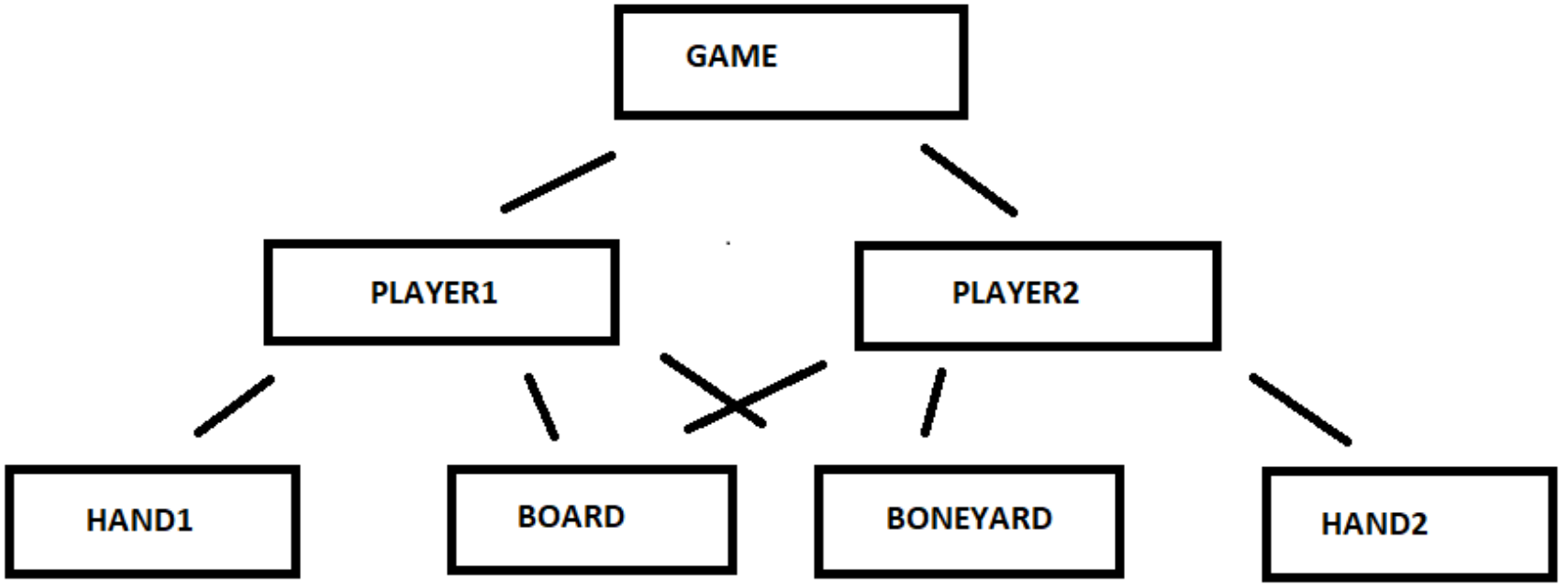
GAME
PLAYER1
PLAYER2
HAND1
BOARD
BONEYARD
HAND2

Tony Nguyen

CS-375

Design sketch stayed the same for version two and three.

Design:

Version3:

Classes / methods

- Game
  - Start() - starts the game.
  - Render() - draws the graphics.
  - HumanPlay()- deals with human player logic.
  - CheckWin() – checks if game is over.
- Boneyard
  - randDraw() – returns random Domino.
  - isEmpty() – checks if boneyard is empty.
- Computer
  - doStrategy() – handles computer play.
- Domino
  - render() – draws a Domino.
- Board
  - playRight() – given Domino make play on the right of the board.
  - playLeft() – given Domino make play on the left of the board.
  - render() – draw board representation onto GUI.
- Hand
  - isEmpty() – checks if hand is empty.
  - render() – draws board representation onto GUI.
- Human
- Player
  - abstract class that contains basic player methods.
- Turn
  - Enum value that represents current turn.
- Value
  - Should represent possible Domino numbers and blank.

Version2:

- Game
  - start() – starts the game.
  - render() – draws the graphics.
  - humanPlay() - deals with human player logic.
  - applyInput() – interprets user input typed from standard in.
  - checkWin() - checks if the game is over.

- Boneyard
    - randDraw() – returns random Domino.
    - isEmpty() – checks if boneyard is empty.
- Computer
    - doStrategy() – handles computer play.
- Domino
    - render() – draws a Domino.
- Board
    - playRight() – given Domino make play on the right of the board.
    - playLeft() – given Domino make play on the left of the board.
    - render() – draw board representation onto GUI.
- Hand
    - isEmpty() – checks if hand is empty.
    - render() – draws board representation onto GUI.
- Human
- Player
    - abstract class that contains basic player methods.
- Turn
    - Enum value that represents current turn.
- Value
    - Should represent possible Domino numbers and blank.

<u>Version1:</u>

- Game
- Board
- Player
- Computer Player
- Boneyard
    - List Arralist<Domino>
    - Boneyard constructor () - creates all possible domino and stores into arraylist.
        - Dominoes
            - 2 values (value)
                - Num1, num2
            - Constructor for dominoes
            - Domino(Value val1, Value val2)
            - getVal1, getVal2
        - Enum class
            - Blank and one - six