja difícil implementar para a maioria dos programadores).

Imagine uma tabela de itens em inglês:

SGL INDEF	SGL DEF	NOME ITEM SGL		PLR INDEF	PLR DEF	NOME ITEM PLR	
uma	a	a	espada	(alg)umas	as	as	espadas
um	o	o	machado	(alg)uns	os	os	machados
uma	a	uma	poção			(alg)umas	poções
uma	a	a	Masamune			as	Masamunes
uma	a	a	Excalibur			as	Excaliburs
...	

Nesta tabela, podemos definir os seguintes *tokens* que utilizam dados de uma ou mais colunas em tempo real conforme o contexto e o desenrolar do jogo. (Usamos "xxx" e "itemx" aqui, mas é possível usar qualquer variável do seu sistema.)

- <SGL_I_NOME_xxx>

Imprime a forma singular do nome do item. (É basicamente o *token* de item padrão em todos os jogos até agora.)

Por exemplo: Você ganhou 1 <SGL_I_NOME_itemx>.

→ Você ganhou 1 espada.

- <PLR_I_NOME_xxx>

Imprime a forma plural do nome do item.

Por exemplo: O ladrão roubou todas as suas <PLR_I_NOME_itemx >!

→ O ladrão roubou todas as suas espadas!

- <INDEF_ART_SGL_I_NAME_xxx>

Imprime o artigo indefinido seguido pelo nome do item singular.

Por exemplo: Richard encontrou <INDEF_ART_SGL_I_NOME_itemx >.

→ Richard encontrou uma espada.

→ Richard encontrou um machado.

→ Richard encontrou a Masamune.

→ Richard encontrou a Excalibur.

- <DEF_ART_SGL_I_NOME_xxx>

Imprime o artigo definido seguido pelo nome do item singular.

Por exemplo: Richard coloca <DEF_ART_SGL_I_NOME_itemx > na mochila.

→ Richard coloca a espada na mochila.

→ Richard coloca o machado na mochila.

→ Richard coloca a Masamune na mochila.

→ Richard coloca a Excalibur na mochila.

- <INDEF_ART_PLR_I_NOME_xxx>

(Provavelmente não é necessário para o inglês, certo? Mais para francês, italiano, alemão e espanhol.)

Imprime o artigo indefinido plural seguido pelo nome do item plural.

Por exemplo: Richard encontrou <INDEF_ART_PLR_I_NAME_itemx >.

→ Richard encontrou algumas espadas.

→ Richard encontrou alguns machados.

(→ Richard encontrou algumas Excaliburs. Provavelmente nunca acontece, apenas para fins de demonstração.)

- <DEF_ART_PLR_I_NOME_xxx>

(Provavelmente não é necessário para o inglês, certo? É mais para francês, italiano, alemão e espanhol.)

Imprime o artigo definido plural seguido pelo nome do item plural.

Por exemplo: Richard coloca <DEF_ART_PLR_I_NOME_itemx > na mochila.

→ Richard coloca as espadas na mochila.

→ Richard coloca os machados na mochila.

(→ Richard coloca as Excaliburs na mochila. Provavelmente não ocorrerá, apenas para fins de demonstração.)

- <NUM_I_NOME_xxx, val_yyy>

Imprime o número no segundo parâmetro val_yyy, seguido pela forma singular se o

número for igual a 1; caso contrário, o plural será usado.

Por exemplo: Compre <NUM_I_NOME_itemx, 1>.

→ Compre 1 machado.

Por exemplo: Compre <NUM_I_NOME_itemx, 2>.

→ Compre 2 machados.

- <DEF_ART_NUM_I_NOME_xxx, val_yyy>

Se o número no parâmetro val_yyy for igual a 1, imprima o artigo definido singular seguido pela forma singular do nome; caso contrário, imprima o número em val_yyy seguido pela forma plural do nome.

Por exemplo: Richard coloca <DEF_ART_NUM_I_NOME_itemx, 1> na mochila.

→ Richard coloca a espada na mochila.

Por exemplo: Richard coloca <DEF_ART_NUM_I_NOME_itemx, 5> na mochila.

→ Richard coloca 5 espadas na mochila.

(Podemos até fazer um *token* semelhante que imprima o artigo definido plural em vez de imprimir o número da forma plural.

Por exemplo: "Richard coloca as espadas na mochila.")

- <INDEF_ART_NUM_I_NOME xxx, val_yyy>

Se o número no parâmetro val_yyy for igual a 1, imprima o artigo indefinido singular seguido pela forma singular do nome; caso contrário, imprima o número em val_yyy seguido pela forma plural do nome.

Por exemplo: Richard compra <INDEF_ART_NUM_I_NOME_itemx, 1>.

→ Richard compra uma espada.

→ Richard compra um machado.

Por exemplo: Richard compra <INDEF_ART_NUM_I_NOME_itemx, 5>.

→ Richard compra 5 espadas.

→ Richard compra 5 machados.

(Podemos até fazer um *token* semelhante que imprima o artigo indefinido plural em vez de imprimir o número da forma plural.

Por exemplo: "Richard compra um machado." e "Richard compra alguns machados.")

- *TOKEN*: MASCULINO/FEMININO

<SE_MASCULINO>...<SENÃO_NÃOMASCULINO>...<FIMSE_MASCULINO>

Se o líder do grupo for um homem, imprime a primeira parte '...', senão imprime a segunda parte '...'.

p.ex. " Olá <SE_HOMEM>menino lindo<SENÃO>menina linda<FIMSE>!", se o personagem principal for homem, então imprime "Olá, menino lindo!", caso contrário imprime "Olá, menina linda!"

Com certeza há algum trecho em que isso pode aumentar a profundidade do texto, e isto pode ser expandido para os outros participantes, se necessário. É que o líder do grupo é na maioria das vezes o mais mencionado nos RPGs.

- *TOKEN*: SOLO/VÁRIOS MEMBROS

<SE_SOLO>...<SENÃO_NÃOSOLO>...<FIMSE_SOLO>

Se o grupo tem apenas um membro ativo, então imprime a primeira parte '...'; senão imprime a segunda parte '...'.

Em inglês, é provável que possamos usar o *you* já que ele abrange situações tanto no singular quanto no plural.

(Já japonês, chinês e coreano apresentam o plural "vocês", e alguns idiomas europeus mudam tanto substantivos para "vocês" quanto sua circundante gramática. Então, isso ajuda muito.)

Se o seu grupo tem apenas um membro, você pode fazer o seguinte.

p.ex. 1) Ei, <SE_SOLO>você<SENÃO_NÃOSOLO>vocês aí<FIMSE_SOLO>!

p.ex. 2) <SE_SOLO> Deveria consertar sua cabeça. <SENÃO_NÃOSOLO> Deveriam consertar a cabeça de vocês.

- *TOKEN*: SINGULAR/PLURAL

<SE_CANTAR val_xxx>.<SENÃO_NÃOCANTAR>.<FIMSE_CANTAR>

Um *token* maravilhoso que pode ser usado em qualquer lugar que tenha um valor numérico.

Se val_xxx==1, imprime a primeira parte '...', senão imprime a segunda parte '...'.

Uso óbvio: Você recebeu <val_1> <SE_CANTAR val_1>pedaço<SENÃO_NÃOCANTAR>pedaços<FIMSE_CANTAR> de ouro.

(Ou: "Você recebeu <val_1> moeda<SE_CANTAR val_1><SENÃO_NÃOCANTAR>s<FIMSE_CANTAR> de ouro.", se preferir concisão

em vez de clareza.)

→ "Você recebeu 1 moeda de ouro"

→ "Você recebeu 2 moedas de ouro"

Até frases comuns podem ser melhoradas.

Por exemplo: se <val_2> contém um valor representando um número de meninos, use então: <SE_CANTAR val_2>Aquele menino nunca vai crescer.<SENÃO>Aqueles meninos nunca vão crescer<FIMSE>.

→ Aquele menino nunca irá crescer.

(se <val_2> ==1); → Aqueles meninos nunca vão crescer... (caso contrário).

Em alguns jogos fomos até capazes de lidar com sentenças loucas como essas.

<Cap><ART_NUM_H_NOME_mês_no, núm_de_meses> <SE_CANTAR núm_de_meses>me deu<SENÃO>me deram<FIMSE> <PLR_I_NOME item_no> <SE_CANTAR núm_de_meses>dele<SENÃO>deles<FIMSE>. (para obter todos os tipos de saída numa mesma linha).

→ Um duende me deu suas maçãs.

→ Cinco duendes me deram suas maçãs.

→ Uma anaconda me deu suas poções.

→ Três anacondas me deram suas poções.

- DIFERENÇA DE NOMES DE ITENS MASCULINO/FEMININO/NEUTRO NA GRAMÁTICA

<SE_NOME_xxx_M>.<SE_NOME_XXX_F>...<SE_NOME_xxx_N>...<FIMSE_NOME_XXX>

(Onde xxx é a variável que estamos testando.)

Para idiomas europeus, adicionamos uma coluna ao lado do item de nomes para indicar M/F/N, para podermos usar o *token* acima em combinação com este dado para criar sentenças mais naturais.

Por exemplo: imagine que você receba um <item_1> e queira usar "este" para se referir a ele.

Em português seria...

Coloque isso em sua bolsa.

Em alemão poderíamos escrever...

*Du tust<SE_NOME_1_M> ihn<SE_NOME_1_F> sie<SE_NOME_1_N>
es<FIMSE_NOME_1> in deine Tasche.*

É claro que se em seu idioma há apenas gênero masculino e feminino e não há gênero neutro, então escreveríamos somente M ou F na coluna ao lado do nome. Na tradução deixaríamos essa opção em branco no *token*, por exemplo, em italiano:

*<SE_NOME_1_M> Lo <SE_NOME_1_F>
La<SE_NOME_1_N><FIMSE_NOME_1>metti nella borsa.*

De forma semelhante, podemos adicionar colunas que contenham sinalizações ou essas palavras:

- inicia ou termina com vogais ou consoantes
- é ou não é um nome próprio
- são itens em pares (por ex., par de calças, meias ou óculos)

Então, podemos criar *tokens* semelhantes a esses acima, que ramificam o texto em tempo real de acordo com as marcações encontradas na linha da tabela dos nomes, em tempo real.

Por exemplo: < SE_VOGAL_x>.<SENÃO_CONSOANTE_x>.<FIM_SE_VOGAL_x>

Se a primeira letra de um determinado nome for uma vogal,
imprime a primeira parte '...',
senão a segunda parte '...'.

Por exemplo: Uma esfera brilha sobre os pés do <líder>

Un orbe luit aux pieds
<SE_VOGAL_HERÓI>d'<SENÃO_CONSOANTE_HERÓI>de<FIM_SE_VOGAL_HERÓI><HERÓI>.

<SE_x_PR>...<SENÃO_x_NÃO_PR>...<FIMSE_x_PR>

Se x é um nome próprio, imprime a primeira parte '...', senão a segunda parte '...'.

(Como alternativa: Se o nome x não tem seu correspondente artigo definido do singular (ou singular indefinido, tanto faz) na tabela de palavras, consideramos então que seja um nome próprio (por exemplo, um "nome de personagem"). Então não é tão necessária uma sinalização, mas ela pode facilitar muito a vida se esta regra não se aplicar a todas os idiomas.)

Não vamos entrar em detalhes aqui sobre os sistemas de caso e declinação da Alemanha e da Rússia, em resumo, são expansões desse conceito básico, que adicionam mais colunas para nomes e artigos, e criam mais *tokens* para acessar as crescentes combinações da tabela em tempo real, transformando tudo em texto do jogo.

Quem tiver interesse pode enviar perguntas ao endereço de e-mail listado no fim deste documento. Mas estejam avisados! Estes projetos não são para os despreparados, pois tendem a ser bem detalhados.

COLABORADORES

Stéphane Bonfils

Domhnall Campbell

Alain Dellepiane

Kate Edwards (fundadora/presidente do SIG)

Erik d'Engelbronner

Jon Fung (editor, versão 2.0)

Richard Honeymoon (Vice-presidente do SIG; autor, versão 1.0)

Rolf Klischewski

Víctor Alonso Lion

Eduardo Lopez

Teresa Luppino

Fabio Minazzi (Vice-presidente do SIG)

Virginia Petrarca

Andrea Santambrogio

Miron Schmidt

Tom Slattery

Davide Solbiati

Steve Williams

Tradução para o português:

Ana Rodrigues

Bella Cardoso de Carvalho

Carolina de Góes

Debora Weirich

Diego Hernandez

Paulo Noriega

Sheila Gomes

Thomas Melo

Revisão da versão em português:

Adriana Primão

Isabela Pedrosa

Thaina Monteiro

Thiago Araujo

Thiago Hilger

Por gentileza, envie mudanças/sugestões etc. para locsig-sc@igdad.org. Suas ideias serão revisadas e implementadas assim que possível.

Este é um documento em construção que continuará sendo atualizado conforme necessário. Este guia de localização de jogos não tem a finalidade de ser o guia definitivo, mas sim de ser a base que inspirará as pessoas a adaptarem os conceitos às suas necessidades de projeto, e também de melhorar a qualidade geral da localização de jogos no mundo inteiro. Para um olhar mais aprofundado das práticas de localização de jogos, recomendamos "The Game Localization Handbook" (ISBN: 0763795933) por Heather Maxwell Chandler e Stephanie O' Malley-Deming (também membros do grupo de localização da IGDA!).

Se você não é um membro do grupo de localização da IGDA, inscreva-se em nossa lista de e-mails hoje mesmo, por meio do site IGDA(igda.org/locsig) e participe das discussões.

Ajude seu grupo de localização da IGDA.