CESA ゲーム開発技術ロードマップ（プログラミング分野）

プログラミング一般

＜最新＞　C/C++で作成。マルチコア CPU で API ベースのスレッド制御、ゲームエンジンを使用した開発環境の普及

＜数年後＞　- メモリの共有・排他レベルの宣言とスレッド生成・同期の簡略化等をサポートする新言語もしくは言語拡張の実用化
参考例）CUDA/Axum/ATIstream/TBB/OpenCL/OpenMP/関数型言語(F#,Ocaml,Haskell)のアプローチ、Apple の GCD, Blocks のような言語拡張
- LLVM/PGO 等にみられる実行時最適化技術の向上
- ゲーム本体部分は、徐々に C#や JavaScript 等の生産性の高い言語に移行

コンピューターグラフィックス

＜最新＞　- ポリゴンベースのモデル＋マッピングのバリエーションが広まっている、また Deferred 系の Rendering が普及

＜数年後＞　- DX11 で可能になった表現の一般化
- Voxel/Micro polygon/NURBS/Displacement Map/Tessellation/Fractal 等を使用した、スケーラブルなジオメトリの実現
- Global Illumination のリアルタイム化、リアルタイム RayTracing の実現
- スマートフォンなどでも現行据え置き型ハードレベルのグラフィックス表現の実現

ＡＩ

＜最新＞　- ＦＳＭのスクリプトベースの実装
- ノードベースでのグラフィカルな AI 実装

＜数年後＞　- プランナ向けのグラフベース、セッティングベースのビジュアルスクリプト
- ソースコード上の条件分岐によらない得点計算、条件判定等による行動選択
参考例）プランニング/Behavior Tree/確率推論等
- 動画、画像、音声、構文解析による自動・半自動コンテント生成

物理

＜最新＞　- 剛体シミュレーション + Constraint Solver、Ragdoll 物理等

＜数年後＞　- セットアップに頼らない破断、壊れ、変形などのリアルタイム処理
- クラウドコンピューティングによる大規模シミュレーション
参考例）海洋など環境レベルの流体表現

アニメーション

＜最新＞　- スケルトンベースのキーフレームアニメーションと IK による自動補完

＜数年後＞　- プロシージャルなアニメーション技術の普及
- 物理シミュレーションと連携したよりリアルな動きの生成