

Testing all the functions in main.js

Testing createDeck()

- this function should create a deck to check this we will call the function and output all the tiles
- the program should output all the tiles in order

```
C:\windows\system32\cmd.exe
{ Value: '7', colour: 'Yellow', Weight: 7 }
{ Value: '7', colour: 'Blue', Weight: 7 }
{ Value: '7', colour: 'Clubs', Weight: 7 }
{ Value: '8', colour: 'Red', Weight: 8 }
{ Value: '8', colour: 'Yellow', Weight: 8 }
{ Value: '8', colour: 'Blue', Weight: 8 }
{ Value: '8', colour: 'Clubs', Weight: 8 }
{ Value: '9', colour: 'Red', Weight: 9 }
{ Value: '9', colour: 'Yellow', Weight: 9 }
{ Value: '9', colour: 'Blue', Weight: 9 }
{ Value: '9', colour: 'Clubs', Weight: 9 }
{ Value: '10', colour: 'Red', Weight: 10 }
{ Value: '10', colour: 'Yellow', Weight: 10 }
{ Value: '10', colour: 'Blue', Weight: 10 }
{ Value: '10', colour: 'Clubs', Weight: 10 }
{ Value: '11', colour: 'Red', Weight: 11 }
{ Value: '11', colour: 'Yellow', Weight: 11 }
{ Value: '11', colour: 'Blue', Weight: 11 }
{ Value: '11', colour: 'Clubs', Weight: 11 }
{ Value: '12', colour: 'Red', Weight: 12 }
{ Value: '12', colour: 'Yellow', Weight: 12 }
{ Value: '12', colour: 'Blue', Weight: 12 }
{ Value: '12', colour: 'Clubs', Weight: 12 }
{ Value: '13', colour: 'Red', Weight: 13 }
{ Value: '13', colour: 'Yellow', Weight: 13 }
{ Value: '13', colour: 'Blue', Weight: 13 }
{ Value: '13', colour: 'Clubs', Weight: 13 }
```

It works

Testing createPlayers(num)

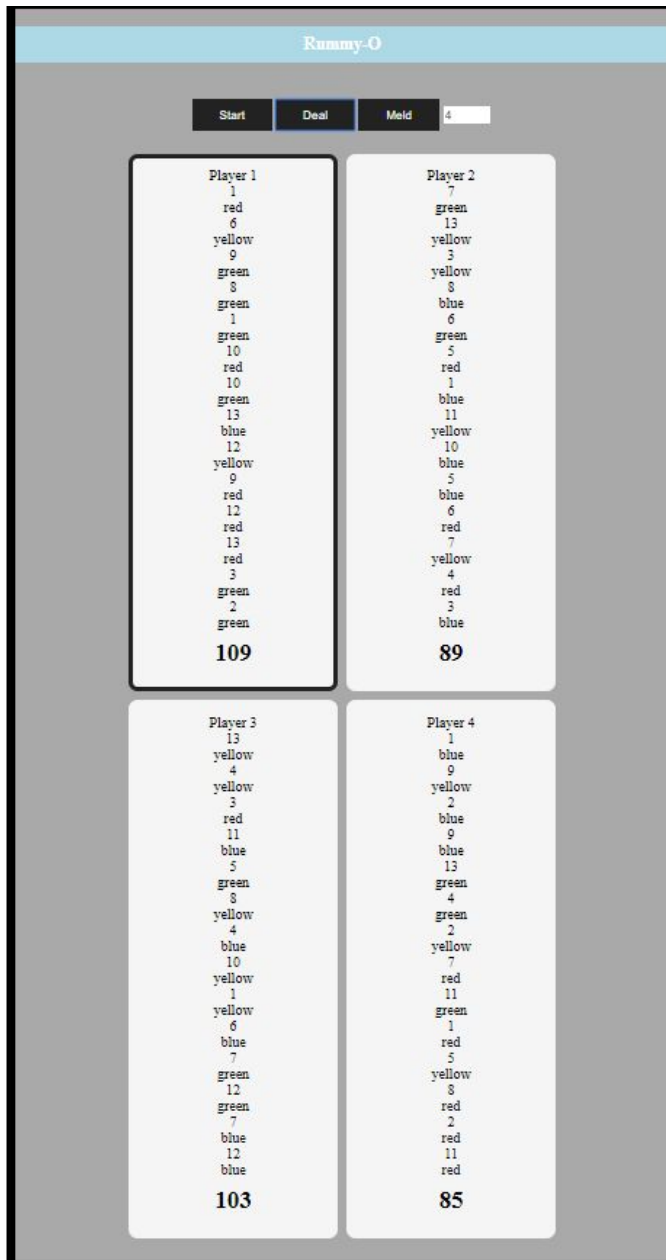
- this function should create the array of player given a number of players
- for the sake of the test we shall call the function with 4 as a parameter and output each player
- the result should be 4 players in the array

```
C:\Users\100703933\Desktop\Code>node t
{ Name: 'Player 1', ID: 1, Points: 0, Hand: [] },
{ Name: 'Player 2', ID: 2, Points: 0, Hand: [] },
{ Name: 'Player 3', ID: 3, Points: 0, Hand: [] },
{ Name: 'Player 4', ID: 4, Points: 0, Hand: [] }
```

Success

### Testing createPlayersUI()

- this is a function that creates a UI with players hands, points, and their id on an html page
- for this test we will run the html page using node.js create the players and deal the cards to each player
- This should all be visible as a UI on the HTML page separately for each player
- for the sake of testing there will be 4 players



The UI displays which passes part of the test however there should only be 1 UI per page

## Testing shuffle()

- this one is self explanatory it should take an existing deck and shuffle it
- to make this happen we will create a deck then call shuffle
- the deck will be outputted and should appear in a random order

C:\windows\system32\cmd.exe

```
{ Value: '12', colour: 'Yellow', Weight: 12 },
{ Value: '4', colour: 'Yellow', Weight: 4 },
{ Value: '3', colour: 'Blue', Weight: 3 },
{ Value: '4', colour: 'Blue', Weight: 4 },
{ Value: '13', colour: 'Red', Weight: 13 },
{ Value: '2', colour: 'Red', Weight: 2 },
{ Value: '9', colour: 'Red', Weight: 9 },
{ Value: '9', colour: 'Blue', Weight: 9 },
{ Value: '2', colour: 'Blue', Weight: 2 },
{ Value: '8', colour: 'Clubs', Weight: 8 },
{ Value: '2', colour: 'Yellow', Weight: 2 },
{ Value: '12', colour: 'Blue', Weight: 12 },
{ Value: '2', colour: 'Clubs', Weight: 2 },
{ Value: '9', colour: 'Yellow', Weight: 9 },
{ Value: '5', colour: 'Clubs', Weight: 5 },
{ Value: '6', colour: 'Clubs', Weight: 6 },
{ Value: '6', colour: 'Red', Weight: 6 },
{ Value: '5', colour: 'Red', Weight: 5 },
{ Value: '5', colour: 'Blue', Weight: 5 },
{ Value: '9', colour: 'Clubs', Weight: 9 },
{ Value: '10', colour: 'Yellow', Weight: 10 },
{ Value: '3', colour: 'Clubs', Weight: 3 },
{ Value: '10', colour: 'Clubs', Weight: 10 },
{ Value: '11', colour: 'Blue', Weight: 11 },
{ Value: '7', colour: 'Blue', Weight: 7 },
{ Value: '7', colour: 'Red', Weight: 7 },
{ Value: '8', colour: 'Red', Weight: 8 },
{ Value: '1', colour: 'Clubs', Weight: 1 },
{ Value: '6', colour: 'Blue', Weight: 6 },
{ Value: '8', colour: 'Blue', Weight: 8 },
{ Value: '7', colour: 'Yellow', Weight: 7 },
{ Value: '5', colour: 'Yellow', Weight: 5 },
{ Value: '13', colour: 'Yellow', Weight: 13 },
{ Value: '1', colour: 'Red', Weight: 1 },
{ Value: '6', colour: 'Yellow', Weight: 6 },
{ Value: '11', colour: 'Clubs', Weight: 11 },
{ Value: '7', colour: 'Clubs', Weight: 7 },
{ Value: '11', colour: 'Yellow', Weight: 11 },
{ Value: '11', colour: 'Red', Weight: 11 },
{ Value: '3', colour: 'Yellow', Weight: 3 },
{ Value: '10', colour: 'Red', Weight: 10 },
{ Value: '1', colour: 'Blue', Weight: 1 },
{ Value: '4', colour: 'Clubs', Weight: 4 },
{ Value: '10', colour: 'Blue', Weight: 10 },
{ Value: '13', colour: 'Blue', Weight: 13 },
{ Value: '12', colour: 'Clubs', Weight: 12 },
{ Value: '8', colour: 'Yellow', Weight: 8 },
{ Value: '12', colour: 'Red', Weight: 12 },
{ Value: '4', colour: 'Red', Weight: 4 },
{ Value: '1', colour: 'Yellow', Weight: 1 }
```

The order is random and no tile has been lost or duplicated

Testing startGame()

-this function is more of a boot function that calls all relevant functions when called

-this includes:

```
createDeck();  
shuffle();  
createPlayers(num);  
createPlayersUI();  
dealHands(1);  
document.getElementById('player_' + currentPlayer).classList.add('active');  
check()
```

-to test such a function the criteria is to have all the functions running and displaying appropriate output on the html page

-this was already shown in created player UI

The function passes the test

Testing dealHands(k)

-This function should k cards to each player

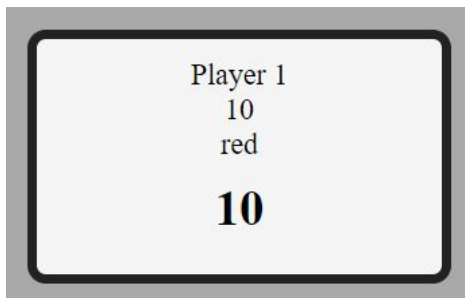
- to test this we will create a deck and create 3 player

-we will then call the function with k being 5

-we should see 3 players with 5 cards each as output

Test failed relies on html element

-new test run deal hand from the html



A card is dealt

A hand is dealt



New test success

Testing check()

- this checks the points of 2 players a simple if statement
- this can be tested by observing any output from within the if statement
- a list of players will be created and check will be called
- the output should be true

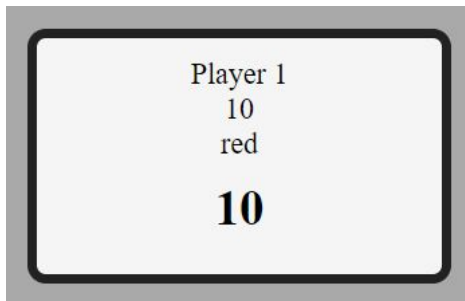
```
C:\Users\100703933\Desktop\Code>node t  
true
```

Test passes

Testing rendertile(title, player)

-this displays a tile from a players hand

-there has already been instances of this working while testing the deal hand function



Getting UI is a function that no longer serves a purpose due to time constraints and will not be tested

Testing getPoints(player)

-this takes all the cards from a players hand and adds them up

-unfortunately this is another function that is tethered to the html

-but there is already an isolated case of it working in the display of deal hand

Test is successful

Update points is another function that will be scrapped no point in testing it

Testing deal()

-deal deals 14 cards to each player in the game

-again it is dependent on the html file

-again it has been working in an isolated environment in the player UI and dealHands

Test is successful

eliminate\_tiles and next\_turn are both incomplete functions that will not run and therefore will not be tested

For the last 3 functions none of them work in html or otherwise due to logical error

#### Testing initialMelds()

- this function should only activate if the player has not made a successful meld before
- it should call the meld function and make sure the melds being made are over 25 points
- if not it denies the meld that is trying to be made
- to test this a hand of 14 cards is made and the the user is then asked how many cards will be in the meld and what they are
- if the melds are valid an surpass 25 a flag would signify that they made a meld
- if it was not valid there would be something stating that there is something wrong

#### Testing switchActive()

- this function gives a status to a player of active if it is their turn so turns can be played out
- this is more of an interface function and can be observed rather than tested for

#### Testing end()

- end is a function only containing an if and else statement
- it compares points to see who the winner is
- to test it define points in the players list and call the function end
- the person with more points has to get the victory message otherwise it would fail the test