Game Option and Just

I am making a puzzle maze game because I can reuse one class for the walls for all the mazes.

The game needs to have multiple mazes which increase in difficulty (levels). The game needs to include OOP elements and abstraction. The target audience is young gamers.

I will make an abstract class for a wall that can interact with other walls. I will then use inheritance to make a “wall” be able to be controlled by keys (player), and a “wall” that chases the player through the maze (enemy).

The objective of the game is to go through three levels with different levels of difficulty. The 1st maze will be simple and only has walls blocking the player. The 2nd level will have some boxes blocking the player. The 3rd level will have an enemy chasing the player through a maze. If making an ai is too hard/time consuming, I will instead make a moving object that players have to dodge, and the door class will use a different move method. Below is an example of maze 3:

Red is the player, green is the enemy

Class diagram

|  |
| --- |
| Class name:  player |
| Attributes:  Speed(int) |
| Methods:  \_\_init\_\_(speed, x\_position , y\_position, width, length)  player\_move()  draw() |

|  |
| --- |
| Class name:  wall |
| Attributes:  x\_position(int)  y\_position(int)  width(int)  length(int) |
| Methods:  \_\_init\_\_(x\_position, y\_position, width, length)  draw()  check\_for\_collision(other)  get\_rect() |

|  |
| --- |
| Class name:  enemy |
| Attributes:  Speed(int) |
| Methods:  \_\_init\_\_(speed, x\_position, y\_position, width, length)  ai\_move()  draw() |

|  |
| --- |
| Class name:  Pushable\_block |
| Attributes:  Is\_pushed(boolean) |
| Methods:  Push(direction, sprites)  #sprites is the list of things the block can collide with/be pushed by |

The player, enemy, and pushable\_block classes inherit from the wall class the following methods:  
draw(), check\_for\_collision(other), get\_rect(), \_\_init\_\_()

The draw and get\_rect methods are meant for creating the sprite, check\_for\_collision method checks for if a sprite has collided with another sprite.

The player\_move method overrides the \_\_init\_\_ method and makes the code redraw the player sprite. The ai\_move method overrides the \_\_init\_\_ method and makes the code redraw the enemy sprite. The push method overrides the \_\_init\_\_ method and makes the code redraw the block sprite after it has moved in the direction an enemy or player sprite has pushed it in. the check\_for\_collision method checks if any sprite has collided with another and stops the sprites if they do.