

## FILES AFFECTED IN EXTENSIVE FORM GUI SCALING

- **Main.mxml**
  - Main class that directly or indirectly uses all other classes.
  - Controls main screen GUI, button placement, what functionality button clicks provide, etc.
- **Game.as**
  - High level class used to represent general games.
  - Extended to provide the basis for strategic and extensive forms.
  - Contains an instance *Player* variable *\_firstPlayer*.
  - *\_firstPlayer* provides access to the entire linked list of players of a game.
- **Player.as**
  - Stores basic information about a player including name and ID.
  - Contains a *nextPlayer* accessor method which points to the next player in the game (given that such a player exists).
- **ExtensiveForm.as**
  - Handles the lower-end functionality of representing a game tree.
  - Deals with non-GUI elements.
  - Internally represents players, nodes, moves, and information sets.
  - Contains functions relating to perfect recall.
- **Node.as**
  - Basic node class used by *ExtensiveForm.as*.
  - Contains information about other relevant nodes (e.g. father node, brother node, and children).
  - Stores data relating to information set, depth, and moves.
- **TreeGrid.as**
  - An extension of *ExtensiveForm.as* that handles many GUI functions.
  - Manages the colours of nodes, the setup of the default tree, the graphical additions of moves and nodes, and the payoffs at leaf nodes.

- **FileSettings.as**
  - Interacts with *SCodes.as* to fetch values such as default player colours.
  - Adds a layer of abstraction between the raw values found in *SCodes.as*.
- **SCodes.as**
  - Stores the raw values of different settings such as fonts and colours.
  - Specifies default settings (e.g. the default colour for Player 1 is 0xFF000, red).
- **TreeGridActionFactory.as**
  - Controls actions performed in the tree, such as changing payoffs, randomizing payoffs, or making the game zero-sum.
- **PayChangeAction.as**
  - Contains the action used to modify a payoff.
  - Used extensively by *TreeGridActionFactory.as*.
- **AbstractPainter.as**
  - Abstract class storing basic methods for drawing extensive and strategic form games.
- **TreeGridPainter.as**
  - Handles the actual drawing of the full tree
  - Uses data such as player colours from *TreeGrid.as* to correctly paint the tree.
- **TreeGridSetPainter.as**
  - Similar to *TreeGridPainter.as*, but is used exclusively to paint the information sets of a tree.