

# 모바일 환경을 위한 지능형 물리엔진 시스템 설계 및 구현

김회창<sup>○</sup>, 신동규<sup>○</sup>, 신동일<sup>○</sup>, 김수한<sup>○○</sup>, 이명수<sup>○○</sup>  
세종대학교 컴퓨터 공학과<sup>○</sup>, 삼성전자<sup>○○</sup>  
realkato@gce.sejong.ac.kr<sup>○</sup>, {shindk, dshin}@sejong.ac.kr<sup>○</sup>,  
{ksoohan, myungsu.lee}@samsung.com<sup>○○</sup>

## Design and Implementation of Intelligent Physics System for Mobile Environment

Hoichang Kim<sup>○</sup>, Dongkyoo Shin<sup>○</sup>, Dongil Shin<sup>○</sup>, Soohan Kim<sup>○○</sup>, Myungsu Lee<sup>○○</sup>  
Dept. of Computer Science, Sejong University<sup>○</sup>,  
Samsung Electronics CO. LTD, Div. Internet Infra Planning Team<sup>○○</sup>

### 요 약

최근 모바일 게임에 있어서 중요한 이슈는 게임상에서 존재하는 물체들이 사실감과 생동감을 유지하면서도 프로그램의 속도 감소 문제를 발생시키지 않는 방안을 찾는 것이다. 본 논문이 제시하는 내용은 모바일 환경에서의 지능형 물리엔진 아키텍처에 대한 것으로 물리엔진으로 구현 한 게임 안의 환경을 인지하고 수집한 수치를 이용, 학습하여 사용자가 컨트롤하는 게임 내의 물체가 최적화 된 움직임을 보일 수 있도록 하는 데 목표를 두고 있다. 이를 위해 모바일 환경에 적합하도록 구현된 물리엔진으로 자동차 주행 시스템 환경 내에서 속성을 추출, 인공지능 모듈에 입력하여 연산량이 비교적 적은 베이지안 네트워크 알고리즘을 통해 분석하며 이를 평가한다.

### 1. 서 론

최근 다양한 하드웨어 및 소프트웨어 기술의 발달로 인해 컴퓨터 상에서 사용자에게 뛰어난 현실을 제공하는 시뮬레이션 프로그램이나 가상 환경이 많이 선보여지고 있으며, 특히 과거와 달리 흔히 물리 엔진과 인공지능 엔진 및 그래픽 엔진 등을 결합한 프로그램을 이용하여 제작되고 있다. 이러한 발전의 추세는 계속해서 가속화 되고 있으며 특히 차세대 킬러 애플리케이션이 될 것이라 주목을 받은 모바일 3D 게임 산업에서도 급격히 나타나고 있다.

3D 게임 콘텐츠의 제작에는 3D 입체 게임 엔진이 필요하다. 게임 콘텐츠를 사실적이고 구체적이며 현실적으로 표현하기 위해서는 물리 엔진의 지원이 필요하다. PC기반 3D 게임과는 달리 모바일 3D 게임은 단말기의 배터리 용량, 디스플레이 크기의 한계, 게임 콘텐츠 용량에서 제약을 받는다.

그리고 물리엔진만이 적용된 게임은 캐릭터의

사실적인 움직임만을 표현할 뿐 캐릭터에게 생명력을 심어주지는 못한다. 본 논문에서는 모바일 물리엔진이 적용된 게임에서 자동차가 주행할 때 발생하는 물리적 요소를 인지하고 수집한 수치를 이용, 학습하여 사용자가 컨트롤하는 가상 환경 내의 물체가 최적화 된 움직임을 보일 수 있도록 하는 데 목표를 두고 있다. 기존의 상황인지에 사용된 알고리즘은 SVM, Neural Network등 고성능이나 CPU의 연산을 많이 필요로 하는 알고리즘 들이 사용되었다. 하지만 본 연구는 비교적 CPU연산을 적게 필요로 하면서 만족할 만한 성능을 보여주는 베이지안 네트워크 알고리즘을 사용하였다.

### 2. 배경연구

#### 2.1 물리엔진

물리엔진(Physics Engine)이란 게임 내의 오브젝트를 대상으로, 질량·속도·마찰·공기 저항 등의 수치를 이용하여 뉴턴 역학 모델을 시뮬레이트하는 프로그램을 통칭하는 것이다[1]. 현실적인 물리 연산을 위해 하복(Havok) 물리 엔진이 사용되었다더라는 등의 얘기를 들으며, 한번쯤은 물리 엔진이 과연 게임 내에서

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2009년도 문화콘텐츠산업기술지원사업의 연구결과로 수행되었음

어떤 기능을 하는 프로그램인지 궁금하게 여긴 사람도 있을지 모른다. '물리 엔진(physics engine)'이라는 명칭을 바탕으로 생각해보면, 게임에 있어 물리 엔진이란 곧 '자연계의 물리 현상을 게임 내에서 흉내 내는(simulate) 프로그램'일 것이다. 그렇다면, 물리 엔진을 이용하여 어떤 종류의 물리 현상을 시뮬레이트하는 것일까? CPU의 연산 능력에는 한계가 있으므로, 모든 물리 현상을 게임 내에서 완벽하게 구현하거나 범지구적인 거대한 자연계의 현상까지 시뮬레이트하는 것은 물론 불가능하다. 앞서 정의한대로, 물리 엔진은 다양한 변수에 기초하여 게임 내의 오브젝트의 운동에 뉴턴 역학(Newtonian mechanics)을 적용하여 시뮬레이트하는 프로그램이다. 즉, 현재의 물리 엔진은 실시간으로 '게임 내'라는 국지적인 범위 안에서 뉴턴 역학 모델을 시뮬레이트하는 것이 주목적이다. 물리학의 발전에 따라 뉴턴 역학과 상대성 역학은 '고전 역학'이 되었고, 현재의 물리학에서는 양자역학(quantum mechanics)에 따라 복잡한 물리 현상을 설명하고 있지만, 우리 눈에 보이고 우리 주변의 생활 속에 있는 사물들의 운동은 일반적인 뉴턴 역학으로도 충분히 설명할 수 있다. 물리 엔진은 이러한 뉴턴 역학을 실시간으로 계산하여, 게임 내에서 다양한 오브젝트에 가해지는 힘과 운동 상태와의 관계를 계산하는 프로그램인 것이다. 하나의 게임 내에는 게임 논리, 렌더링 엔진(rendering engine), 네트워크, 물리 엔진, 사용자 입력 처리 등의 수많은 프로그램이 결합되어 있다. 물리 엔진은 이중 게임 내부에 논리적으로 뉴턴 역학 공간을 구성하고, 이 공간 안에서 일어나는 물리 현상을 시뮬레이트하는 일을 하게 된다[1]

### 2.2 충돌반응

충돌반응은 두 개 이상의 물체들이 충돌했을 때의 반응을 나타낸다. 예를 들면, 두 개의 당구공이 충돌하여 다른 방향으로 이동하는 반응을 말할 수 있다. 이러한 충돌반응은 당구시뮬레이션 게임과 같은 곳에서 주로 적용되며 또한 3차원입체 액션 게임에서 총으로 강통을 향해 총알을 발사했을 경우 강통의 움직임과 같은 충돌효과를 나타낼 수 있다. 두 개 당구공의 질량과 충돌전의 속력벡터들과 당구공 재질의 복원 계수가 주어지면, 충돌 후의 움직임은 운동역학의 에너지 보존 방정식과 복원계수의 수식을 통해 두 개의 당구공의 충돌직후의 속력벡터들을 결정할 수 있다.

### 2.3 모바일 기기를 위한 자동차용 물리엔진 설계

모바일 장치의 특성상 최소한의 수학 연산을 수행하며 레이싱 게임의 요소를 만족시키는 모델이 필요하다. 본 논문은 Ted Zuvich [2]의 연구를 기본으로 하고 있다. Ted Zuvich는 레이싱 게임을 위한 단순화된 시뮬레이션 모델을 제시하였다 [3]. 따라서

상대적으로 적은 계산량을 요구하여 계산 능력이 한정되어 있는 모바일 기기에 적합하다. 다음은 레이싱 게임을 위해 단순화된 시뮬레이션 모델의 몇 가지 예이다.

그림 1은 자동차에 미치는 중요한 힘들을 나타내고 있다. 앞으로 나아갈 때의 힘, 측면에 미치는 힘, 지면에 미치는 힘이 바로 그것이다. 이들 힘을 표시하기 위하여 SAE(Society of Automotive Engineers) 표준 좌표 시스템을 사용하는데 이 시스템은 자동차의 무게중심 CG(Center of Gravity)를 기준으로 하며, 오른손 좌표계가 이용된다.

자동차가 정지된 상태이고, 평지에서서의 힘의 분산은 자동차의 움직이는 상태에서 미치는 힘의 분산을 나타내기 위한 기본이 된다. 그림 2에서처럼 수평으로 정지된 상태에 미치는 힘의 분산을 살펴보자. 그림에서 나오는 바퀴 부분 A, B에 힘의 분산이 이루어진다.

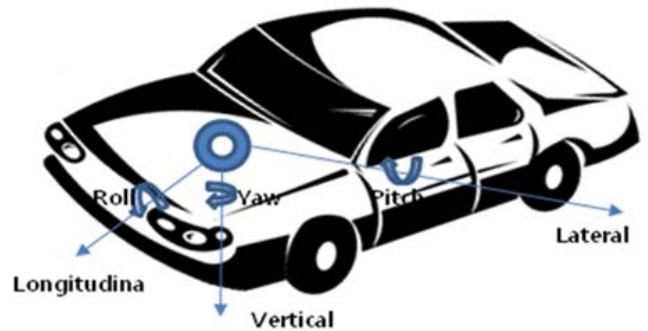


그림 1 자동차 좌표 시스템

$$W_{fs} = W \frac{c}{L}$$

$$W_{rs} = W \frac{b}{L}$$

$W_{fs}$  = Front Wheel Axis에 걸리는 기본 힘  
 $W_{rs}$  = Rear Wheel Axis에 걸리는 기본 힘  
 $b$  = CG로부터 Front Wheel Axis에 미치는 거리  
 $c$  = CG로부터 Rear Wheel Axis에 미치는 거리  
 $L$  = Front Wheel와 Rear Wheel 의 자축거리

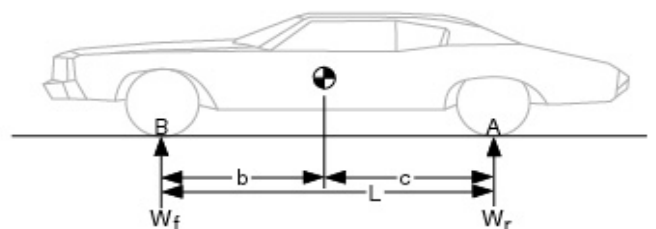


그림 2 지평 정지상태에서의 힘의 분산

자동차가 가속/감속시, 자동차에는 앞바퀴, 뒷바퀴, 회전시 측면에 힘이 분산되게 된다. 이들 힘의 분산은 가속도에 의해 영향을 받게 되며, 이 식은 다음과 같다.

$$W_f = \frac{c}{L}W - \frac{hW}{Lg}A_x$$

$$W_r = \frac{b}{L}W + \frac{hW}{Lg}A_x$$

Wf = Front axle load

Wr = Rear axle load

### 2.4 베이저안 네트워크

베이저안 네트워크(Bayesian Network)는 특정분야의 영역지식(Domain Knowledge)을 확률적으로 표현하는 대표적인 수단으로, 변수들 간의 확률적 의존관계(Probabilistic Dependency)를 나타내는 그래프와 각 변수별 조건부확률로 구성된다[4]. 따라서 하나의 베이저안 네트워크는 각 노드마다 하나의 조건부 확률표(Conditional Probability Table)를 갖는 하나의 비순환 유향 그래프(Directed Acyclic Graph: DAG)로 정의할 수 있다. 이때 각각의 노드는 이벤트의 발생을 의미하고 노드를 이어주는 DAG는 이벤트간의 관계(Relationship)를 의미한다.

$$A \rightarrow B: A \text{ causes } B \quad (1)$$

일반적으로, 하나의 베이저안 네트워크는 다른 노드들에 배정된 값들을 기초로 특정 노드가 가질 값에 대한 조건부 확률을 계산하는데 이용할 수 있다. 따라서 하나의 베이저안 네트워크는 한 개체의 다른 속성들의 값이 주어졌을 때 분류 클래스 노드(Classification Node)의 사후 확률 분포(Posterior Probability Distribution)를 구해 줌으로써 개체들에 대한 하나의 자동 분류기(Classifier)로 이용될 수 있다[5]. 즉 하나의 데이터 집합으로부터 베이저안 네트워크를 학습할 때 베이저안 네트워크의 각 노드는 데이터 집합의 각 속성을, 각 아크는 속성들 간의 의존성을 표현하게 되며, 이렇게 학습된 베이저안 네트워크를 기초로 분류 클래스를 확률적으로 예측할 수 있다. 예를 들어 변수 X1~X10이 있고 각 변수들이 TRUE(1)와 FALSE(0)의 값을 갖는다고 할 때 베이저안 네트워크는 각 변수들에 대한 의존성을 그래프로 표현할 뿐 아니라 각 변수별로 표 1과 같은 조건부 확률도 함께 표현할 수 있다. 표 1은 10개의 변수 중, 변수 X8이 가지고 있는 X2와 X6에 대한 조건부 확률표(Conditional Probability Table)를 나타내고 있다. 그림 1에서 보듯이 변수 X8은 변수 X2와 X6에 대해서만 종속성을 가지며 따라서 조건부 확률표의 각 행(Row)은 이 두 변수 X2와 X6에 배정 가능한 값들을 나타내며 각 열(Column)은 변수 X8이 가질 수 있는 값들을 나타낸다. 결국 표 상의 각 셀은 두 변수 X2와 X6의 값들에 의해 X7이 가질 수 있는 값들의 확률을 표현

하고 있는 것이다.

$$P(X8 | X2, X6) = 0.185$$

$$P(\sim X8 | X2, X6) = 0.815 \quad (2)$$

와 같은 의미를 지닌다.

표 1. Probability Distribution Table For X8

X2	X6	0	1
0	0	0.135	0.865
0	1	0.285	0.625
1	0	0.579	0.421
1	1	0.815	0.185

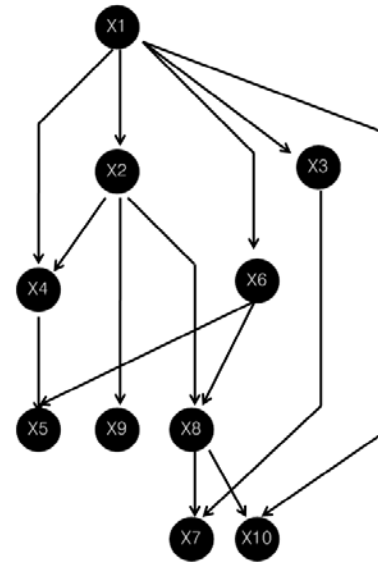


그림 3 베이저안 네트워크

베이저안 네트워크는 조건부 확률 계산에 식(1)과 같은 베이저 정리(Bayesian Theorem)를 이용한다. 베이저 정리는 관측된 데이터 D로부터 가설 h가 옳을 확률 P(h|D)와 가설 h의 사전확률(prior probability) P(h)을 기초로 계산할 수 있는 방법을 제시한다[6].

$$P(h | D) = \frac{P(D | h)P(h)}{P(D)} \quad (3)$$

### 3. 지능형 물리엔진을 사용한 주행 시스템

모바일 3D 레이싱 게임을 위한 이 실험에서는 자동차와 같은 물체모델링과 레이싱 트랙 및 장애물 같은 지형모델링을 구현, 각자의 성질에 맞는 속성을 제공한다. 본 논문이 제시하는 것은 베이저안

알고리즘을 적용, 시뮬레이션된 측정 정보를 추출하여 가장 최적화된 이동정보를 사용자에게 제공하여 물리 엔진 기반에서의 게임 속 차량이 물리 수치를 인식함에 따라 운전자에게 쾌적한 주행을 돕는 것에 그 목적을 두고 있다. 아래 표 2는 이 실험에서 사용된 의사코드이다.

표 2 구현 코드 예제(pseudo code)

Pseudo Code	Explanation of Code
double start_line geomSetposition(start_line, x, y, z) drawbox start_line if(collision start_line) start timecount	시작 포인트의 선언 및 구현 (충돌 시에 발생하는 이벤트 구현 포함)
double CarSensor[3]; geomSetposition CarSensor(front, left, right)	전면, 양 옆을 체크하기 위한 센서 선언 및 구현 (화면에는 표시되지 않음)
if(collision CarSensor(front, left, right) Select(Track & obstacle angle) Decision AreaType	센서로 앞과 양 옆의 공간을 체크 충돌이 일어났을 시 충돌 방향과 장애물의 회전각을 이용 구역 유형을 판단.
GeomGetposition&rotation(CarBody) Print screen(position, rotation, CarSpeed)	자동차의 현 위치와 회전각 및 속도를 측정 화면에 표시함

### 3.1 차량 주행 시뮬레이션 실험

이 장에서는 모바일 환경에서의 지능형 물리엔진에 기반하여 게임 속 차량을 사용해 실험하는 방법에 대하여 논한다.

지능형 물리엔진을 사용한 주행 시뮬레이션을 적용하기 위해 기본적으로 고려해야 하는 사항은 차량의 이동 위치 및 속도, 각도의 입력 값과 코스에 따른 경과 시간을 알아내야 하는 것이다

그림 4는 다음에 설명하는 실험 과정에 따라 주행 시스템 내의 측정된 정보가 학습 과정을 거쳐 사용자에게 전달되는 것을 표현하고 있다.

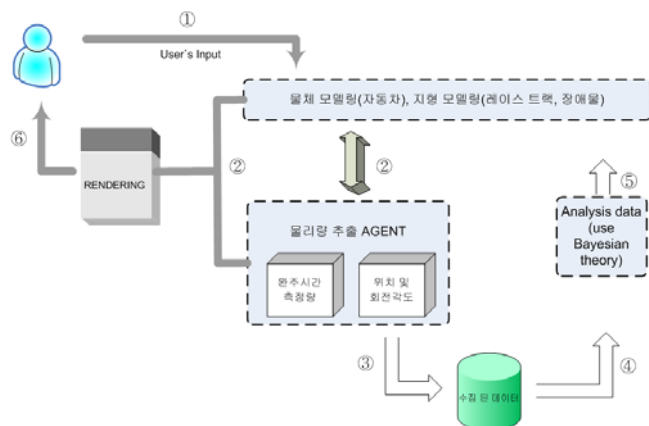


그림 4 모바일 환경 주행 시스템 실험 과정

먼저 응용 프로그램을 통하여 사용자의 힘을 입력 받는다. 이 때 게임에 설정된 가상 환경에 따라 장애물 등의 물체와 지형의 물리량을 측정하여 현재 자동차의 위치 및 회전각 등의 다양한 값들이 생성된다. 생성된 값을 토대로 물체와 지형모델은 실시간으로 변화된 위치와 힘을 렌더링 엔진에 제공하는 한편, 물리량을 추출 하는 모듈에서 요구하는 원주시간 및 위치와 회전각도를 전달한다. 물리량 추출 AGENT에서 정보를 수집하고 이 수집된 개인 사용자의 구간별 경과시간과 각 위치에 따른 회전각도 정보를 베이지안 알고리즘에 적용시켜 현재 사용자의 위치에서 가장 최적화된 회전각도를 구한다.

### 3.2 모바일 환경을 위한 지능형 물리엔진 구현

그림 5는 지능형 물리엔진 모듈 구현도를 나타낸다. 그림 5와 같이 자동차가 시작점을 지나 주행을 하는 동안 자동차의 포지션은 XY축의 절대 값으로 표현되며 주행하는 자동차는 주변을 탐색하여 충돌체크만 할 수 있도록 설정된다. 여기서 중요한 점은 모든 지형의 위치 및 회전각은 사전 정보로 모듈에 입력된 상태로 시작된다는 것으로 자동차의 충돌을 통해서 지형과의 거리가 감지되고 충돌이 일어날 경우에만 지형과 자동차의 회전각을 체크, 비교함으로써 모듈의 수집과 분석을 가능하게 한다는 것이다. 이는 모바일 환경에서의 계산 과부하를 줄여 나가는 효과를 가진다.

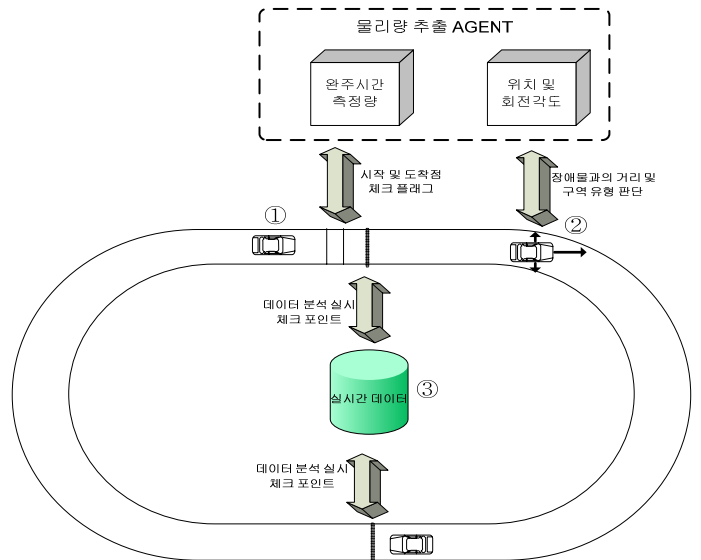


그림 5 지능형 물리엔진 모듈 구현도

이 실험에서 제공되는 수치 중에서 회전 각을 선택하는 기준은 장애물과 차량의 회전 각이 수평이 될 때 가장 안전한 상태로 인식하며 장애물과의 각도를 기준 점으로 삼아서 충돌 에너지, 즉 피해량을 아래와 같이 계산하여 조건부 확률 변수를 정했다.

F: 충돌량  
 W: 차의 중량  
 V: 차의 속도  
 $\Theta$ : 장애물과의 회전각도 차이  
 $F = W*V*cos(90-\Theta) = W*V*sin \Theta$

위 식은 각도에 따른 충돌에너지를 계산하기 위해 차량의 무게와 속도는 동일한 상태라는 통제하에 상태 유지 했다.

이 실험에서는 상황 판단을 위하여 내부적으로 베이지안 이론을 적용하였으며 각 위치별로 가장 확률상 충격을 적게 줄 수 있는 각도가 계산되어 회전 시 속도의 계산을 통해 주행에서 요구되는 최적의 속도와 회전각의 값이 화면에 표시되게 하였다. 이로써 사용자는 더 빠른 시간 내에 자동차 코스를 완주할 수 있으며 자신에게 가장 적합한 주행법을 파악할 수 있게 된다.

**4. 모바일 환경 지능형 물리엔진 실험결과**

본 논문이 제시하는 실험은 물리엔진이 적용된 모바일 게임환경 속에서 차량을 주행할 때 발생하는 상황에 따른 물리량을 베이지안 네트워크를 이용하여 예측하고 이를 통해 조작 감을 개선 해 낼 수 있는지에 대한 실험이다.

NBN(네이브 베이지안 네트워크)을 이용한 이 실험에서는 사고 위험성에 관한 분석을 위해 속도 및 회전 각도 등의 조건부 관계 변수를 이용, 사고 가능성을 계산하여 사고 가능성이 높을 경우에는 화면에 경고 표시와 함께 장애물과 현재 차량의 회전 각도를 이용해 계산 된 추천 각도를 표시하는데 사용하였다.

실험은 NaïveBayes 알고리즘을 사용하여 특징 중 코스 선택에 관한 속성을 분류 기준으로 삼은 뒤, 10개의 속성을 가지는 데이터의 양을 150개로 정하고 베이지안 네트워크로 구성한 후에 Cross Validation 방식을 통해 검증 하였다.

표 3. Training Set 크기에 따른 결과

	K 지 수	Traini ng Set 개수	Correct 개수	성공 확 률	평균
Naïve Bayesian Network	5	120	116	96.66%	89.37%
	3	100	111	92.50%	
	2	75	104	86.66%	

\*k 지수는 k-fold cross validation의 k 지수를 뜻함

네이브 베이지안 분류기의 실험 결과는 위의 표 3과 같다. 전체 데이터에서 Training Set을 정해 모델링 하고 나머지를 Validation Set으로 사용하였는데 검증 방법은 k-fold Cross-Validation 방법을 택하였다.

일반적으로 데이터를 그룹화하는 k-fold의 숫자가 높을수록 좋은 검증 결과를 보이는데, 이 실험에서도 역시 k=5로 설정한 5개의 그룹으로 나누어 30개의 Validation Set으로 120개의 Training Set에 관한 검증을 했을 시에 가장 높은 성공 확률을 보였다.

그림 5는 모바일 환경에서 물리량을 추출해 화면에 표시하는 시뮬레이션 환경을 구축 한 것을 보여주는 화면이다.

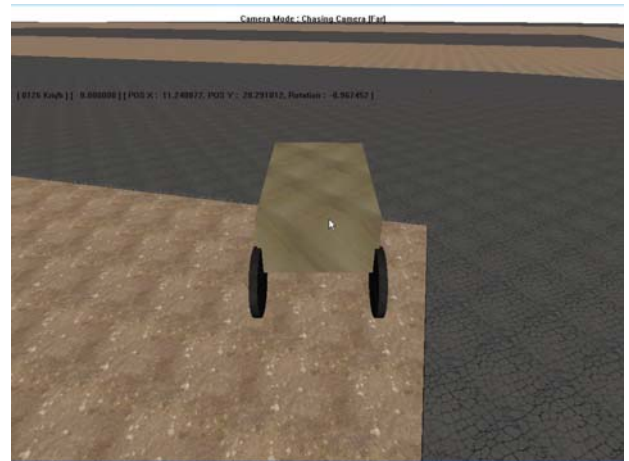


그림 5 모바일 지능형 물리엔진 구현 화면

그림 6은 본 연구팀에서 구현한 물리엔진 시스템이 적용된 또 다른 모바일 3D 게임 데모 화면이다. 이는 모든 부분에 있어 최적화된 물리적 현상이 적용 되지는 않았으나 모바일 환경에 물리엔진을 적용한 좋은 예로 볼 수 있다.

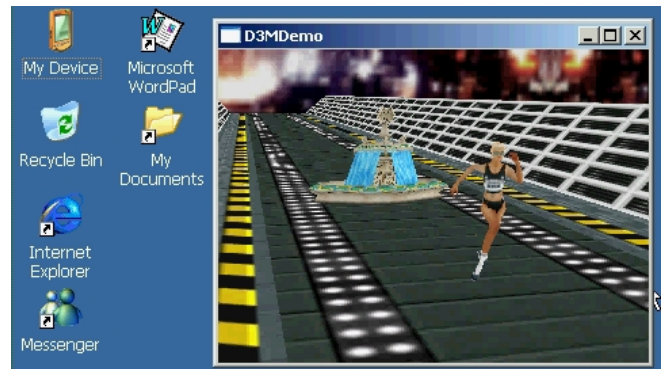


그림 6 3D 모바일 환경 물리엔진 데모 화면

**5. 결론**

본 논문에서는 유저가 게임 환경에 쉽게 익숙해질 수 있도록 하기 위해 게임 내에서 최적화 된 학습정보를 도모하는 방법인 지능형 물리엔진 주행시스템을 보였다.

지능형 주행 시스템을 만드는데 있어 주요 사항 중에 하나는 주요 특징들을 데이터베이스화 하는 것이나 본 연구의 모바일 3D 레이싱 시뮬레이션에서는 CPU 연산의 부담이 적고 빠른 처리를 필요로 하는 모바일

3D게임의 특성을 고려하여 데이터베이스 시스템을 사용하지 않았다. 모든 지형의 위치 및 회전각을 사전 정보로 모듈에 입력하여 자동차의 충돌을 통해서 지형과의 거리가 감지되고 충돌이 일어날 경우에만 지형과 자동차의 회전각을 체크, 비교함으로써 모듈의 분석을 가능하게 설계하였다. 이는 모바일 환경에서의 계산 과부하를 줄이기 위함이며 본 연구팀이 위 실험을 PC 환경에서 수행했던 경험을 바탕으로 모바일 환경에 맞도록 수정, 보완한 과정에 기인한다.

현재 많은 게임에서의 물리엔진 적용이 이루어지고 있으므로, 향후에는 이러한 물리엔진이 적용된 게임에서의 물리적 상황인식이 게임에서는 중요한 요소로 자리 잡을 것이다. 본 논문에서의 연구는 자동차 게임에 한정되어 있다. 앞으로 다른 장르의 게임에 이러한 지능형 물리엔진의 적용이 이루어져야 할 것이며 또한 전체 게임의 성능이 증가될 수 있도록 본 연구에서 사용된 알고리즘 이외의 인식 알고리즘이 적용 되어야 한다.

## 5 참고 문헌

- [1] D. H. Eberly, 유채곤 / 차미리 역, "Game physics (게임 물리 바이블)," 사이텍미디어, Nov. 2004, pp. 223-444.
- [2] T. Zuvich, "Vehicle Dynamics for Racing Games," Game Developers Conference Proc., 2000.
- [3] 이동춘, "실시간 자동차 동작을 표현하기 위한 단순화된 시뮬레이션 모델", 한국컴퓨터게임학회 논문지, 2004
- [4] Jensen, F. V., An Introduction to Bayesian Networks, New York : Springer-Verlag, 1996
- [5] Kevin Patrick Merphy, "A Brief Introduction to Graphical Models and Bayesian Networks," Technical Report, Department of Computer Science, UC Berkley 2001
- [6] Jiawei Han and Micheline Kamber, Data Mining : Concept and Techniques, Morgan Kaufmanm, 2001