1.In the provided code, discover all the points where an '@Override' annotation could be added on a method and list them.

fileName, class and method name, line

Monster.java , Monster.location() , line 10

Monster.java , Monster.location(Point) , line 11

Monster.java , Monster.ping(Model), line 15

Monster.java, Monster.draw(Graphics, Point, Dimension), line30

Keys,java , keys.keyTyped(KeyEvent) , line 17

Keys,java , keys.keyPressed(KeyEvent) , line 18

Keys,java , keys.keyReleased(KeyEvent) , line 22

Camera.java , Camera.location() line11

Camera.java , Camera.location(Point) line12

Camera.java , Camera.ping(Model), line14

Camera.java , CameraMonster.draw(Graphics, Point, Dimension) line17

Sword.java , Sword.location(Point) , line14

Sword.java , Sword.ping(Model) , line24

Sword.java , Sword.draw(Graphics, Point, Dimension) , line33

Viewport.java , Viewport.paintComponent(Graphics),line 16

2.Consider Direction, Controller, Camera and Sword. (a) What is the name of the design pattern encoded by those classes? (b) Explain in the detail the meaning of " setAction(KeyEvent.VK_W, c.set(Direction::up), c.set(Direction::unUp)) " What is this code doing? How is this code allowing for the player to control the hero?

(a) composite pattern

(b) the type of c is Camera. So this function is controlling player to move,c.set(Direction::up) mean when the key pressed, player move up, ,c.set(Direction::unUp) means when the key released the player will stop.

3. Explain the behavior of Cells.forAll.

(a) How is Cells.forAll currently used?
(b) Can you describe some feature that we could add to our game that would take advantage of this
method?

(a) this function is used to create the map for the game, if the place is out of range, it will be stone.

(b) When we want to add some decorations to the game ,

4.In Compact, the concept of changing phase encodes a programming pattern.

(a) What is this programming pattern?
(b) How is this implemented? What classes serve what roles in that programming pattern?

(a)Strategy pattern

(b)Using different Phases to choose which strategy to use.

5.Explain why using “SwingUtilities.invokeLater(Compact::new)” is needed and why“assert SwingUtilities.isEventDispatchThread();” is a good check to have in this setting.

(a) In Swing, if we want a gui Interfaces to run our program , we need to give a instance for gui. So, first step is to Compact and make sure that everything works as expected.

(b) SwingUtilities.isEventDispatchThread(); this function is used to check if system dispatched a thread for Swing. If assert true means Swing have a dispatched a thread which means invokeLater work. So this function can use to check.

6.Explain why the provided code was forced to use “addWindowListener(new WindowAdapter(){..})” instead of a Lambda.

This code can provide a windows event messages, In this way, we can exit normally when we click close window. Lambda expressions can replace functional interfaces, but for WindowAdapter it have more than one function, so we can not use lambda here.

7.Currently there is a bug: if you press the ‘up’ key the hero keeps going up even after the key has been released.

(a) Explain what is happening with this bug.
(b) Fix the bug changing a single character. Report changed code and the line number here.
(c) What features could help to avoid this and similar bugs?

(a) this is beacuse in Direction the funtion name is unUn, So when the key released, Camera didn't get the move up order.

(b) Direction.unUp() line 8

(c)@Override