Notes on scenarios that envision potential capabilities for multi-core processing support for Computer Games and and Persistent Online Virtual Worlds in 2010-2012

Walt Scacchi
Game Culture and Technology Laboratory
University of California, Irvine
Irvine, CA 92697-3455
20 June 2007

The starting assumption we make is that computer games and virtual worlds like *Second Life* are currently limited to a conventional single processor-based computing architecture. In the future 3-5 years forward, we anticipate that an alternative computing architecture in the form of multi-core processors that may include 32-128 cores will become more widely available, and thus become the processing architecture of choice for computer games and virtual worlds like Second Life. This situation however will likely change both the ways how such games and worlds are developed, and also the kinds of processing capabilities and game-play mechanisms that may be supported.

With this in mind, we begin by identifying a number of possible scenarios for computer games and virtual worlds that assume, exploit, or otherwise require the existing of multi-core processing architectures, with the number of cores possible spanning 32-128. These scenarios are numbered for identification purposes only, and not to indicate a priority or preference.

1. Games or virtual worlds (GVWs) that provide higher resolution of visual realism than currently available—from flat shaded polygonal forms and textures to photo-realistic forms, surfaces, reflectances, shadows, etc. For example, this might entail the use of ray-tracing visualization pipelines or other rendering methods that are amenable massive or multi-core parallelism.
2. GVWs that provide higher resolution of object-centered physical realism than currently available—from simple physical forces and displacement/movement mechanics to complex physical behaviors and dynamic interactions between reactive/deformable objects. For example, this might entail the use of object-centered physical modeling that internalize externally observable (visual) behavior with some internal mechanism (e.g., game characters or avatars with "muscles" that compress or stretch in response to physical loads, and thus display deformable body/muscle contours). Such physical modeling techniques are amenable to massive or multi-core parallelism.
3. GVWs that provide higher resolution of complex multi-object behaviors than currently available—from simple or no "collision detection" when objects/avatars touch (or unintentionally "pass through" one another) to complex collisions between objects that may result in the object/avatars being fractured, deformed, or shattered into pieces in ways similar to real-world collisions among physical objects (e.g., dropping a glass of water should both spill/disperse the water in a fluid manner, as well as fracture the glass into pieces who flying shards may follow trajectories

determined by the glass's mass, velocity, angle of incidence with object/surface of collision, deformation/resilience (firmness) of the collision object surface, etc. Such physical collision modeling techniques are amenable to massive or multi-core parallelism.

4. GVWs where game characters/avatars are able to coordinate their gaze direction and visualize emotional states. Most game characters or avatars are unable to "look into the eyes" of others that can gaze back—they lack the ability to be able to look eye-to-eye. This in turn reduces or inhibits the conveyance of emotional expressions like attraction, interest, dis-interest, fear, etc. Instead, game characters/avatars act more like lifeless dolls whose visual expression or appearance is persistent, as well as un-reactive to presence of others. Game characters and avatars lack the ability to perform "gaze tracing" (like ray-tracing from the eyes outward into the surrounding world or towards others nearby) that can require eye movements, as well as head movement (e.g., head turns, nodding), and coordinated facial expressions (smiling, frowning, grimacing, etc.). Such visualization and coordination of visual gaze and emotional expression among multiple interacting game characters or avatars requires techniques amenable to multi-core, multi-tasking or multi-processing.
5. Same as 4, but with the gaze direction determined through computer vision techniques that employ digital cameras that monitor human player gaze direction, facial expression, head movement, etc. This also requires techniques amenable to multi-core, multi-tasking or multi-processing.
6. GVWs that support persistent (semi-) autonomous bots, avatars, or game characters with internal mental state, disposition, ability to act with discretion, self-identity, resources and game-play capabilities. In MMOGs like *World of Warcraft*, common game play entails groups of players acting as teams or clans engaging in shared quests or competitive battles against other teams/clans. Each quest/battle currently entails a multi-processing game play experience where each human's in-game character is guided, controlled, and monitored through a client-specific view port on a human players computer. If a battle entails two clans each with 16 human players, then there are 32 in-game characters, each associated with a human (client-side) computer (hence 32 processing cores) whose collective action (the articulated battle with emergent outcome) is coordinated and synchronized through a shared server. Here, we simply envision a scenario whereby the human player can experience a similar game-play experience with one or more teams/clans of (semi-) autonomous, artificially intelligent in-game character bots that are capable of believable role-based game play. Each in-game character/bot thus requires or benefits from at least one dedicated processing core to control and monitor the AI-based game-play behavior in a multi-bot (or mixed multi-bot, multi-human) game battle. Further, associated with each game character bot may be a persistent viewport that can record game-play experience from the perspective of an individual bot, thereby enable a kind of multi-channel concurrent "game camming", replay, or machinima capability.
7. Similar to 6, but where the bots now have "enterprise skills" or "business AI capabilities". In this regard, a game player can create, instantiate, or customize their own virtual enterprise as a collective of multiple bots (agents or actors) that can be collectively tasked to perform "administrative services" or similar, depending on the AI that the bots are programmed to perform. This would draw on known methods for

multi-agent process enactment, which has been used in computational studies of complex organizations. As before, each bot/agent requires a dedicated processing core for its AI, and perhaps zero or more additional cores for management of their visual appearance, physical behavior, etc.

8. GVWs that support or require interoperable multi-system commerce. For example, in a VW where a player can shop in-world for objects to acquire and purchase with a real-world payment instrument (e.g., credit card), payment processing may be performed in one or more systems that are external to the VW. The goal however is to maintain a seamless end-user/player experience. For instance, if a young person is shopping in a retail clothing store depicted in an online world like "Virtual Laguna Beach" and finds a object to acquire (e.g., "a nice shirt"), but requires a credit card (which they don't have, but their parent does), then the young person would like to be able to contact their parent from within the GVW, request and receive a credit card purchase authorization (perhaps after some negotiation via instant messaging), then take possession of the in-world object (put the new shirt on the young person's avatar). Such a capability requires multi-tasking, multi-system coordination, and network transactions that might best be enacted, managed, and tracked (e.g., for non-repudiation purposes) by multiple processors on the server side, or possibly on the client side.
9. GVWs that support multiple game engines for graphics, physics, AI, audio, networking, etc.
10. GVWs that support multiple physical game engines for physics, chemistry, biology, psychology, sociology, cosmology, etc. that span multiple levels of physical realism (from micro to macro).
11. Physically embodied user interfaces from heterogeneous devices to GVWs, such as SL to cell phones, computer-controlled musical instruments, scientific instruments (remote telescopes that capture images that are digitally pasted into a virtual sky on-demand).