

TODO: Clean up `__init()` so it's not as much a wall of code.

1 class ArcLineArc

1.1 start

Vec3. The starting point for the path.

1.2 end

Vec3. The ending point for the path.

1.3 start_tangent

Nonzero Vec3. The tangent to the path at **start**.

1.4 transition1

Vec3. The location of the transition from the first line segment to the arc.

1.5 radius

Float. The signed radius of the arc. Positive for CW, negative for CCW.

1.6 current_state

TODO: Add this argument or `follow_path` won't work. **CarState**. The current state of our car, probably from `game_info.me`.

1.7 self.center

Vec3. The location of the center of the circle containing the arc.

1.8 self.end_tangent

Vec3. The tangent vector to the path at **end**.

1.9 self.transition1

Vec3. The location of the transition from the arc to the second line segment.

1.10 self.is_valid

Boolean. True when the LineArcLine is valid, using conditions added as seen fit. TODO: Add checks that we don't leave the floor of the stadium.

1.11 self.find_lengths()

Returns three floats: The length of the first line segment, the length of the arc, and the length of the second line segment.

1.12 self.draw_path()

Uses the RLBot renderer to draw the LineArcLine path on screen. Disable for tournament versions.

1.13 self.check_validity()

Returns a boolean, True when the LineArcLine is valid. Currently always returns True. TODO: implement actual checks.