

PA1459/PA1460 Example II

Mikael Svahnberg

March 1, 2021

Contents

1	System Description	1
2	Class Diagram	1
3	Description of Classes	2
4	Pseudocode	3
5	Discussion of GRASP patterns	3
6	Usage of GRASP patterns	4
	pattern State Pattern GRASP1 Creator GRASP2 Information Expert	

1 System Description

Desktop Ponies är en applikation som låter MLP ponies springa runt på skärmen.

Varje Pony byter slumpvis beteende. Varje beteende är ett **tillstånd**, ett **state**.

2 Class Diagram

PonyContainer - "*" Pony

```
abstract class PonyBehaviour <<Abstract State>> {  
    +enter()  
    +execute()  
}
```

```

+exit()
}

```

```

Pony - "*" PonyBehaviour

```

```

Pony : -List<PonyBehaviour*> myBehaviourCollection
Pony : -PonyBehaviour* myCurrentBehaviour

```

```

PonyBehaviour <|-- Stand
PonyBehaviour <|-- Walk
PonyBehaviour <|-- Jump

```

```

Pony -- PonyBehaviourFactory

```

```

PonyBehaviourFactory : +getBehaviourByName(String ponyName)

```

```

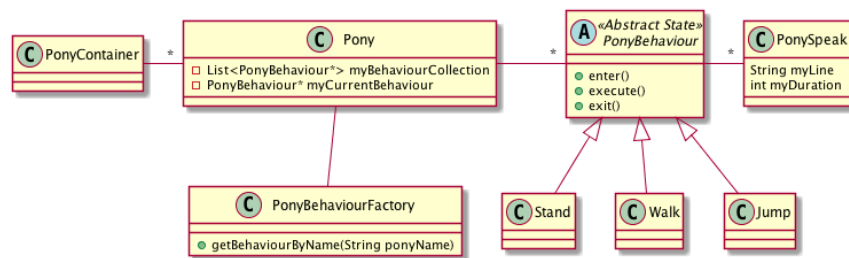
class PonySpeak
PonySpeak : String myLine
PonySpeak : int myDuration

```

```

PonyBehaviour - "*" PonySpeak

```



3 Description of Classes

Pony är i State pattern. Den äger en samling av **PonyBehaviour** och vet vilket som är **current**.

PonyBehaviour är , den erbjuder det gränssnitt som alla konkreta beteenden måste implementera.

{Stand, Walk, Jump} är . Var och en implementerar beteendet för ett tillstånd.

PonyContainer Innehåller samlingen av alla Pony.

PonySpeak En enskild replik som en Pony kan säga i ett visst tillstånd.

PonyBehaviourFactory creates objects of the subclasses.

4 Pseudocode

```
Pony::create() {
    List<String> behaviourNames = PonyInitFile::getBehaviours(ponyName);

    behaviourNames.forEach( (n) => {
        PonyBehaviour* pb = PonyBehaviourFactory::getBehaviourByName(n);
        myBehaviourCollection.append(pb);
    });

    myCurrentBehaviour = myBehaviourCollection[0];
    myCurrentBehaviour.enter();
}

Pony::setBehaviour(String newBehaviourName) {
    PonyBehaviour* pb = myBehaviourCollection.findByName(newBehaviourName);

    if(pb) {
        myCurrentBehaviour.exit();
        myCurrentBehaviour = pb;
        myCurrentBehaviour.enter();
    }
}
```

5 Discussion of GRASP patterns

Context är **information expert** på vilka tillstånd som finns och vilket som gäller just nu.

Context kan vara **creator** av de konkreta tillstånden.

Abstract State är **information expert** på gränssnittet för alla tillstånd.

Concrete State är **information expert** på vad det innebär att vara just det tillståndet.

6 Usage of GRASP patterns

PonyContainer är **information Expert** på vilka Pony som finns.

Pony är, som **Information expert** på vilka **PonyBehaviour** en viss Pony har och vilket som gäller just nu.

PonyBehaviourFactory är **Creator** som ansvarar för att skapa objekt av de konkreta **PonyBehaviour**-implementationerna.

PonyBehaviour är som **Information expert** på gränssnittet för alla tillstånd.

PonyBehaviour är **information expert** på vad man kan säga (**PonySpeak**) i ett visst beteende.

{Stand, Walk, Jump} är **information expert** på att vara respektive beteende.

PonySpeak är **information expert** på att säga en specifik sak.