|  |  |
| --- | --- |
| **PLAYER CLASS** | |
| **Properties:** | |
| Player\_index - integer | The players unique index in turn order. 0 is player 1, 1 is player 2 etc |
| Position - integer | The space the player is currently on |
| Automated - bool | The player is a player agent. See lower down. |
| Money - integer | The amount of money the player has |
| Passed\_go - bool | If the player has passed go |
| GOOJ\_cards – bool array | Array of the get out of jail cards the player has. True is an opportunity knock card, False is potluck |
| In\_jail - bool | Is in jail |
| Jail\_turns - int | The amount of turns the player has been in jail. Resets when they leave |
| Owned\_properties – array | An array of the properties the players own. This will store the objects directly. |
| **Methods:** | |
| \_\_init\_\_  Index – integer  mainWindow | Saves the players index and stores the main window class so that it can access it later. |
| Move  Roll - integer | Will recursively move the player’s space one by one with a 200ms delay. Will move the icon on the main window and also handle passing go. |
| Go\_to  Position - int | Moves the player directly to a space without the delay between movement. Still handles passing go. |
| Attempt pay  Amount – integer  RETURN integer | The method takes the amount/price of the transaction. The return value is how much the player can pay. If the player has enough, the full cost is returned, and the money is decreased. If they don’t, however much they can pay is returned, and the player is bankrupt. |
| Pay\_player  To – player  Amount - integer | Uses the attempt pay method to pay a designated player. |
| Property purchase  Space - space | Assumes that the player already has enough money. Removes the correct amount of money from the player, and sets the owner to the player, and appends the property to owned properties |
| Go\_to\_jail | Moves the player to jail, and sets the position of the icon. |

Although the player agent is not defined as its own class, it will be represented here as such since it makes more sense. For more information, see the player agent write-up.

|  |  |
| --- | --- |
| **PLAYER AGENT** | |
| **Properties:** | |
| Decision chance – int | The chance that the agent will buy something. The higher the value, the more likely |
| Poor decision chance - integer | The chance that the agent will buy something within the poor threshold. The higher the value, the more likely |
| Poor threshold – float | the difference between the cost and the amount of money that the poor decision is used. |
| Absolute\_no\_dist – integer | The amount of remaining money after a purchase to make the agent not buy something 100% of the time |
| group\_preference – float | How much more likely an agent is to buy something if thay have a property of the same suit. |
| Jail\_benefit – integer | How much the agent wants to leave jail. |
| **Methods:** | |
| Setup\_agent | Sets up the variables to create the agents “personality”. |
| Decide\_property  \_benefit  Property – property  RETURN - float | Takes a propery as an argument and will use the properties benefit along with the group preference to decide how beneficial the space is. This benefit value can be used in the decision later on. |
| Agent\_decision  Benefit – float  Cost – int  RETURN - bool | Uses the properties to decide whether the purchase should be completed based on how much money they have, the chances defined earlier, and the benefit given to the method. For more information, see the write-up. |

|  |  |
| --- | --- |
| **SPACE CLASS** | |
| **Properties:** | |
| Space\_index – integer | The index of the space. Can refer to the position on the board, so 0 is go etc. |
| Name - string | The space’s name |
| Action – integer | the action index – used to determine what to do when landing on this space. |
| Is\_property – bool | The property can be purchased |
| Position – int array | Acts as a Vector2 position for the player’s icon to move to |
| Owner - player | The player that owns the space |
| Group – integer | The suit/colour of the space. Includes stations and utilities |
| Cost – integer | The cost to buy the property |
| Benefit – float | The benefit to buying – used for player agents |
| Current\_level – integer | The level – how many houses the space has currently |
| House\_cost - integer | The cost to buy a house |
| Unlevelled\_rent – int | The rent when there are no houses |
| Rent – integer array | The rent when its in a group. 0 is no houses, 1 is one house etc. |
| **Methods:** | |
| \_\_init\_\_  Space - int | Sets up the space to fit the index. It will read data from the space array/dictionary, and copy the data to itself to reduce the read calls, and make the data more useable |
| Get\_price  Roll - integer | Returns the rent of the space based off of the suit, housing, etc. the roll passed is for the utilities. |