



AI-based Air Hockey game

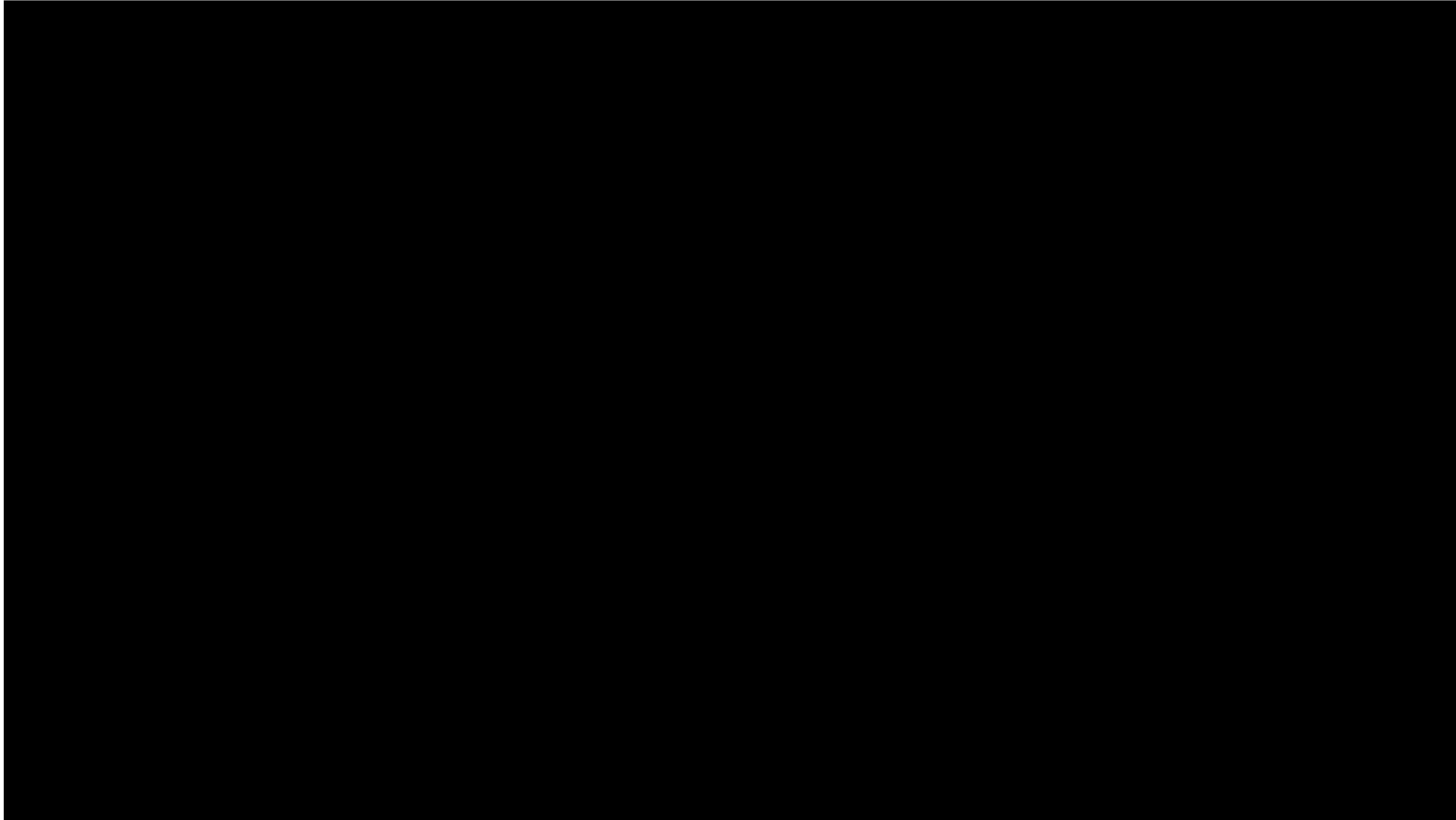
by Leonardo Chang, Ph.D., lchang@tec.mx



contenido

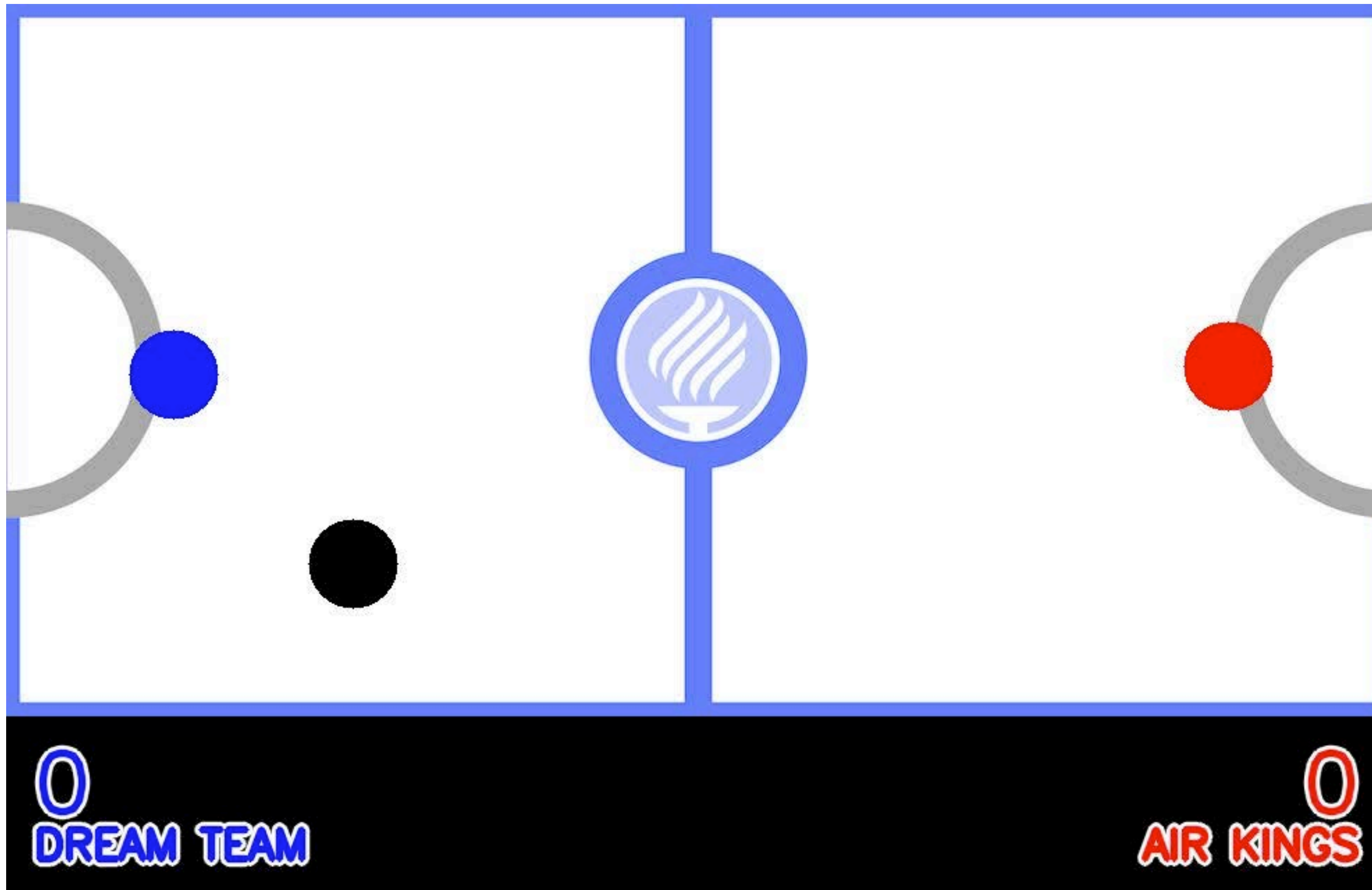
- 01 Objetivo del reto**
¿qué haremos durante la semana i?
- 02 Características del reto**
arquitectura de alto nivel, qué debo hacer.
- 03 Asunciones del sistema**
¿cuáles son las bases sobre las que se define el problema?
- 04 Reglas**
qué debo hacer para ganar, cuáles faltas debo evitar
- 05 Herramientas**
códigos y plataformas de apoyo

el verdadero air hockey



Fuente: <https://youtu.be/Br-8jxfLql4>

nuestra versión de air hockey



<https://youtu.be/Hr5h6PfBy8M>



objetivo del reto



Crear un algoritmo que permita **determinar los movimientos del mazo buscando anotar gol** en la portería contraria.

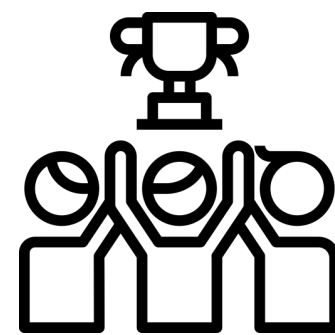
Se podrá utilizar todo tipo de algoritmos, desde algoritmos que implementen estrategias propias de los programadores, algoritmos adaptados de otros problemas o algoritmos de aprendizaje automático.

*Equipos de hasta **3 estudiantes**.*

competencia



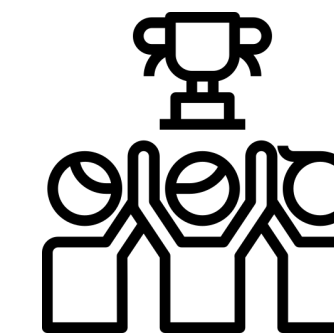
**un equipo ganador
absoluto de la competencia**



un equipo ganador
por categoría
semestres 1, 2, 3

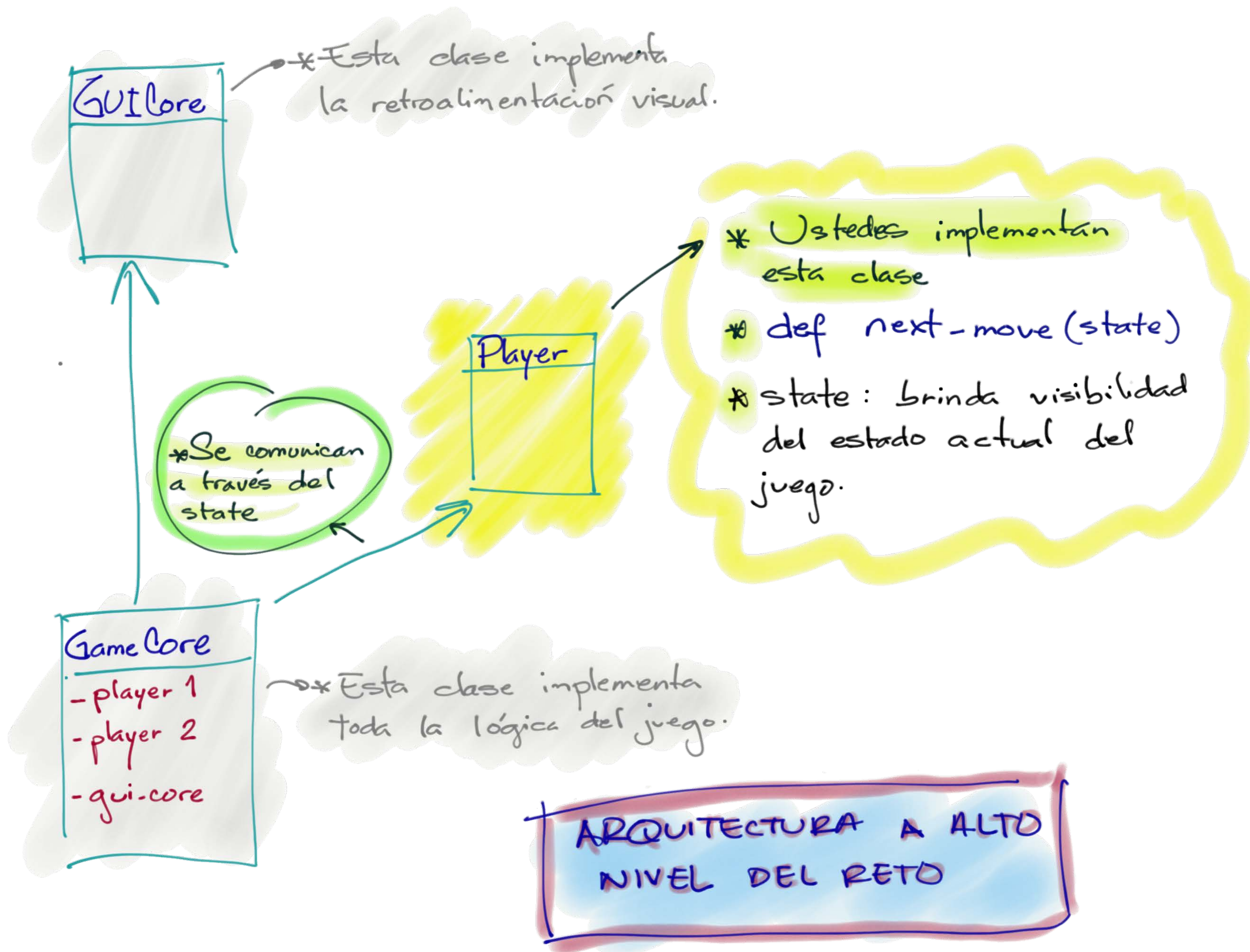


un equipo ganador
por categoría
semestres 4, 5, 6



un equipo ganador
por categoría
semestres 7, 8, 9







¿dónde aplico toda mi grandeza?

```
class Player:
    def __init__(self, paddle_pos, goal_side):
        # set your team's name, max. 15 chars
        self.my_display_name = "AIR KINGS"

    def next_move(self, current_state):
        """
        return {'x': paddle_x, 'y': paddle_y}
```

- Los estudiantes implementan la clase **Player**.
- y la función **next_move** dentro de ella.
- que recibe como **entrada** en cada iteración el **estado actual del juego**.
- y **devuelve** la **nueva posición** de su paddle.



¿qué recibo como entrada?

un diccionario con las siguientes llaves

<code>delta_t</code>	Resolución de tiempo, i.e., frecuencia de cada jugada
<code>board_shape</code>	Tamaño de la mesa de juego, tupla (<code>alto</code> , <code>ancho</code>)
<code>goals</code>	Puntaje, diccionario <code>{'left': 0, 'right': 0}</code> Que indica la cantidad de goles de cada lado.
<code>goal_size</code>	Amplitud de la portería respecto al alto de la mesa
<code>puck_radius</code>	Radio del puck, en pixeles
<code>puck_pos</code>	Posición del puck, diccionario <code>{'x', 'y'}</code> . El origen de coordenadas se asume en la esquina superior-izquierda
<code>puck_speed</code>	Velocidad del puck, diccionario <code>{'x', 'y'}</code> . Dado por la componente de la velocidad en <code>x</code> y <code>y</code> , respectivamente.



¿qué recibo como entrada?

un diccionario con las siguientes llaves (continuación)

`paddle1_pos`

Posición del paddle del jugador cuya portería es la izquierda, diccionario `{'x', 'y'}`. El origen de coordenadas se asume en la esquina superior-izquierda

`paddle2_pos`

Posición del paddle del jugador cuya portería es la derecha, diccionario `{'x', 'y'}`. El origen de coordenadas se asume en la esquina superior-izquierda

`paddle1_speed`

Velocidad del paddle1, diccionario `{'x', 'y'}`. Es `{'x': 0, 'y': 0}` durante todo el juego.

`paddle2_speed`

Velocidad del paddle2, diccionario `{'x', 'y'}`. Es `{'x': 0, 'y': 0}` durante todo el juego.

`paddle_radius`

Radio del paddle, en pixeles

`paddle_max_speed`

Velocidad máxima a la que puede moverse el paddle

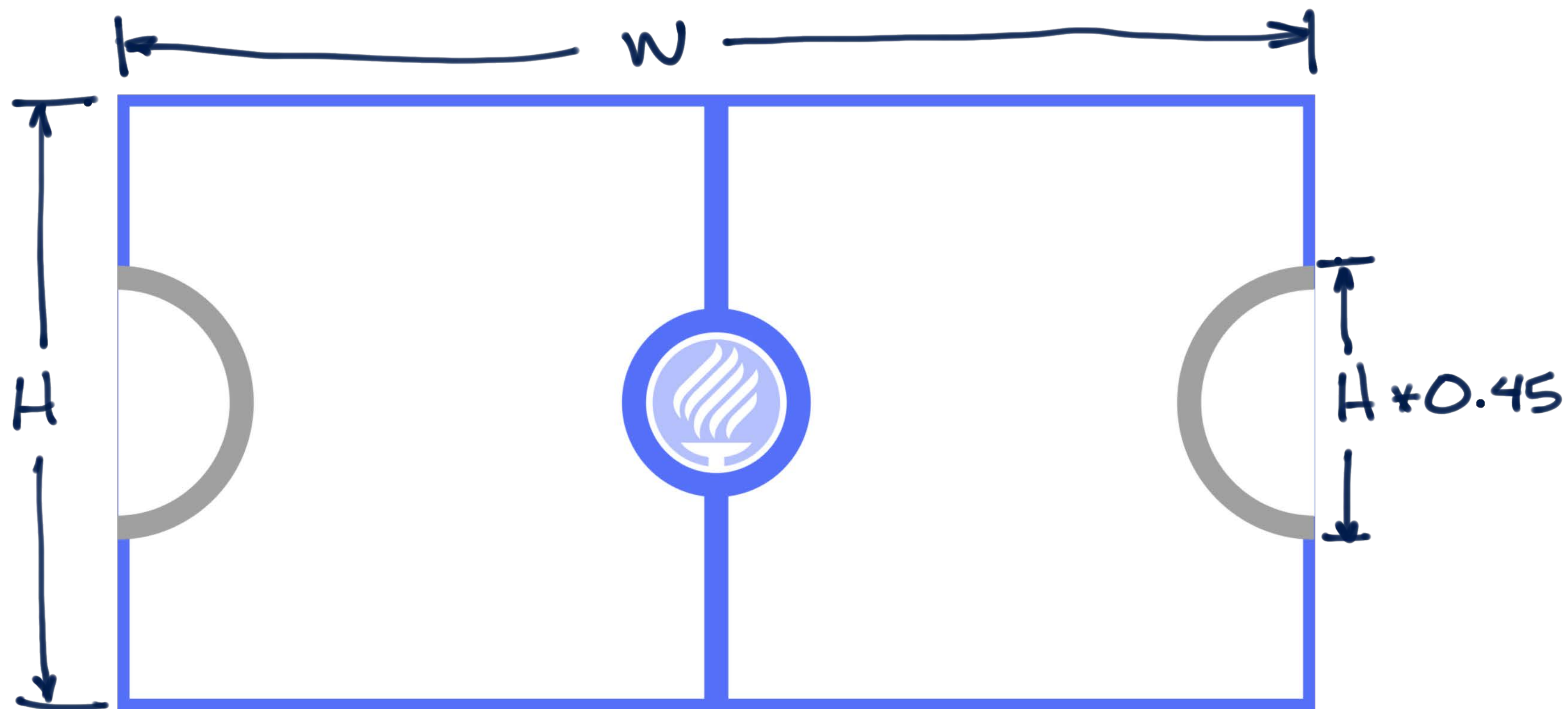
`is_goal_move`

Indica si el presente movimiento fue goal (not `None`), en caso positivo, indica donde `'left', 'right'`. Esta bandera solo dura una jugada.



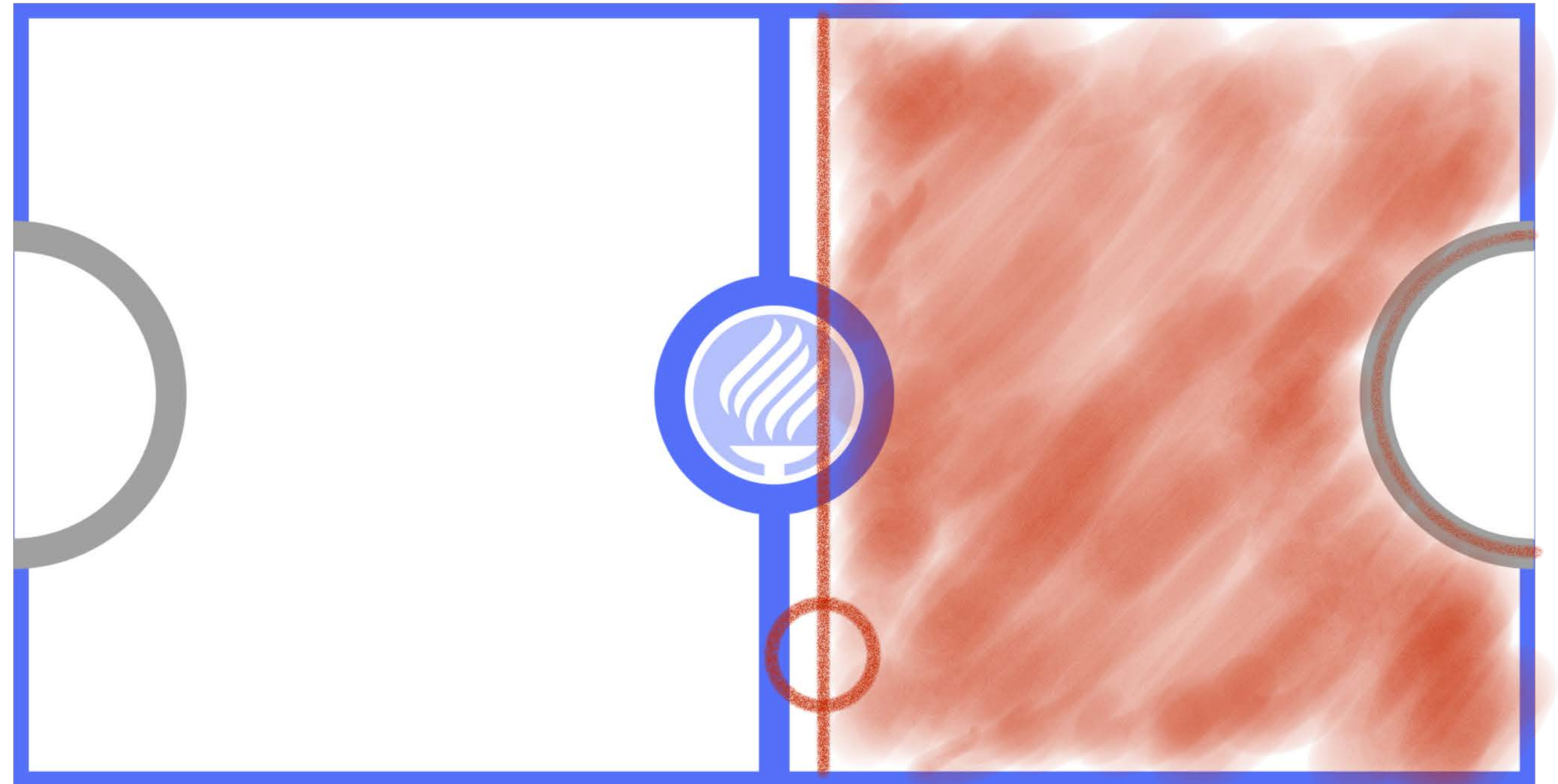
¿qué podemos asumir?

de la mesa de juego



¿qué podemos asumir?

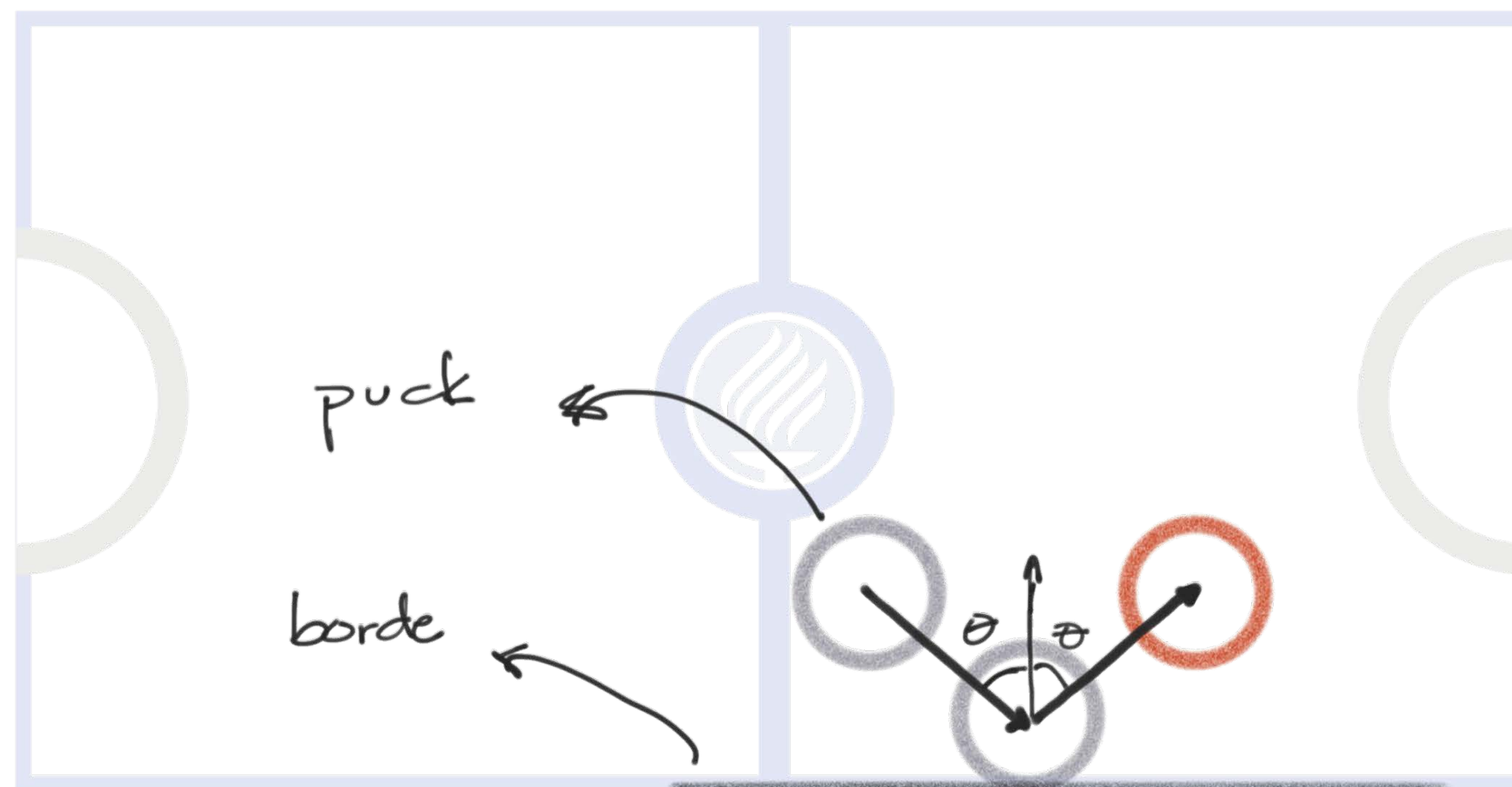
de mi área de movimiento





¿qué podemos asumir?

sobre el rebote del puck contra los bordes



* Un rebote contra los bordes mantiene la velocidad del puck y solo cambia la dirección en igual ángulo. Para este ejemplo: $(v_x, v_y) = (v_{0x}, -v_{0y})$

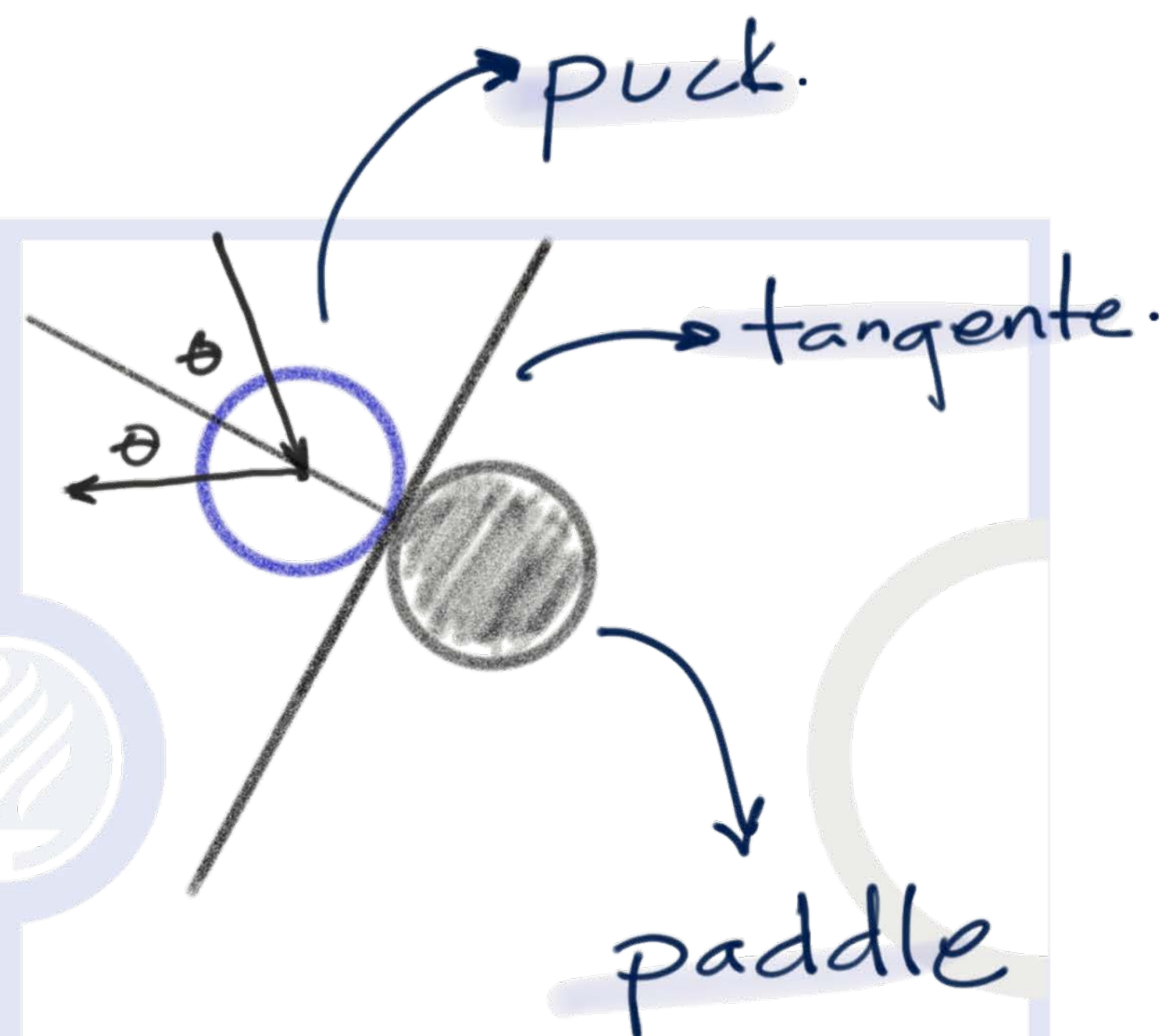


¿qué podemos asumir?

sobre el rebote del puck contra el paddle

* El puck rebota en el paddle como si chocara con una superficie plana cuya pendiente se define por la tangente de la circunferencia que representa al paddle en el punto de impacto.

* El puck mantiene una velocidad constante durante toda la partida.

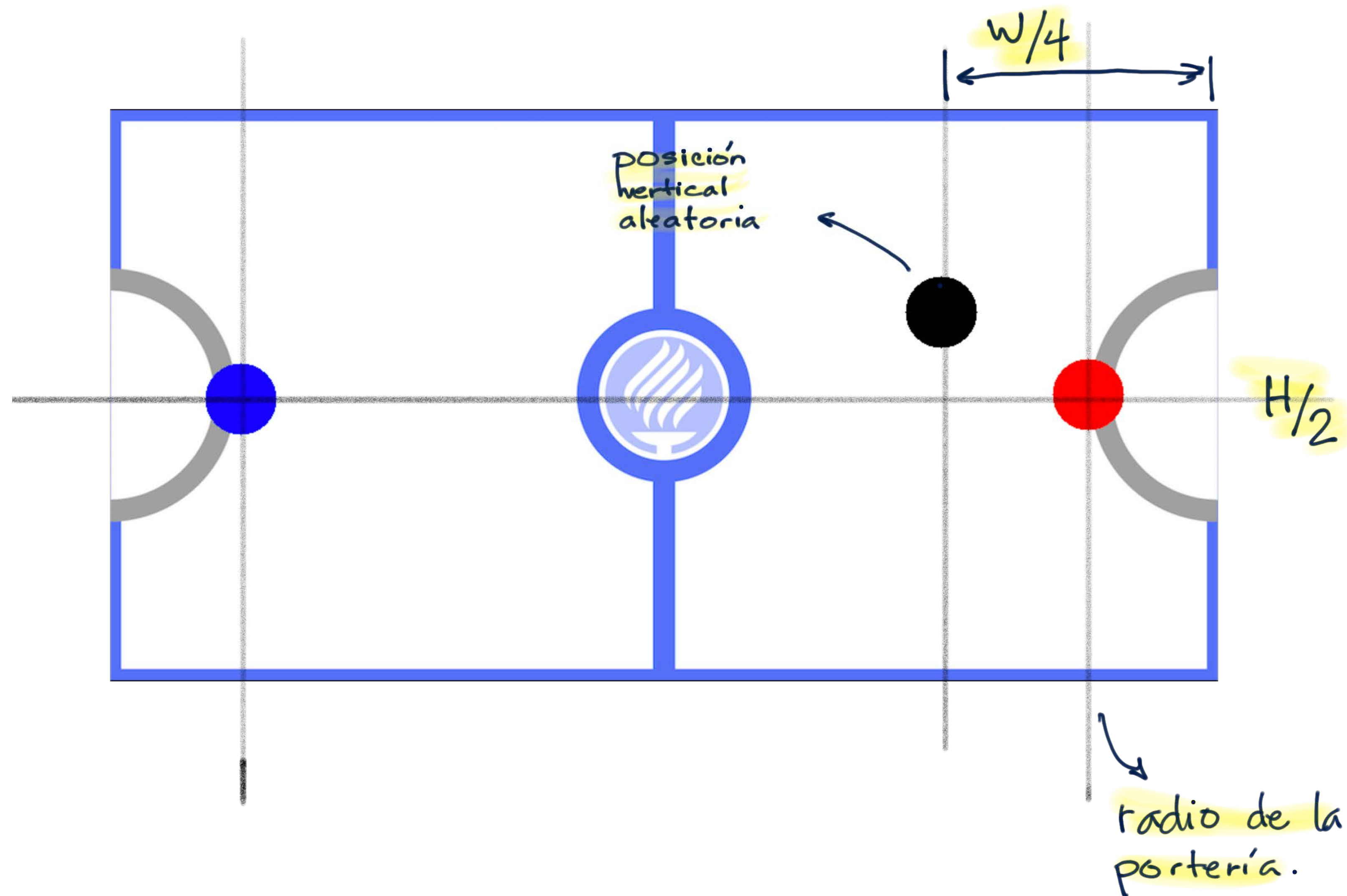


* al momento del impacto se asume que el paddle no tiene movimiento, i.e., $v_x = 0$, $v_y = 0$.



¿qué podemos asumir?

sobre el estado inicial y después de gol





¿qué podemos asumir?

haciendo más difícil el problema

- Luego de cada movimiento de cada jugador, se le añadirá una perturbación aleatoria.
- El tamaño máximo será predefinido.



¿cuándo termina la partida?

- Uno de los dos jugadores **alcanza el máximo de goles**.
- **Luego de un tiempo t** , gana el robot con mayor cantidad de goles.
- Si hay empate, se da un **tiempo extra** de $t * 0.3$. Si luego de ese tiempo siguen empatados, termina en empate.
- Si algún jugador comete **falta grave**, termina la partida y pierde la misma.



es falta grave si:

y termina la partida, perdiendo el autor de la falta

- Se mueve el **paddle más rápido que su velocidad máxima.**
`state['paddle_max_speed']`
- Se mueve el **paddle fuera de los límites del tablero.**
`state['board_shape']`
- El **paddle toca la línea divisoria** central.
- **La duración de una jugada** es mayor que tiempo predefinido. i.e., tiempo de ejecución de la función `Player.next_move()`
- Cualquier otro **error en tiempo de ejecución.**



otras faltas:

- Luego de la salida, **no hacer contacto** con el puck luego de un tiempo predefinido.
 - *Se le anota gol al autor de la falta y el puck pasa a la cancha del jugador contrario.*
- Cualquier variante o manifestación de **trampa**.
 - *Expulsión de la competencia y calificación de 0 en semana i.*



¿qué herramientas tenemos?



<https://github.com/leonardochang36/ai-airhockey>

- Código para el reto.
- Levantar issues respecto al código base.
- Publicaremos las soluciones ganadoras

¿qué herramientas tenemos?



Google
Classroom

qmart3g

- Por favor, inscríbanse en Classroom
- Publicaremos anuncios e información importante



¿qué herramientas tenemos?

10.49.64.57:5000

10.49.64.58:5000

- Servicios para probar vuestras soluciones contra las de otros equipos, de manera segura.
- Por favor, no saturar, son dos PCs de características de gama media.



¿cómo instalar OpenCV?

<https://pypi.org/project/opencv-python/>

- Windows
- Mac OsX
- Ubuntu



calendario

Lunes AM:

- Presentación de la actividad

Lunes PM, martes, miércoles, jueves:

- Trabajo en equipo

Viernes:

- Competencia



"Great Dialogue", by Karel Nepraš, Museum of Modern and Contemporary Art, Prague

Contacto

Leonardo Chang, Ph.D.

Ichang@tec.mx

Tecnológico de Monterrey,

Campus Santa Fé

