Computer Visions of Computer Games: analysis and visualization of play recordings

Jeremy Douglass
U. California San Diego
9500 Gilman Drive
La Jolla, CA 92093-0037

jeremydouglass@gmail.com

ABSTRACT

Computer gameplay poses unique challenges and unique opportunities for the artists and researchers who seek to represent it in novel ways. The extreme variability of interactive visual media such as computer games greatly complicates both the process of sampling elements from individual gameplay sessions and the process of comparing, combining, or mashing up excerpts from different sessions in meaningful ways. This MAST demonstration session models the use of live gameplay video recording, simple computer vision techniques, and image processing software in order to produce a series of novel representations of gameplay, including both graphs and maps of gameplay activity and mash-ups of gameplay screenshots into timelines, montages, and images that superimpose gameplay modes into average "eigenmodes."

[Note to organizers: This proposal is described as a demonstration session, which would allow participants to briefly play five minutes of a game, process the gameplay video, create their own novel representations, and then study and discuss them. However, it could be presented as a video-augmented poster session or as a short talk instead.]

General Terms

algorithms, measurement, design, experimentation

Keywords

games, computer vision, visualization, cultural analytics, software studies

1. GAMEPLAY AND RECORDING

During the first stage, live computer game play is recorded as digital video. Any of a number of techniques can be used to accomplish this, from inline pocket recording devices to combinations of emulation and screen capture software. The only requirement for good results is that the video be a direct recording of the game output signal -- not a live recording from a video camera pointed at a screen. Indeed, as long as this condition is met, the subsequent steps in this process work equally well with gameplay videos taken directly from various internet archives. This is important for artists and researchers interested in working with either found gameplay or massive amounts of gameplay data.

The approach to recording at the Re:Game research lab is to develop PlayDVR, a modified Linux-based PVR (personal video recorder) that records MPEG2 files directly to a web-accessible directory where it can be downloaded for processing. The design and use of the recorder is outside the scope of the demonstration, but it is a modification of the open source software MythTv and the open hardware specification Dragon 2.0.

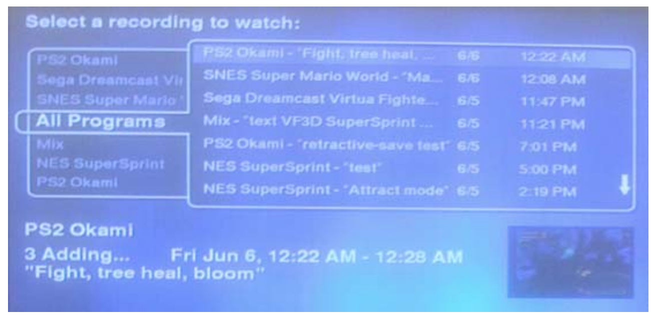

Figure 1: PlayDVR used as a gameplay video server.

It is also worth noting that, while this demonstration is primarily focused on data mining the video of recorded gameplay, many other kinds of data can be collected simultaneously for later classification and visualization, from logs of keys pressed to records of heart rate or galvanic skin response.

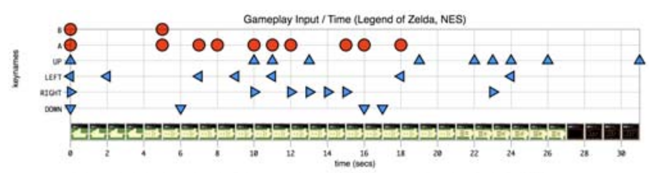

Figure 2: game controller key presses visualized as a "tablature" of gameplay

2. IMAGE PROCESSING

During the second stage, the downloaded video is split into periodic keyframes, simple color and motion data is collected. This is a largely automatic batch process enabled by Vilinx, which is built on top of the open source libraries ImageMagick and ffmpeg. Vilinx is being developed in-house at the UCSD Software Studies Initiative as part of the Cultural Analytics research program.

The basic output is directories of simple PNG image files (one per keyframe) as well as comma-separated value text files of statistical information which may be automatically passed on to graphing software.

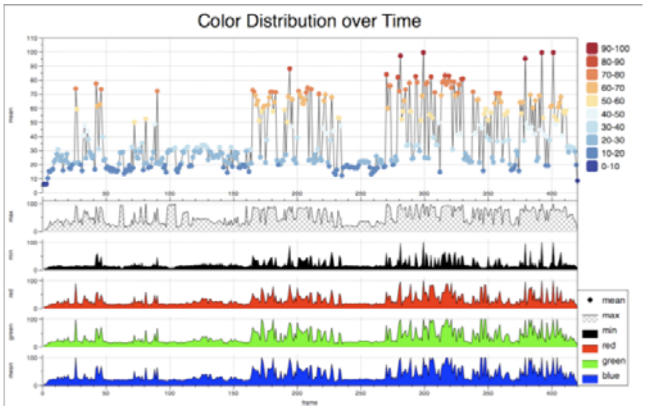

Figure 3: Basic information on gameplay video frames such as RGB levels are graphed over time.

In addition to producing frames and data, Vilinx can automatically generate a variety of simple representations, including both image series and high-speed video synopses of filmstrip views, montages, differential motion views, and histograms. These outputs can become the raw assets for artworks from digital mashups to VJ sets. They also inform the analytic process of exploring gameplay data and customizing the computer vision algorithms for any given game.

Figure 4: Vilinx generated movies and montages of *Fatal Frame 2*, with red versions indicating differential motion.

3. MODE ANALYSIS

During the third stage, the gameplay states of keyframes are classified in order to enable the production of timelines, maps, or more meaningful montages and averages of selected frames. Game mode analysis by the classification of simple states in a keyframe is one of the simplest kinds of computer vision application, but results are often spectacular due to the fact that computer game video output is *highly structured* because it is the product of software operations. Thus the nature of various game elements, affordances, user interfaces, and other signifiers is often to be identical in color and pixel location.

In order to more easily identify invariant elements in the structured output of video games, I developed an approach here termed "eigenmodes."[1] In it, many frames are superimposed, creating "average" images that highlight recurrent visual artifacts in the game software. These images may be beautiful in themselves, and provide unique insights into patterns of experience in gameplay.

Figure 5: Example frames (left) and "eigenmodes" (right) for the game modes of fighting, cut scenes, and free exploration in Fatal Frame 2.

Furthermore, the recurrent elements highlighted by eigenmodes can then further be used by the system as classifiers, creating new images and further refining the ability to detect and depict different aspects and states of the game. Once a set of simple classifiers has been identified for a given game, these can be used by Vilinx and client applications to create timelines, render maps, or in the power interactive gameplay exploration demos or art installations.

[1] "Eigenmode" is so called after the "eigenface" technique in computer face recognition, in which a given face may be classified in relation to a small set of images resulting from massive averaging of normalized faces. In the eigenmode, a screening effect often occurs as the modes "own" recurrent image values pop out of a sea of unstructured noise.

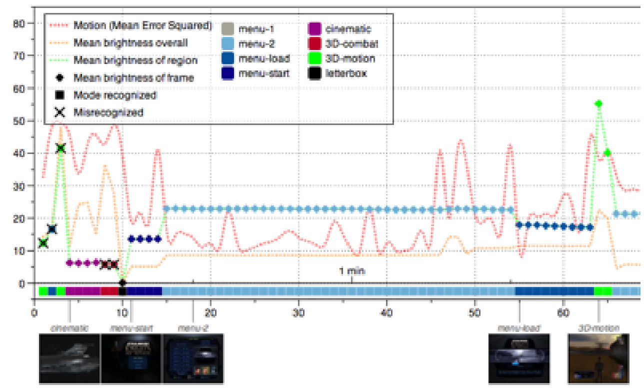

Figure 6: A 30 minute recording of *Knights of the Old Republic* is auto-catagorized by mean brightness of a screen region.

4. CONCLUSIONS

The full range of uses for simple computational approaches to gameplay recording has barely been considered, but the potential for new kinds of artistic representations and analytic insights about games is huge. Current prototypes being explored by Software Studies Initiative collaborators include massive maps of classic game worlds with gameplays embedded, visualizations of full levels with many gameplays simultaneously superimposed, and branching timelines that track save points and variable play experiences. PlayDVR game video recording and Vilinx image processing represent two initial technical approaches to the computational analysis and visualization of gameplay, but the future is wide open – for games, for interactive visual media, and most generally for the visual outputs of all software systems.

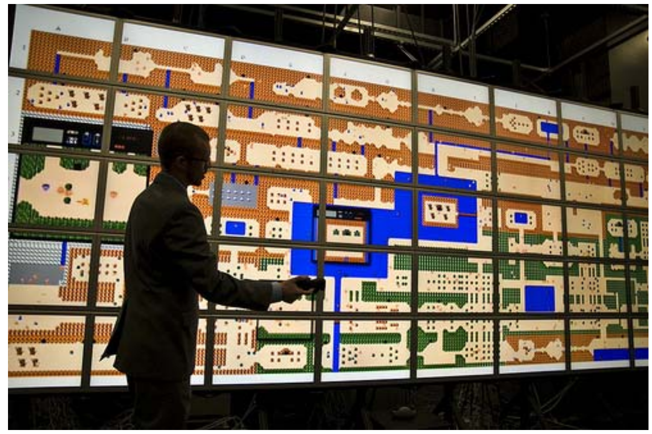

Figure 7: a prototype video map tracks the current position of *Legend of Zelda* gameplay in a multiscreen gameworld grid.

5. ACKNOWLEDGMENTS

Thanks to Lev Manovich and Noah Wardrip-Fruin of the Software Studies Initiative for their collaboration and supporting of my research, to William Huber and Colin Wheelock for work on FF2 and KotOR, to Daniel Rehn, Derek Lomas, and the rest of Re:Game for their feedback and inspiration, to the Center for Research in Computing and the Arts for their generous support, and to the HIPerWall and HIPerSpace research teams for access.