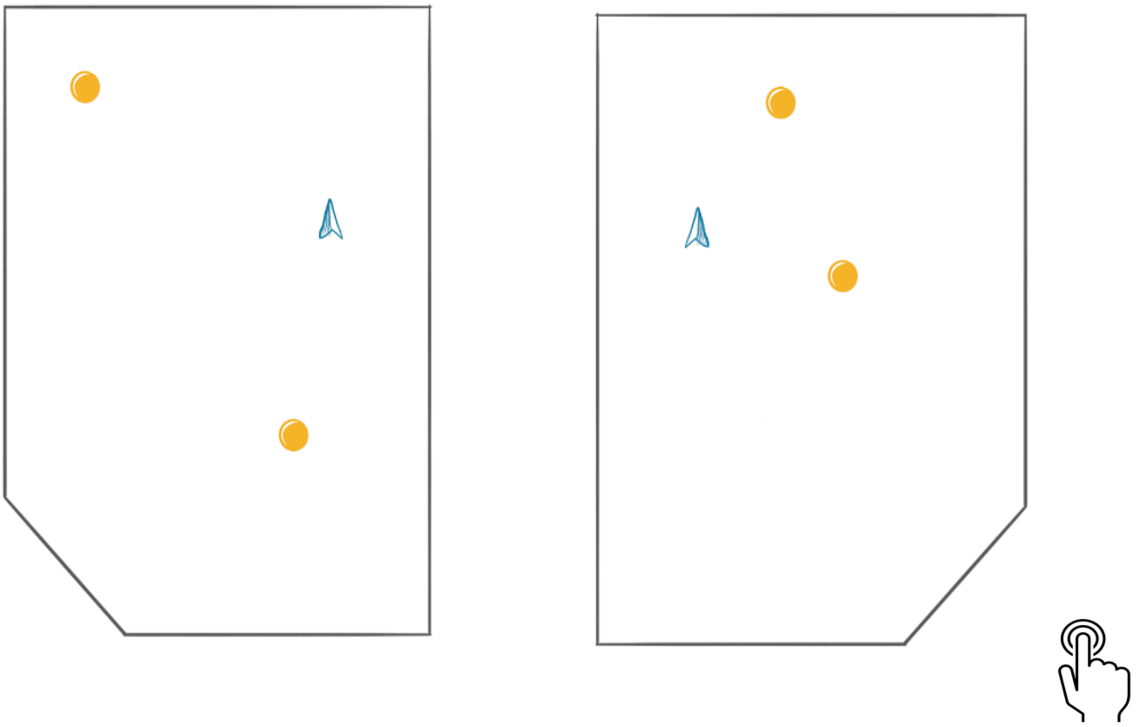
End semester report of Transform Game

Introduction:

We are making python-based game named “Transform-Game”. Our game can be played by two people. Our main aim is to provide users new concept with Great entertainment. Simple Python-based game. Here multiple objects interacting in the XY plane using positions and vectors - can be visualized and be controlled using keys. It is example of a problem-solving game that uses LA in the background to check for the optimal solution or winning criteria, etc. Collision detection is the computational problem of detecting the intersection of two or more objects. Collision detection algorithms can be divided into operating on 2D and 3D objects. Here we are using with 2D objects.

Illustration



nd:

Making a game using Linear Transformation at its core. The player can move in his 2-dimensional space. On the right is another copy of the world which is skewed (Linearly transformed concerning a matrix (which can be manually modified)). And it will detect collision in both dimensions. So that user will not be set with the environment of game as this feature will make user more active.

Here two particles moving on the same plane and there is one player to operate them. Similarly, there is another plane which is a mirror image of first one and in this plane there are also two particle which is operate by the second player. Quantity of particles in both the plane will be changed as user goes on upper level. There is a 45' wall so that particle interact and change its direction and vector.

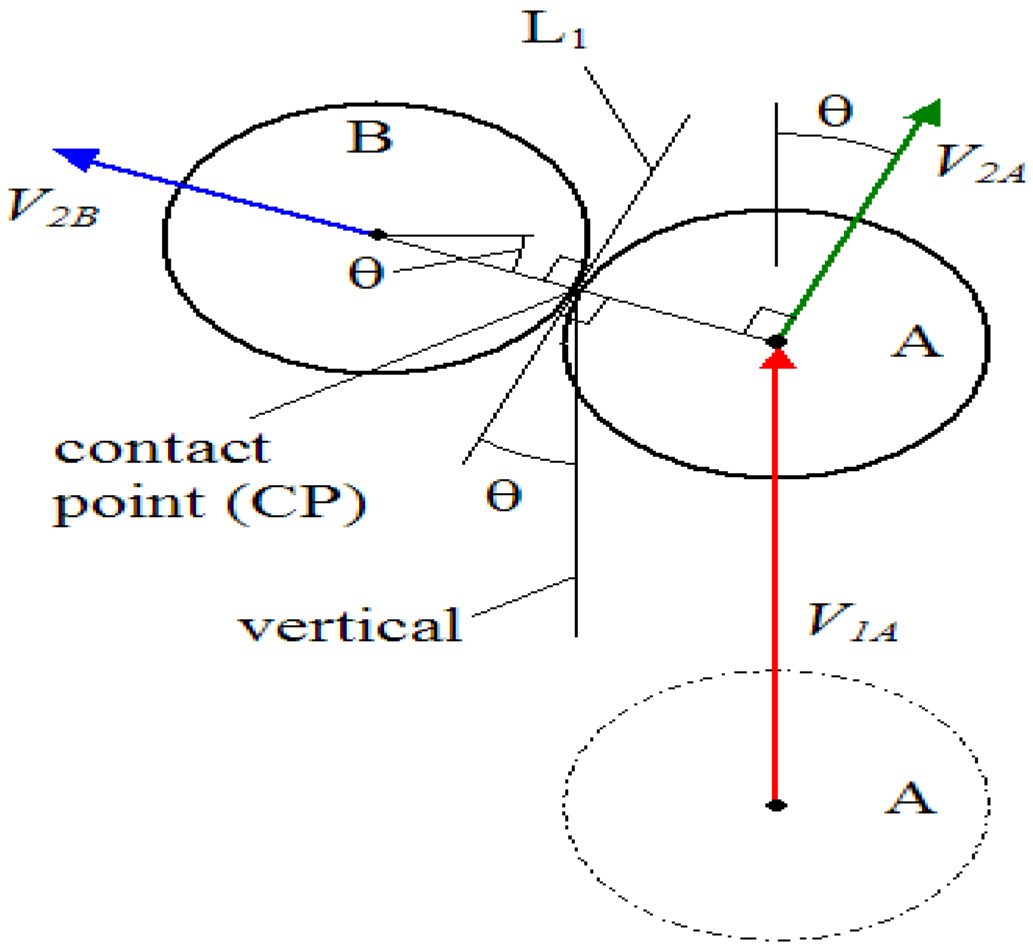
Motivation:

Learning Linear Algebra while practicing to code. Using vectors in 2 separate coordinate systems to find position of each entity and finding if they collide with a range around each entity (Hit Box). It is intended to make the brain become more adaptable.

Literature survey:

We are optimizing collision detection in this project.

Billiards balls hitting each other are a classic example applicable within the science of collision detection.



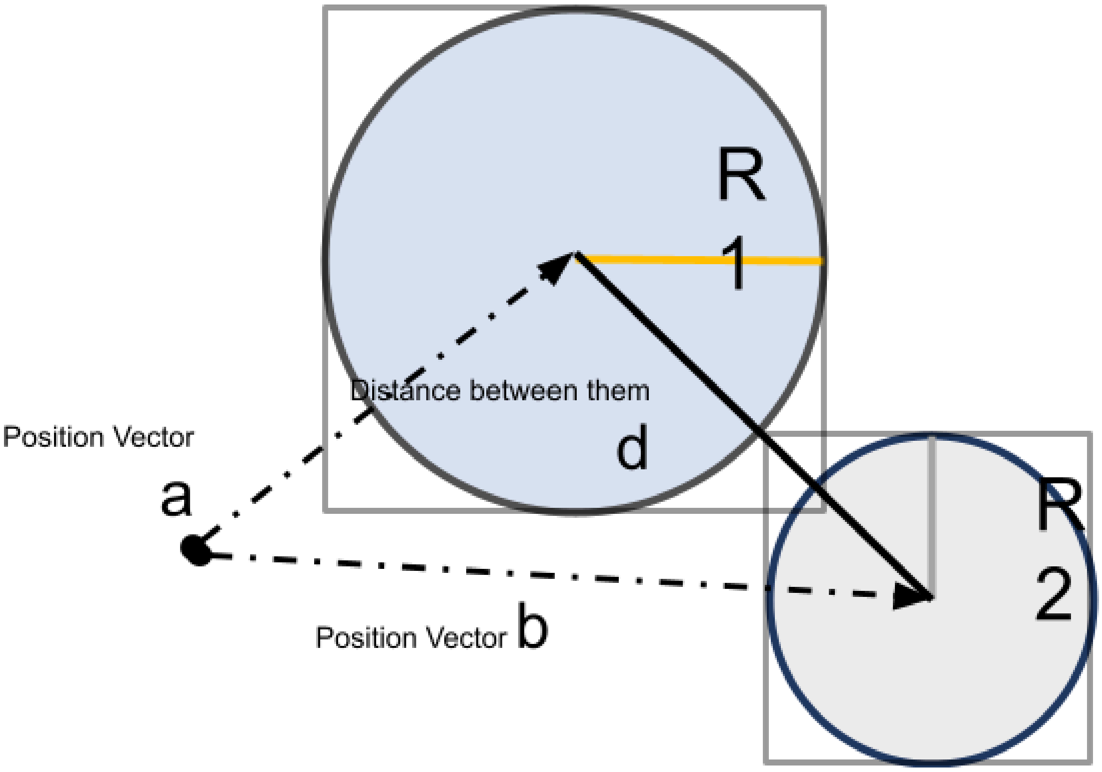
Transform game is similar to all this rules, collision and angle details in billiard game. In our game white ball will look like a person and other coloured ball in billiard will be particles in this game.

Our survey mainly focused on collision detection and it’s implementation in python because our concept is different and other details can’t available on web sites, books, and other sources. As we know that Collision detection algorithms optimize the task of detecting collisions by maintaining a set of neighbours for every particle in the set and only checking for collisions between neighbouring particles. This small things makes lot expects while implementation of logic in game.

If the number of entities increase, the number of calculations also increases before each frame is rendered.

$$O(n) = (numberofentities)!$$

To shorten the process of checking if two entities are colliding, first a square hitbox is checked and if they intersect with each other, it will check with greater accuracy whether it is inside circle or outside.



Contributions

Idea/Theme of game.	Aharnish
Vector based movements (Game Engine)	Aharnish, Kenil, Jay
Collision detection based on ALA.	Aharnish
Graphical User Interface using python (TKinter)	Kenil, Viraj
Random Entity generation	Jay
Report	Viraj,Kenil
Power point slides	Kahan

Mathematical model

We will refer our defined bounding shape of a visible object as an object. Let us consider collision between an object in motion and another object in still.

If the square hitbox is colliding then $R1+R2<d$ then collision is detected.