Mortal Kombat — Requirements Document

1. General

- Main panel size: 1280 x 720 pixels
- Refresh rate: 60 FPS
- No more than one window

2. Main menu

- **Name:** Text field for username
- **Host**: Starts a LAN server on port 50418
- **Join**: Joins the game hosted by the IP inputted on the adjacent text field
- **Scan**: See Scan (page)
- **Help**: Opens the interactive help demo screen

3. Scan (page)

- Scans the LAN for servers using port 50418 and sends a message to check if they are hosting Mortal Kombat
- Populates a table with a list of servers and corresponding join buttons

4. Chat

- Press ‘c’ to open the chat overlay
- Press ‘Esc’ to close the chat overlay
- Messages are displayed in format ‘**Name (Fighter):** Message’
- See Network Message Design document

5. Fighters

Name	HP	DMG (punch)	DMG (kick)	DMG (uppercut)	Special Move
Scorpion	1000	50	70	140	Spear (150 DMG)
Subzero	1000	50	70	140	Freeze (30

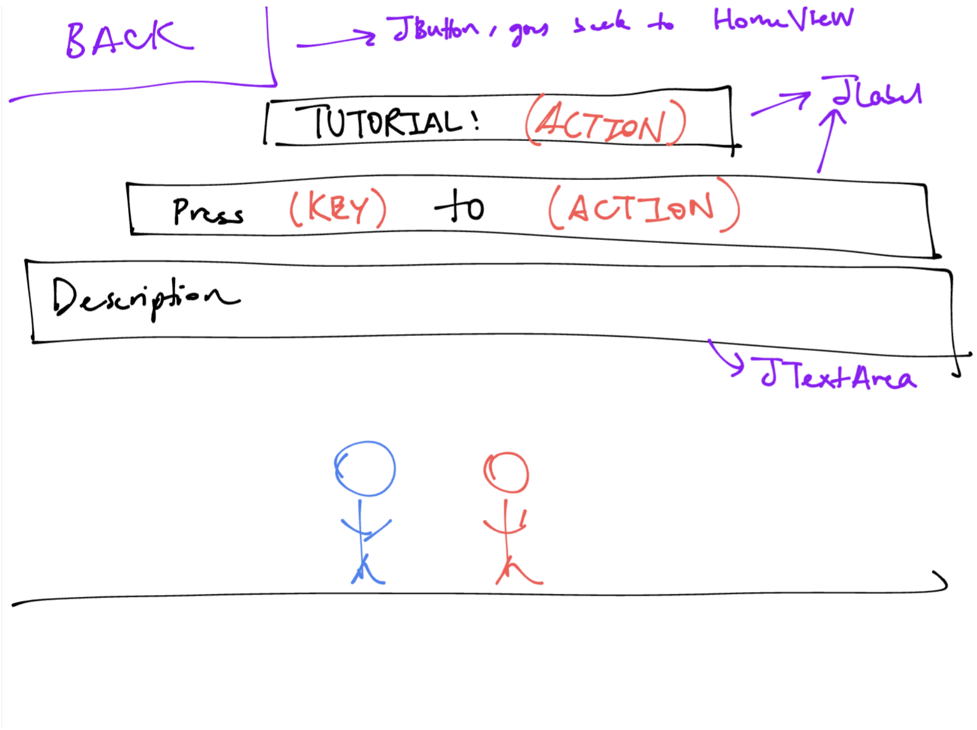
					DMG)

6. Game Mechanics

- When the player takes a hit and is not blocking, they get a stagger animation that pushes them back a couple X. Uppercut additionally launches player into the air. While staggered, players cannot move or counterattack
- Players have reflected animation images if they are switching sides
- Players have WAD-key movement functionality for jump, left, right movement and IJKL-key attack functionality for special move, punch, uppercut, kick, and U-key for blocking
- Ice Ball launched by Sub-Zero's special move freezes the opponent in place for a set amount of time
- Spear launched by Scorpion's special move does large amount of damage and pulls the opponent close to him

7. Help

- Interactive help screen shows players how to move, attack, block, and use their special move by telling them which button to press.
- This interactive screen has 9 frames to display the 8 moves that players have and how to chat



8. Classes

Model

- Fighter
  - Properties
    - HP (int)
    - name (String)
    - idleLeft, idleRight, punchLeft, punchRight, kickLeft, kickRight, uppercutLeft, uppercutRight, specialLeft, specialRight, staggerLeft, staggerRight, blockLeft, blockRight (BufferedImage)
    - WIDTH, HEIGHT (int) – *BufferedImage dimensions*
    - isSpecialBeingUsed (boolean)
  - Methods
    - void punch()
    - void kick()
    - void uppercut()
    - void block()
    - void takeDamage(int DMG)
- <Specific>Fighter extends Fighter
  - Properties
    - String name = <Specific>Fighter.name;
    - specialLastUsed (long) // UNIX timestamp
  - Methods
    - void specialMove()
- Player
  - Properties
    - name (String)
    - fighter (Fighter)
    - movementDisabled (boolean)
    - isKeyPressed (boolean)
    - isAttacking (boolean)
    - isBlocking (boolean)
    - hasRun (false) – *prevents looping issues in paintComponent*
    - currentX (int)
    - currentY (int)
    - currentAnimationImg (BufferedImage)
    - currentAction (String)
  - Methods
    - void chooseFighter(String fighter)
    - void punch(Player opponent)
    - void kick(Player opponent)
    - void uppercut(Player opponent)
    - void specialMove(Player opponent)

        - void block(Player opponent)
        - void blockRelease(Player opponent)
- GameState
    - Properties
        - playerOne (Player)
        - playerTwo (Player)
        - currentPlayer (Player)
        - isGameOver (boolean)
        - ipAddress (String)
        - ssm (SuperSocketMaster)
        - listener (ActionListener)
        - iceBall1, iceBall2 (IceBall)

- IceBall
    - Properties
        - iceBallLeft, iceBallRight (BufferedImage)
        - iceBallX (int)
        - toRender (boolean)
        - WIDTH, HEIGHT (final int)

- Spear
    - Properties
        - spearLeft, spearRight (BufferedImage)
        - spearX (int)
        - toRender (boolean)
        - WIDTH, HEIGHT (final int)

View

- MainView
    - Properties
        - frame (static JFrame)
        - cardLayout (static CardLayout)
        - state (GameState)
        - panel (static JPanel)
        - homeView (HomeView)
        - characterSelectionView (CharacterSelectionView)
        - gameView (GameView)
        - helpView (HelpView)

- HomeView
    - Properties
        - title (JLabel)
        - usernameField (JTextField)
        - labelList (JLabel[]) – *Stores serverList*

        - buttonList (JButton[]) – *Stores buttons to join the serverList servers*
        - serverList (ArrayList<String>)
        - hostButton (JButton)
        - helpButton (JButton)
        - imgBackground (BufferedImage)
        - state (GameState)
        - ms (model.MulticastServer)
        - mc (model.MulticastClient)
        - ipAddress (String)
        - timer (Timer)
        - joinButtonPressed (boolean)
        - hostButtonPressed (boolean)
    - Methods
        - @Override void paintComponent(Graphics g)
        - @Override void actionPerformed(ActionEvent evt)
        - void addServer(String readText)

- CharacterSelectionView
    - Properties
        - screenTitleLabel (JLabel)
        - hostNameLabel (JLabel)
        - clientNameLabel (JLabel)
        - player1ChoiceLabel (JLabel)
        - player2ChoiceLabel (JLabel)
        - startingGameLabel (JLabel)
        - chooseScorpionButton (JButton)
        - chooseSubzeroButton (JButton)
        - state (GameState)
        - gameView (GameView)
        - hostChoice, clientChoice (String)
        - hostReady, clientReady (boolean)
        - imgScorpSelection, imgSubZeroSelection, imgEnlargedScorp, imgEnlargedSub, imgBackground (BufferedImage)
    - Methods
        - @Override paintComponent(Graphics g)
        - void startGameTimer()
        - void drawScorpion(Graphics g, int x, int y)
        - void drawSubZero(Graphics g, int x, int y)
        - @Override void actionPerformed(ActionEvent evt)

- HelpView
    - Properties
        - IFW = JComponent.WHEN_IN_FOCUSED_WINDOW (final int)
        - helpScorpion (ScorpionFighter)

- helpSubzero (SubzeroFighter)
- scorpX, scorpY (int)
- subX, subY (int)
- currentScorpionImg (BufferedImage)
- currentSubzeroImg(BufferedImage)
- tutorialFramePassed (boolean[])
- isDisplayed (boolean)
- timer (Timer)
- title (JLabel)
- instruction (JLabel)
- description (JTextArea)
- backButton (JButton)

- Methods
  - @Override void paintComponent(Graphics g)
  - @Override void actionPerfomed(ActionEvent evt)
  - int checkFrameNumber(boolean[] arr)
  - void dFrame(Graphics g)
  - void aFrame(Graphics g)
  - void wFrame(Graphics g)
  - void jFrame(Graphics g)
  - void lFrame(Graphics g)
  - void kFrame(Graphics g)
  - void uFrame(Graphics g)
  - void iFrame(Graphics g)
  - void altCFrame(Graphics g)
  - void jumpTimer(Graphics g)
  - void attackTimer(String attack, int index)

- GameView
  - Properties
    - state (GameState)
    - hitbox (Hitbox)
    - player1NameLabel (JLabel)
    - player2NameLabel (JLabel)
    - IFW = JComponent.WHEN_IN_FOCUSED_WINDOW (final int)
    - imgBackground (BufferedImage)
    - timer (Timer)
  - Methods
    - @Override void paintComponent(Graphics g)
    - @Override actionPerformed(ActionEvent evt)
    - void forceFrame()
    - void jumpTimer()
    - void attackTimer(String attack)
    - void animateScorpionSpear(Graphics g)

        - void animateIceBall(Graphics g)

- GameOverView
    - Properties
        - state (GameState)
        - title (JLabel)
        - winner (JLabel

- ChatView
    - Properties
        - state (GameState)
        - controller (ChatController)
        - chatHistory (JTextArea)
        - chatBox (JTextField)
        - IFW = JComponent.WHEN_IN_FOCUSED_WINDOW (final int)
    - Methods
        - @Override actionPerformed(ActionEvent e)
        - updateChat (void)

Controller

- Listener (SSM)
    - Properties
        - state (GameState)
        - String msg = state.ssm.sendText();
        - String[] msgComponents = msg.split(“,”);
    - Methods
        - @Override actionPerformed(ActionEvent evt)

- Hitbox
    - Properties
        - state (GameState)
    - Methods
        - String punchHitbox()
        - String kickHitbox()
        - String uppercutHitbox()
        - String IceBallHitbox()
        - String spearHitbox();

- ChatController
    - Properties
        - state (GameState)
        - player (Player)
    - Methods
        - void sendMessage(String message)