

The Maze Game – OOP Review

Step 1: Highlight all nouns (green underlined), adjectives/properties/numerals referring to them (bold orange) and verbs (italics blue). Those not important in a sentence can be ignored.

- **6** player *run* around in a **2D** maze, as **2** teams of **3**.
- They can *change* the team they are on at any time by *stepping* onto a button in the **middle** of the maze. *Stepping* on the button *swaps* the player's team, and has an initial **10** frames cooldown, *increasing* by **2** every time a player *uses* the button.
- player can *tag* opponents by *doing* a “tag” action **when next to 1** opponent. Tagged players *are out* of the game and *they lose*. If **2** players *tag* each other in the same game tick, *they* both *lose*. If player A *tags* player B in the same tick when *they* themselves *were tagged*, both player A and B *lose*. Tag actions *are invalid* if a player *is adjacent* to **more than 1** opponent (adjacency *does not consider* diagonals).
- The **last** player standing *wins*.
- The game also ends **after 1000 game** ticks. If multiple players from the same team *are alive* at the end, but only **1** player from the **other** team, the **1 solo** player *wins* and everyone else *loses*. If multiple players from both teams *are still alive*, everyone *loses*. If **2** player *are alive at the end* and on opposite teams, they *tie*.

The Maze Game – OOP Review

Step 2: remove all other unimportant words and remove duplicates (or those with similar meaning e.g. change team/swap team here). All nouns in singular form. Can structure things a little for clarity:

- player -> 6 run (in) 2D maze
- player -> 2 team of 3
- player -> swap team
- player -> step on button
- button -> (middle) maze
- button -> press to swap player team
- button -> cooldown, increase by 2 (when used)
- player -> tag player (opponent) when next to 1 opponent [+other rules]
- player -> lose when tagged
- player -> win if last at the end
- game -> end after 1000 ticks [+other rules for player win conditions]

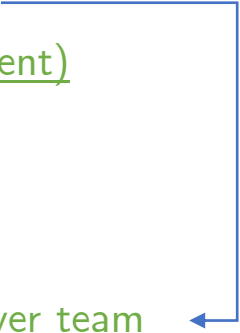
The Maze Game – OOP Review

Step 3: for simplification, remove all orange items. These are important for coding the problems, but not for an abstract representation of the OOP problem. Group bullet points by noun on the left side.

- player -> *run* (in) maze
- player -> team
- player -> *swap* team
- player -> *step* on button
- player -> *tag* player (opponent)
- player -> *lose*
- player -> *win*
- button -> maze
- button -> *press* to *swap* player team
- button -> cooldown, *increase* cooldown
- game -> *end*

The Maze Game – OOP Review

Step 4: Group all elements corresponding to one noun on the left (an object!) under the same bullet points, removing any duplicate properties or logic

- maze
 - player
 - *run* (in) maze
 - team
 - *swap* team
 - ~~*step on* button~~
 - *tag* player (opponent)
 - *lose, win*
 - button
 - (in) maze
 - *press to swap* player team
 - cooldown
 - *increase* cooldown
 - game
 - *end*
- 

The Maze Game – OOP Review

Step 5: Find any implied properties currently not explicitly stated and add them. Some things might need to be renamed as well.

- maze
- player
 - *run* (in) maze -> *changePosition*
 - position (in) maze ←
 - team
 - *swap* team
 - *tag* player (opponent)
 - *lose, win* -> *changeWinStatus*
 - winStatus ←
- button
 - position (in) maze ←
 - *press* to *swap* player team
 - cooldown
 - *increase* cooldown
- game
 - *end*

The Maze Game – OOP Review

Step 6: Group the green items in the sub-bullet points (properties/variables) and the blues (behaviours/methods)

- maze
- player
 - position (in) maze
 - team
 - winStatus
 - *swap* team
 - *tag* player (opponent)
 - *changeWinStatus*
 - *changePosition*
- button
 - position (in) maze
 - cooldown
 - *increase* cooldown
 - *press* to *swap* player team
- game
 - *end*

The Maze Game – OOP Review

Step 7: Rewrite some of the names for clarity. For methods, any green elements left are arguments, or properties modified (can be removed and just point to with an arrow). If other methods are referred to within other methods, they can also be removed and pointed to with an arrow.

- maze
 - player
 - position
 - team
 - winStatus
 - *swapTeam* ←
 - *tag (player=opponent)*
 - *changeWinStatus*
 - *changePosition*
 - button
 - position
 - cooldown ←
 - *increase* ←
 - *press* ←
 - game
 - *end*
- 

The Maze Game – OOP Review

That's it! Other interactions may be represented as well (e.g. player pressed button), but these are the key elements for abstracting this problem with objects and their properties and behaviours. For implementation, we'll need to bring back some of the elements removed earlier (in orange), but this gives us an idea of what happens in the program.

- maze
 - player
 - position
 - team
 - winStatus
 - *swapTeam*
 - *tag (player=opponent)*
 - *changeWinStatus*
 - *changePosition*
 - button
 - position
 - cooldown
 - *increase*
 - *press*
 - game
 - *end*
- 