

平成 29 年度 学士論文

仮想空間のインタラクティブなバーチャルペットの開発

千葉工業大学
情報科学部 情報ネットワーク学科
1432104 中島 葉瑠奈

指導教員 _____ 菅原 研次
_____ 真部 雄介

目次

第 1 章	序論	1
1.1	背景	1
1.2	関連研究	1
1.3	目的	2
1.4	本論文の構成	2
第 2 章	利用技術	3
2.1	Unity	3
2.2	HTC Vive	3
2.3	Maya	3
2.4	Visual Studio	4
第 3 章	先行研究	5
3.1	概要	5
3.2	問題点	7
第 4 章	提案手法	9
4.1	概要	9
4.1.1	仮想空間	10
4.1.2	ユーザ	10
4.1.3	バーチャルペット	10
4.2	ユーザ管理部	11
4.3	ペット管理部	11
第 5 章	実験と評価	12
5.1	実験環境	12
5.2	実験用バーチャルペット	12
5.3	実験方法	13
5.4	評価方法	13
5.5	実験結果	13
5.5.1	アンケート	13
第 6 章	結果と考察	14
6.1	結果	14
6.2	考察	14
6.3	今後の課題	14
付 録 A	コルーチン処理による Unity-chan の発生	17

図 目 次

2.1	HTC Vive	3
2.2	コントローラ	3
3.1	ちょぼにゃんの外観	5
3.2	ちょじにゃんの外観	6
3.3	反応行動	6
3.4	実験のタイムチャート（心理的効果）	6
3.5	実験のタイムチャート（生理的効果）	7
4.1	提案システムの構成	9

表 目 次

3.1	画像による表の例	8
4.1	スクリプト一覧	10
4.2	コントローラの入力一覧	10
4.3	ペットの動作一覧	10
5.1	開発、実験環境	12
5.2	ペット条件の比較	12

第1章 序論

本章では、本研究における背景と目的を中心として述べる。

1.1 背景

近年、精神疾患を抱える患者が増加している。その要因として、人間関係や環境変化・身体の不調等から来るストレスがある。ストレスはうつ病といった精神疾患のみならず、神経性胃炎・十二指腸潰瘍を引き起こすなど身体に悪影響を及ぼす。ストレス社会とも呼ばれる現代を生きる上でメンタルヘルス対策への取り組みは必要不可欠といえる。

ストレス対処法の1つとして、アニマルセラピー（Animal Assisted Therapy）がある。動物と触れ合うことにより精神・身体機能の向上を図る治療法で、（アニマルセラピーの効果事例）のように、効果が確認されている。しかし、この治療法には生きた動物と接するが故のリスクが存在する。例えば不特定多数の人間が集まる医療施設では、アレルギーを持つ人や動物に苦手意識がある人への配慮が必要である。更に、セラピーに使用する動物には吠える・噛むといった問題行動をしないよう徹底的な訓練を行わなくてはならず、また動物を管理するため医師や看護師とは別に専門的な訓練を受けた人間が必要になる。

アニマルセラピーの代替案として、近年ペットロボットやバーチャルペットの研究が進められている。ペットロボットとは、実用性よりも娯楽性に重きを置いて設計されたロボットのことである。動きなど目で楽しませるものから、対話を行うことができるものまでその機能は幅広い。当初はペットロボットという名前通り動物の形を模したものが多く存在したが、現在は人の形のロボットも開発されている。中でも産業技術総合研究所が開発したアザラシ型ロボット「パロ」は2002年に世界一の癒しロボットとしてギネスブックにより認定されており、現在においても医療機関・介護施設等で活躍している。2017年にはSonyによって「aibo」が発表されるなどその発展は目覚ましい。

一方、バーチャルペットはゲームソフトやスマートフォンアプリなど、気軽に触れることのできるコンテンツが多く存在する。最も知名度の高いものとして、たまごっちがあげられる。画面上に存在する「たまごっち」と呼ばれる架空の存在を飼育し、成長させることを目的としたキーチェーンゲームで、1990年代において社会現象を巻き起こした。現代ではステージである家の庭先に猫を集めるスマートフォンアプリ「ねこあつめ」が人気を博し、2015年にはCEDEC AWARDS2015ゲームデザイン部門最優秀賞を受賞している。

1.2 関連研究

林らはペットロボットとバーチャルペットとのふれあいによるセラピー効果の差異を身体性の観点から比較検証を行った。身体性とは身体が持つ性質を指すが、その定義は分野ごとに変わる。林らは実体を有することにより、ユーザや周辺環境との物理的な相互作用を可能とすることを身

体性として定義している。実験の結果、バーチャルペットに比べてペットロボットの方が心理的・生理的にストレス緩和効果が高いことが分かった。その要因として、ペットロボットが身体性を有することを述べている。

しかしながら、バーチャルペットにおいても VR 技術を用いることにより、疑似的ではあるが身体性を持つことは可能であると考ええる。身体性がセラピー効果と深い繋がりを持つことから、身体性を有することができればバーチャルペットのセラピー効果向上が期待できる。

1.3 目的

そこで本研究では、VR 技術を利用して身体性の特性である接触フィードバックの付与・現実感の向上を行うことで身体性を有するバーチャルペットの開発を目的とする。

1.4 本論文の構成

本論文は、本章を含め 7 章から構成される。第 1 章では、序論として背景、目的と、本論文の構成について述べた。第 2 章では、関連技術として、エージェントシステム、エージェント通信言語、パーソナルアシスタントエージェントについて述べる。第 3 章では、エージェントプラットフォーム OMAS の概要、日本語対応に向けた課題と解決方針、評価システムについて述べる。第 4 章では、日本語対話エージェントシステムの構成、動作、機能設計について述べる。第 5 章では、実装環境、使用した技術、作成プログラム、日本語対話エージェントの利用法について述べる。第 6 章では、履修アドバイスをドメインとした対話実験について述べる。第 7 章では、まとめと今後の課題について述べる。

第2章 利用技術

本章では，本研究に利用する各技術について述べる．

2.1 Unity

Unity とは，ユニティ・テクノロジーズが開発した総合開発環境を内蔵するゲームエンジンである．モバイル，VR,AR, デスクトップ，コンソール，Web と数多くのプラットフォームに対応しており，その柔軟性は極めて高い．簡易なものであればプログラミングを行わずとも 3 D ステージの構築や物理エンジンの導入が可能である．また 3 D だけでなく，2 D 描画にも対応している．更に，アセットストアと呼ばれるショップで他のユーザが作成したプログラムや 3 D CG モデル，テクスチャといったデータがアセットという形で提供されており，プロジェクトに直接インポートして使用することができる．3 D 空間の作成が容易であることから，本研究では U n i t y を使用する．

2.2 HTC Vive

HTC Vive とは，HTC と Valve Corporation の共同で開発されたバーチャルリアリティ向けヘッドマウントディスプレイである．部屋の二箇所にベースステーションを設置することで，最小 1.5m × 2m，最大 3m × 4m の範囲でユーザの動きをリアルタイムにトラッキング，仮想現実反映できる．VR 専用に設計されたコントローラが付属しており，直感的な操作とジェスチャーを実現可能．本研究では現実感を高めるために HTC Vive を，接触フィードバックの実装にコントローラを使用する．



図 2.1: HTC Vive



図 2.2: コントローラ

2.3 Maya

Maya はオートデスク社によって開発されたハイエンド 3 次元コンピュータグラフィックスソフトウェアである。ハリウッドをはじめとした映画からゲーム，CM の制作に使用されている。映画の VFX 市場で圧倒的なシェアを誇り，ゲーム市場においても Maya と同じく 3 次元コンピュータグラフィックスソフトウェアである 3ds Max と 50 % 50 % であるとオートデスク社は述べている。本研究で使用したペット，ユーザの手のアバターは Maya を用いて作成した。

2.4 Visual Studio

Visual Studio は Microsoft 社が開発した統合開発環境である。C #，Visual Basic，Python といった 8 種類のプログラム言語に対応している。また，Windows だけでなく iOS や Linux などの OS にも対応可能である。Unity には MonoDevelop という統合開発環境が標準で備わっているが，Visual Studio がこのスクリプトエディタよりも高機能であることから，本研究では Visual Studio を利用する。

第3章 先行研究

本章では、本研究における先行研究を述べる。

3.1 概要

林らは、バーチャルペットと比較して実体を有するペットロボットの方がセラピー効果が高いと考え、ふれあいによる心理・生理的なストレスの緩和効果の差異を比較・検証した。

検証に使用するペットロボットは、林らが開発した柔らかい触感を有するセラピーロボット「ちょぼにゃん」で、頭部に配置された接触を検知するセンサ部、人工筋肉でできた尻尾に位置する感情表出部、感情表出部を制御する制御部から構成されている。



図 3.1: ちょぼにゃんの外観

バーチャルペットは「ちょぼにゃん」を模した3DCGキャラクターの「ちょじにゃん」を開発、実験に使用した。「ちょじにゃん」はユーザの接触動作をペットロボットに出来る限り近付けるために、Leap Motionを用いる。これによりトラッキングされた手の位置は仮想ハンドとして画面上に反映され、ちょじにゃんへの頭を撫でる、叩くといったユーザの接触を検知できる。



図 3.2: ちょじにゃんの外観

ペットロボット、バーチャルペット共にユーザが頭を撫でた時は尻尾を左右に振り喜びを表現を、ユーザが頭を叩いた時には尻尾を垂れ下げ悲しみ表現を表出するといった二種類の反応行動を行う。

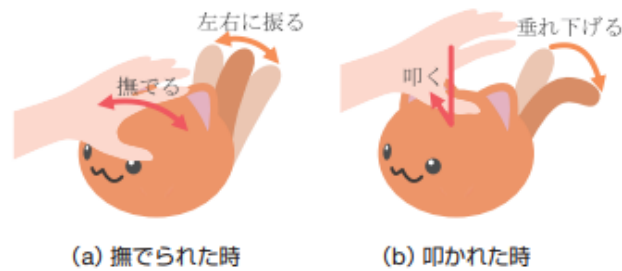


図 3.3: 反応行動

セラピー効果を比較するため、心理的、生理的の二面からストレス緩和効果の比較検証実験を行った。実験参加者は両実験共に 25 歳から 45 歳までの男女 18 名（内、女性 8 名）である。

心理効果の評価実験では、まず実験参加者にストレスを与えるために、被験者に計算問題を課した。240 個の 1 桁の正整数乱数を羅列した計算用紙を用意し、隣り合う数字を加算する作業を課題として 2 分間行った。計算課題終了後、POMS と呼ばれる気分プロフィール検査を実施した。POMS は成人用に全項目版と短縮版の 2 種類が用意されているが、短期的な気分変化を測定するために一時気分尺度 18 項目に対し 5 件法で回答を求めた。質問用紙を回収後にペットロボット、もしくはバーチャルペットと 2 分間自由にふれあってもらい、再度 POMS 検査を行った。最後に、ふれあったペットの印象の評価として SD 尺度法を用い、特性形容詞尺度 20 項目に対して 7 件法で回答を求めた。以上の流れを、条件を変えて 2 回実施した。

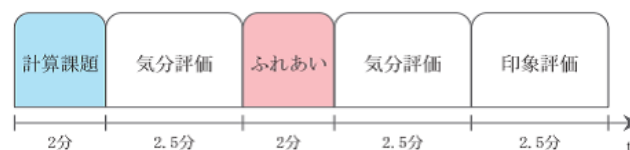


図 3.4: 実験のタイムチャート（心理的効果）

実験の結果、ペットロボットとバーチャルペットともに心理的なストレス緩和効果があることが分かった。しかしながら、ペットロボットはバーチャルペットと比較して有意な変化が認められた尺度が多く、特に緊張の緩和効果と活気の尺度において高い効果を示した。印象評価は親密性、力本性、信頼性の3次元?? 親密性ではペットロボットの方で高い値が確認されたが、力本性と信頼性においてはバーチャルペットの方が高い。以上より、ペットロボットとバーチャルペットが与える印象の性質に差があると記述としている。

(因子負荷得点平均の比較の図挿入)

生理的効果の評価実験では、心理的効果測定と同様にまず計算課題を実施する。その後、図に示す MindWave Mobile を用いて被験者の安静閉眼時の脳の状態を1分間計測した。MindWave Mobile とは、国際10-20法のFp1領域の脳波を計測可能な簡易脳波計である。次に、ペットロボット、もしくはバーチャルペットと2分間ふれあってもらい、その後再度被験者の安静閉眼時の脳状態を1分間計測した。以上の流れを、条件を変えて2回実施した。

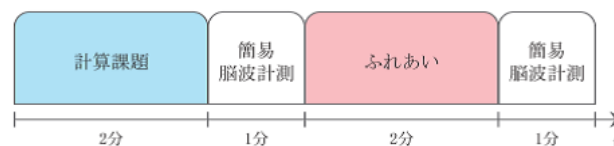


図 3.5: 実験のタイムチャート（生理的効果）

実験の結果、ペットロボットには α 波の含有率の増加、 β 波の減少が確認された。 α 波はリラックスしている時に、 β 波は精神的に緊張しているときに多く出現するとされている。一方バーチャルペットには、 α 派、 β 波共に変化が殆ど確認できなかった。このことから、ペットロボットは生理的なストレス緩和効果を持つがバーチャルペットは持たないということが分かった。

(脳波の変化量の比較図挿入)

以上二つの評価実験の結果から、生理的、心理的どちらの側面においてもペットロボットの方が高いストレス緩和効果を得ることができた。検証した全ての比較結果を下の表に示す。このような結果になった要因として、「接触フィードバック」「現実感」といった二点のキーワードをあげている。まず、接触フィードバックとはユーザの接触に対するペットからのフィードバックのことである。ペットロボットは現実存在しているために、接触すればその感触が返ってくる。しかしながら、バーチャルペットは架空の存在であるために感触が返ってこない。実験参加者の9割が、自身の手が接触しているのかどうか分かりづらく、戸惑ったと答えている。林らは接触フィードバックは身体性が持つ特性の一つであると記述している。

次に現実感について、実験に使用されたバーチャルペットはタブレット上に表示された3DCGキャラクターであり、このことから実験参加者の8割がゲームで遊んでいるような感覚だったと答えた。一方ペットロボットとのふれあいにおいては、実験参加者の7割が動物と接している感覚に近く、親しみやすかったと答えた。現実感についても身体性が持つ特性の一つであると記述している。

3.2 問題点

林らはペットロボットと比較して、バーチャルペットは身体性を有さないためにセラピー効果が低いとしている。しかしながら、バーチャルペットにおいても利用技術によって身体性を付与することは可能であると考えられる。本研究では、ヘッドマウントディスプレイを用いて360度全方

セラピー効果	一時的気分尺度	緊張	ペットロボット
		抑鬱	ペットロボット
		怒り	ペットロボット
		混乱	-
		疲労	ペットロボット
		活気	ペットロボット
	脳波含有率	α 波	ペットロボット
		β 波	ペットロボット
印象		親密性	ペットロボット
		力本性	バーチャルペット
		信頼性	バーチャルペット

表 3.1: 画像による表の例

位に映像を出力することで疑似的な現実感をユーザに与え、更にコントローラの振動を利用して接触フィードバックを実装した身体性を有するバーチャルペットの開発を行う。

第4章 提案手法

本章では、本研究で開発するバーチャルペットについて述べる。

4.1 概要

本研究のシステム構成図を 4.1 に示す。本システムは HTC Vive、コントローラ、ヘッドホンから成るデバイス部と Unity により実装したペットシステム部から構成される。



図 4.1: 提案システムの構成

ユーザは HTC Vive、ヘッドホンを装着し、コントローラを手に持った状態で仮想空間に没入する。仮想空間にはペットとユーザの2つのアバターと、環境オブジェクトが存在する。ユーザの頭部の方向は HTC Vive に、手の動きはコントローラによってトラッキングされ、仮想空間上に反映される。コントローラはトリガーもしくはボタンが入力されれば、その情報をユーザ管理プログラムへと渡す。ユーザアバターがペットアバターとの衝突を検知すると、ユーザ管理プログラムとペット管理プログラムへそれぞれ衝突情報が渡される。ユーザ管理プログラムからはコントローラへ振動命令が出され、コントローラが振動することでユーザに接触フィードバックを起こす。ペット管理プログラムからはペットアバターのアニメータに尻尾

一連の動作を以下の図に示す。システムを起動すると、ペットが自由に動き回る待機状態から開始する。ユーザがトリガーを深く引くと、ペットはユーザの位置まで移動し、ユーザの手前で停止する。この状態を触れ合い可能状態と呼ぶ。触れ合い可能状態では、ペットに触れることが可能になる。この状態で再度トリガーを深く引くと、ペットは待機状態へと戻りフィールド内を自由に歩き回るようになる。

4.1.1 仮想空間

仮想空間の構築には、Unity を使用した。

仮想空間内の環境構築には Unity のアセットストアで提供されている Nature Starter Kit 2 を使用し、森林空間を表した。より現実感を高めるため、WindZone を配置することで風を吹かせ草などの背景オブジェクトが揺れるようにした。更に NHK が配布している環境音素材である木々のざわめき、鳥のさえずりの二種類の音楽データを重ねて再生することで、森林らしさを追求した。

表 4.1: スクリプト一覧

名前	説明
PetController	ペットの動作管理
PlayerController	ユーザーの動作管理
MovePet	待機状態の動作
HandController	アバターの衝突検知
HeadLookController	ペットの視線移動
DirectionManager	ペットの視線移動における補完

4.1.2 ユーザ

ユーザの手の 3DCG モデリングには Maya を使用した。ユーザの入力を表に示す。

表 4.2: コントローラの入力一覧

入力	動作
トリガーを深く引く	待機状態と触れ合い可能状態への遷移
メニューボタンを押す	着席位置のリセット
グリップボタンを押す	シーンの切り替え

4.1.3 バーチャルペット

バーチャルペットの 3 DCG モデリングには Maya を使用した。待機、歩く、走る、尻尾を振るの 4 種類のアニメーションが可能である。ペットの動作を 4.3 に示す。

表 4.3: ペットの動作一覧

動作一覧
歩く
走る
尻尾を振る
視線先変更

4.2 ユーザ管理部

ユーザ管理部は PlayerController と HandController の 2 つのスクリプトによって構成される。PlayerController は SteamVR の ~ にアタッチされており、コントローラ入力全般の処理を行う。コントローラのトリガーが深く引かれると、口笛の効果音を鳴らし終えた後クラス変数の代入を行う。このクラス変数の値に応じてペット管理プログラムはペットの動作処理を実行する。コントローラのメニューボタンが押された場合は HMD の着席位置のリセットを、グリップボタンを押された場合はシーンを切り替える。このシーンの切り替えは後述する実験で使用する。HandController は ~ の子オブジェクトである Hand にアタッチされており、アバターの衝突情報を親オブジェクトである PlayerController に渡す。衝突情報を受け取った PlayerController は、コントローラに振動命令を出す。HTC Vive のコントローラの振動の値は最大 3999 まで設定が可能だが、本研究で想定するペットに触れるという動作において強い振動は好ましくない。人が振動を感知できる程度の弱い振動を求め値を 200 で実装する。

4.3 ペット管理部

ペット管理部は PetController、MovePet、HeadLookController、DirectionManager の 4 つのスクリプトによって構成される。PetController は主にペットの動作全般の制御を行う。PlayerController でユーザの動作に応じて代入した変数を参照し、移動、回転、視線移動、アニメーションの再生及び停止とペットの動作を制御する。MovePet は待機状態におけるペットの動作を制御する。待機状態でペットはまず目標位置を設定しそこへ移動を開始する。その後一定距離移動するごとに目標位置を再設定しその位置に移動、と目標位置の設定と移動を繰り返す。触れ合い状態に移行するとき、MovePet は停止する。再び待機状態に戻る際にはまた動作を再開する。HeadLookController はアセットストアから入手したスクリプトで、空間内の指定されたポイントに振り向くように動作する。本研究ではペットアバターの首から頭にかけてのボーンを指定し、s DirectionManager はペット管理

第5章 実験と評価

本章では、本研究における関連研究を述べる。

5.1 実験環境

本研究のバーチャルペットの開発、および実験環境を（）に示す。開発、実験共に同じコンピュータで行った。

表 5.1: 開発、実験環境

CPU	
メモリ	
Video Card	
OS	
SoftWare	

5.2 実験用バーチャルペット

本研究で開発したバーチャルペットを元に、機能を減らし先行研究のものに可能な限り近付けた実験用バーチャルペットを作成した。本研究で開発したバーチャルペットは身体性を有すると仮定して以降は身体性あり条件と呼び、実験用バーチャルペットは身体性なし条件と記述する。二つの条件を比較したものを??に示す。

表 5.2: ペット条件の比較

	身体性あり条件	身体性なし条件
デバイス	HTC Vive コントローラ イヤホン	コントローラ
接触フィードバック	振動	なし
ペットの動作	移動 視線移動 尻尾を振る	尻尾を振る
環境	森林	Unity のデフォルト空間に白い床
環境音	あり	なし

使用するデバイスはコントローラの1種類で、本研究で開発したものと同様に HTC Vive 付属の VR 専用コントローラを使用する。そのため、カメラとして HTC Vive を使う必要がある。実

験参加者には HTC Vive を膝に置き、安定させた上で右手にコントローラを持ち画面を見てふれあってもらう。ペットの動作はユーザと接触すると尻尾を振る一種類のみである。ユーザが離れると一定時間経過後に尻尾を振る動作を停止する。

5.3 実験方法

実験を行う前にバーチャルペットの身体性を測る実験であるという趣旨と操作方法を説明し、身体性なし条件→身体性あり条件の順で時間を定めず自由にふれあってもらう。その後、身体性に関するアンケートを実施した。アンケートは二つのペットを比較した上で、身体性ありペットをどのように感じたかを回答する比較部と、身体性ありペットにおける各要素に対しどれ程現実感を感じることが出来たかを回答する要素部の二部で構成される。

5.4 評価方法

5.5 実験結果

5.5.1 アンケート

第6章 結果と考察

本章では，本研究における関連研究を述べる．

6.1 結果

6.2 考察

6.3 今後の課題

謝辞

本研究に際し，多大なるご指導，ご支援をいただきました指導教員である菅原研次教授，真部雄介准教授，ならびに大変ご多忙であるにもかかわらず，数多くのご意見やご協力をいただきました菅原・真部研究室の各氏，実験にご協力いただいた各氏にも深く感謝いたします。

参考文献

- [1] H. D. I. Abarbanel, “Analysis of observed chaotic data,” Springer-Verlag, New York, 1996.
- [2] M. Adachi and M. Kotani, “Identification of Chaotic Dynamical Systems with Back-Propagation Neural Networks,” IEICE Trans. Fundamentals, Vol. E77–A, No. 1, pp. 324–334, 1994.
- [3] K. Aihara Eds., T. Ikeguchi, T. Yamada and M. Komuro, “Fundamentals of Chaotic Time Series and Its Application,” Sangyo-Tosho, 2000, in Japanese.
- [4] A. M. Albano, J. Muench, C. Schwartz, A. I. Mees and P. E. Rapp, “Singular-value decomposition and the Grassberger-Procaccia algorithm,” Phys. Rev. A Vol.38, 30174–3026, 1988.
- [5] K. T. Alligood, T. D. Sauer and J. A. Yorke, “Chaos — An Introduction to Dynamical Systems,” Springer-Verlag, 1996.
- [6] R. Andrews, J. Diederich, and A. B. Tickle, “Survey and critique of techniques for extracting rules from trained artificial neural networks,” Knowledge-Based Systems, Vol. 8, pp. 373–389, 1995.
- [7] P. J. Angeline, “Evolving Predictors for Chaotic Time Series,” S. Rogers, D. Fogel, J. Bezdek and B. Bosacchi Eds. , ”Proc. of SPIE (Vol. 3390): Application and Science of Computational Intelligence,” pp.170–180, SPIE, Bellingham, WA, 1998.
- [8] M. Arisawa and J. Watada, “A Structural Learning Algorithm for a Neural Network with Fuzzy Reasoning,” SICE Trans. , Vol.33, No.11, pp.1087–1092, 1997, in Japanese.
- [9] V. Babovic, M. Keijzer and M. Stefansson, “Optimal Embedding using Evolutionary Algorithms,” Proc. of the International Conference on Hydroinformatics, Iowa City, 2000.
- [10] J. M. Benitez, J. L. Castro, and I. Requena, “Are Artificial Neural Networks Black boxes?,” IEEE Trans. on Neural Networks, Vol. 8, No. 5, pp. 1156–1164, 1997.
- [11] G. P. Zhang, “Neural Networks for Classification: A Survey,” IEEE Trans. on Systems, Man and Cybernetics, Vol. 30, No. 4, pp. 451–462, 2000.
- [12] Z. H. Zhou, Y. Jiang and S. F. Chen, “Extracing Symbolic Rules from Trained Neural Network Ensembles,” AI Communications, Vol. 16, No. 1, pp.3–15, 2003.

付 録 A コルーチン処理による Unity-chan の発生

```
using UnityEngine;
using System.Collections;

public class hassei: MonoBehaviour
{
    public GameObject cube;

    void Start()
    {
        StartCoroutine("Sample");
    }

    IEnumerator Sample()
    {
        yield return new WaitForSeconds(1.0f);
        for (int i = 1; i < 10; i++)
        {
            yield return new WaitForSeconds(1.0f);
            if (i == 3)
            {
                float x = Random.Range(0.0f, 1.0f);
                float y = Random.Range(0.0f, 0.0f);
                float z = Random.Range(0.0f, 50.0f);
                Instantiate(cube, new Vector3(x, y, z), Quaternion.identity);
            }
        }
    }

    void OnTriggerEnter(Collider other)
    {
        Destroy(gameObject);
    }
}
```

付 録 B NavMesh に従い目的地に移動する

```
using UnityEngine;
using System.Collections;

public class hassei: MonoBehaviour
{
    public GameObject cube;

    void Start()
    {
        StartCoroutine("Sample");
    }

    IEnumerator Sample()
    {
        yield return new WaitForSeconds(1.0f);
        for (int i = 1; i < 10; i++)
        {
            yield return new WaitForSeconds(1.0f);
            if (i == 3)
            {
                float x = Random.Range(0.0f, 1.0f);
                float y = Random.Range(0.0f, 0.0f);
                float z = Random.Range(0.0f, 50.0f);
                Instantiate(cube, new Vector3(x, y, z), Quaternion.identity);
            }
        }
    }

    void OnTriggerEnter(Collider other)
    {
        Destroy(gameObject);
    }
}
```