ABSTRACT

The movement of enemies in a game especially to find or chase players is something that is difficult to do by experimenting with existing routes. The time needed to solve these problems depends on the complexity of the obstacles available. Of course, obstacles that have high complexity will be very difficult and also take a long time to be resolved manually. The results of previous studies have proposed an approach to solve this problem by using vector algorithm calculations, but there are still various deficiencies in terms of effectiveness and response of non-playable characters to players who have lags. In this research, the proposed solution is to use a collision detection algorithm. The collision detection algorithm is an algorithm that can check whether several spatial objects are overlapping or not. By using collision detection algorithms, obstacle problem solving can be done easily and quickly. Apart from using the collision detection algorithm to chase players, NPC uses the random number generator algorithm. The random number generator algorithm plays a role in the process of randomizing the items that will be received by the player when the player defeats the monsters in the game. The random number generator was chosen because it can solve problems that arise in the process of randomizing items so that the occurrence rate of recurring items becomes smaller.

***Keywords**: collision detection algorithm, game, non-playable character, random number generator algorithm*

ABSTRAK

Pergerakan musuh dalam sebuah game khususnya untuk menemukan atau mengejar pemain merupakan sesuatu yang sulit dilakukan dengan cara mencoba-coba rute yang ada. Waktu yang dibutuhkan untuk menyelesaikannya permasalahan tersebut sangat bergantung dari kompleksitas rintangan yang tersedia. Tentu saja rintangan yang memiliki kompleksitas yang tinggi akan sangat sulit dan membutuhkan waktu yang lama pula untuk diselesaikan secara manual. Hasil penelitian sebelumnya telah mengusulkan pendekatan untuk menyelesaikan masalah ini dengan menggunakan perhitungan algoritma vector, namun masih terdapat berbagai kekurangan dalam hal efektifitas dan respon *non-playable character* terhadap player yang memiliki jeda. Di dalam penelitian tugas akhir ini solusi yang diusulkan adalah dengan menggunakan algoritma *collision detection.* Algoritma *collision detection* merupakan salah satu algoritma yang dapat melakukan proses pengecekan apakah beberapa buah objek spasial saling bertumpuk atau tidak. Dengan menggunakan algoritma *collision detection*, penyelesaian masalah rintangan dapat dilakukan dengan mudah dan cepat. Selain pemanfaatan algoritma *collision detection* untuk melakukan pengejaran terhadap player, *npc* memanfaakan algoritma *random number* generator. Algoritma *random number generator* berperan untuk melakukan proses pengacakan terhadap item-item yang akan diterima oleh pemain ketika pemain mengalahkan monster yang ada didalam permainan. *Random number generator* dipilih karena menyelesaikan masalah yang muncul pada proses pengacakan item sehingga tingkat kemunculan item-item yang berulang menjadi semakin kecil.

Kata Kunci: algoritma *collision detection*, algoritma *random number generator*, *game*, *non-playable character*