Text-to-speech Designed For a Massively Multiplayer Online Role-Playing Game (MMORPG)

Mike Rozak
μXac, Darwin, NT, Australia
Mike@mXac.com.au, http://www.CircumReality.com

Abstract

CircumReality is a niche-market massively-multiplayer online role-playing game (MMORPG or MMOG) that relies heavily on text-to-speech (TTS) for narration, non-player character (NPC) speech, and text chat between players. Associated speech technologies enable voice chat and voice transformation/disguise. Most contemporary TTS engines are geared towards hand-held devices or telephony, resulting in technology that is not ideally suited for games. This paper discusses how a game-oriented TTS engine differs from a telephony or device-oriented TTS engine, and how those differences affect the technology.

1. Introduction

In the early 1980's, most computer games relied on "sprites" to generate their visuals. [1] Sprites are pixelated images that are moved around the screen by offsetting their X and Y origin. They are animated by quickly redrawing different images in the same location. Very few games used real-time 3D rendering in the 1980, the most memorable being Atari's Battlezone.[2] Most real-time 3D rendering was used for CAD/CAM[3] and military flight simulators.[4] Twenty-seven years later, real-time 3D rendering in games is ubiquitous, and sprites are rarely used.

In 2007, most computer games use recorded speech, or even just text without any audio component. Recorded speech is expensive to produce, and requires enormous amounts of storage when used with a computer role-playing game (CRPG) or MMORPG. Everquest 2, a recent MMORPG, recorded over 200 actors [5] to produce 130 hours of voice-acting audio.[6]

No contemporary MMROPGs or CRPGs use TTS, just as in 1980, very few games used real-time 3D rendering. Current TTS isn't acceptable for game developers because of the lack of emotion, and because contemporary TTS engines target telephony and handheld devices, resulting in some features that are net negatives for games. Recorded speech vs. TTS in 2007 is analogous to sprites vs. 3D rendering in 1980.

CircumReality is an experimental niche-market MMORPG [7] that is designed to survive financially with only a small base of players. Since recorded speech is so expensive and produces such large downloads, TTS is employed as a cost saving technology, as well as an enabler of new gameplay. Because of CircumReality's niche-market nature, unemotional TTS, while still an issue, is not the show-stopper that it would be with mass-market games, where players demand the best visual and audio effects.

CircumReality also lets players create their own "world" that other players can enter. Based on statistics from older MMORPG development toolkits, [8] around 1% to 0.1% of the players are likely to create content. Many of these amateur authors will also wish to create their own TTS voices since that is part of the fun.

When CircumReality was first envisioned, a survey of TTS engines was conducted, but significant anti-game design decisions were found in existing engines:

1. Many MMORPGs, such as the popular Runescape [9], are distributed to players for free. Such products earn money from advertising or virtual item sales.[10] Most TTS engines are licensed on a per-unit royalty, which isn't acceptable for free-to-play games because they only earn income from around 10% of their players, and only from around 1% to 0.5% of all the copies of their software that are downloaded.[11]

2. MMORPGs and CRPGs include up to a thousand computer-controlled non-player characters (NPCs).[12] While having a unique voice for every NPC isn't necessary, enough voices should exist that players don't notice their re-use. Contemporary TTS engines focus on producing a few high-quality voices that, for the purposes of telephony, can easily require one hundred or more megabytes of RAM per voice.[13][14][15] MMORPGs require around a hundred voices, most procedurally derived from a few real voices, all fitting in a few hundred megabytes of RAM and download.

3. In the case of CircumReality and many other games, some players will become authors and create their own content. This practice is often called "modding" [16] I expect that many authors will create their own text-to-speech voices, using their own voice or their friends' voices for training. Most TTS engines don't distribute their voice-creation tools, or make them easy enough for non-experts to use.

4. Many of the authors will want to create non-English content. Precisely because CircumReality is targeting

niche markets, many of the players who speak less-common languages will want to create a content that represents their language and culture. Such languages are not supported by contemporary TTS engines because the market it too small. Although not fully implemented, CircumReality's TTS toolkit will allow motivated authors to create their own language with a minimum of linguistic knowledge.

This paper will examine how game-specific requirements affect TTS technology, and how the resulting design decisions impacted the Blizzard 2007 Challenge results.

The majority of the game-specific design decisions affect:

1. The acoustic feature set – The voice must be stored in a compact format that is easily mutable into new voices.
2. Speech recognition – ASR is not only used for segmentation, but for eliminating bad units and selecting the "best" units to keep.
3. Unit concatenation – The acoustic feature set's design has consequences for unit concatenation.
4. Prosody model – The prosody model must be automatically generated with a minimum of linguistic knowledge.

2. Acoustic feature set

MMORPG-oriented TTS needs to be able to synthesize enough different voices that players don't notice duplications, while still retaining a download of a few hundred megabytes. Not only are approximately one hundred voices required, but some must be exotic-sounding voices, such as growling dragons and bell-like fairies.

To meet these flexibility requirements, the CircumReality TTS engine encodes voices using an acoustic feature set based on a two spectral envelopes: one for voiced and one for unvoiced, along with an F0, and per-harmonic phase. No residual is stored since residuals aren't readily modified during voice transformation.

A screenshot of the feature set appears below. The top lines are the transcription, with F0 appearing underneath. The top histogram represents the voiced audio, and the bottom unvoiced. Each horizontal band in the histograms is one octave, with the lowest frequency being 100 Hz and highest 12.8 kHz. The bottom portion of the display is the colour-coded phase angle for 64 harmonics, with the fundamental at the bottom.

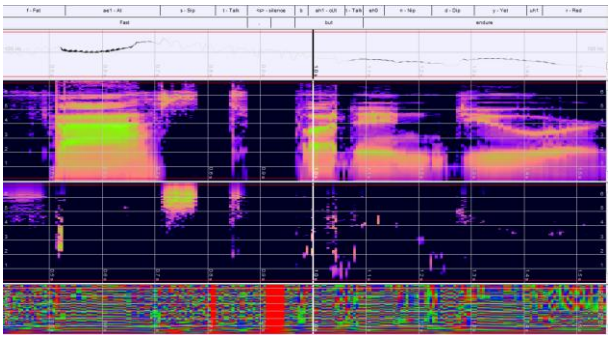

Figure 1: Acoustic feature set for "Fast, but endure".

The original signal is encoded by detecting F0 as accurately as possible since even a small error in F0 detection causes severe errors in the encoded signal. To generate the spectral envelopes, a pitch period is stretched in time domain to a power-of-two width needed for an FFT. Adjacent pitch periods are also analysed, with differences in energy and phase in each harmonic being used to guestimate how much of the signal is voiced or noise; Noise has more variation in the harmonic's energy and phase. Additional heuristics are used to minimize errors due to incorrect F0 estimation, such as using a windowed FFT for higher frequencies, as well as voiced vs. noise calculation. Phase is easily obtained from the FFT. All phases are rotated so that F0 always has a phase angle of 0.

The voiced signal is regenerated using additive sine-wave synthesis, adjusting for an interpolated phase. The unvoiced signal is synthesized using three sine waves per harmonic, with a randomly perturbed frequency whose variation is a function of the noise-to-voiced ratio. Using three randomized sine waves allows for smoother transitions from voiced to unvoiced than simply adding in a filtered noise signal, and provides more flexible monster-like voice transformations. The phase angles are also used for resynthesis; ignoring the phase information produces a "buzzy" voice.

This feature set has one major disadvantage: Because there is no residual, all of the voices have a slight "vocoder" sound. In the case of the voice used for the Blizzard 2007 Challenge, the vocoder sound was particularly noticeable.

Despite the feature set's flaws, it has many game-specific advantages:

- Voice variation:
  - Voice transformation/disguise algorithms can easily modify the spectral envelopes and produce new human-sounding voices.
  - Non-integer harmonics can be employed to generate unusual bell-sounding voices.
  - Looped waveforms can be used instead of additive sine-wave synthesis, good for growling monsters.
  - Converting portions of the voiced energy to unvoiced can also be used for monsters.
  - Some "emotions" such as whispering, speaking softly, or shouting also involve transformations of the acoustic feature set.
- The feature set is easily compressed with a lossy encoding, producing smaller versions of a voice. Players with more memory and bandwidth can download larger uncompressed voices.
- The feature set affects unit concatenation. See below.
- Re-use of technology:

- The same feature set, without F0 and phase, is used for ASR's segmenter and unit scoring. (See below.)
- The same feature set is used for encoding of voice chat. Voice chat includes the ability to transform/ disguise the players' voices, using the same algorithms that modify the TTS voices.
- The same feature set is used for automatic lip sync of player's characters when they speak using voice chat.

3. Speech recognition

Some players will create their own TTS voices as part of the effort to create content for their world. Either they or their friends will record several thousand utterances.

Consequently, the process of creating a voice must be reasonably automatic, and can't require any speech expertise. For example: Authors can't be required to review all the units and select the best one, nor can they be expected to speak exactly what they're prompted to speak.

CircumReality's voice-development toolkit employs ASR not only as a segmenter, but also as a filter to eliminate bad units, and to automatically select the best unit. ASR is included in the toolkit's install and fits seamlessly into the toolkit's user interface.

Traditional HMM-based speech recognizers split a unit into three-to-six time-slices, and create a frequency-domain mean and variance for each windowed time slice.[17, pp. 307-413] While this is adequate for dictation and phoneme segmentation, the speech recognition score can't be used to accurately identify the "best" TTS unit for a given context. Most times, a simple mean/variance will choose a distinctly "poor" unit.

Poor selection happens because the "best" units often have the brightest, most prominent formants, narrow energetic peaks in the spectrum. The peak's central frequency varies significantly throughout the frames of an individual unit, and more so across thousands of units. Averaging several thousand spectrums with narrowly-peaked formants together produces a single spectrum with blurry, ill-defined formants. A speech recognizer that uses the blurry formants to pick the "best" unit out of thousands inevitably chooses the unit that is most similar, which ends up sounding "muffled" and difficult to understand.

Many TTS systems avoid this problem by using speech recognition to eliminate the worst candidates, but leave the selection of the "best" up to luck and ASR-independent scores.[17, pp. 817] ASR-independent heuristics are used to identify the "best" candidate, such as unit duration, energy, and F0. ASR's tendency to select muffled units is further minimized when synthesizers store contiguous unit sequences from the original data and select the longest contiguous candidate; a contiguous sequence of five or six phonemes only has a couple of other candidates as competitors, so even if mean/variance speech recognition were used to select the "best", the phoneme sequence wouldn't have enough available candidates to reliably select the most muffled candidate.

CircumReality's TTS cannot leave unit selection up to chance, and it cannot rely on expert speakers or expert prompt reviewers. Amateur speakers are likely to misspeak or mispronounce the prompts. They are unlikely to review what they have recorded, failing to check for bad segmentation or pronunciations. Because of memory constraints for the voices, voice files won't have long stretches of contiguous phonemes. Consequently, relying on ASR's mean/variance to select a unit fails under the conditions that CircumReality will encounter.

CircumReality's TTS voice generator attempts to work around muffled units by stochastically storing 48 exemplar spectrum-samples per sub-segment instead of storing one mean and variance.

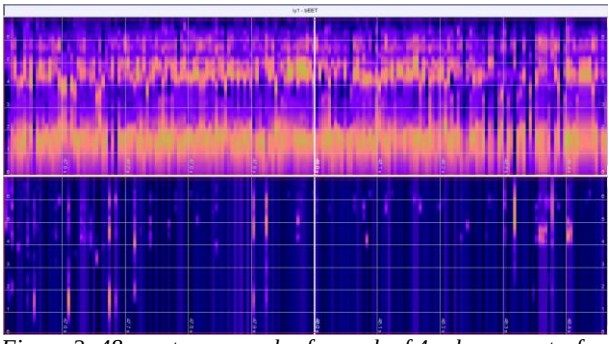

Figure 2: 48 spectrum samples for each of 4 sub-segments, for "iy1" phoneme. Voiced is on top, with unvoiced below.

To produce a score for a feature-set spectrum time-slice, ASR performs a difference comparison between the spectral envelope in question and each of the 48 spectrums stored for the sub-segment of the recognition unit. Difference is calculated by performing a dot-product on the energy-normalized spectrums. A penalty is added based on the difference in total energies. The 48 scores are sorted, and the top 12% of the scores are combined together to produce the final score, using a weighted average.

By using only the top 12%, the speech recognizer is fairly confident that the score is representative of the unit. Considering the extremes, if scores from all 48 comparisons were averaged, the result would be similar to the single mean (and no variance) comparison common to ASR, preferring muffled formants. If only the top score were used, then incorrect units and alignments in the training set would easily derail the recognizer. Using the top 12% is a compromise.

Even using the stochasticly chosen spectrums, ASR has a tendency to choose muffled formants. To further minimize this propensity, the spectrum comparison algorithm is augmented to allow small frequency shifts in the formant peaks. Each octave of spectrum can be shifted plus or minus half an octave without penalty, as well as a +/- 3 dB energy change. This causes peaks with slightly different centre frequencies or energies to be treated as identical.

Even with these adjustments, ASR still tends to select the more muffled formants, particularly with the Blizzard Challenge 2007 voice. (See "Lessons learned".)

CircumReality's phoneme segmenter uses a Viterbi search, hypothesizing over all word pronunciations, and all possible phoneme durations. To create a score for a hypothesized phoneme, individual time-slice scores are averaged.

When building a TTS voice, a context-dependent (CD) ASR model is built for each phoneme, stochastically selecting the

48 x 4 exemplar spectrums from all of the matching CD phonemes in the model. If the data set is large, the context is the left and right phonemes, if small, the acoustic groups for the left and right phonemes. The ASR model of a CD phoneme is used to represent the "ideal" for the phoneme.

A "quality" score is generated for each unit to determine how accurately it matches the "ideal" for the CD phoneme. As per typical TTS unit-selection, differences between the duration, F0, and energy of the unit and those of the mean unit, are included as a penalty. The CD ASR score is also incorporated as a major portion of the unit's quality score. The final quality score, which includes the CD ASR score, is stored with the unit to aid the concatenation Viterbi search.

Individual units, diphones, and contiguous sequences of units are selected using a weighted average of the individual unit scores. For a given sequence, the candidate with the highest aggregate quality score is retained.

4. Unit concatenation

CircumReality unit concatenation is fairly traditional. [17, pp. 804-817] A Viterbi search tries to find the highest scoring sequence of units. Demiphones are used.

Target costs include:

- The unit's quality score, a major portion of which is the CD ASR score, is used.
- Differences between the original and target unit's F0, F0 delta, energy, and duration have minimal effects.
- Penalties for CD phoneme mismatches are also included. Small penalties are used if the left/right phonemes differ from the desired context only in phoneme stress. Larger penalties are used is the left/right phonemes differ, but are from the same acoustic group.

Join costs include:

- The difference between the two edge-spectrums, calculated using the same difference measure employed by ASR.
- Weighting to encourage contiguous units is important. Penalties for non-contiguous units are amplified if either of the units is plosive since the acoustic feature set does a better job of joining non-plosive unit halves. Breaks are preferred between two non-plosive units, or in the middle of a single non-plosive unit.
- F0 and the harmonic's phase are not used.

Concatenated units are smoothed by pitch-shifting and energy-adjusting the formants at the boundaries. These adjustments, while easy to perceive in the visualized feature set, provide only a minor improvements to the voice.

Phase is blended at non-contiguous unit boundaries. Around $1/16^{th}$ of a second is blended for the lowest harmonics. Higher harmonics do not require blending. Failure to blend phase is most noticeable in the low-F0 Blizzard voice.

5. Prosody model

A prosody model is learned from the voice. This is critical for two reasons: (1) It helps make each voice sound more unique, and (2) Learned prosody simplifies localization, important for authors customizing CircumReality to their own language; they won't have professional linguists to hand-generate prosody rules.

To train prosody: F0 and energy curves are calculated for the utterances. Phoneme segmentation, word segmentation, part-of-speech, and punctuation are also used. A list of syllables is extracted from the audio sample, with punctuation being treated as a silent syllable. The following information is extracted for each syllable:

- The average F0 of the syllable, the rise of the F0 over the syllable, and the amount of "bend" in the F0. Three values produce a second-order approximation of the F0 curve. These numbers are relative to the sentence's average F0.
- The average energy of the syllable, relative to the sentence's average energy.
- The duration of the syllable relative to the duration predicted by the syllable's phonemes, and the duration of the syllable relative to the average duration of all syllables.
- The number of phonemes in the syllable.
- What word the syllable came from, particularly important for function words.
- Whether the syllable is stressed or unstressed.
- The syllable index number within the word.
- Whether there is silence before the beginning of the syllables that begin words.
- The depth of the word in the context-free grammar (CFG) parse tree used for part-of-speech disambiguation.

The syllable information is used to create a number of simple prosody models, such as:

- How stressed/unstressed syllables affect F0, duration, and energy.
- How the location within the sentence of length N affects F0, duration, and energy.
- How the previous and subsequent parts-of-speech and stresses affect F0, duration, and energy.
- The silence bit is used to determine the probability of a short pause before the word given the part-of-speech context, often indicative of a phrase break.

Just as using a single mean and variance for ASR produces muffled units, using simple models for prosody produces uninteresting and bored-sounding prosody. To work around this problem, the prosody generated by the simple prosody models is "subtracted" from the original sentences, which are then stored in the prosody model as "residuals".

When prosody is synthesized, the synthesizer finds the longest contiguous segment of residual that's the best match. A "contiguous segment" requires an exact match for the syllables' part-of-speech and stressed/unstressed values. The quality of the match is determined by differences in the numbers of phonemes in each syllable, the syllable index within the word, and the original word (important for function words).

If only the longest/best match is chosen, however, the synthesized prosody becomes a somewhat erratic, speaking some words far too quickly, slowly, loudly, or quietly.

To stabilize the system, the top residual choice is compared to the next seven-best residual choices. A difference score is calculated by comparing the difference in the synthesized prosodies for each of the candidates, using F0, duration, and energy. The four most-similar prosodies are averaged together, largely eliminating the erratic prosody, while still maintaining interesting prosody.

To regenerate the prosody, the simple prosody models are first run, and then the prosody residual is incorporated. Adjustments are made where the synthesized sentence doesn't exactly match the sentence used to generate the residual. For example: If the synthesized prosody has a syllable with three phonemes, but the chosen residual had only two phonemes for that syllable, then the syllable will be lengthened according a database value that was generated as part of the prosody model.

This prosody model has several advantages:

- Every voice has its own prosody, making the voices sound more unique.
- It requires no linguists (or any input) for localization. Unfortunately, part-of-speech determination still requires a context-free grammar with a few dozen hand-programmed linguistic rules.
- Prosody can be transferred or merged between voices.
- Components of the prosody model can be procedurally adjusted to create new prosody models from the original.
- If an author has a problem with a common phrase not being spoken properly, he merely has to record how it should be spoken, and add that recording to the TTS or prosody-model training set.
- The residual match scores are randomly perturbed so that a sentence is spoken differently every time it is heard, which is particularly important in a game where players may repeatedly hear the same sentence, such as "You open the door."

6. 2007 Blizzard challenge

The CircumReality TTS engine didn't perform well in the 2007 Blizzard Challenge, coming in last for the mean opinion score and similarity tests.

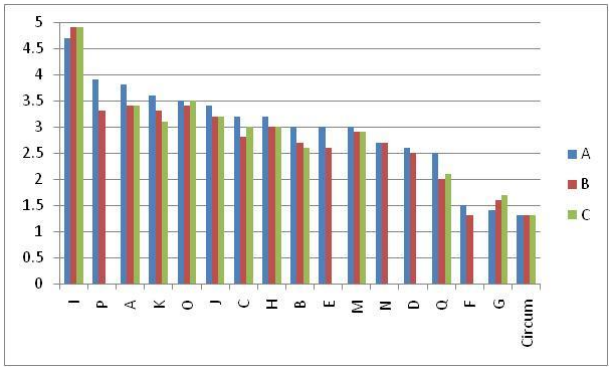

Figure 3: Mean opinion score for all listeners.

The major problems with the voice were:

- Existing voice encoding/decoding algorithms didn't work well with the Blizzard Challenge voice, producing a noticeable "vocoder" effect. While the vocoder effect occurred with other test voices, it was much more noticeable with the Blizzard 2007 voice.

  Switching to a different method for encoding the voice, or adding a residual, would eliminate the vocoder effect, but it would also hinder the engine's ability to procedurally create a variety of voices from a single source.

- Speech recognition tended to select muffled units for the voice. I think this happened because the brightness of the formants varies greatly between units, a useful skill for a professional voice talent trying to create an expressive sentence. However, ASR ends up learning that the "best" brightness for a unit is someplace between bright (and easy to understand) and muffled, which results in slightly muffled units, suboptimal for a TTS voice.

Oddly, the engine did relatively better with the word error rate:

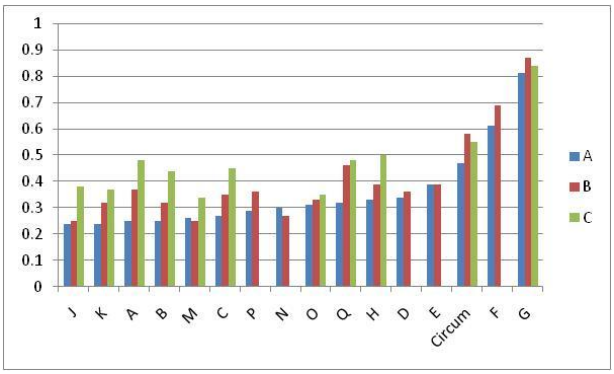

Figure 4: Word error rate for all listeners.

One explanation for this discrepancy is that despite the "vocoder" distortion caused by encoding to the acoustic feature set, and unit selection preferring "muffled" units, ASR successfully chose one of the "better" muffled units, although not the "best" un-muffled unit.

Synthesized prosody performed better than I expected. Prosody quality wasn't isolated out by any of the Blizzard tests, but my own non-scientific comparison of the CircumReality prosody to that of other engines puts it closer mid-range. It sounded less stiff and more natural than many of the engines, probably because it doesn't use any hand-generated prosody rules. Ironically, natural-sounding prosody was also a minus: The Blizzard speaker's volume often trailed off near the end of a sentence, and so too did CircumReality's synthesized prosody. CircumReality's synthesized prosody performs less well on long sentences though.

7. Lessons learned

Since the listening test samples for the Blizzard challenge were submitted, CircumReality's algorithms have been improved so that the Blizzard 2007 voice sounds noticeably

better. The improvements that made the greatest difference were:

- Due to memory constraints, the voice generator tool couldn't produce a voice larger than 20,000 units on 32-bit Windows. This wasn't a problem for a game-oriented voice since 20,000 units, around 100 megabytes uncompressed, is at the extreme upper range of acceptable voice size. Multiplied by 8 voices, this produces an 800 megabyte download, which is far too large for most players. After installing 64-bit Windows Vista and recompiling the toolkit for 64 bits, a better-sounding larger voice was produced.

- The algorithms to extract the acoustic features from the Blizzard 2007 voice needed improving. The voice's combination of low F0, well defined, crisp formants, large dynamic range, large F0 range, and occasional vocal fry all proved troublesome. The improved feature extraction algorithms noticeably reduced the "vocoder" sound of the voice, although it still exists. Even though storing a residual would eliminate this, doing so would impair the flexibility of voice transformation, an important feature for games.

- Unit scoring has been improved so that when the "ideal" units are trained using ASR, training is weighted by the unit energy, causing ASR to prefer the louder units, which also tend to have brighter and more well-defined formants.

- Unit selection has been augmented to store longer unit sequences, as well as store multiple versions of a unit sequence based on F0. Again, such affordances produce better-sounding but larger voices, less suited for games.

  Frequency shifting a unit of the Blizzard 2007 voice had significantly worse effects on quality than the other voices that CircumReality was tested with. I think this is because of errors in the harmonics' phases: Looking at the visual representation of the feature set, it's obvious that phase tracks with the formants. Harmonics at the peak of the formant incur the least amount of phase delay, while those at the edges more. Importantly, harmonics outside a formant are so quiet that their phase is undetectable or completely inaccurate. When a unit's F0 is shifted, the phase relationship with the formant is broken; the harmonic at the peak of the formant no longer has the lowest phase delay. Even worse, some of the undetectable harmonics are frequency-shifted into the formant, and consequently resynthesised with either a random or 0-degree phase. The Blizzard 2007 voice illuminated the problem because it had a low F0 and narrower formants.

8. Future work

Although acoustic feature extraction still induces noticeable errors, the largest problem with the CircumReality TTS engine, in terms of games, is the synthesized prosody. Even if prosody were as good as the best prosody example in the Blizzard Challenge, it wouldn't be good enough; even the best TTS engines produced speech that sounds like a "bored telephone operator", destroying any illusion that the NPCs are real.

Recorded speech is much more emotionally powerful. To experience an example of NPC interaction with emotionally recorded speech, see the Facade [18] interactive storytelling [19] demo. Facade's 167 megabyte distribution also illustrates the problems with recorded speech: Most of the download size is used for speech audio recordings of only two NPCs speaking in an extremely limited domain. Extrapolating to a thousand NPCs with a much broader conversation domain reveals the ultimate impossibility of using recorded speech, despite its emotional quality.

For the next 10-20 years, I suspect the best solution to this problem will be to use transplanted prosody wherever possible. Of course, many phrases that NPCs speak are procedurally generated and must still revert to synthesized prosody.

The key to good transplanted prosody is the integration of the transplanted prosody tools into the game development toolkit. Recording an utterance for transplanted prosody needs to be so convenient, quick, and easy that authors automatically and effortlessly record the transplanted prosody utterance when they type in a sentence for a NPC to speak.

Thus, the key to better game TTS isn't just the algorithms, but also TTS's integration into the game-development toolkit.

The tools for producing a CircumReality TTS voice are publicly available as part of the "3D Outside the Box" software package on http://www.mXac.com.au/m3d. The CircumReality game, still not finished, is available from http://www.CircumReality.com. It uses many of the open-license voices from the 2005 Blizzard Challenge.

9. References

[1] http://en.wikipedia.org/wiki/Sprite_%28computer_graphics%29
[2] http://en.wikipedia.org/wiki/Battlezone
[3] http://en.wikipedia.org/wiki/CAD/CAM
[4] http://en.wikipedia.org/wiki/Flight_simulator
[5] http://pc.gamezone.com/news/08_02_04_10_57AM.htm
[6] http://www.mobygames.com/game/everquest-ii
[7] http://en.wikipedia.org/wiki/Mmorpg
[8] http://www.mudconnect.com/ - Around 1650 text MUDs have been created by a community of 50,000 to 200,000 players.
[9] http://www.runescape.com/
[10] http://en.wikipedia.org/wiki/Real-money_trading
[11] http://en.wikipedia.org/wiki/Shareware
[12] http://en.wikipedia.org/wiki/Non-player_character
[13] http://www.cepstral.com/downloads/
[14] http://www.naturalvoices.att.com/products/tts_data.html
[15] http://www-306.ibm.com/software/pervasive/voice_server/technical_details/
[16] http://en.wikipedia.org/wiki/Mod_%28computer_gaming %29
[17] Huang, Xuedong, Acero, Alex, and Hon, Hsiao-Wuen, *Spoken Language Processing, A Guide to Theory, Algorithm, and System Development*, 2001, New Jersey, Prentice Hall PTR.
[18] http://www.interactivestory.net/
[19] http://en.wikipedia.org/wiki/Interactive_storytelling