Christian Dummer cjdummer@iu.edu
Samantha Sharp sharpsam@iu.edu
Adam Wittenberg adamwitt@iu.edu

Server

<<interface>>
ModelRMI extends java.rmi.remote

+getPlayerPositions(int except) :
HashMap<Integer, Pair<Integer,
Integer>>
+sendPlayerPosition(int playerID, int
positionX, int positionY) : void
+gatherResource(Player player) : void
+updateResources() : void
+getResources() : int[][]
+setResources(int[][] resources) : void

ForagerModel

+ board: int[][];
+ players: ArrayList<Player>

+ForagerModel(int [][] board): void
+getPlayerPositions(int except) :
HashMap<Integer, Pair<Integer,
Integer>>
+sendPlayerPosition(int playerID, int
positionX, int positionY) : void
+gatherResource(Player player) : void
+updateResources() : void
+getResources() : int[][]
+setResources(int[][] resources) : void
+main(String[] args): void

UnicastRemoteObject

Controller

+model: ForagerInterface
+view: View
+player: Player
+board: int[][]
+timer: Timer
+playerPositions: HashMap<Integer,
Pair<Integer, Integer>>
+seconds: int = 1

+ Controller(model)
+main(String[] args): void
+keyPressed(KeyEvent e)
+ updateBoard(controller c) :void
+startTimer(): void
+getSeconds(): int
+keyTyped(KeyEvent e) : void
+keyPressed(KeyEvent e) : void
+keyReleased(KeyEvent e) : void

<<interface>>
KeyListener

+keyTyped(KeyEvent e) : void
+keyPressed(KeyEvent e) : void
+keyReleased(KeyEvent e) : void

Extends

Observable

PlayerDecorator

+PlayerDecorator(Tile
decoratedTile): void
+draw(Graphics g): void
+ getX(): int
+getY(): int

Pair

-a: A
-b: B

+pair(A a, B b): void
+getFirst(): A
+getSecond(): B

Player

-playerID: int
-positionX: int
-postionY: int
-collectedResources: int

+player(int x, int y, int resources):
void
+getID(): int
+getPosition(): Pair<int, int>
+setPostion(int x, int y): void
+getResources(): int
+setResources(int resources):
void

View

+ jframe: JFrame
+ jpanel: JPanel

+View();
+update(Observable o, Object
args):void
+makeBoard(int [][] input): void
+updateBoard(int [][] input): void
+getJFrame(): JFrame

<<interface>>
Observer

+update(Observable o, Object
args):void

<<interface>>
Tile

+draw(Graphics g): void
+getX(): int
+getY(): int

TileDecorator

+decoratedTile: Tile

+tileDecorator(Tile decoratedTile):
void
+draw(Graphics g): void

TileZero

• positionX: int
• positionY: int

+ TileZero(int x, int y): void
+ getX(): int
+ getY(): int
+draw(Graphics g): void

TileOne

• positionX: int
• positionY: int

+ TileOne(int x, int y): void
+ getX(): int
+ getY(): int
+draw(Graphics g): void

TileTwo

• positionX: int
• positionY: int

+ TileTwo(int x, int y): void
+ getX(): int
+ getY(): int
+draw(Graphics g): void

TileThree

• positionX: int
• positionY: int

+ TileThree(int x, int y): void
+ getX(): int
+ getY(): int
+draw(Graphics g): void

TileFour

• positionX: int
• positionY: int

+ TileFour(int x, int y): void
+ getX(): int
+ getY(): int
+draw(Graphics g): void

TileFive

• positionX: int
• positionY: int

+ TileFive(int x, int y): void
+ getX(): int
+ getY(): int
+draw(Graphics g): void

TileSix

• positionX: int
• positionY: int

+ TileSix(int x, int y): void
+ getX(): int
+ getY(): int
+draw(Graphics g): void

TileSeven

• positionX: int
• positionY: int

+ TileSeven(int x, int y): void
+ getX(): int
+ getY(): int
+draw(Graphics g): void