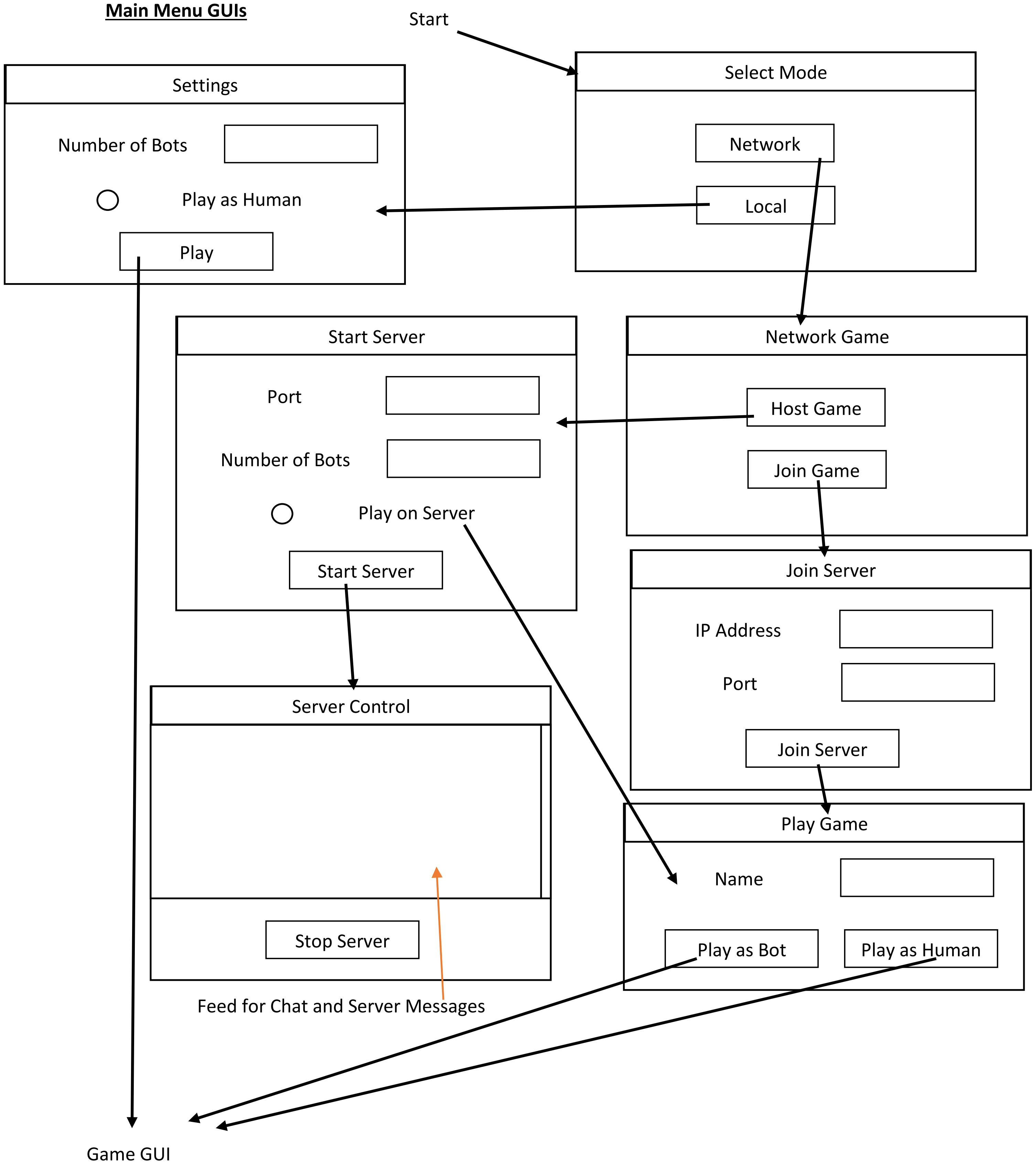
Introduction

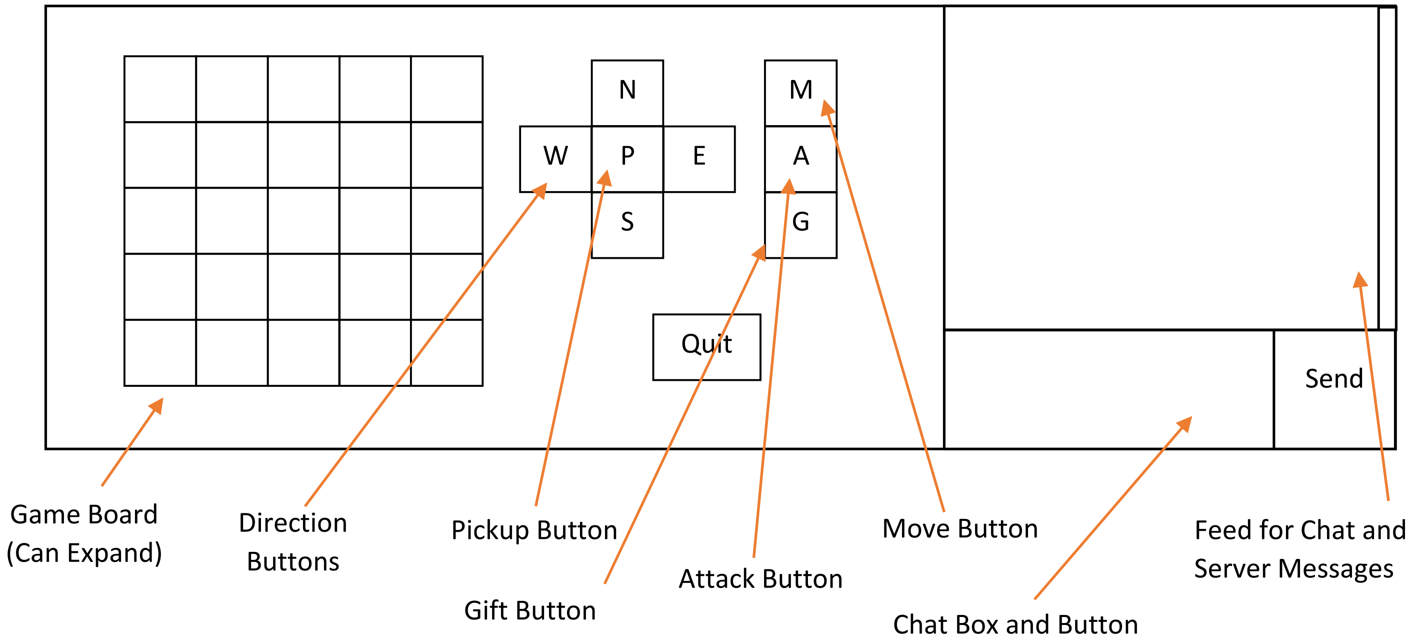
In the following report I shall talk about the structure I intend on my code, I shall firstly talk about GUI design to help me visualise the flow of control in the program and then talk about new classes I shall introduce

Main Menu GUIs

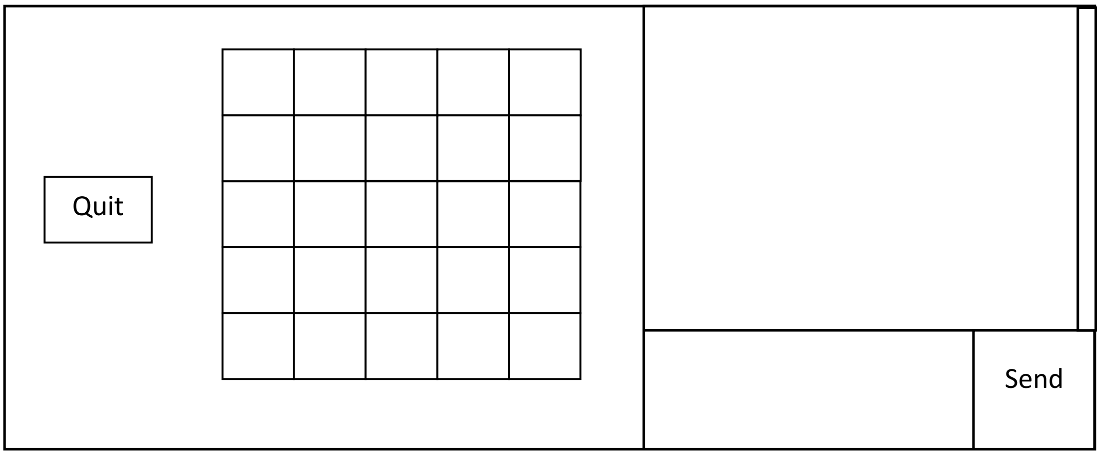


Game GUIs

This is the main GUI for human players



The BOT GUI is slightly adapted from this:



GUI Code Layout

The main idea is to have set classes for each GUI:

- MainMenus
  - Contains the main initial menus for selecting game modes as shown in the first GUI section, this excludes the Server GUI
- MessageFeedGUI

    - Designed to be inherited contains code for creating the scrollable message feed seen on client and server GUIs, it also has code for updating messages on feed
- ServerGUI
    - Inherits from MessageFeedGUI
    - Creates the Server GUI menu as seen in the first GUI section
- PlayerGUI
    - Inherits from MessageFeedGUI
    - Represents a generic player and provides tools for creating and updating the Game Board
- HumanPlayerGUI
    - Inherits from PlayerGUI
    - Provides code for creating the controller buttons and organising the game page as seen in the Game GUIs section
- BotPlayerGUI
    - Inherits from PlayerGUI
    - Organises the page to look like the bot game page from Game GUIs section

MainMenus is not involved as it essentially decides what GUI to run. Another activity of the MainMenus Class is to decide how the GUI sends and recives information, either locally or over a network.

BotPlayerGUI will posses an instance of the class BotLogic which represents the bots decisions through the game.

Communication Classes

These classes exist as a communication layer between the GUI classes and the game. There are two methods of communication here local communication and network communication. The communication layer will be instanciated by main menu and passed and accessed by the Game GUIs. The Game GUIs need to work regardless of the method of communication so a parent class is needed. The communication class will also hold a reference to the the Game GUI in order to update the GUI.

HumanPlayerGUI will send button press responses to the Communication Class to communicate with, BotPlayerGUI will use BotLogic to send commands and any updates will to BotPlayerGUI will be fed through BotLogic to interpret.

The three communication classes are:

- GameCommunicator
    - Abstract Class allowing some shared functionality but all together allows GUI classes to treat communicator classes arbitrarily
- LocalGameCommunicator
    - Inherits GameCommunicator
    - Allows a Local Player or Bot to play the game it does this by calling functions directly in the responder classes
    - Responder classes also directly call output functions in this class to reply
- NetworkGameCommunicator
    - Allows a player or bot to play by sending command strings over a socket for the responder classes to read and reply

So the inheritance diagram goes:

Responder Classes

These are made to deal with commands either locally or other a network. They communicate Directly with GameLogic class which was given in CW2 and modified to add additional functionality.

There will be three classes in this:

- User
    - This is the User class which has modified code from the code given in CW2 it has code that will take strings as commands and use GameLogic to get an appropriate response
    - It is abstract where the output and input responses are left abstract
- LocalUser
    - Inherits user
    - Provides functions to be called by LocalGameCommunicator which simply pass strings in to be intrepreted
    - It also contains a reference of LocalGameCommunicator which it uses to reply to the object
- NetworkUser
    - Inherits user implements runnable
    - Listens to a given socket for string commands which tI interprets and outputs using the same socket, it implements runnable as it will need to block on the socket input in order to read it, so concurrency is needed to allow both an input and output from this class

The remaining classes were all supplied in CW2 and have only been modified to allow multiple players, allow attacking, gifting and death.