**User stories**

**1.Bewegung der Spielfigur(Pinguin)**

Der Benutzer soll seine Figur nach links und nach rechts bewegen können, damit die Spielfigur  den Hindernissen ausweichen kann.

**Task für das Nutzen der Pfeiltasten**

**Task 1:**Erzeuge die Klasse **Spielfigur\_model** (SpielObjekt)und die Klasse **Controller**

**Task 2:** Die Klasse Controller soll **Keylistener** implementieren. Dazu muss **view** im Konstruktor die Methode **addKeylistener** aufrufen (davor werden werden die benötigten Attribute zunächst initialisiert und im Konstruktor deklariert).

**Task 3:**Implementiere schließlich die Klasse Test, in der im Bezug zur Bewegung der Spielfigur ein neuer **ActionListener** erzeugt wird.In dieser Klasse wird die Methode **actionPerformed** überschrieben mit dem Parameter **ActionEvent.**Hier wird zunächst ein zufälliges Hindernis hinzugefügt, das Hindernis wird dann aktualisiert, die Spielfigur informiert alle und schließlich wird auf der Konsole „Code im Takt ausgeführt” ausgegeben)

**Task 4:**Implementiere die Methode **keyPressed.**Hier soll festgelegt werden ,dass sich die Spielfigur nach rechts bewegen soll ,falls die rechte Pfeiltaste gedrückt wurdeAußerdem soll sich die Spielfigur nach unten bewegen, ,falls die Pfeiltaste nach unten gedrückt wird .Die Spielfigur soll sich nach links bewegen, falls die linke Pfeiltaste gedrückt wurde.Die Methoden **keyReleased** und **keyTyped** bleiben in diesem Fall leer.

**Task 5**: In der Klasse **Spielfigur\_model** soll nun die Methoden **nachRechtsBewegen** und **nachLinksBewegen** und **nachUntenBewegen** implementiert werden ,indem man von der x-Koordinate jeweils 10 abzieht/hinzufügt (für links und rechts) und von der y Koordinate 10 hinzufügt(für die Verschiebung nach unten) und schließlich alle informiert.

Dafür müssen die Getter- Methoden für x und y Koordinaten erstellt sein und ihnen muss ein Wert im Konstruktor zugewiesen sein

**Task für das Erstellen der Hindernisse Task 1**: Erzeuge die Klasse **Hindernis\_model** mit den Attributen: Höhe, Breite, Verschiebung, Name, Hindernisliste und Hbeobachter

**Task 2:** Im Konstruktor soll den Attributen ein neuer Wert zugewiesen werden und bei den Attributen mit dem Datentyp Arraylist soll eine Arraylist erzeugt werden.

**Task 3:** Getter- und Settermethoden, die Methoden**„add“** (hinzufügen),**entfernen,anmelden.geschwindigkeitSetzen,abmelden,aktualisieren** und **nachObenRücken** sollen implementiert werden

**Task 4:** In der Klasse **Controller** soll die Methode **zufälligesHindernisHinzufügen()** implementiert**,** damit man das erzeugte Hindernis ganz zufällig im gesamten Spiel verteilt

**2.Niederlage**

Der Benutzer verliert das Spiel ,falls die Figur ein Hindernis berührt(Schneemann,Hütte und Salomonfahne)

Das Spiel stoppt und auf dem Bildschirm erscheint ein ..

**“Game Over“**

**3.Münzen sammeln**

Der Benutzer kann kann im gesamten Spiel Münzen sammeln, die zufällig auf der Spielfläche verteilt sind.Falls die Spielfigur mit der Münze kollidiert, erhöht sich der Highscore um 10.