

## 슈팅게임을 위한 충돌 처리 알고리즘

서정만\*

### Algorithm of collision processing for a shooting game

Jeong-Man Seo \*

#### 요 약

본 논문에서는 기존의 사각형을 이용한 충돌처리 기법 알고리즘에 대하여 알아보고, 문제점을 제시하며, 단순한 사각형 충돌의 단점을 보완한, 작은 사각형 단위의 충돌 체크 기법을 제안하고, 실험과 실제 구현한 게임 화면 디자인을 통하여 제안한 알고리즘이 실제 게임에서 적용할 수 있음을 보인다.

#### Abstract

In this paper, I review the collision processing algorithm using existing quadrilateral unit and present problems with it. Then I propose a collision check technique of a small quadrilateral unit, by which the defects with simple quadrilateral collision has been made up for. I finally show that the proposed algorithm can be applied to real games in terms of presenting the experiment results and screen design for implemented real games.

▶ Keyword : collision technique, collision algorithm

---

• 제1저자 : 서정만

\* 한국재활복지대학 컴퓨터게임개발과 교수

## I. 서론

게임이라는 용어는 “흥겨우게 뛰다”라는 인도 유러피안 계통의 “ghem”에서 파생된 단어로 “흥겹다”는 정신적으로 재미 또는 즐거움을 느낀다는 뜻과 “뛰다”라는 동작을 나타내는 동사적 의미의 단어가 합성된 용어이다(1). 문화콘텐츠의 중요성이 증대되면서 영상, 음반, 애니메이션, 게임 등 문화콘텐츠산업은 전세계적으로 급속하게 성장하고 있다. PC게임, 비디오게임, 업소용게임, 온라인게임, 모바일게임 등 게임산업의 각 장르도 그 성장률은 계속 높아지고 있다. 특히 첨단 제작기술 및 이동통신 기술의 발전에 따라 게임이 구현되는 플랫폼도 기존 오프라인과 온라인으로부터 모바일에 이르기까지 다양해지고, 이에 따라 게임제작도 더욱 첨단화되고 있다(2). 인터넷을 이용한 온라인게임이나 온라인 기능을 탑재한 비디오게임기의 등장 및 모바일게임의 보급은 게임이라는 문화콘텐츠의 접근이용도를 높이고 이는 결국 앞으로 보다 거대한 게임시장의 형성으로 이어지게 될 것이다. 또한 우리나라 모바일기기의 보급률과 온라인게임의 선전을 고려했을 때, 모바일게임에 대한 투자와 지원은 게임산업을 통한 국내산업의 견인과 모바일게임의 해외수출 선점 효과로 이어질 것이다. 오락실에서 이루어지고 있는 게임들은 대부분 아케이드 게임의 형태에 속한다고 볼 수 있다. 인터넷을 이용하지 않은 PC에서 이루어지고 있는 슈팅게임은 형태가 여러 가지이지만, 대부분 총알이나 어떤 형태의 무기를 발사하고, 그것이 적이나 아군이 맞으면 충돌이 되어서 폭발되거나 없어지도록 하는 것이다. 여기에 슈팅게임에서 충돌이 되었는지를 검사하거나 프로그래밍하는 것은 쉬운 일은 아니다. 본 논문에서는 기존의 사각형을 이용한 충돌처리 기법 알고리즘에 대하여 알아보고, 문제점을 제시하며, 단순한 사각형 충돌의 단점을 보완한, 작은 사각형 단위의 충돌 체크 기법을 제안하고, 실험과 실제 구현한 게임 화면 디자인을 통하여 제안한 알고리즘의 우수성과 향후 연구과제에 대하여 을 논한다.

## II. 기존 연구

충돌 처리란, 말 그대로 두 물체가 서로 충돌 관계에 있는지를 판별하는 작업을 의미 한다. 이러한 처리가 필요한 이유는, 디자인과 프로그램을 멋지게 구현하였다도 화면에 등장하는 아군 캐릭터 그리고 화면 상단에 존재하는 적군 캐릭터들, 멋진 등장 뒤에 멋있게 총알 발사 버튼을 누르고, 그리고 그 화면 상단으로 날라가는 총알을 바라보며 그 총알에 맞아 죽게 될 적을 상상하지만, 만약 충돌처리가 이루어지지 않는다면 아무리 총알을 쏘고 그 총알이 적들에게 날라가도 그 총알에 맞아 죽지 않는 적들을 본다면 아마도 허무감이나 허탈감, 아니 이런 게임이 다 있는가 하고 생각할 것이다. 충돌 처리에는 여러가지 기법들이 존재한다. 여러가지 충돌 처리에 대해서 소개하면, 2D에서 가장 정확한 판단을 할 수 있는 픽셀 처리 방법, 이미지의 전체 영역을 작은 박스 단위로 나누어서 비교적 정확한 판단을 할 수 있는 방법, 회전 사각형 충돌 처리, 그리고 가장 빠르고 쉽게 구현할 수 있는 사각형 충돌 기법들이 있습니다. 그 밖에 3D에서 사용할 수 있는 바운딩 박스에 의한 충돌 기법, 바운딩 구에 의한 충돌 기법들이 있다. 충돌 처리에 이 처럼 여러가지 기법이 존재하는 이유는, 보다 정확한 충돌 처리를 하기 위함과 정확하지 않지만, 특정 게임의 생명이라 할 수 있는 속도 문제로 인해 이렇게 여러 다양한 기법들이 존재하게 된다(3). 격투 게임과 같이 비교적 많은 객체들이 존재 하지 않고 정확히 특정 부분에 충돌이 발생 했는지를 알아 보려면, 속도는 느리지만 픽셀 단위 혹은 작은 사각형단위의 충돌 처리가 적합할 것 이고, 빠른 그래픽 처리 속도를

요구하는 슈팅게임 등에서는 비교적 정확하진 않지만 코드가 단순해서 빠른 처리를 할 수 있는 사각형 충돌 처리가 적당할 것이다(3). 물론 이 두가지 장단점을 보완한 충돌처리 기법도 존재한다. LSP 충돌처리라는 기법이 있는데, 이는 라인 단위로 충돌 처리를 하는 기법으로 픽셀 처리 보다 빠르고 사각형 충돌 처리 만큼이나 속도도 나타내는 기법이다.

### III. 사각형 충돌처리 알고리즘

충돌상황이 발생하는 경우는 첫째로, 아군 무기에서 총알 발사시 적 캐릭터와 충돌, 둘째로, 적 캐릭터의 무기 발사시 아군 캐릭터와의 충돌, 셋째로, 아군 캐릭터와 적 캐릭터와의 충돌이다.

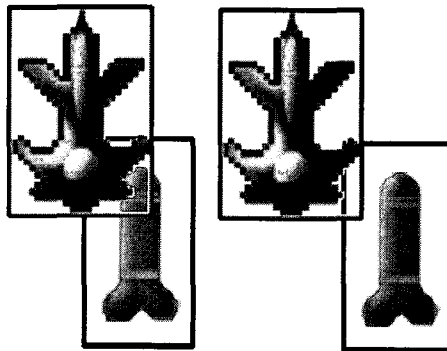


그림 42. a) 충돌 캐릭터, b) 충돌되지 않은 캐릭터  
Fig 1. a)character collision b)character uncollision

현재 게임에서 사용하는 사각형 충돌에 대해서 알아 보면, 사각형 충돌은 상당히 단순한 구조를 가지고 있다. 또한 쉽게 구현이 가능한 만큼 여러 부분에서 사용이 가능하며, 그림1에서 보면 사각형 충돌이 어떠한 것이라는 것을 알 수 있다. 왼쪽의 그림 a)을 보면, 정확히 충돌이 되었다는 것을 알 수 있다. 그러나, 오른쪽의 그림 b)을 보게 되면, 이미지들의 주위에 둘러져 있는 사각형은 충돌이 되었지만, 실제 이미지들, 캐릭터 2개는 상호간에 충돌되지 않았음을 확인할 수 있다. 그림1의 b처럼 충돌되지 않았음에도 불구하고, 충돌된 것으로 처리됨으로 인하여 게임의 신뢰성이나 게이머로부터 불만을 가지게 될 수 있다. 이 처럼 사각형 충돌은 픽셀 단위 비교가 아닌 이미지가 가지고 있는 좌표 값과 이미지의 X,Y 크기를 가지고 만든 사각형으로 두개의 이미지가 충돌이 발생 했는지 판단하는 것으로 오른쪽 그림과 같은 문제점을 발생하게 된다. 이러한 문제에 대하여 단순한 사각형 충돌의 단점을 보완한, 작은 사각형 단위의 충돌 체크 기법을 제안한 그림이다. 큰 이미지의 영역을 작은 사각형 단위로 잘라 이 잘린 영역 별로 충돌을 비교하는 이 비교 방법은 위의 사각형 충돌 시 발생했던 문제점을 해결할 수 있다.

알고리즘은 다음과 같다.

만약 캐릭터 2개를 각각 A, B라고 가정하면

If (A.Left >= B.Left) And \_

$$(A.Left <= B.Left + A.Width) \dots\dots (1)$$

If (B.Top >= A.Top) And \_

$$(A.Top <= B.Top + B.Height) \text{ Then } \dots\dots 2)$$

알고리즘에서 식(1)과 식(2)를 동시에 만족하면 2개의 A,B 캐릭터가 출동된 것이다. 식에서 Top, Width, Height등을 점차로 잘게 쪼개여 반복 처리한다면 실제적으로 충돌처리 여부를 세밀한 부분까지 처리할 수 있다. 여기서 각각의 설명은 다음의 표에 나타나 있다.

【표 1】 캐릭터 속성  
Table 1. character Attribute

속 성	의 미 설 명
Hight	컨트롤의 크기, 틀 전체의 높이를 의미.
Width	컨트롤의 전체 넓이를 의미.
Top	컨트롤의 상단 좌표를 의미, 품의 최 상단은 0.
Left	컨트롤의 왼쪽 좌표를 의미, 품의 최 좌측 좌표는 0.

#### IV. 실험 및 구현

##### 4.1 화면 디자인

〈그림 18-3〉를 보고 디자인 한다[4]. Form 1개, Picture Box 5개, Timer 4개, MMC 컨트롤 3개, Command Button 2개, Label 2개, Text Box 2개로 구성되어 있다. MMC 컨트롤은 배경음악과 미사일을 발사 할 때의 소리와 충돌이 되었을 때의 소리를 위한 개체이다.

## 4.2 실험환경

실험환경으로는 프로그램 언어는 visual basic을 사용하였고, 펜티엄4, 윈도우 XP, 메인메모리 512메가바이트이다. 실제 구현한 화면에서 정상적으로 충돌처리 됨을 알 수 있었다.

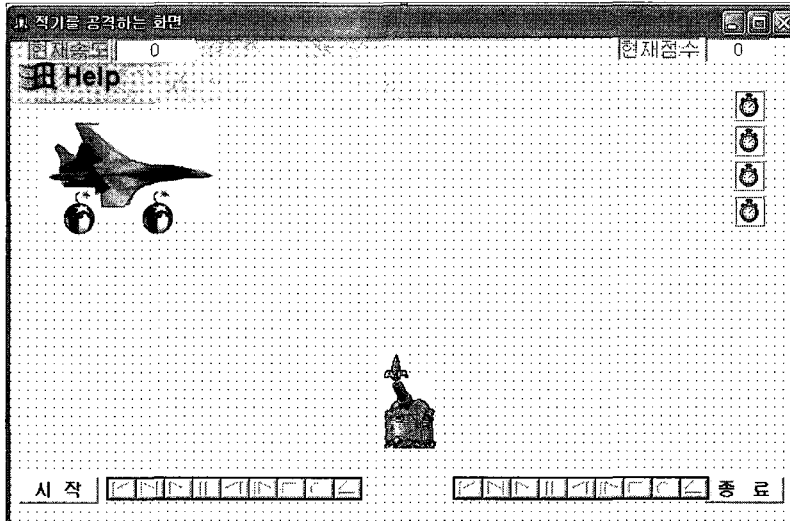


그림 43. 화면 디자인  
Fig 2. window design

## V. 결론

본 논문에서는 PC환경에서 슈팅게임 시에 충돌처리 알고리즘에 대하여 소개하였고, 기존의 사각형 충돌처리 기법에서 단순한 사각형 충돌의 단점을 보완한, 작은 사각형 단위의 충돌 체크 기법을 제안한 그림이다. 큰 이미지의 영역을 작은 사각형 단위로 잘라 이 잘린 영역 별로 충돌을 비교하는 이 비교 방법은 위의 사각형 충돌시 발생했던 문제점을 해결할 수 있음을 보였다. 향후 연구과제로는 보다 세밀한 충돌처리기법에서 3차원 게임에서 충돌처리기법에 대한 구현이다.

## 참고문헌

- [1] 민용식, 이동희, "게임학개론", 도서출판정일, pp. 14, 194, 2002
- [2] 성재환, "모바일 게임 산업 동향과 발전방안연구", (재)게임종합지원센터, pp.11, 2001
- [3] <http://www.cizi.pe.kr/Lec/Invader19.htm>, "충돌처리"
- [4] 서정만, "비주얼베이직과 게임만들기", 도서출판정일, pp. 319-322, 2005

### 저자소개



서 정 만

2003년 충북대학교 컴퓨터공학과 공학박사

2002년 ~ 현재 : 국립한글재활복지대학 컴퓨터게임개발과 조교수

〈관심분야〉 데이터베이스, 게임프로그래밍, 실시간처리