

# **Chess Notes**

James Otto

11/15/22

# Table of contents

<b>Preface</b>	<b>3</b>
<b>I Openings</b>	<b>4</b>
<b>1 The Scotch</b>	<b>5</b>
1.1 Introduction . . . . .	8
1.2 Main Line (4.Nxd4) . . . . .	8
Classical Variation (4...Bc5) . . . . .	8
Steinitz Variation (4...Qh4!?) . . . . .	8
1.3 Scotch Gambit (4.Bc4) . . . . .	8
<b>2 Fried Liver Attack</b>	<b>9</b>
2.1 Introduction . . . . .	12
2.2 Continuing . . . . .	12
<b>Appendices</b>	<b>12</b>
<b>References</b>	<b>13</b>

# Preface

These are my notes on various topics in chess, aggregated from a variety of sources.

This is a Quarto book. To learn more about Quarto books visit <https://quarto.org/docs/books>.

# **Part I**

## **Openings**

# 1 The Scotch

```
board_setup = function(data, title) {

  var game = data.find(d => d.title === title)

  // if there's just one variation, need to put in array
  var variations = game.variations

  var pos = game.start - 1
  var moved = false

  var onDragMove = function() {
    moved = true

    d3.select('#reset-' + title)
      .classed('btn-outline-secondary', false)
      .classed('btn-success', true)

    d3.select('#analyze-' + title).property("disabled", true)
    d3.select('#prev-' + title).property("disabled", true)
    d3.select('#next-' + title).property("disabled", true)
  }

  // Note: boolean arrays don't survive transpose(), need to use characters ("show" and "h
  var board = Chessboard(title, {showNotation: gamenotation === 'show', position: game.po

  d3.select('#prev-' + title).property("disabled", pos == 0)
  d3.select('#next-' + title).property("disabled", pos == game.positions.length - 1)

  $('#next-' + title).on('click', function () {
    if (pos < game.positions.length - 1) {
      pos = pos + 1
      board.position(game.positions[pos])
    }
  })
}
```

```

d3.select('#analyze-' + title).property("disabled", false)
d3.select('#prev-' + title).property("disabled", false)
d3.select('#next-' + title).property("disabled", pos == game.positions.length - 1)
})

$('#prev-' + title).on('click', function () {
  if (pos > 0) {
    pos = pos - 1
    board.position(game.positions[pos])
  }

  d3.select('#analyze-' + title).property("disabled", false)
  d3.select('#prev-' + title).property("disabled", pos == 0)
  d3.select('#next-' + title).property("disabled", false)
})

$('#reset-' + title).on('click', function() {

  // if pieces have not been moved, reset to beginning of line
  if (!moved) {

    // if looking at a variation, reset to main line
    if (!(variations === undefined) && variations.includes(game.title)) {
      game = data.find(d => d.title === title)

      d3.select('#reset-' + title)
        .classed('btn-primary', false)
        .classed('btn-outline-secondary', true)
    }

    pos = game.start - 1

  }

  board.position(game.positions[pos])

  d3.select('#analyze-' + title).property("disabled", false)
  d3.select('#prev-' + title).property("disabled", pos == 0)
  d3.select('#next-' + title).property("disabled", pos == game.positions.length - 1)

  d3.select('#reset-' + title)

```

```

        // .classed('btn-outline-secondary', true)
        .classed('btn-success', false)

        moved = false
    })

    $('#analyze-' + title).on('click', function() {
        window.open(game.link[pos], '_blank');
    })

    // support for variations:

    if (!(variations === undefined)) {

        if (!(Array.isArray(variations))) {
            variations = [variations]
        }

        for (let k = 0; k < variations.length; k++) {
            $('#' + variations[k]).on('click', function() {
                game = data.find(d => d.title === variations[k])
                pos = game.start - 1
                board.position(game.positions[pos])

                d3.select('#analyze-' + title).property("disabled", false)
                d3.select('#prev-' + title).property("disabled", pos == 0)
                d3.select('#next-' + title).property("disabled", pos == game.positions.length - 1)

                d3.select('#reset-' + title)
                    .classed('btn-outline-secondary', false)
                    .classed('btn-success', false)
                    .classed('btn-primary', true)
            })
        }
    }
}

```

```
scotch = transpose(scotch_t)

scotch.filter(d => d.type === 'main')
  .map(d => board_setup(scotch, d.title))
```

## 1.1 Introduction

## 1.2 Main Line (4.Nxd4)

### Classical Variation (4...Bc5)

After 4.Nxd4 Bc5 there are three main lines:

5.Nxc6

5.Be3

5.Nb3

### Steinitz Variation (4...Qh4!?)

- Black wins a pawn here but gives white a lead in development and ends up with their king in a slightly awkward spot.

## 1.3 Scotch Gambit (4.Bc4)

This line transitions into the Italian Game

Black can further accept the gambit with 5...dxc3

4...Bb4+ is the London Defense



## 2 Fried Liver Attack

```
board_setup = function(data, title) {

  var game = data.find(d => d.title === title)

  // if there's just one variation, need to put in array
  var variations = game.variations

  var pos = game.start - 1
  var moved = false

  var onDragMove = function() {
    moved = true

    d3.select('#reset-' + title)
      .classed('btn-outline-secondary', false)
      .classed('btn-success', true)

    d3.select('#analyze-' + title).property("disabled", true)
    d3.select('#prev-' + title).property("disabled", true)
    d3.select('#next-' + title).property("disabled", true)
  }

  // Note: boolean arrays don't survive transpose(), need to use characters ("show" and "h
  var board = Chessboard(title, {showNotation: gamenotation === 'show', position: game.po

  d3.select('#prev-' + title).property("disabled", pos == 0)
  d3.select('#next-' + title).property("disabled", pos == game.positions.length - 1)

  $('#next-' + title).on('click', function () {
    if (pos < game.positions.length - 1) {
      pos = pos + 1
      board.position(game.positions[pos])
    }
  })
}
```

```

d3.select('#analyze-' + title).property("disabled", false)
d3.select('#prev-' + title).property("disabled", false)
d3.select('#next-' + title).property("disabled", pos == game.positions.length - 1)
})

$('#prev-' + title).on('click', function () {
  if (pos > 0) {
    pos = pos - 1
    board.position(game.positions[pos])
  }

  d3.select('#analyze-' + title).property("disabled", false)
  d3.select('#prev-' + title).property("disabled", pos == 0)
  d3.select('#next-' + title).property("disabled", false)
})

$('#reset-' + title).on('click', function() {

  // if pieces have not been moved, reset to beginning of line
  if (!moved) {

    // if looking at a variation, reset to main line
    if (!(variations === undefined) && variations.includes(game.title)) {
      game = data.find(d => d.title === title)

      d3.select('#reset-' + title)
        .classed('btn-primary', false)
        .classed('btn-outline-secondary', true)
    }

    pos = game.start - 1

  }

  board.position(game.positions[pos])

  d3.select('#analyze-' + title).property("disabled", false)
  d3.select('#prev-' + title).property("disabled", pos == 0)
  d3.select('#next-' + title).property("disabled", pos == game.positions.length - 1)

  d3.select('#reset-' + title)

```

```

        // .classed('btn-outline-secondary', true)
        .classed('btn-success', false)

        moved = false
    })

    $('#analyze-' + title).on('click', function() {
        window.open(game.link[pos], '_blank');
    })

    // support for variations:

    if (!(variations === undefined)) {

        if (!(Array.isArray(variations))) {
            variations = [variations]
        }

        for (let k = 0; k < variations.length; k++) {
            $('#' + variations[k]).on('click', function() {
                game = data.find(d => d.title === variations[k])
                pos = game.start - 1
                board.position(game.positions[pos])

                d3.select('#analyze-' + title).property("disabled", false)
                d3.select('#prev-' + title).property("disabled", pos == 0)
                d3.select('#next-' + title).property("disabled", pos == game.positions.length - 1)

                d3.select('#reset-' + title)
                    .classed('btn-outline-secondary', false)
                    .classed('btn-success', false)
                    .classed('btn-primary', true)
            })
        }
    }
}

```

```
fried_liver = transpose(fried_liver_t)

fried_liver.map(d => board_setup(fried_liver, d.title))
```

## 2.1 Introduction

## 2.2 Continuing

## References