Polimorfismo: El polimorfismo ocurre cuando distintos objetos reciben el mismo mensaje y lo entienden, pero tienen distintas formas de actuar a partir del mismo. En nuestro juego el mensaje polimórfico es "**efectoAlChocar(jugador)**" este es implementado por obstáculos que se encuentran en el camino del jugador, de los cuales algunos debe intentar evitarlos y otros intentar chocarlos. Al recibir el mensaje efectoAlChocar(jugador) los objetos realizarán las siguientes acciones:

- Los ObjetoGrande: Aplastan al jugador, haciendo que este pierda. Acción: **nombreDelJuego.perder()**
- Los ObjetoMediano: Empujan al jugador para atrás 2 posiciones, haciendo que esté quede más cerca de la policía. Acción: **personaje.moverPersonaje(personaje.position().left(2))**
- Los ObjetoChico: Empujan al jugador para atrás 1 posición. Acción: **personaje.moverPersonaje(personaje.position().left(1))**

Colecciones: En el código se utilizan colecciones a la hora de sumar las estrellas para evaluar la condición de victoria. Se usa la operación con efecto **add()** para modificar la colección y agregar los distintos tipos de estrella a la lista de coleccionadas que tiene el jugador. Luego, se utiliza el método **sum()** para que, de todas las coleccionadas, se calcule el puntaje que dan para evaluar si el jugador ganó y mostrar los puntos que obtuvo.

Clases: Las clases las utilizamos en el juego, ya que hay diferentes objetos, por ejemplo las estrellas y los obstaculos, que utilizan todos la misma clase, ya que de esta forma se evita repetición de código; hay configuraciones que son para todos los objetos la misma, y esta se declara en la clase general. Luego, para especificar cosas de cada objeto, uno puede declarar más métodos o re-declararlos.

Por ejemplo en la clase ObjetosAEsquivar, nosotros definimos los métodos aparecer, la posición y moverse. Mientras que en el ObjetoGrande, que hereda de ObjetosAEsquivar, también se le define dentro los métodos: chocarCon, velocidad y la imagen.

Herencia: Ocurre la herencia en por ejemplo Portal y PortalEspacio, que portal espacio hereda toda la información que está en Portal, por ejemplo velocidad e imagen, y PortalEspacio, por ejemplo, sobrescribe la información de la imagen, pero se queda con la información del Portal de la velocidad. Es decir, en la herencia, uno puede utilizar la información de quien hereda, o la puede reescribir. Utilizamos en este caso la herencia, para no hacer repetición de código innecesario, y también para vincular estos objetos; Portal se vincula con PortalEspacio; a la vez, Portal hereda de ObjetosAEsquivar, ya que cuando comienza el juego en el espacio calle, el portal aparece como un objeto más. Por ende, mostramos así un tipo de relación entre objetos.

Composición: Utilizamos composición en la clase Estrella, interactuando con los objetos modoMayor y modoMenor. La clase estrella delega en estos objetos para que sean ellos los que se encarguen de decir cómo se realiza la tarea que tienen en “común”, que en nuestro caso, es sumarle un puntaje al jugador y mostrar un mensaje por pantalla. De este modo, la Estrella no debe preocuparse de cuánto puntaje debe otorgar ni qué mensaje debe enviar, sino que esto dependerá del modo en el que se encuentre. No lo hacemos como herencia ya que no vale la pena tener distintas clases de estrellas si todas van a comportarse de igual manera a excepción del mensaje y el puntaje, además, al tener una sola clase Estrella que pueda entender esos 2 comportamientos distintos le otorga una mayor flexibilidad al código