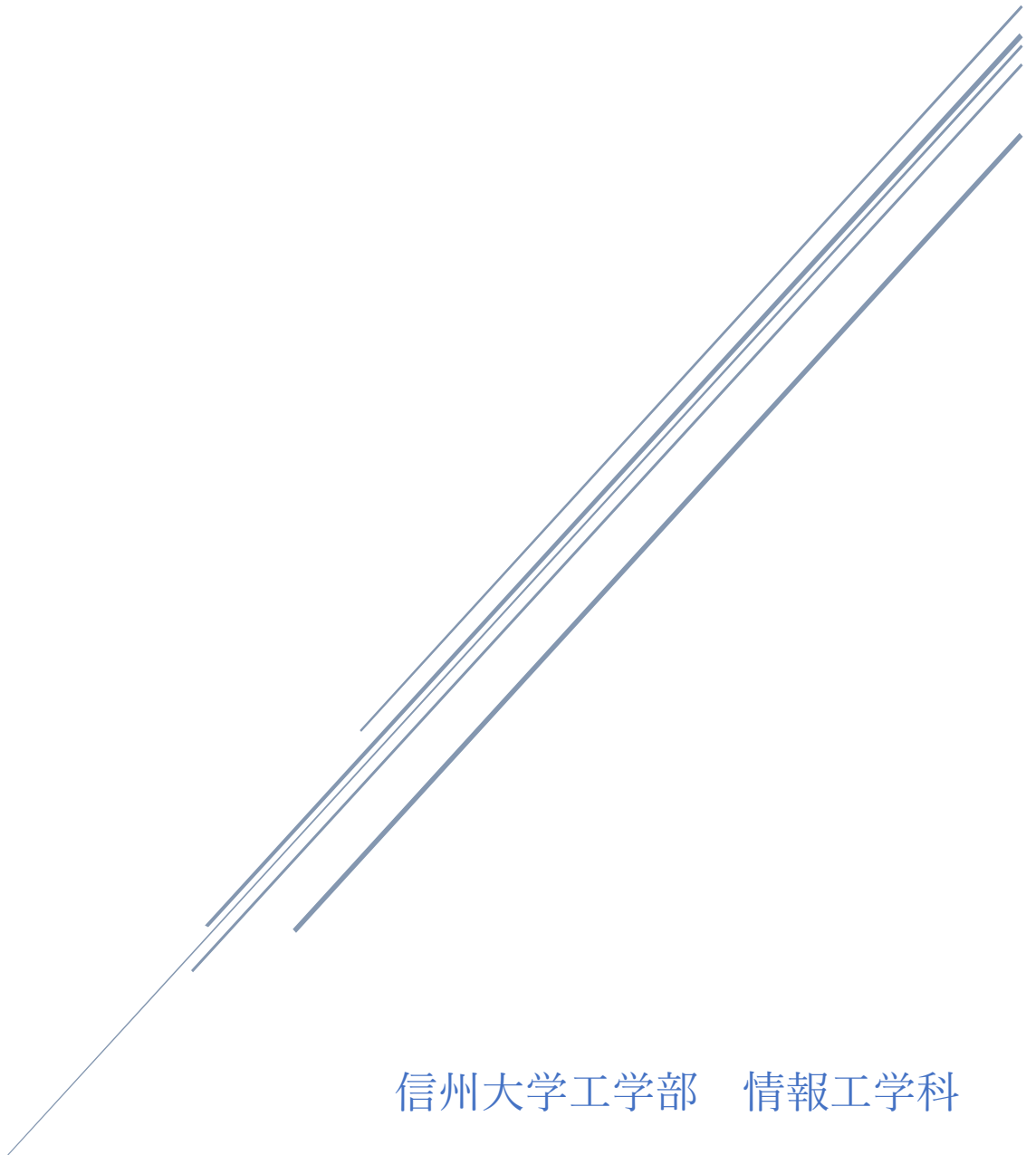


3D だるま落としの制作

2017 年度 コンピュータグラフィックス 課題2 レポート



信州大学工学部 情報工学科

15t5801g 宇野 貴士

コンピュータグラフィックス プログラミング課題2 レポート

信州大学工学部情報工学科

15t5801g 宇野 貴士

・課題タイトル

「3D だるま落とし」

・プログラムの説明

本プログラムは物理演算を用いた 3D 空間においてだるま落としを行うためのプログラムである。近年正月も飲食店や食料品店、娯楽施設が開店していることもあり、正月らしい正月を過ごさない家庭や、自らも仕事をしているなどにより、正月文化全般が減衰の一途をたどっていることを感じている。正月の遊技の代表格であるだるま落としも、こうした背景に後押しされ減衰の一途をたどっていることは言うこともなく、現代において、だるま落としを持っていないという家庭が多くあることについては想像に難くない。よって本プログラムはこうした点に着眼し、家庭で道具がなくとも気軽にだるま落としを楽しめる。また、3D 空間で行うことで、既存のだるま落としにはない、新たな楽しみの創生と文化の存続を行うことが可能であると考え、3D 空間で物理演算を行い、だるま落としを行うプログラムの作成を行った。本プログラムでは、現実空間のだるま落としでは難しい、摩擦の設定や、自由な形状の土台を定義している。

・プログラムの機能説明（マニュアル）

本プログラムの最大の注意点としてはブラウザによって動作しないという問題がある。特に Google Chrome において動作しない。これは、physijs の example においても同じ現象が発生するため、この外部の javascript に依存した問題であると考えられる。現在 Microsoft Edge においては、問題が発生しないことを確認しているため、もし使用しているブラウザで動作しなかった場合、**Edge にて動作確認を行っていただきたい**。ただし、サーバーアップロードかローカルホストにおいて実行時においてはこの問題は発生しない。また本プログラムは、複数の難易度を用いるために、いくつかの HTML ファイルがフォルダ内に存在している。どの HTML ファイルにアクセスしても、すべてリンクしているが、**「15T5801-2.html」がスタートアップファイルである**。

まずスタートした画面の状態を以下に示す。

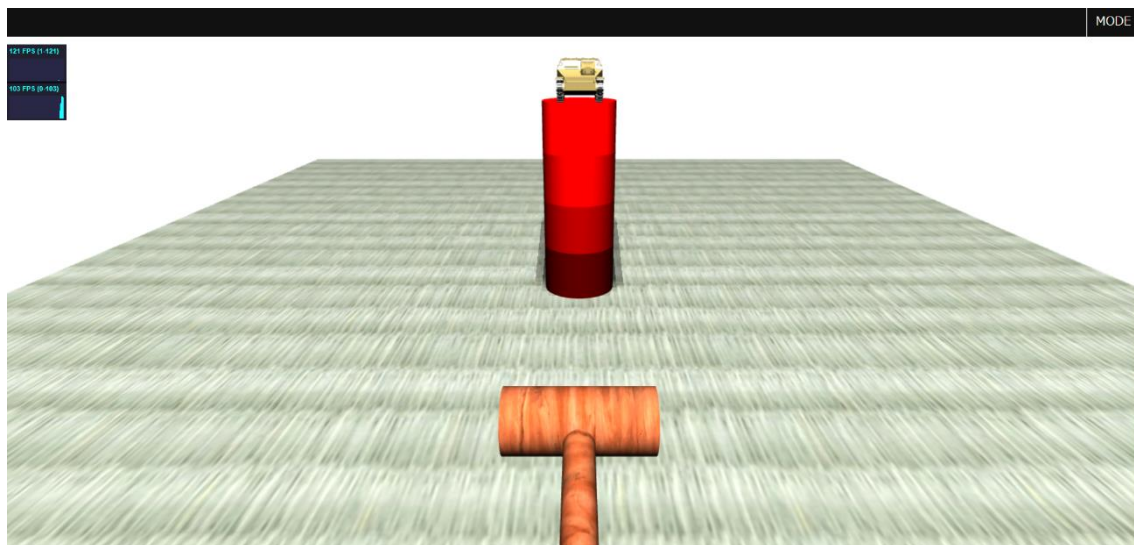


図1 プログラムの実行初期状態

左上にあるのが fps を表示している。

右上にある、MODE から複数の難易度のモードにアクセスすることができる。このMODEをクリックすると、上からメニューが開き、ほかのモードへのリンクが現れる。

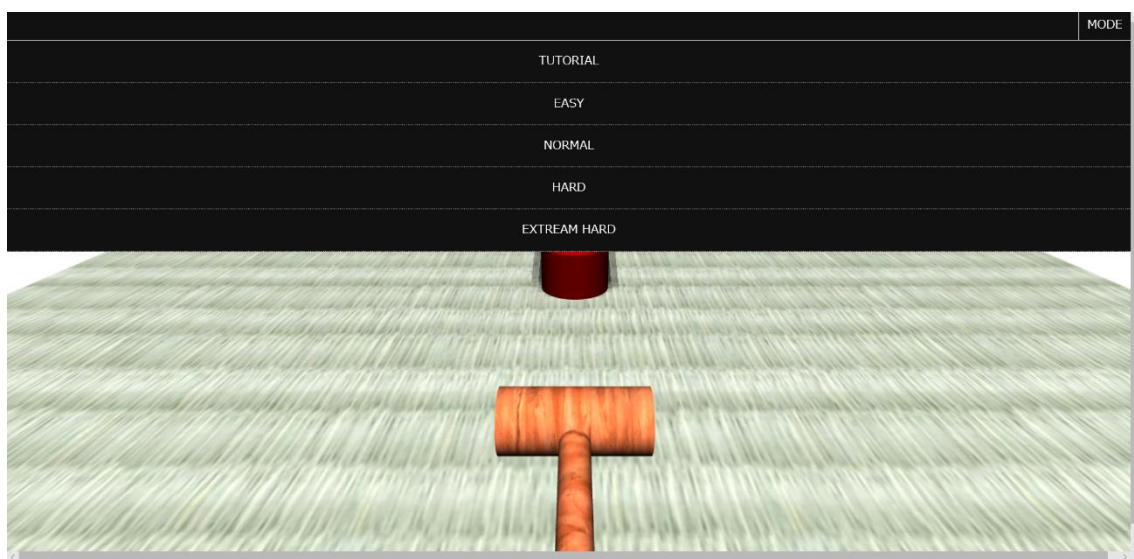


図2 メニュークリック時の動作

モードとしては、TUTORIAL、EASY、NORMAL、HARD、EXTREAM HARD を用意している。これは実際のだるま落としでは、難易度は一定であり、土台の形状を変化させることは難しいため、3DCG の利点を生かし、複数の難易度を作成し、瞬時に切り替えられるようにしてあるのである。また実際のだるま落としでは積む作業の難しい形状でも、3DCG 上では、数値を入力しておけば瞬時に再現することができる。そのため、既存のだるま落と

しにはない利点を生み出している。

だるま落とし自体の機能はシンプルであり、ハンマーの柄の部分をクリックすることで、ハンマーを移動できる。ハンマーやだるま落とし本体は物理演算を行っているため、衝突や重力によってオブジェクトがする。上に乗せるだるまのグラフィックが、未完成であるために仮のオブジェクト（以下 CV33 とする）を配置している。これを地面まで落とすことを目的としたゲームである。モードごとに、摩擦や積み上げる形状などを変更しているが、操作やだるま落とし以外のオブジェクトの配置、機能については共通である。また、勝敗判定やターンなど、実際のだるま落としに則し行わないものとしている。これは地方ごとにローカルルールが存在していることを想定し、システム的な制限を行わないことで、プレイヤーの楽しみ方を尊重するためである。実際のプレイ画面を以下に示す。

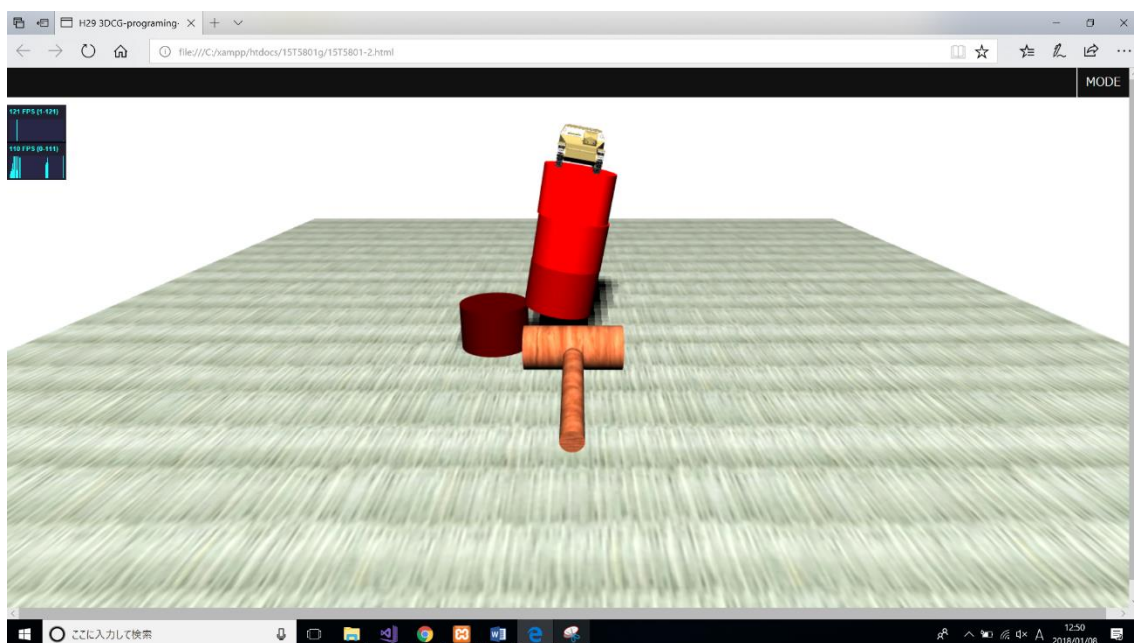


図3 プレイ画面(衝突時)

このプレイ画面は EASY モードであり、ドラックによって十分な勢いでハンマーと衝突させることで、下段のみを抜き、落下させることができる。モードによって、だるま落とし下段の形状だけでなく、摩擦係数や重力などのパラメータを操作することでも、難易度の調節を行っている。

また HARD モードは大量の glass モデルを使用している。このモード多くのモデルを読み込むために、非常にローディングが重いので十分なスペックの PC でないと動作しない可能性がある。

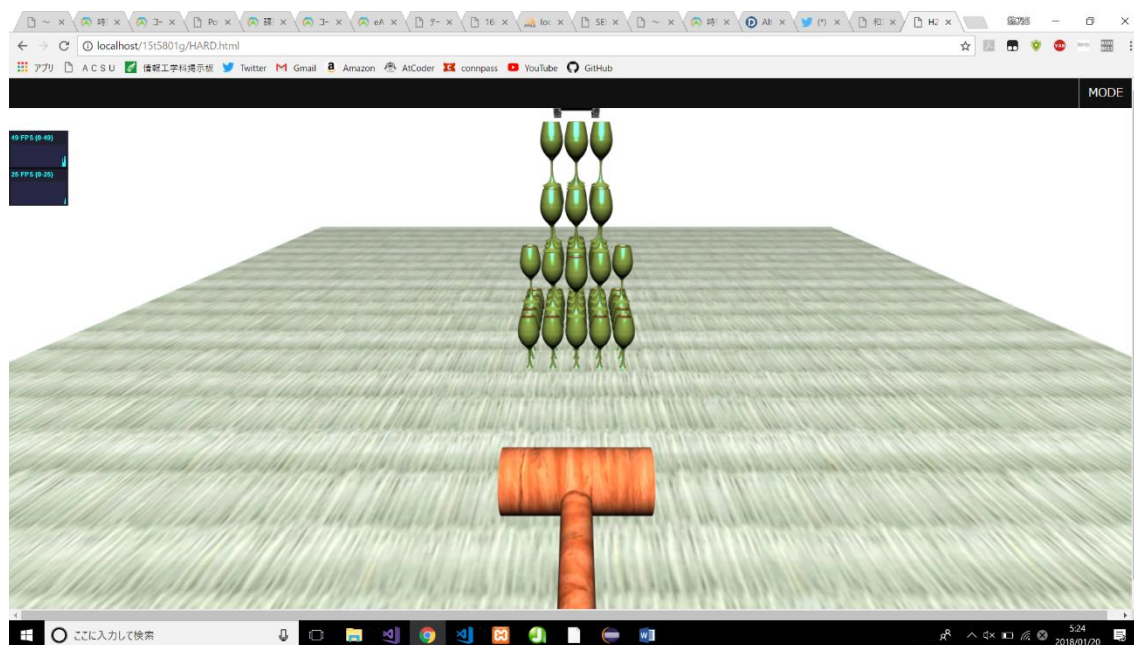


図4 ハードモードの実行時

- ・上記の機能を three.js 等の機能をどのように使って実現しているかの説明

これらの機能 three.js と physijs によって実装している。physijs は物理演算シミュレーションを可能にする three.js のプラグインである。three.js でオブジェクトを作成するのとあまり大きな違いなく、物理演算可能なオブジェクトを生成することができる。以下に今回大量に用いた円筒の実装例を示す。

```
var cylinderGeometory_daru=new
THREE.CylinderGeometry(cylinder_r, cylinder_r, cylinder_height, 100,
100);

var cylinderMaterial_daru = Physijs.createMaterial(
    new THREE.MeshLambertMaterial({
        color:0x440000,
        shininess: 100
    }),.1,.1
);

var cylinders=new
Physijs.CylinderMesh(cylinderGeometory_daru,cylinderMaterial_daru);
```

Geometory は three.js のクラスをそのまま利用可能である。Material は physijs のクラスであるが、第一引数に three.js のクラス、第二引数に摩擦係数、第三引数に反発係数を書けばよい。形状だけでなく、この反発係数と摩擦係数の値を利用して、より反発させることや、より抜けにくくすることで各モードの難易度の調節を行っている。Mesh の作成は physijs のクラスで行うが、three.js と書き方は同じである。だるま落としの台座は cylinder、地面は box に畳のテクスチャを使用している。ハンマーは cylinder 二つにより定義を行い、柄の子として、ハンマーヘッドを定義し、木のテクスチャを張っている。

CV33 は obj ファイルである。obj ファイル自体は three.js に loader があるため容易に読み込むことが可能であるが、読み込んだオブジェクトは、そのままでは衝突判定ができない。そのため、透明にした box オブジェクトで CV33 を囲い、CV33 を box オブジェクトの子として親子関係の定義を行うことで、あたかも CV33 が衝突しているように見せている。HARD モードの glass も同様の方法で衝突判定を行っている。

Scene の作成については、ほぼ同じであるが重力の設定を行わなくてはならない。Scene の作成と重力の設定を行うソースコードを以下に示す。

```
scene = new Physijs.Scene({ fixedTimeStep: 1 / 120 });
scene.setGravity(new THREE.Vector3( 0, -100, 0 ));
```

Scene の作成時の fixedTimeStep は一つのシミュレーションステップの時間である。この値が小さいほど、より正確なシミュレーションが可能となるが、動作が重くなる。setGravity によって重力のベクトルを示すことができる。今回はオーソドックスに y 軸下向きのベクトルとしている。この値はモードによって変えている。強すぎる場合自重による崩壊を招くが難易度が低下し、弱すぎると、下からのわずかな衝撃で上のオブジェクトが跳ね飛ばされてしまう。

ドラッグによってハンマーを移動させる方法については、基本的には後述する physijs の examples にある jenga.html にある処理を利用している。この操作は主に MouseMove,MouseDown,MouseUp の3つのイベントハンドラを定義している。今回クリックする点については、柄のみであるが、複数のブロックに対して、クリックによる移動を可能とするかを定義することができる。

本プログラムは上で説明した要素を組み合わせ実現している。

- ・ 参考にした事例があれば何をどの程度参考にして作成したかの説明

-threejs 関係

<http://chandlerprall.github.io/Physijs/examples/jenga.html>

Physijs の examples にある jenga.html を非常に参考になっている。

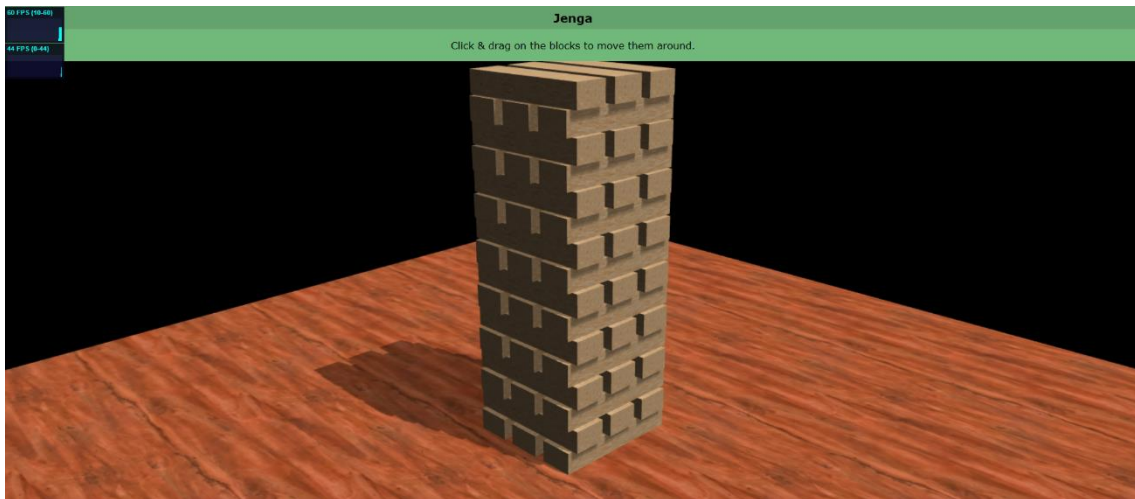


図5 jenga.html 実行画面

このプログラム物理演算により、摩擦、重力、衝突などを定義し、ジェンガをたて、クリックでブロックを動かすことができるプログラムである。このプログラムから、クリックによりオブジェクトを動かす方法をハンマーにのみ適応すること、および物理演算をこのプログラムを参考になっている。

-モデルデータ

以下はプログラムに組み込んだモデルデータのソースである。

CV 3 3 model データ

<http://seiga.nicovideo.jp/seiga/im5439530>

glass モデルデータ

<https://free3d.com/3d-model/wineglass-48650.html>

-BGM 音源

以下は組み込んだ音源のソースである。

和風音源

http://www.hmix.net/music_gallery/image/asian.htm

メタル音源

<https://maoudamashii.jokersounds.com/list/bgm2.html>