

Genshi報告書

1. Genshiとは...

「日々これ狩り」といった非常に単純な「暮らし」をVR世界で表現し、そこで遊ぶことによって現代の複雑化し過ぎて方向性の定めにくくなった「暮らし」を見直すキッカケを作ろうというのが、この企画での目的である。

この企画で創造されるVR世界は馬鹿馬鹿しいまでにデフォルメ化された原始人世界である。装置も三輪車コントローラ、武器デバイス、母音ヘッドホンなど、理性を吹き飛ばし本能を刺激することを徹底的に狙っている。

野生を呼び覚ます本能的空間を味わうことにより、現代の抑制された倦怠感漂う「暮らし」に喝を入れられるような作品を提供することが本企画の目的である。

2. 企画内容

プレイヤーはVR世界で原始人となり、制限時間以内（夜明けから日没まで）に食料となるマンモスを狩る。プレイヤーは3人まで同時にVR世界に参加でき、協力プレイ、チームに別れての対戦プレイ、全員が別チームに別れるバトルロイヤルプレイなど、プレイヤーが遊びたいように遊べる。プレイヤーはヘッドマウントディスプレイを装着した状態で三輪車に乗る。移動は三輪車のペダルを漕ぐことで行い、漕ぐスピードに応じて速度が変化し、またハンドルを切ることで曲がる事が出来る。

攻撃には投げ槍を使う。投げ槍を操作するデバイスは筒型の形状をしており圧力センサーが仕込まれている。槍コントローラーを振るとその勢いを圧力センサーが検出し、その勢いに応じた速度で槍を発射する。発射する方向はプレイヤーの顔の向きとなる。

プレイヤーにはヘッドホンとマイクを付けてもらいVR世界の音を聞いてもらう。この時マイクから入ってきたプレイヤーの声は、DSPを通して母音のみの音声に変換してから他のプレイヤーのヘッドホンに送る。原始人らしさを演出するとともにチームプレイを行う時に一工夫が必要になり、何らかのゲーム性アップが図れるのではないかと狙っている。

VR世界は崖や木をあちこちに配置し退屈しない画面にする。プレイヤーは自由に移動出来てしまうので、ある程度進むと行き止まりになるようにしておく。

3. システムの構成

ゲームは3人まで同時プレイが可能。プレイヤーにはHMD、ヘッドホン、マイク、武器デバイス（位置センサー+ボタン）、三輪車コントローラーが与えられる。

HMDに表示する画像やそれぞれの位置センサーの入力、ヘッドホンへの再生を行う為にプレイヤー一人に1台の割合でウィンドウズマシンが割り当てられる。

観客が現在のVR世界の状況を見られるようにプロジェクタにゲーム画面をライブ形式で表示。この観客用モニタの画像描画とプレイヤーマシン間のデータの統合処理を行う為にサーバマシンを1台用意する。

音声の変換用にDSPを3つ用意。これはプレイヤーマシンに接続しておく。DSPで加工されたプレイヤーの音声はミキサーによりVR世界の効果音と合成され、各プレイヤーのヘッドホンに送られる。サーバ及びクライアントマシンはそれぞれ以下のような処理を担当する。

<サーバマシンで行うこと>

マンモス、槍、肉の制御。

プレイヤー、マンモス、槍、肉など動くもの同士の当たり判定。

槍、自転車のペダルなどのセンサーの読み取り。

<クライアントマシンで行うこと>

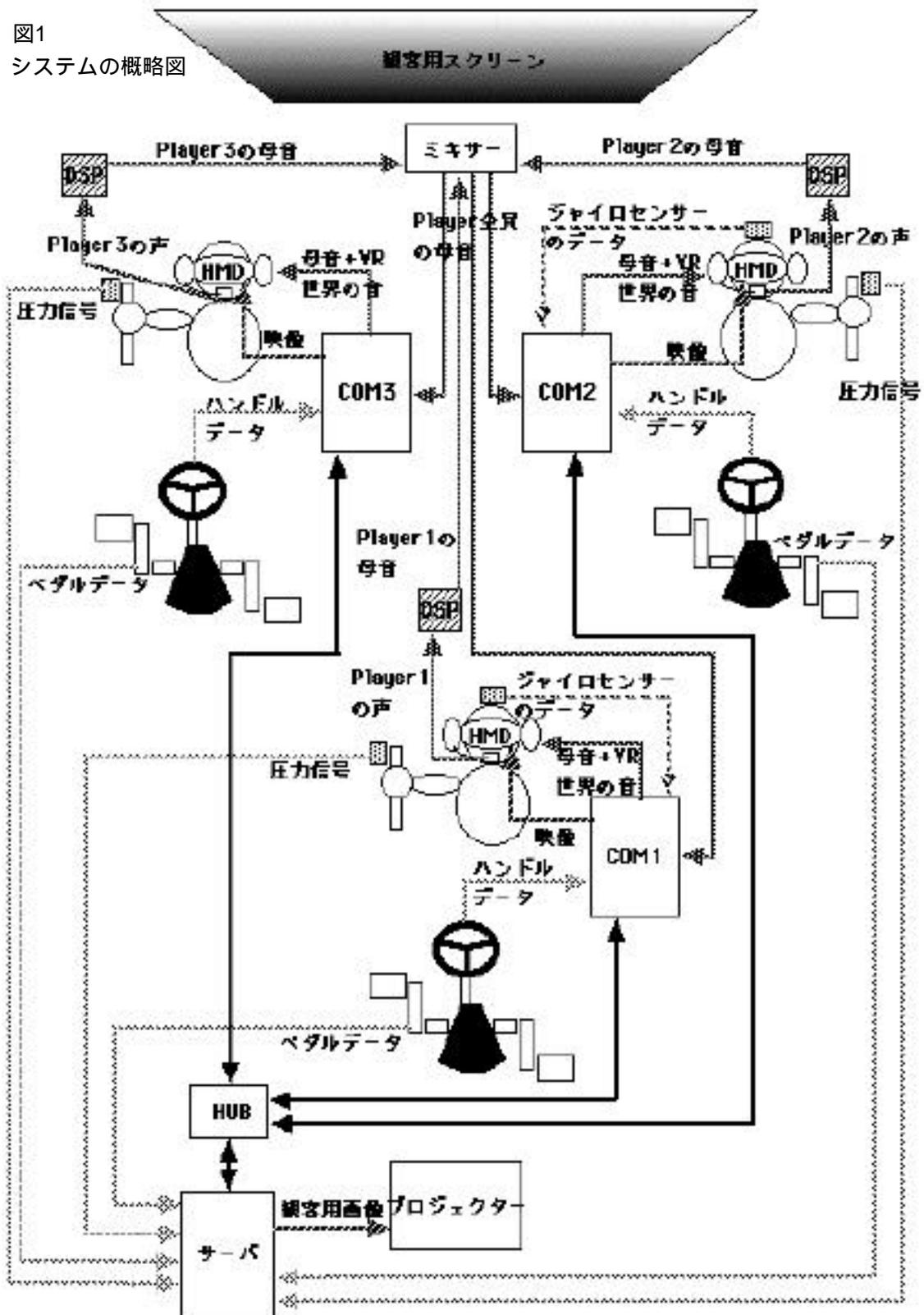
プレイヤーの制御。

地面、木といった静止物とプレイヤーの当たり判定。

自転車のハンドルの読み取り。

システムの概略図を図1に示す。

図1
システムの概略図



4. 機材紹介

1) Windowsマシン4台

CPU	:	ペンティアム166MMX以上
メモリ	:	64MB以上
ビデオカード	:	Riva128 (4MB) 以上
イーサカード	:	10Base-T以上

2) HMD3台

プレイヤーに提供され、プレイヤー視点で見たVR世界を表示する。

3) 圧力センサー3機

投げ槍コントローラーの勢いを読み取る装置。

4) 三輪車コントローラー用自転車シャシー3台

三輪車コントローラーの基礎となるパーツ。パイプを工作して自転車を安定させる。

5) 自転車ライト用発電機+センサー3台

タイヤの回転によりダイナモで発電された電気をコンデンサーに貯め、フレーム毎にそのコンデンサーに溜まった電圧を読み込むことで速度センサーとする。

7) ハンドルコントローラ

ハンドルのきり具合を検出する。市販のパソコン用ハンドルコントローラを使用。

8) ヘッドセット3つ

VR世界の音声を聞いたり、自分の声をVR世界に伝える為に各プレイヤーに装着してもらう。

9) DSP3つ

プレイヤーの音声を母音のみの音声に変換するためのフィルタを作成する。

10) ミキサー1つ

全プレイヤーのヘッドホンに流れるVR世界の音声と各プレイヤーの母音音声を合成する為に使用。

11) 5ポートHUB1台

各プレイヤーの情報をやりとりするネットワークを構築する為に使用。

12) 液晶プロジェクタ1台(貸与品)

観客用モニタとして使用。

13) ジャイロセンサー2台(貸与品)

プレイヤーの頭の方向を読み取り、それによりHMDに表示する画像を変化させる。

5. チーム報告

5.1. 全体のシステム

開発の遅れから、プレイヤーキャラクタの選択、槍以外の武器の導入、肉の取得数の管理、狩り終了時のデモ、狩り終了後のミニゲーム、その他、細かい演出のいくつかをカットした。プログラム出来なかった部分で、企画の根幹に関連する狩り終了時のデモなどに関してはプレゼンに状況解説を加えることでフォローした。

またセンサーの読み取りをサーバに行わせてしまったことで、ネットワークや槍の読み取り処理などのレスポンスが落ちてしまったのは失敗だった。次回こういうシステムを考える際には、センサー読み取り専用のマシンをおくようにしたい。

5.2.CG班

Genshiの企画において一つの核となるのが3Dグラフィックスの表現力である。体験者が創造世界にいきいきと引き込まれるような画像、動き回るだけで楽しく感じられる画像、つい見とれてしまうような画像、以上のような3DCG描画システムの実現がCG班に課された課題であった。

5.2.1.開発環境

3D描画エンジン...DirectX6
ジオメトリエンジン...自作ライブラリ
開発言語...Microsoft Visual C++ 5.0

5.2.2.狙い

以上の目的実現の為に掲げたポイントは以下の3点である。

リアルタイムレスポンスの追求
迫力あるマップの作成
環境効果の実現

5.2.3.制約条件と実現要件

<最低ハードウェア性能>

CPU...Pentium166MMX
メモリ容量...64MByte
グラフィックボード...Riva128(4MB)

<実現目標>

30fps (秒間30フレーム) の実現。

5.2.3.ソリューション

高速化の為にやったことは色々あったのだが、代表的な手法が以下の3点である。

1)クリッピングテスト処理

Genshiのゲーム画面は広く(ランドスケープ形式で $(128+1) * (128+1)$ の頂点を持つ)、全ての地図を毎回描画するわけにはいかない。そこで $8*8$ マス分を1エリアとしてまとめ、 $16*16$ の独立したエリアでマップを構成した。それらを描画前にクリッピングテストし、カメラに描画される可能性のあるエリアのみを描画している。

2)モデルの最適化

30fpsの書き換え速度表示や、表示媒体がHMDであることからオブジェクトの細かいディテールを作り込むのではなく、グーローシェーディングやフォグ、エンチエイリアシングを利用し、クオリティ的に問題が出る寸前まで削れるポリゴンを削り高速化するという事を行った。

3)リニアシンクロの採用

速度の違うマシンで動作させる場合や、描画するシーンによってフレーム数が不安定になるのがリアルタイム3Dの一つのネックである。それを補完する手法としてリニアシンクロを採用した。これはオブジェクトの制御を基準キーフレーム単位で処理するようにしておき、前回の処理にかかった時間を係数として基準キーフレームにかけて動作の補完をしてやるという手法である。

5.2.4. プログラムを作成して

現時点の3D描画環境はジオメトリ演算をCPU、ピクセル描画処理を3Dアクセラレータボードで行なうという構成になっている。現状では3Dボードの方がCPUに比べて性能が高い傾向にあるので、オブジェクトの頂点数が増えてジオメトリ演算数が増えるのがパフォーマンスを落とす最大の原因となる。つまり処理速度を稼ぐにはクリッピングの仕方などを工夫して頂点演算回数を減らすのが最も効果的となる。

今回色数は16bit (65,536色) を標準とした。現在のアクセラレータボードが一番高速に動作するのがこのモードだからである。色数に対する不満は殆ど無かったが唯一問題となったのは空のような単色グラデーションの表現であった。6万色では色数が不足しマッハバンドが発生してしまい、ディザ処理で誤魔化したが見え透き通るような空は演出できなかった。

VRAMの容量は4Mbyteを基準に設計した。バッファには表示用バッファが1枚、トリプルバッファリングを行なったのでバックバッファが2枚、16bitZバッファが1枚の合計4枚を使用した。バッファ1枚は640*480pixelで16bitだから614,400byte。バッファ4枚で2,457,600byteだから、残り約1.5Mbyteがテクスチャ用となる。今回はテクスチャに1.5Mbyteもあれば問題無かったが、空などの解像度は下げなければならなかった。

CG班としての目的は大方達成出来た。今後は細かい演出の強化、ライブラリの使いやすさなどを改良していき、今後に備えたい。

5.3. AI班

AI班ではGenshiにおいてマンモスの動作の制御を担当した。基本的に用いたのはNeural Networkだが、教師信号を必要とせず学習が比較的容易なGA (Generic Algorithm) という手法を用いて、淘汰と突然変異によってのみ学習をさせた。開発言語にはBorland Delphi 3.0を使用し、DLL化することでGenshi本体への組み込みを行った。

5.3.1. マンモスの見ている要素

マンモスは餌、壁と障害物等、プレイヤーの位置、仲間の位置の4種類の入力を受ける。出力は前方、後方、前右、前左の4つである。

5.3.2. 人工生命の育て方

人工生命を育てるには、段階を踏んでいく必要がある。

はじめは設定をかなり甘くする。そうすることにより4種類ある入力層の内の1つだけ、例えば餌の入力層のみ見えるようにする。

次にその個体を敵のいる設定の中に放し、敵が見えるようにする(ここでいう「見える」とはきちんとした反応が出来るということ)。つまり餌が見えることは他の生物との差にならなくなってしまふのである。

このように甘い設定から開始して徐々に新しい設定を加えていくことで、最終的に全ての入力層に対してきちんと反応する個体が出来てくるのである。いきなり厳しい設定で学習を行わせようとするのは、アフリカのサバンナに生まれたての赤ちゃんを放ってしまうようなもので、学習する前に全滅してしまう可能性が高いのである。

5.3.3. 遺伝子のみを使うときの注意

今回のGenshiでは遺伝子と判断を担当したのだが、育成用のプログラムとGenshi本体の設定(例えば壁にぶつかったときの処理など)の違いにより実装がうまく行かなかった。もったきちん

と処理について話し合う必要があった。また一度きちんと動く遺伝子（ある程度で良い）でテストしておくべきだった。このように制作者が異なる場合は話し合いとテストは不可欠である。

5.3.4.GAの長所と短所

<長所>

マシンをセッティングしてしまえば、あとはコンピュータが勝手に学習してくれるので、人がつきっきりにならなくて良い。

意外な行動が実は良かったりする。

判断基準の定めにくい問題を解くのに効果的。

<短所>

時間がかかる。

どのように判断してその行動を起こしているのかが分かりにくい。

遺伝子の数が増えると優秀な個体を作るのに非常に時間がかかる。

初めは、全滅の可能性が大きい。

5.3.5.まとめ

今回は会場ではじめて遺伝子をGenshiに組み込んだため、学習結果がちゃんと反映されたかどうかを判断する時間がなかった。やはり一度テストしておくべきだった。

ミュートーションの値をいじるよりは、エリート制を導入した方が効果的だったのではないかと思った。

今回は「こう動いて欲しい」という前提に基づいて条件を作っていたが、次は「どう動くのだろう」という問題をAIで作りたい。

5.4.DSP班

5.4.1.開発環境

DSP...TEXAS INSTRUMENTS社のTMS320C5X

使用言語...TMS320C5X用アセンブラ

5.4.2.原理

人の声は基本的に母音と子音により発声される。母音とは「あいうえお」つまりローマ字で表すと「AIUEO」のことである。子音とは「かきくけこ」では「K」、「さしすせそ」では「S」などと表現出来る。これを声として出した時、子音が先に出て母音が後に続くのである。

基本的に子音は波形が小さく母音は波形が大きい傾向にある。つまりDSPに小さい波形をカットするようなフィルタをプログラミングすれば母音ヘッドホンは実現出来るのである。

5.4.3.作成した結果

結果的には期待した効果が得られなかったシステムになってしまった。原因としてはいくつか挙げられるが最大の問題は使用したヘッドホンとマイクの性能にあった。ヘッドホンは外界の音をシャットアウト出来ず、またマイクがプレイヤーの声以外の音を拾ってしまった為、VR世界の音声をヘッドホンから流すという当初の目的を実現できなかったからである。

DSPのフィルター自体も、母音ヘッドホンというよりは原始人ヘッドホンになった。これはDSPのフィルターでカットする音の領域が発声する人の声の高さによって変化してしまうことなどから

当初考えていたほど単純には母音のみの切り出しが出来なかったからである。

理屈では出来るはずだということからスタートしたシステムだったが、DSPや音響に関する経験不足から最適な機材の選択と作成手法を採れず、今後の課題が多く見つかったということになった。

5.5. ネットワーク班

Genshiに使用している通信コンポーネントはTCP/IPプロトコル環境上での運用を想定している。また、小規模なLAN上で運用することを想定しているため、データの欠損や輻輳はほとんど発生しないものと仮定し、データの欠損等のトラブルに対する対策は行っていない。

5.5.1. 開発環境

Genshiのグラフィック、音声周りはDirectXを使用しているが、この通信コンポーネントの作成はBorland Delphi 3.0を使用している。Genshi本体への組み込みはDLL化することで行った。

5.5.2. ネットワーク処理

データを送信する際、データと共にデータサイズを送信するが、受け取り側は、最初に送信されたデータサイズを参考に、バッファに収納されたデータの切り分けを行う方式を取っており、データサイズは最初に取り除いて別の配列に收容してしまう為、実際に扱うのはパケットサイズのデータのみを送受信することになる。これによって、TCP/IPではよく発生する、一度に複数のデータを受信した場合も正確なデータを読み出すことが出来る。

やり取りされるデータはあらかじめ設定されたバッファに貯えられるが、バッファはリングバッファ方式を採用している。通常のバッファは連続してデータが送られてきた場合、前のデータを上書きしてしまうか、データを積み重ねる場合、読み込んだ後に次のデータを先頭に移動させていかなければならないので、非常に遅くなる。リングバッファ方式は、データ自体を移動させるのではなく、書き込み・読み込みポイントを移動させていく。したがってデータの移動が無い分、高速化する。

5.5.3. ネットワーク環境を作成して

ネットワークの送受信などの一般的な処理はDelphiのコンポーネントを利用することで比較的簡単に作成出来たが、Genshiへの組み込みの際に問題が発生した。特に苦労したのがデータフローの問題である。使用したコンピュータの性能が違っている上にサーバとクライアントで処理内容が違う為、一通りの処理を完了するまでにかかる時間に差が発生し、状況次第では数個のパケットが溜まってしまうことがある。データフロー対策を完全に行おうとすると非常に複雑なチェックルーチンが必要となり全体の処理速度も落ちてしまうことから、今回は応急処置としてリングバッファの容量を溢れないであろうという分だけ確保することで対応した。

5.6. 会場設営

ブースは竹で櫓を組み、それを黒いメッシュ状の布で覆い、ライトを下から上へ照らして原始時代の雰囲気を出すよう工夫をした。ブースに入った瞬間から作品の雰囲気に浸ってもらうことで、より深い感情移入を演出できたのではないかと思う。

待っている人たちに現在の状況を伝える為にサーバマシンの画像を液晶プロジェクタで壁に映し出した。会場でのプレゼンはこのサーバ画面を中心に行い、大きな効果が得られた。残念だったのは開発時間が足りなくてサーバ画像のカメラアングルを種類しか用意出来なかったことである。

当初は今回入れた上空からプレイヤーを追うショットの他にプレイヤーの視点からの画像、マンモスからの画像、定点からプレイヤーを追跡するショットなどを入れる予定であった。これらがあればより大きな効果が得られたと思われる。

展示日の後半から各マシンに接続していたデバッグ用のモニターをそのままつけておき、各プレイヤーのしている画像を待っている人たちにも見えるようにした。トラブル解決の際に偶然思い付いた演出で、待っている人たちが事前に作品の内容が分かる、作品の迫力が伝わるなど、思った以上の効果があった。

5.7. 自転車コントローラ

当初はペダル以外の全てを自作する予定であったが、知識と経験が足りなかったことからシャシーの強度確保やデザインを実現する為に適した材料や加工の方法が判らず、結局倒れないように支えを付けた既存の自転車に市販のハンドルコントローラとダイナモを利用したペダル速度測定センサーをつけることで代用した。

シャシーの自作が出来なかったことから、乗り心地の不安定感、プレイヤーの身長の変化に対応出来なくなった、センサーがペダルの回転をダイレクトに読み取らない構造になってしまうことでブレーキや後進の操作がペダルのみで出来なくなったなどの問題が発生した。

5.8. 槍コントローラ

1m弱の長さをもつ棒の先に圧力センサーを仕込んだ槍投げコントローラ。当初はポヒマスセンサーを使い、コントローラの位置と加速方向、それに加速度をそれぞれ測定し、槍を投げ始める位置、投げる方向と初速を決定しようとしていたが、開発期間の都合から初速のみを読み取る構造の単純な圧力センサーを使用することになった。

強度不足からの故障の多発や、データの読み取りをサーバマシンに行かせたことからのレスポンスの遅さなどの問題が発生したが、圧力センサーを利用したことやコントローラとしてのユニークさは良い評価を受けた。

5.9. コンピュータ

展示には本展示用に4台、子供用に考案した簡易体験版用に1台の計5台のコンピュータを使用した。使用したOSは、殆どのマシンが自作Windowsマシンだったこともありデバイスドライバの安定度を考えてWindows98ではなくWindows95に統一した。

会場では使用する予定であったコンピュータのうちの1台がHDDトラブルとグラフィックボードのトラブルを発生させ、デバイスドライバを会場に持ち込んでいなかったことから復旧が遅れ、結局3人同時プレイを実現することが出来なかった。

使用するコンピュータに要求されることは、3Dアクセラレータを搭載している高速なグラフィックボードを積んでいること、10Base-TのLANボードを搭載していること、それからジョイスティックを接続出来るようにサウンドカードを搭載していることである。これらを満たす為、グラフィックボードとサウンドカードをいくつか購入することになった。

通常使う分には問題のなかったコンピュータも、展示させる為の最適な設定を施そうとすると様々なトラブルが発生し、その調整に時間がかかった。

5.10. 会場でのプレゼン

原始人をイメージした服を着たスタッフを置き、操作の説明、ストーリー、狩りの実況を行い、

雰囲気盛り上げた。また会場で配布したパンフレットも好評で、展示2日目で持ってきた全てのパンフレットがはけてしまうほどであった。

プレゼンに関しては体力の配分をしなかったことから最終日にトーンダウンしてしまったのが悔やまれるが、IDESEMIの色を良く出したという点で成功だったと感じている。

6. 総括

様々な想定のもとに開発した「Genshi」であったが、展示により得られたフィードバックは色々であった。

全体としては予想を上回る良好な反応を得られ、特に外国の方々と子供連れの家族からの反響は大きかった。自転車を漕ぐ、槍を投げる、マンモスを狩るというように要素を単純なものに絞り込んだことが成功をもたらしたと考えている。

「操作性が悪い」「不親切」という指摘も少なからずあった。自転車コントローラが万人向けのサイズではなかったこと、操作性を調整仕切れなかったこと、マップを十分に作り込めなかったこと、システムの設計不足から槍コントローラの反応が鈍くなってしまったことなどが原因である。また様々なものの故障から結局同時に二台しか動かせず、試乗の回転率が悪くなってしまったのも問題だった。沢山の人が見に来てくれた時に一部の人にしか体験してもらえなかったことは残念だった。

まとめると以下のようなになる。

<良かった点>

- エンターテインメント性重視にしたこと
- 要素を単純なものに絞り込んだこと
- 自転車という身近なコントローラを採用したこと
- 会場でのプレゼン

<悪かった点>

- 操作性の調整不足
- マップの作り込み不足
- 各ルーチンの統合時の行き違い
- 回転率の悪さ

Genshiの一番の問題点は、開発期間に比べて企画の要素が多過ぎて細かいところを仕上げる時間が無かったことであろう。しかし企画の方向性としては間違っていなかったという実感はある。企画の目的である「現代の抑制された倦怠感漂う「くらし」に喝を入れられるような作品を提供すること」は、多くの体験者の反応と自分達で作品を体験してみた感触から達成出来たと判断している。

7.多摩大学「IDESEMI」メンバー紹介

CG班

田中 雄（代表） u6215yt@elm.timis.ac.jp

近藤 聡明

中島 幸史

並木 良夫

黒瀬 洋

作曲担当

森 有樹

マップ作成

伊藤 賢一

イメージング担当

佐藤 信吉

マップイメージ作成

増倉 陽子

AI班

高木 俊克

小島 宏之

DSP班

喜多 恒

深澤 久

高橋 香奈

Network班

古川 雅也

上山 洋一

電子工作班

古作 太郎

工作班

岩松 勝義

益子 健司

池田 智之

プレゼンテーション

笠 明日香

青山 晋祐

被服

針山 愛子

コーディネート担当

大和田 健人

ドクター

出原 至道