볼링게임을 위한 2차원 충돌기법의 설계 및 구현

조용곤, 권순각, 김태석

동의대학교 컴퓨터소프트웨어공학과

An Implementation of 2-Dimension Collision Method for Bowling Game

Yong-gon Cho, Soon-kak Kwon, Tai-suk Kim

Department of Computer Software Engineering, Dongeui University

E-mail : janus0105@nate.com, {skkwon, tskim}@deu.ac.kr

요 약

볼링게임에서 충돌기법은 중요한 요소이다. 충돌기법의 구현을 위해서, 본 논문에서는 두 개의 구를 틀 안에서 움직이면서 틀의 벽에 부딪혔을 때 또는 구와 구끼리 충돌이 일어났을 때의 상황을 고려 한다. 구 와 구끼리의 거리차를 계산하여 충돌시점을 알리며 각각의 구의 성질(무게, 속도, 방향 등)에 따라 변하는 모습을 구현한다. 실질적인 응용을 위하여 볼링 핀의 클래스를 만들고 공과 핀끼리의 충돌, 핀과 핀끼리의 충돌을 해석하는 볼링게임을 직접 구현하였다.

키워드

2차원 충돌기법, 볼링게임

Ⅰ. 서 론

충돌기법이란 어느 물체가 충돌하였을 때, 발생하는 속도변화와 위치변화를 계산하는 기법이다. 충돌기법을 위해서는 운동량에 대한 기본 지식이 필요하며 충돌의 순간을 위하여 피타고라스의 정리가 필요하다. 볼링게임을 단순하게 말하자면 공을 던지고 그 공으로 핀을 많이 쓰러뜨리는 게임이다. 이 말은 '볼링은 충돌이다.' 라고 말해도 될만큼 가장 중요한 요소다.

충돌 중에서도 가장 중요한 것은 볼과 핀끼리의 충돌, 핀과 핀끼리의 충돌이다. 그 각각의 상황을 파악하여 속성값들이 변하게 된다.

공의 무게, 속력, 방향, 위치, 회전의 각 속성 모두가 충돌에서 큰 변화를 가져오게 한다. 공이 무거울수록 속력이 빠를수록 변화는 커지고 가벼울수록 속력이 느릴수록 변화는 작다.

본 논문에서는 실질적인 충돌기법을 구현하는 방법, 볼링게임에 직접 적용하는 방법을 제시한다.

Ⅱ. 충돌기법 동작원리

먼저 충돌후의 속력을 나타내는 식은 다음과 같다[1].

v'(a) = (m(a) - m(b) * e) / (m(a) + m(b)) * v(a) + (m(b) + m(b) * e) / (m(a) + m(b)) * v(b)

v'(b) = (m(a) + m(a) * e) / (m(a) + m(b)) * v(a) + (m(b) - m(a) * e) / (m(a) + m(b)) * v(b)

우선 밑그림과 같이 평면상을 vb로 운동하던 B의 물체에 va로 운동하는 A가 점선(A의 중심과 B의 중심을 이은 점선)상에서 충돌한다.

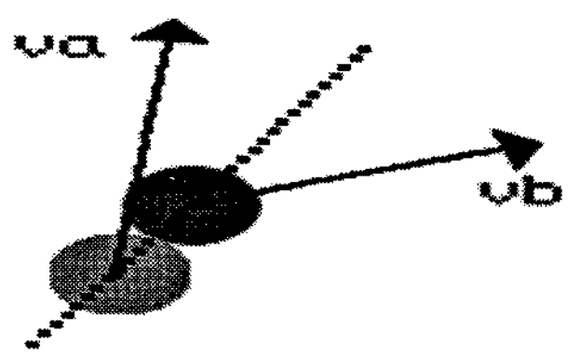

그림 1. A와 B의 충돌

이 운동을 보다 세밀하게 살펴보기 위해 공 A와 B의 속력을 각각 분해하여 vax, vay와 vbx, vby로 나눌 수 있다.

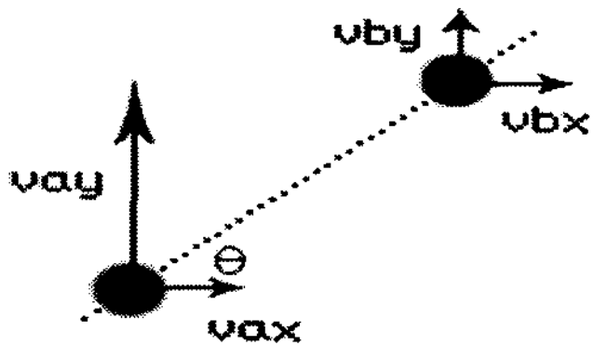

그림 2. A와 B의 속력분해

공 A와 B가 충돌하는 선상을 새로운 x축(x'축)으로 놓고 vax와 vay의 x'성분을 합해준다. 그러니까 x'선상의 속도에 의한 또 다른 1차원 충돌이다. 따라서 공 A의 x'선상의 속도는 vax*cosθ +vay*sinθ가 됩니다. 같은 방법에 의해 공 B의 x' 선상의 속도는 vbx*cosθ+vby*sinθ가 된다.

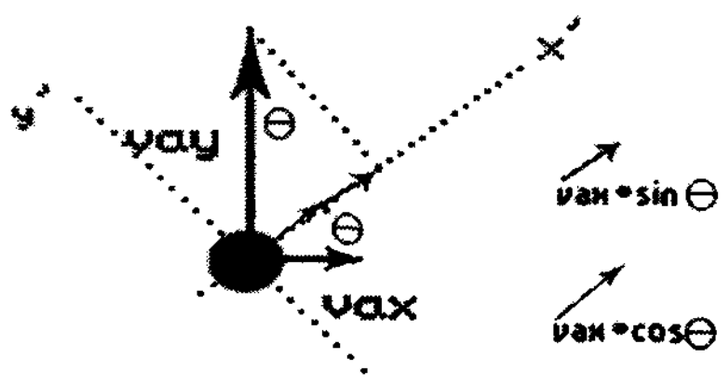

그림 5. 충돌선상의 모습

그럼 충돌후 A의 x'선상의 속도를 vaxp이라고 한다면 앞의 충돌후의 속력변화 공식에 의해

vaxp = (ma-e*mb)/(ma+mb)*

(vax*cosθ+vay*sinθ) +(mb+e*mb)/(ma+mb)*

(vbx*cosθ+vby*sinθ) 되고 또한 충돌 후 B의 x'선상의 속도를 vbxp라 한다면

vbxp = (ma+e*ma)/(ma+mb)*

(vax*cosθ+vay*sinθ) +(mb-e*ma)/(ma+mb)*

(vbx*cosθ+vby*sinθ) 가 됩니다. 이것은 x'축 선상에서의 운동량 보존과 반발 계수를 적용한다.

x'축과 마찬가지로 y'축 역시 운동량이 보존된다. 다만 충돌할 때 받는 두 공의 중심 간의 상호작용으로 공 A가 받는 힘과 공 B가 받는 힘은 서로 작용 반작용의 관계로서 x'선상에 있게 된다. 충돌 전후에 속도가 변하는 이유는 바로 이 힘 때문이다.

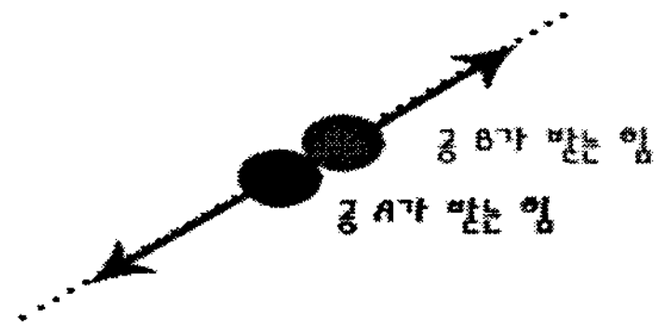

그림 6. 충돌 후의 A와 B가 받는 힘

결국 y'방향의 속도에는 아무런 변화가 없으므로 vayp는 **vayp** = vay*cosθ - vax*sinθ 가 되고 vbyp는 **vbyp** = vby*cosθ - vbx*sinθ 가 된다.

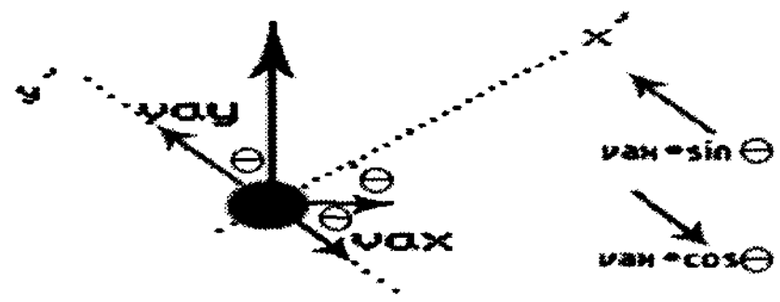

그림 7. y축으로의 속도변화

마지막으로 운동량 보존법칙에 의해 구해진 vaxp, vayp, vbxp, vbyp는 x'와 y'선상의 속도이다. 그러므로 x와 y축으로 환산 해 주어야 한다.

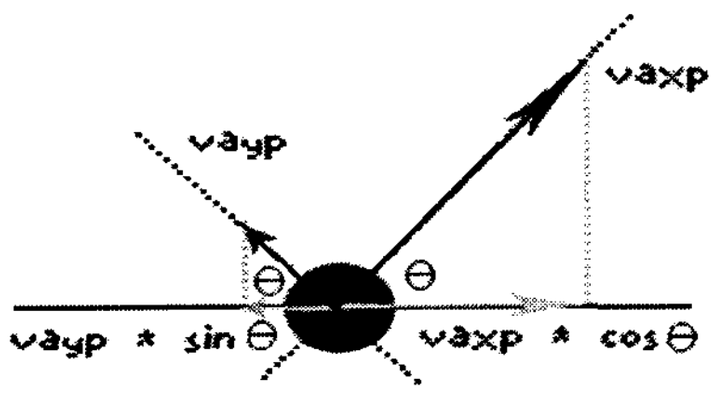

그림 8. x축과 y축으로의 환산

따라서 충돌 후 x축 방향의 속도를 vax라 하면 **vax** = vaxp*cosθ- vayp*sinθ 가 되고 y축 방향의 속도를 vay라 하면 **vay** = vaxp*sinθ+vayp*cosθ 가 된다. 같은 방법으로 충돌 후의 B의 속도는 x축과 y축 각각

vbx = vbxp*cosθ- vbyp*sinθ 와

vby = vbxp*sinθ+vbyp*cosθ 가 된다.

III. 충돌기법을 이용한 볼링게임 구현

III-1 충돌기법 모듈 구현

충돌 여부를 확인하기 위해서 그림 7과 같은 피타고라스의 정리를 이용하여 충돌을 확인한다.

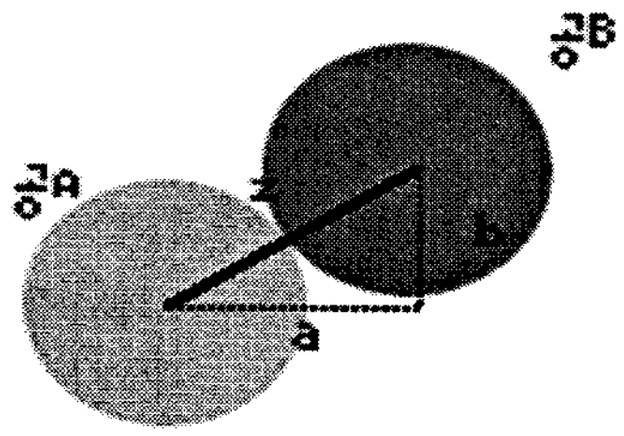

그림 9. 피타고라스의 정리를
이용한 충돌확인

공A와 공B의 반지름 길이를 정해놓거나 임의의 길이로 설정할 수 있다. 공A의 반지름의 길이와 공B의 반지름의 길이의 합이 Z의 길이와 같아진다면 둘은 충돌했다고 본다.

즉 $\sqrt{a^2+b^2}=Z$ 가 된다.

이러한 식으로서 공 2개의 충돌의 충돌 여부를 판단할 수 있다.

```
e = 1;
a = m_ball.ball.p_position.x -
    m_pin.pin.p_position.x;
b = m_ball.ball.p_position.y -
    m_pin.pin.p_position.y;
z = sqrt(a*a+b*b);
costheta = a/z;
sintheta = b/z;

//충돌 확인 부분
if (z <= (A_size + B_size))
{
```

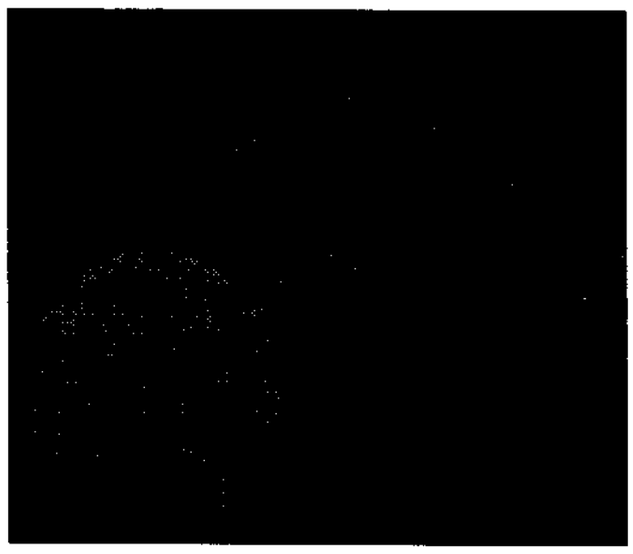

그림 10. 구현된 충돌 전 상황

그림 11. 구현된 충돌 후 상황

III-2. 볼링게임 구현

볼링게임을 통하여 충돌기법을 구현하였다.

```
typedef struct _Object_bowl
{
        float acc;
        float size;
        float mass;
        double static_y_position;
        MAKE_POINT velocity;
        MAKE_POINT position;
        MAKE_POINT first_position;
        MAKE_POINT    velocity_after;

}BALL;
```

핀은 볼과 같은 속성으로 배열로 10개의 핀을 설정 하였다. 볼링게임에서의 필요한 충돌 부분은 다음과 같다.

- 공과 핀의 충돌
- 핀과 핀의 충돌

충돌여부를 확인하기 위하여 다음과 같이 볼과 핀, 핀과 핀끼리 계속적으로 **_Collision()**함수를 불러 충돌여부를 검사한다.

```
for(int i=0;i<10;i++)
    _Collision(one,two.pin[i]);
 //볼과 핀과의 관계//
for(int j=i+1;j<10;j++)
    _Collision(two.pin[i],two.pin[j]);
 //핀과 핀과의 관계//
```

_Collision()함수는 충돌구현 함수로서 충돌에 대한 내용을 담고 있다.

볼과 각의 핀을 _Collision()함수로 보내어 충돌여부확인하였으며, 핀과 핀끼리 역시_Collision() 함수로 보내어 충돌여부를 확인하였다.

그림 10은 본 논문에서 구현한 인터페이스화면으로서, 공의 무게, 속력, 방향, 위치, 회전 등을 설정할 수 있게 하였다[2].

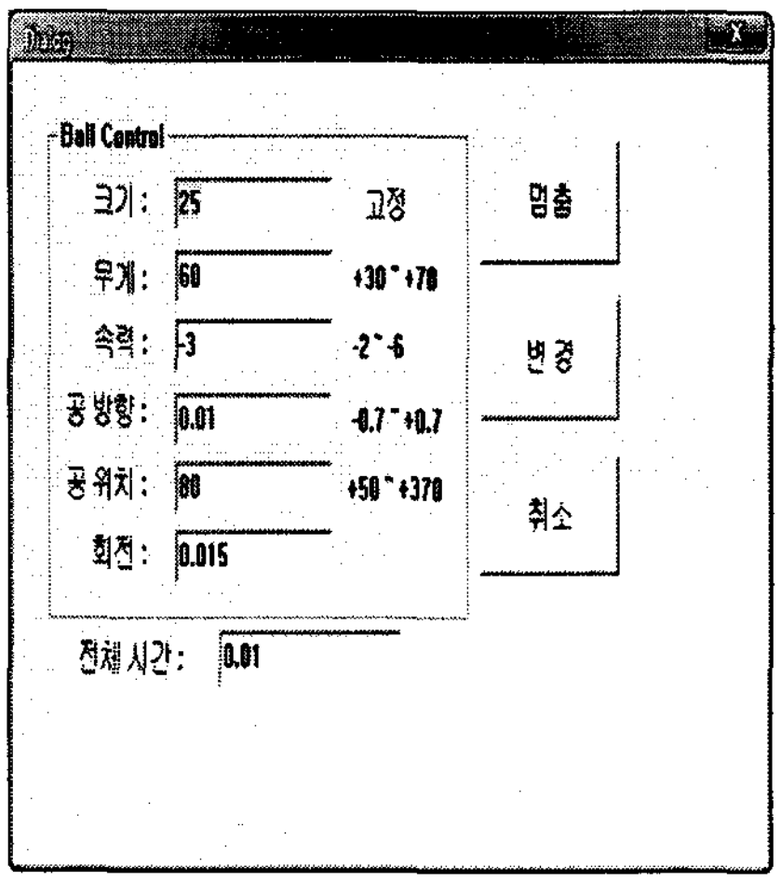

그림 12. 공의 속성 컨트롤다이얼로그

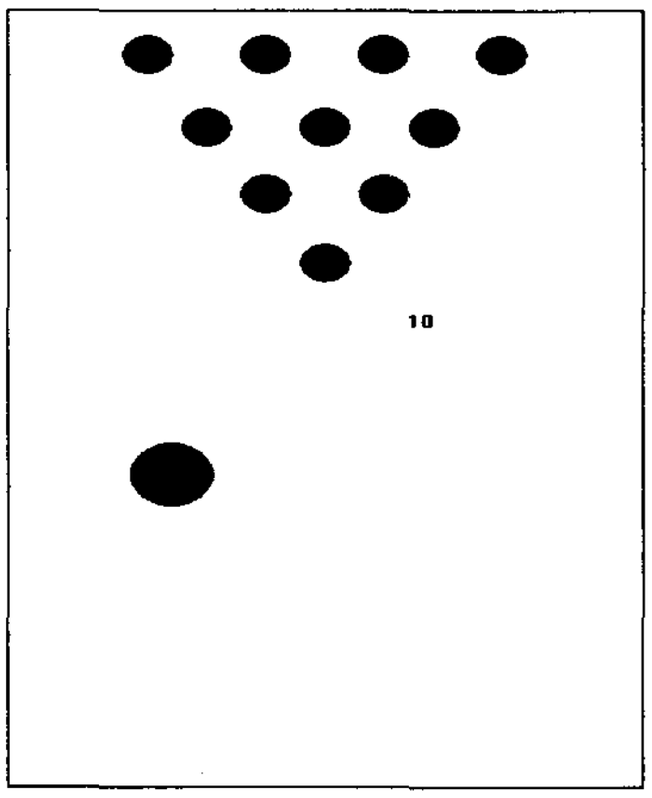

그림 13. 구현된 볼링게임

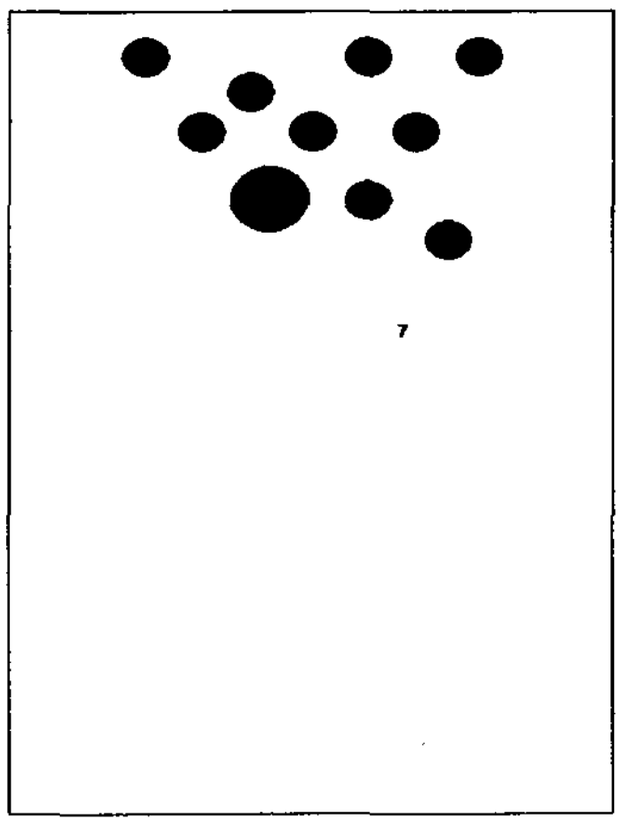

그림 14. 볼과 핀의 충돌 모습

그림 11, 12는 볼이 움직이면서 핀을 치는 모습을 나타낸다. 숫자는 공과 핀끼리 또는 핀과 핀끼리 충돌한 핀의 개수를 나타낸다.

Ⅳ. 결 론

본 논문은 충돌 기법을 이용한 게임을 구현하기 위하여, 충돌 모듈을 설계하였고, 볼링게임에 직접 적용하였다.

공의 무게를 변화시켰을 때, 속력을 변화시켰을 때 등의 변화를 주면서 그때그때 상황이 어떻게 다른지를 알 수 있도록 구현하였다.

참고문헌

[1] http://cafe.naver.com/flashdev.cafe
[2] 김상형, Windows Application programming interface API정복, pp.19-214
[3] 김선우, 신화선, 윈도우 프로그래밍 Visual C++ MFC Programming, pp.403-409