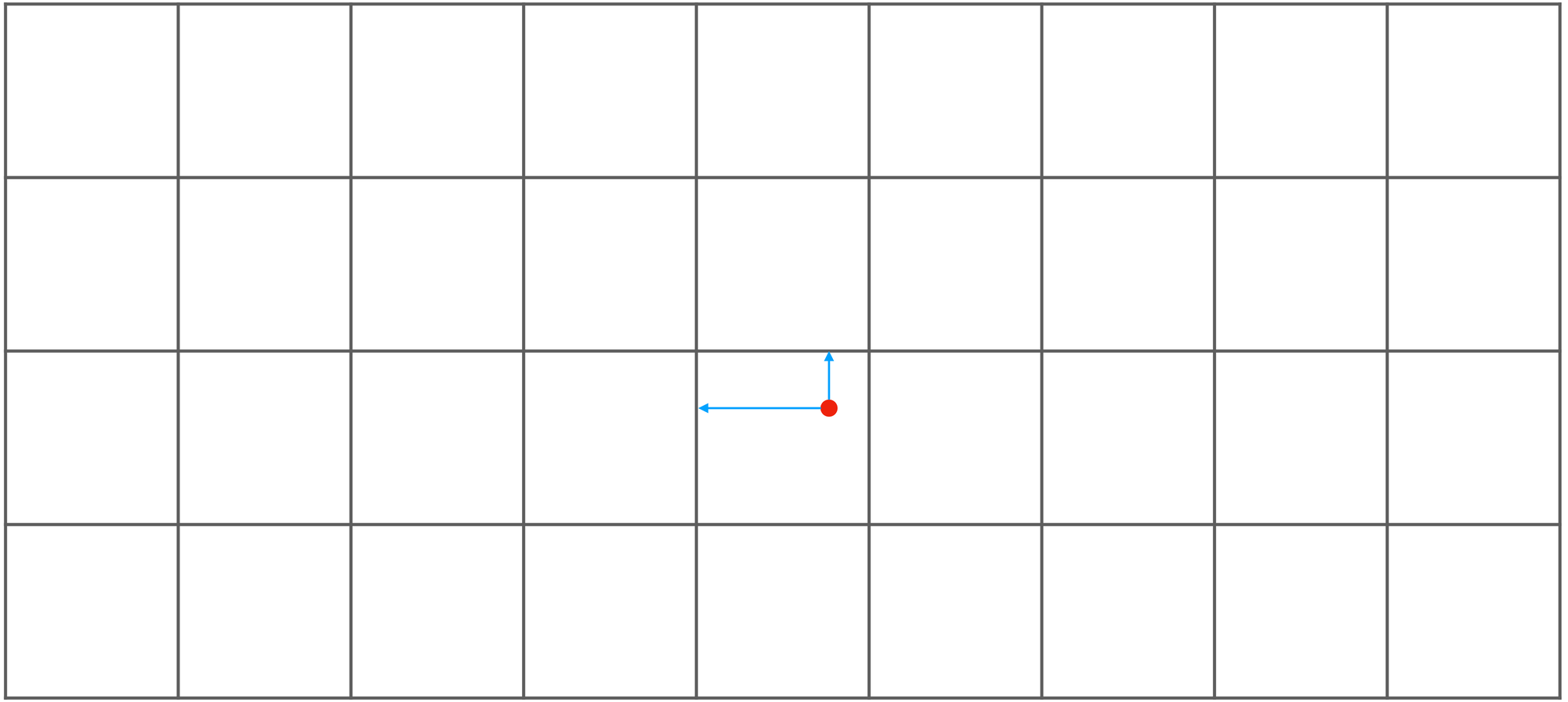
Tile-based collision

Observation: we can very easily determine in which tile we are, by dividing and rounding down the float coordinate.

```
bool collision_check(sf::vec2f position) { … }
```



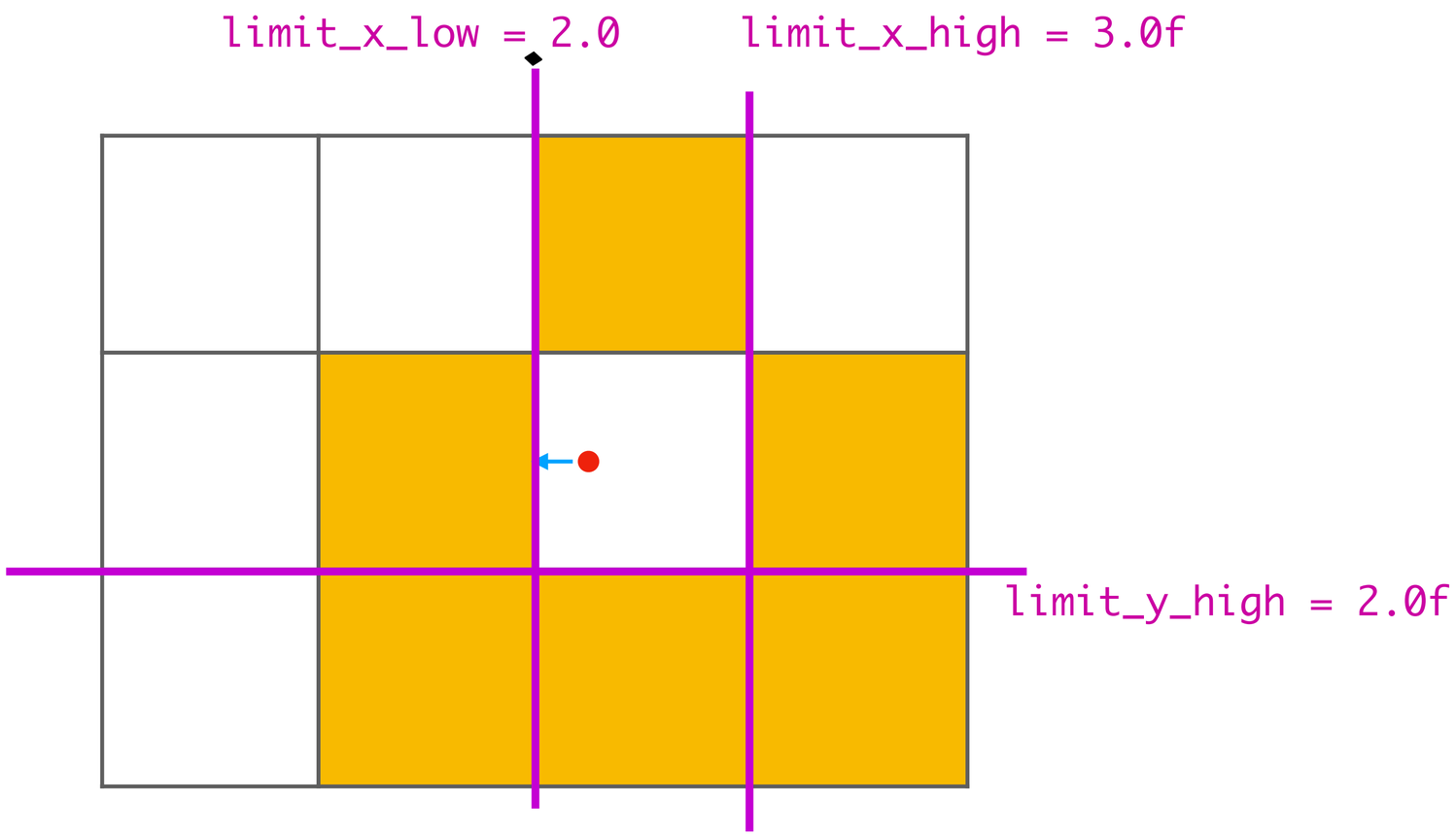
If we pretend our player is just a point with no thickness at all, can you think of a simple way to make it “collide” with the solid tiles? Implement a function `move_player(sf::vec2 wanted_delta)` that will move the player, but only if the destination position doesn’t collide.

Gravity, controls and juice

- Even though our collision is only a point, we can already make physics feel fun to play with and functional
- Using the concepts you learned in the last weeks, implement gravity, jumping, sideways movement.
- Hint: when the player collides, also reset its vertical velocity, so they don't accelerate downward infinitely
- Make a test level and play trough it

Exact collision

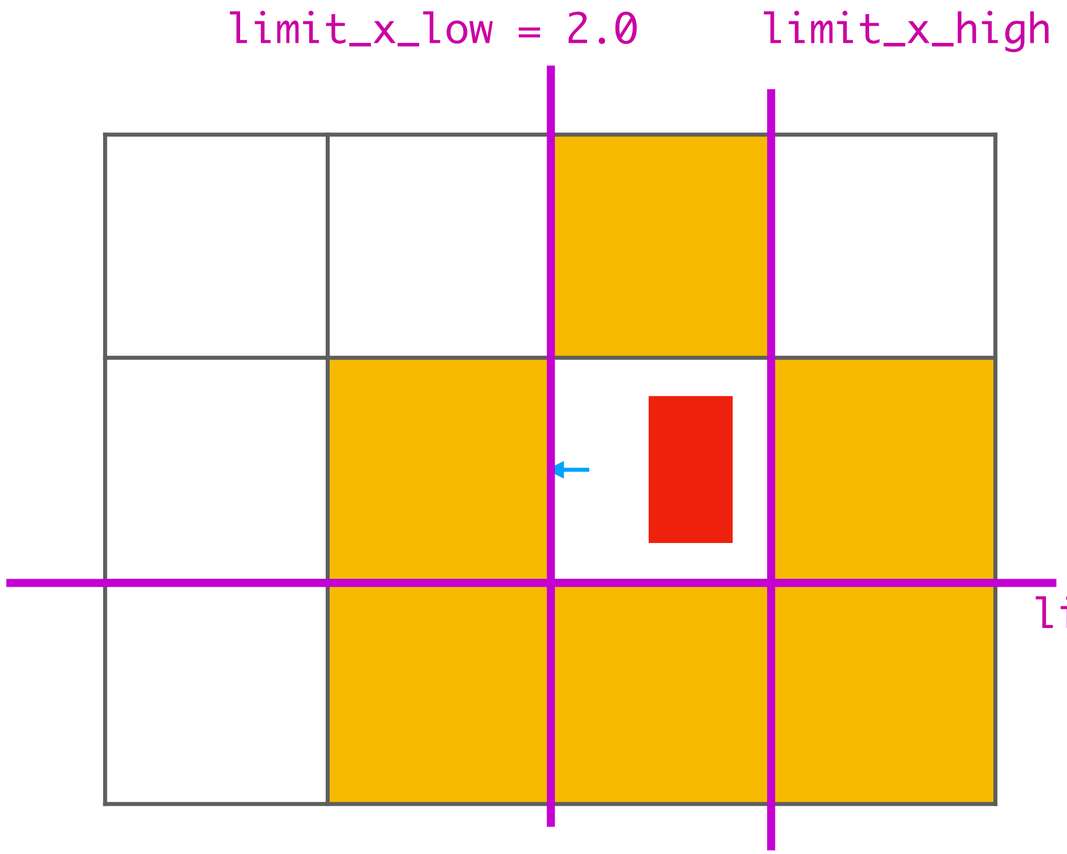
- Until now, we just disallow the player to move to a solid tile and teleport it back to its previous position. Ideally we would block the player exactly at the edge.



- We can do this by limiting how far the player can move in each direction (limit_x_low is the left limit for example).

- To set the limit, we check the tiles to the left, right, top and bottom of the "current" tile. If there's a solid tile there we set the limit to the relevant border of that tile.

Width and height



- Once the single point movement is nailed down, we can easily add support for width and height. When limiting the movement in each direction we just add an extra distance proportional to the size of the player.