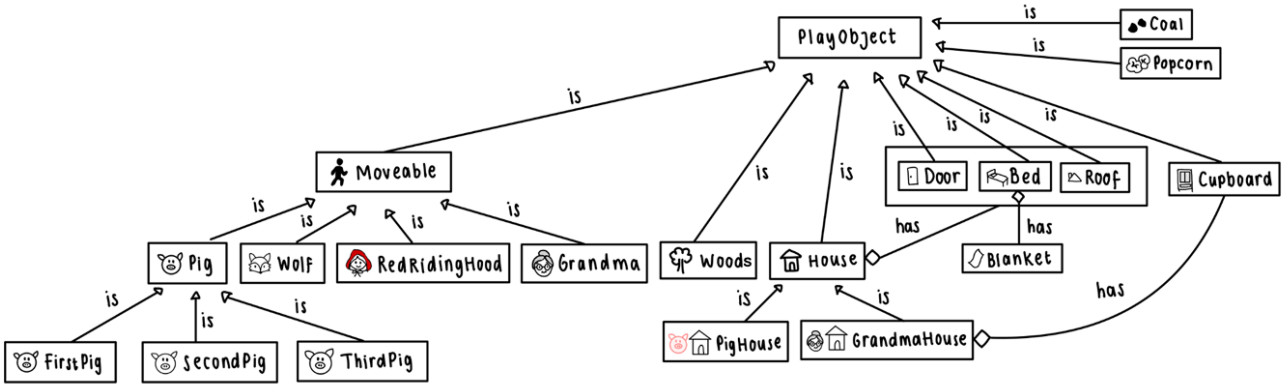
Adelia Putri | 20210782 | adelia@kaist.ac.kr

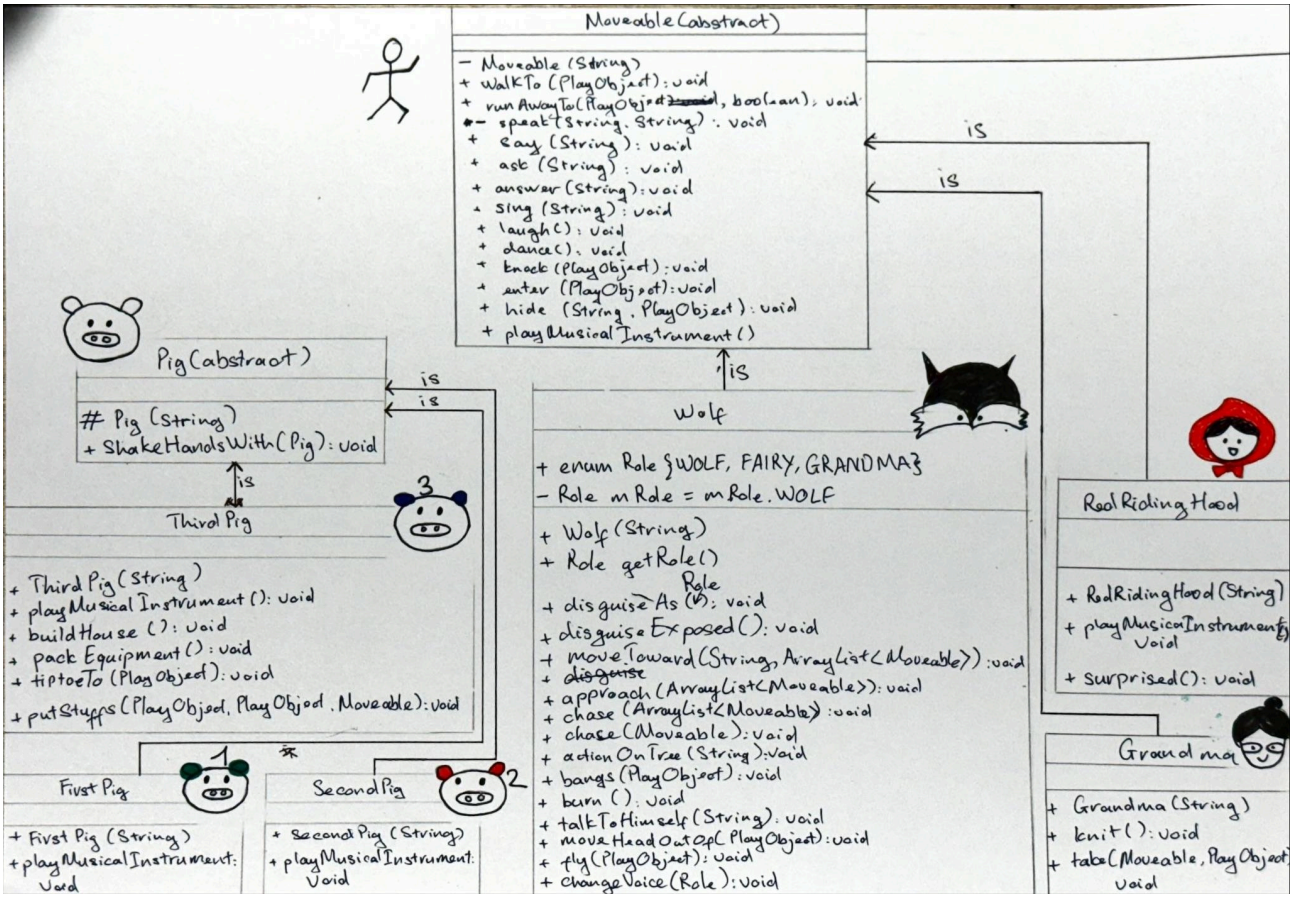
Thi Ha Giang Nguyen | 20256333| jana.nthg@kaist.ac.kr

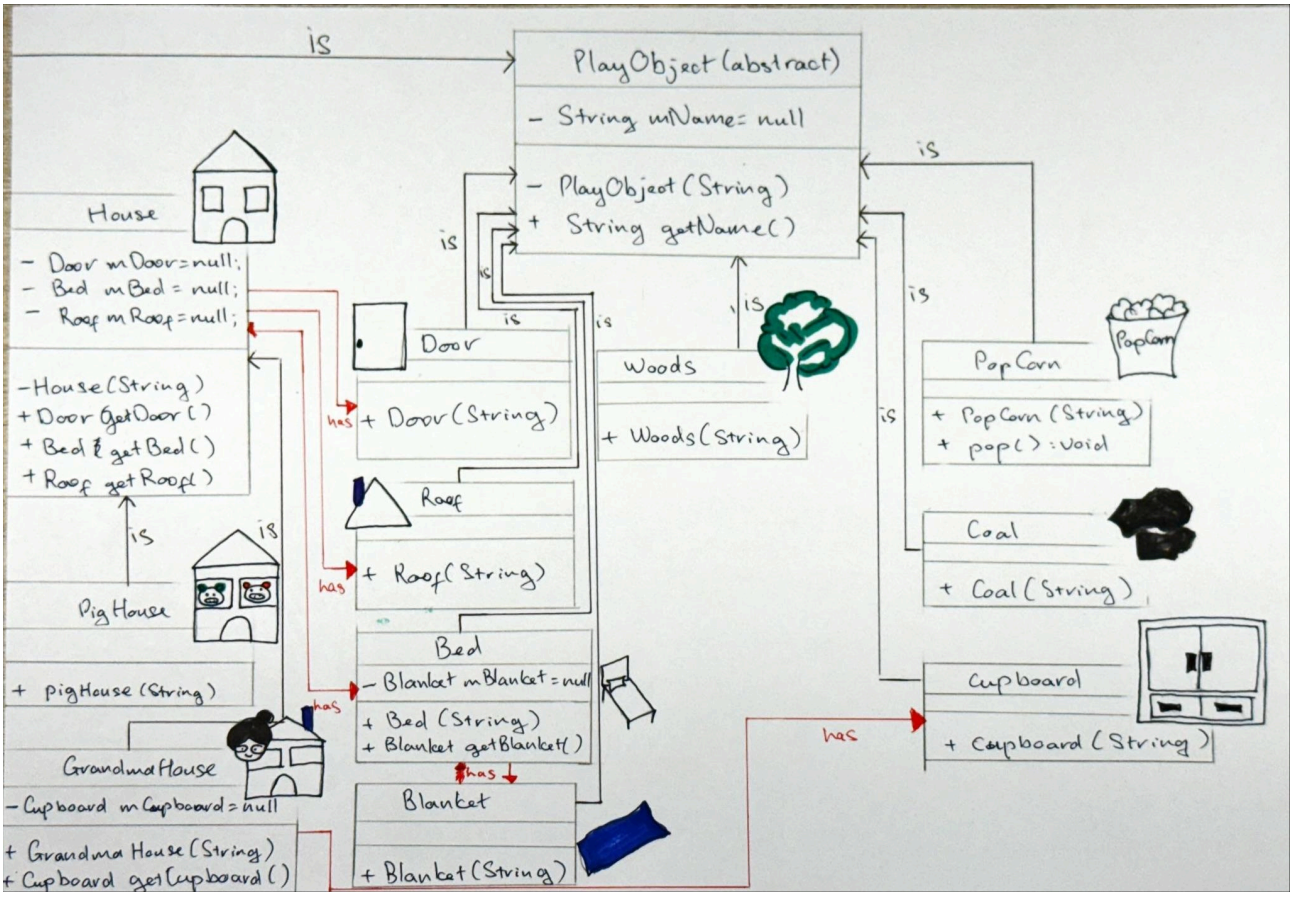
<Class design (with free-hand drawings)>

Below is the overall class design in our implementation.



The details of each class is shown in the diagram below.




is
PlayObject (abstract)
- String mName= null
- PlayObject (String)
+ String getName()
House
- Door mDoor=null;
- Bed mBed = null;
- Roof mRoof=null;
-House(String)
+Door getDoor()
+Bed getBed()
+Roof getRoof()
Door
+ Door(String)
Roof
+ Roof(String)
Bed
- Blanket mBlanket=null
+ Bed(String)
+ Blanket getBlanket()
Blanket
+ Blanket(String)
Woods
+ Woods(String)
PopCorn
+ PopCorn(String)
+ pop():void
Coal
+ Coal(String)
Cupboard
+ Cupboard(String)
PigHouse
+ pigHouse(String)
GrandmaHouse
- Cupboard mCupboard=null
+ GrandmaHouse(String)
+ Cupboard getCupboard()
has
is

<Output result>

<<The Big Bad Wolf>>

=== Red Riding Hood meets Three Little Pigs ===

Red Riding Hood walks to Three Little Pigs' house.

Adam plays his flute.

Brian plays his fiddle.

Charles builds his house of stones and bricks.

Adam says, "Good morning, Little Red Riding Hood."

Brian says, "Good morning, Little Red Riding Hood."

Red Riding Hood says, "Good morning, how do you do?"

Charles says, "Good morning, Miss Riding Hood."

Adam asks, "Where are you going Red Riding Hood?"

Brian asks, "What's in the basket, something good?"

Red Riding Hood answers, "I'm bringing Grandma cakes and wine.

She's awful awful sick. I'm in a great big hurry too.

I've got to get there quick."

Adam says, "You get there quick Red Riding Hood.

If you take the shortcut through the woods."

Brian says, "You get there quick Red Riding Hood.

If you take the shortcut through the woods."

Charles says, "There's danger in the woods. Be aware! Wilson is lurking there!

Better to be safe than sorry!

Shortcuts are not always good. Take the long road around the forest while Wilson is in the woods."

Adam laughs.

Adam says, "That old wolf is just a sissy."

Brian laughs.

Brian says, "That old wolf is just a sissy."

Adam says, "All he does is huff"

Brian says, "and puff."

Adam says, "We'll go with you and protect you. Come along! We'll call his bluff."

Brian says, "We'll go with you and protect you. Come along! We'll call his bluff."

Adam sings, "Who's afraid of the Big Bad Wolf?"

Adam plays his flute.

Brian sings, "Who's afraid of the Big Bad Wolf?"

Brian plays his fiddle.

=== Red Riding Hood goes to the woods ===

Adam walks to woods.

Brian walks to woods.

Red Riding Hood walks to woods.

Wilson hides behind a tree.

Wilson disguises as Fairy.

Adam plays his flute.

Brian plays his fiddle.

Wilson approaches Adam and Brian and Red Riding Hood.

Wilson hangs on a tree.

Wilson sings, "I'm Goldilox the Fairy Queen. Spirit of the woods in my whoops.

You better fly from tree to tree. Come come my dear. You're safe with me."

Wilson fell off a tree.

Wilson's disguise as Fairy is exposed.

Adam runs away.

Brian runs away.

Wilson chases Red Riding Hood.

Wilson stucks on a tree.

Red Riding Hood runs away.

=== Wolf goes to Grandma's House ===

Wilson walks to Grandma's house.

Grandma knits.

Wilson knocks on Grandma's house's door.

Grandma says, "Come in!"

Wilson enters Grandma's house.

Wilson chases Grandma.

Grandma hides in Grandma's house's cupboard.

Wilson bangs on Grandma's house's cupboard

Red Riding Hood walks to Grandma's house.

Red Riding Hood knocks on Grandma's house's door.

Wilson disguises as Grandma.

Wilson says, "Come in!"

Red Riding Hood enters Grandma's house.

Red Riding Hood says, "Good morning Grandma, how do you feel?"

Wilson says, "Terrible!"

(Wilson changes to Grandma's voice.)

Wilson says, "Not so hot, deary"

Red Riding Hood is surprised.

Red Riding Hood asks, "Ooh grandma, what big eyes you've got!"

Wilson says, "All the better to look you over deary"

Red Riding Hood is surprised.

Red Riding Hood asks, "Ooh grandma, what a big nose you've got!"

Wilson says, "All the better to hachchchh"

Wilson hides under blanket.

(Wilson uses his own voice.)

Wilson talks to himself, "How am I doing"

Wilson laughs.

Wilson moves head out of blanket.

Red Riding Hood is surprised.

Red Riding Hood asks, "Ooh grandma, what a big mouth you've got!"

(Wilson uses his own voice.)

Wilson says, "You ain't seen enough of me deary?"

Wilson laughs.

Wilson chases Red Riding Hood.

Red Riding Hood runs around Grandma's house.

=== Pigs' House ===

Adam runs to Three Little Pigs' house.

Brian runs to Three Little Pigs' house.

Adam hides under Three Little Pigs' house's bed.

Brian hides under Three Little Pigs' house's bed.

Adam says, "The Wolf! He's got her!"

Brian says, "The Wolf! He's got her!"

Charles packs equipment into his bag.

Charles runs to Grandma's house.

=== Grandma's House ===

Wilson chases Red Riding Hood.

Grandma takes Red Riding Hood into Grandma's house's cupboard.

Red Riding Hood hides inside Grandma's house's cupboard.

Grandma hides inside Grandma's house's cupboard.

Wilson bangs on Grandma's house's cupboard

Charles enters Grandma's house.

Wilson says, "Open the door and let me in!"

Charles tiptoes to Wilson

Charles puts pop corn and hot coal into Wilson's pants.

Wilson burns.

Wilson flies off Grandma's house's roof.

The pop corn pops.

Wilson runs away.

=== Grandma's House After Chasing The Wolf Away ===

Grandma knits.

Adam shakes hand with Brian.

Adam plays his flute.

Brian plays his fiddle.

Charles plays the piano.

Red Riding Hood plays the piano.

Adam sings, "Who’s afraid of the big bad wolf. He’s the biggest sissy"

Adam dances.

Brian sings, "Who’s afraid of the big bad wolf. He’s the biggest sissy"

Brian dances.

<Discussion>

In designing the class hierarchy, we started by identifying shared behaviors across the classes. For example, we created a superclass called Moveable, which contains methods for common actions like walkTo, hide, and say. Characters that can perform these actions (e.g., Pig, Wolf, Red Riding Hood, and Grandma) were assigned as subclasses of Moveable.

Initially, we considered grouping Woods and House under a common superclass named Place, but these two classes did not share enough similarities to justify that structure. So, we ended up extending Woods and House directly from PlayObject.

We also designed a hierarchy for furniture inside the House to make references in the print statements clearer. For example, when a character knocks on a door, the program specifies whether it is Grandma House's door or the Pig House's door. However, for smaller items like Blanket, we decided to keep its name simple ("blanket") instead of creating a nested reference like "Pig House's bed's blanket" to keep it concise.

To simplify repetitive behaviors, we combined certain methods with similar structures. For example, we merged the chase() and approach() methods into a single method called moveTowards(), since both actions shared the same logic pattern. Similarly, the Wolf has different interactions with trees like "stuck", "hang", and "fell". So, instead of implementing each one as a separate method, we combined them into a method called actionOnTree(String action).

While combining some methods, we learned that method intuitiveness in OOP is important. Even though merging methods might seem efficient, it might cost the intuitiveness of the code. For example, we planned on merging ask(String something) and say(String something) into a generic method speak(String speechType, String something), but it makes the code less clear. From a user's perspective, calling ask("something") is far more intuitive than speak("ask", "something"), so we decided to keep these separate.

In the Moveable class there is a method called playMusicalInstrument(). This is a placeholder for the class's subclasses to override it later. The reason that we use this dummy method instead of making it into an actual abstract method is because not every subclass of Moveable will need this method (e.g., Wolf). While abstract methods will force all the subclasses to override, using a dummy method as a place holder will help as to control which subclasses need to override the method.

In the class ThirdPig, we created a method called putStuffs(PlayObject, PlayObject, Moveable) to describe the Third Pig's attack on the Wolf. Initially, we planned to implement a simple method like attack(PlayObject) that would directly print a message such as "The Third Pig puts popcorn and hot coal into the Wolf's pants." However, after considering future scalability and flexibility, we decided to design it in a more object-oriented way. We introduced separate classes which are Popcorn and Coal, and allowed the putStuffs() method to take different PlayObject instances as parameters. This approach makes the method more reusable and adaptable, as it can now handle different types of objects and interactions rather than being limited to a single hardcoded action.

Additionally, we added a state to the Wolf using an enumeration (WOLF, FAIRY, and GRANDMA). In the story, the Wolf takes on multiple roles, each with different behaviors and voices. By tracking the Wolf's current state, we could adjust its actions and speech accordingly using a switch-case statement.

What we learned from this assignment is the importance of planning ahead before writing any code. OOP in Java requires us to think carefully about the structure of our program like what classes we need, how they relate to each other, and what responsibilities each one should have. We had to consider which classes act as “parents” and which ones are “subclasses,” and how inheritance can help us avoid repeating code. Creating or visualizing a UML (Unified Modeling Language) diagram helped us see the relationships between classes more clearly. Overall, the assignment taught us that good OOP design is not just about writing code, but about thinking in terms of objects and relationships before coding.

It is worth mentioning that even though we tried to convey the whole story, we actually focus on the scene when the Wolf comes to Grandma’s house and the story after that. It is because we believe that this is the main part of the story where the tension and interaction between the characters become most dramatic. We focused on describing character expression and detailed action such as how the Wolf hides his face under the blankets and changes his voice, how Red Riding Hood was surprised because of Wolf’s appearance, etc. By centering our work on this part, we were able to highlight the key emotions, actions, and morals of the story more effectively.

In contrast, we intentionally kept the details in other scenes (e.g., those in the pigs’ house and the woods) to a minimum so that they would not overshadow the main storyline. For these sections, we avoided adding too many new methods and kept the implementation simple.