

High Level Architecture

The position and speed of each game object are represented by 2-dimensional vectors, which simulates these physical properties. At each game cycle the position of a ball is changed according to the formula:

$$\vec{x} \leftarrow \vec{x} + \vec{v} \cdot \Delta t$$

The boundaries of a game object are represented by a rectangle, which defines the left, right, top and bottom edges. This representation applies even for non-rectangular objects, such as the ball. All objects are contained into a much larger rectangle, which is the canvas provided by the game engine — an ultimate boundary for the movement and existence of the objects.

The main game loop iterates over each object and calls their `update()` and `display()` methods. The `update()` method is responsible to refresh the object's logic, physical calculations and state machine (if any). The `display()` method sole purpose is to draw a polygonal representation of the object on the screen.

Collision Detection

The collision detection logic is implemented by modularized and reusable objects. However, as of now, the only movable object in the game is the ball. At each cycle, the ball iterates over other elements in the game (including other balls) and checks if there exists an intersection between its rectangle and theirs. If any intersection is detected, the horizontal and/or vertical components of the ball's speed vector are reversed accordingly.

In order to check if two rectangles intersect we compare their edges positions. This detection is implemented in two parts, which check independently for horizontal intersections (left and right edges) and vertical intersections (top and bottom edges) between the rectangles. The rectangles intercept if and only if horizontal and vertical intersections happen simultaneously.

The following diagrams shows the necessary comparisons to find intersections between two rectangles, A and B. It also shows the change in the horizontal and vertical directions of the speed vector of A that results from the collision with B.

