

Example past (CPSC 219) assignment employing inheritance

URL to full assignment description if needed:

<https://pages.cpsc.ucalgary.ca/~tamj/2012/219W/assignments/assignment5/index.htm>

Synopsis:

For this assignment you will be creating a computer game that simulates a portion of Professor JRR Tolkien's novel, "The Fellowship of the Ring" (see Figure 1).

```
Boromir statistics:
-----
Hit points: 20
Hobbit statistics:
-----
Hit points: 10
  0 1 2 3 4 5 6 7 8 9
0!B!h! ! ! ! ! ! ! ! !
1! ! ! ! ! ! ! ! ! !
2! ! ! !U! ! ! ! ! !
3! ! ! ! ! ! ! ! ! !
4! ! ! ! ! ! !g! ! ! !
5! ! ! ! ! ! ! ! ! !
6! !g! ! ! ! ! ! ! !g!
7! ! ! ! ! ! ! ! ! !
8! ! ! ! !U! ! ! ! ! !
9!U! ! ! ! ! ! ! !X!
  -----
MOVEMENT OPTIONS
-----
  7 8 9
  4 5 6
  1 2 3
Enter 5 to pass on movement
Enter a negative value to quit the game
Enter direction: _
```

Figure 1: Breaking of the Fellowship (Boromir defends the Hobbits)

The Gondorian hero Boromir 'B' must ensure that the hobbits 'h' escape from the horde of evil orcs: goblins 'g' and the elite Uruk-hai 'U'. The hobbits will automatically flee to the exit 'X' while Boromir is controlled by the player. Hobbits first move all the way to the east and then straight down to the south. The goblins will be controlled by the computer (random movement) and use their scimitar swords to attack Boromir or the hobbits if they come within striking range. The elite Uruk-Hai carry bows and can hit targets from afar. If the players have bought enough time for the hobbits reach the exit (bottom right) then the game is won (even if Boromir is slain). If the hobbits are slain the game is lost. The hobbits may be physically weak but they can blend into terrain (i.e., "hide in shadows") so half of the attack will miss.

Some of the class descriptions

MEObject: represents all the entities in the game world. Each instance of the class will have an appearance attribute. Also each MEObject will have a 'hitPoints' attribute that indicates the amount of damage that the entity can take before being slain (or destroyed for the non-living objects). This class will have a number of class constants: one for the appearance, one for the default hit points and one to determine the threshold of life (at or below this level the entity is longer living). Each child class should have its own class constant to indicate the default appearance of the instances of that class e.g., `class UrukHai { public static final DEFAULT_APPEARANCE = 'U'; ... }`

Humanoid: a child class of MEObject, it represents living and moving occupants of the world. Consequently instances of this class will have four attributes: the current row/column coordinate of the humanoid, the attack range, and a boolean flag that indicates whether the object has moved yet during the turn. Additionally Humanoids have methods that allow them to move (randomly generates a value that determines which of the adjacent squares that it will attempt to move to - the MEWorld class will determine whether that location is a valid: is it in bounds and is it empty). Finally humanoids can attack one opponent each turn. The default damage is 2 points although the child classes will inflict differing amounts of damage by overriding the attack method.

Hobbit: one of the heroes in the game but is controlled by the computer.

Minimum required attributes: several constants will be included to model the hobbits behavior: `HIDE_TERRAIN` (the percentage chance that a particular attack on the hobbit will miss), `MAX_HIT_POINTS = 10` (hobbits may not be strong but they are tougher than the average humanoid so they have a higher default value), `EAST/SOUTH` (indicates which direction that the hobbit will travel). At the start of the game the hobbit will continue moving east until either an obstacle is encountered, it's been slain or it has reached the eastern most column (in the latter case the hobbit will then move south).

Minimum required behaviors: include some overridden methods. There should be a new attack method to reflect the range of damage that a hobbit can inflict (1 - 6 points). Also this class should have a new method of deducting damage that determines whether or not the hobbits take damage from a particular attack (JT: this is actually a helpful hint for implementing the 'hide in shadow's capability).

Human: the hero in the game that is controlled by the player.

Minimum required attributes: a constant `MAX_HIT_POINTS = 20` (humans are physically larger and stronger than many of the other inhabitants of Middle Earth so this value is set higher).

Minimum required behaviors: an overridden attack method to reflect the range of damage that a human can inflict (4 - 13) points.

Goblin: the weakest of the orc-kind. Humanoids that are villains that are controlled by the computer.

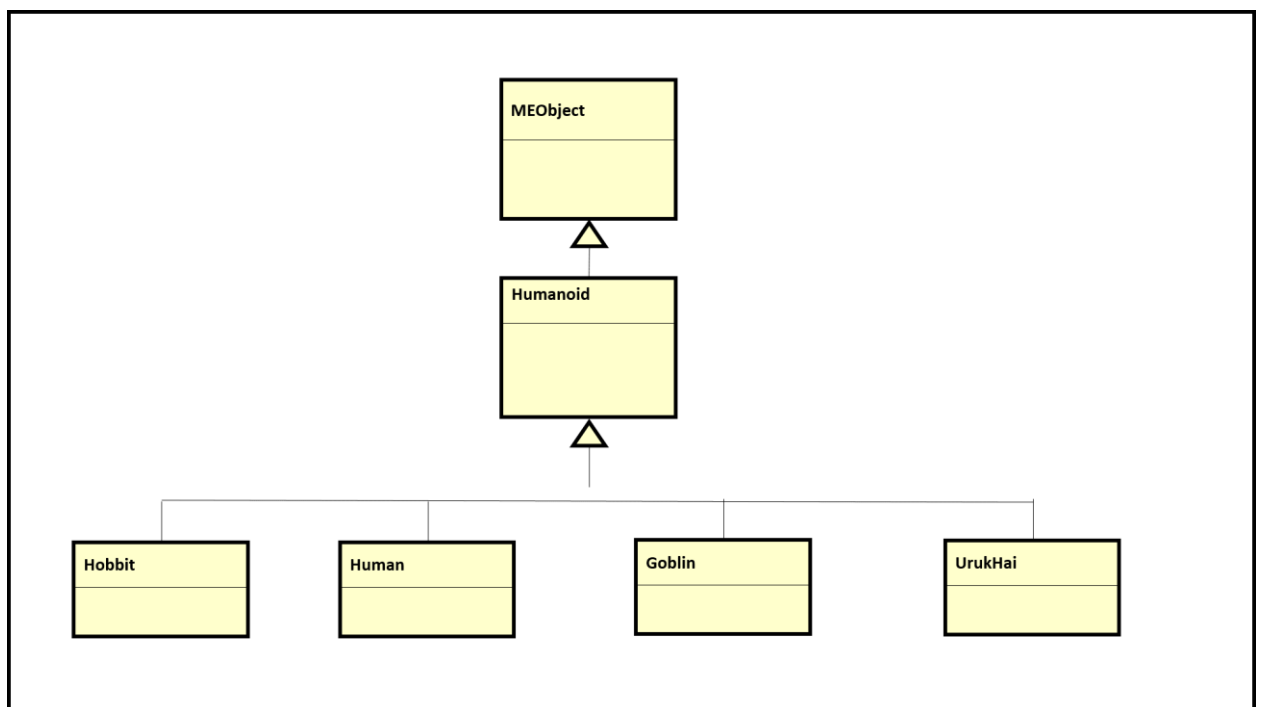
Minimum required behaviors: an overridden attack method to reflect the range of damage that a goblin can inflict (1 - 8 points)

UrukHai: the elite shock troopers in the orc ranks. They are also computer-controlled.

Minimum required attributes: a constant to reflect their increased size and fighting skill `MAX_HIT_POINTS = 14`, and another constant `DEFAULT_RANGE` to reflect the fact that they carry bows and can damage opponents up to two units distant (up or down two rows, over two columns, all other humanoids that attack can only reach adjacent opponents).

Minimum required behaviors: similar to class `Goblin` there should be an overridden attack method to reflect the range of damage that an `UrukHai` can inflict (3 - 9 points).

Summary of class attributes and abilities



MEObject

```
-appearance : char
-hitPoints: int
+DEFAULT_APPEARANCE : char
+DEFAULT_HIT_POINTS : int
+DEAD : int
+MEObject() << constructor >>
+MEObject(anAppearance:char) << constructor >>
+deductDamage(damage:int) : void
+getAppearance() : char
+getHitPoints() : int
+setAppearance(anAppearance:char) : void
+setHitPoints(hitPoints:int):void
```

Humanoid

```
+DEFAULT_APPEARANCE : char
+ATTACK_DAMAGE : int
+DEAD : int
-aGenerator : Random
-row: int
-column: int
-range: int
+Humanoid(anAppearance:char,
           row:int,
           column: int) << constructor >>
+move(): Coordinate
+directionToLocation(direction:int) : Coordinate
+getHitPoints() : int
+setAppearance(anAppearance:char) : void
+setHitPoints(hitPoints:int):void
```

Attribute description (Humanoid)

1. Range: how far the humanoid can attack, all but the UrukHai have a range of 1 (only attacks adjacent squares)

Method descriptions (Humanoid)

1. Move: determines randomly determined direction
2. directionToLocation: given a compass direction (1-8) the method returns a (row,column) location

Hobbit

```
+DEFAULT_APPEARANCE : char
+HIDE_TERRAIN : int
+MAX_HIT_POINTS : int
+attack():int
+move(): Coordinate
+remainsHidden(): boolean
+deductDamage(damage:int)
```

Method descriptions (Hobbit)

1. Move: moves the hobbit east to the right and then south to the bottom.
2. remainsHidden: determine if the hobbit hides during a turn
3. deductDamage: when hobbit is attacked but hidden no damage is deducted from the hit points

Human

```
+MAX_HIT_POINTS : int
+attack():int
```

Goblin
DEFAULT_APPEARANCE : char +attack():int

UrukHai
+DEFAULT_APPEARANCE : char : int +MAX_HIT_POI : int +DEFAULT_ATTACK_RANGE : int +attack():int

Attribute descriptions (UrukHai)

1. DEFAULT_ATTACK_RANGE = 2
(can use bow to attack at range e.g. up to 2 rows up)