Screen:

* Size of screen
* Background colour of screen
* Size of pieces
* Colour of pieces
* Size of board
* Colour of board
* Size of board holes
* Number of rows and columns on board
* How will the start screen look like?
* What will show up on the screen when a player wins or loses?
* Will there be animation for a piece “falling” into the correct slot after a player makes a move?
* When a player wants to make a move, where will the piece needed to make a move appear on the screen? Will the piece be the cursor? Or will the piece to be used appear on some area of the screen?

Game Play:

* Two Players. Each player takes turns to make a move until the board has four pieces of the same colour in a row, column or diagonal. The player that does this will win.
* After every move, check if a player has won.
* A move is a player clicking on the column they want their piece to drop into
* Should the player click on the board or slightly above the board to make a move? Should the player drag and drop the piece instead?

How the Work Break Down Structure Could Look Like:

**[Connect-Four]**

**[1.1Board - Member Assigned to Task]**

Create a window that shows the board and the positions of the stones on the

Board.

.

1.GUI

**[1.2Stones - Member Assigned to Task]**

Game is able to show the stones on the board.

**[2.1algorithm - Member Assigned to Task]**

Game is able to decide the winner based on the algorithm .

2.Playing

**[2.2players - Member Assigned to Task]**

There are 2 players, players are able to make a move on the board.

**[2.3Game Roles - Member Assigned to Task]**

A position in the board can be empty or filled, if the postion is empty no position above it can be filled.

**Links I found Useful/Was Inspired By**

------------------------------------------------------------------------------------------------------------------------------------------

Using Headings Effectively: Parallel Structure

<http://colelearning.net/rw_wb/module4/page14.html>

In the “organization and structure” part of the grading scheme, we must use “parallel structure” for the headings.

------------------------------------------------------------------------------------------------------------------------------------------

A Beginner-Friendly Guide to Work Breakdown Structures (WBS)

<https://www.workamajig.com/blog/guide-to-work-breakdown-structures-wbs>

Link has some nice examples of Work Breakdown Structures. Gives a detailed explanation of how to write tasks as well.

------------------------------------------------------------------------------------------------------------------------------------------

Tutorial 2: Connect 4 (alpha 0.2.3)

<http://jogre.sourceforge.net/connect4_tutorial_a023.htm#mvc>

Shows how to implement “turn-based game of Connect 4” using the MVC (Model-View-Controller) design pattern in java.

------------------------------------------------------------------------------------------------------------------------------------------

Model View Controller Pattern

<https://www.tutorialspoint.com/python_design_patterns/python_design_patterns_model_view_controller.htm>

Using MVC may make dividing work easier. However, it may make the implementation of the game somewhat harder.

------------------------------------------------------------------------------------------------------------------------------------------

An Introduction to Scrum (a type of Agile Development)

<https://axiom.utm.utoronto.ca/~207/19f/lectures/scrum/>

This isn’t really necessary now, but it might come in handy later.

------------------------------------------------------------------------------------------------------------------------------------------

Project Charter: Website Redesign

<http://s.casual.pm.s3.amazonaws.com/toolkit/WebsiteDesign.pdf>

This was one of the links given in the project plan handout. I based the project plan template on its style. Feel free to change up the rough template I made.