# Convexity based collision detection in a 2D game engine – **FIRST DRAFT**

## Introduction

Collision detection is used in many fields from robotics to games. The goal of collision detection is to find out whether objects are intersecting, in games this is vital … Therefore the intention of this essay is to analyse three convexity-based collision detection algorithms for use in a 2D game engine. The first section will detail the chosen algorithms, compare them and then make a recommendation based on the given data.

## Algorithm 1

Gilbert et al propose their collision detection algorithm the ‘Gilbert- Johnson – Keerth’ (GJK) algorithm. [GJK] The algorithm is designed to work efficiently for 3D collision detection however it also works for 2D. The intention of this algorithm is to determine whether objects intersect by calculating the Eucildean distance between a set of convex objects. According to Gilbert the distance between two convex objects is the same as the distance between those object’s Minkowski difference and the origin. If that difference is zero then the objects are intersecting each other and have collided.

The original paper gives run times for the algorithm on a selection of 3D shapes, the minimum being 0.13 seconds and the maximum being 19.81. However these are being run in 3D so it is likely it would take less time on 2D convex objects and this paper was published in FIND YEAR so the results are restricted by the technology of that time, running those tests again on modern hardware would most likely produce smaller CPU times.

This leads to the fact that an issue with the GJK algorithm could be its age, [FIND PAPER] there are many papers that recommend and detail improvements that could be made to GJK to make it faster or more efficient. As it was originally proposed in 1988???

However there are many techniques that could be used to improve the run time of the GJK algorithm such as LOOK UP.

## Algorithm 2

The second algorithm is designed for collision detection in 2d grapple games. This paper was published in 2010 making it more recent than the GJK algorithm however it is limited to 2D. In contrast GJK could work in m-dimensions, the paper detailed it’s use for three dimensions but it is also viable in 2D. Guo [Guo] proposes a method which bounds the game objects in axis aligned rectangles and circles. A collision has occurred if there is an overlap region between the shapes.

In this algorithm the detection area is the area inside of the bounding objects such as the rectangles and circles.

Data such as the coordinate systems are stored in data files beforehand and then loaded into the detection module while running.

The evidence presented shows that this is a fast???? Algorithm. However an issue could be that this algorithm is specific to 2D grapple games, it may be unsuitable for use in other games or require some adaption. In contrast the GJK algorithm is more open to general use as it can be used in 2D or 3D and is not just limited to use in games.

The techniques detailed above may also be applied to this algorithm???

## Algorithm 3

The third technique has been proposed by Sulaiman et al. The algorithm is for distance computation in narrow phase collision detection

## Recommendation

## Conclusion