Final Reflection

Theme of Game: Mindhunter –The Game (based on Netflix show sort of)

Game design

User is an FBI agent on the hunt for serial killer named Donald "the Splitsville Splitter" (known to split victim's bodies in half.)

The goal is to catch Donald the Splittsville splitter. Time limit of 24 hours before killer leaves town.

The setting is the town of Splitsville.

Space class

1) Splittsville Police department. (top)
   class Detective -- interact with detective on case.
   create a random competence function for detective
   the more competent the better the evidence collected.
   1 = incompetent, 2 = somewhat incompetent 3 = competent
   if 3 – best evidence collected etc.

2) The junk yard (where victim body found).
   class interrogate junk yard manger
   if junk yard manager
   says he saw van with yellow stripes.
   go to apartment of killer.
   else go back to police station.

3) Apartment of killer (bottom)
   if killer in house arrest him
   chances of arresting him are greater if detective is competent and van with yellow stripes parked outside.

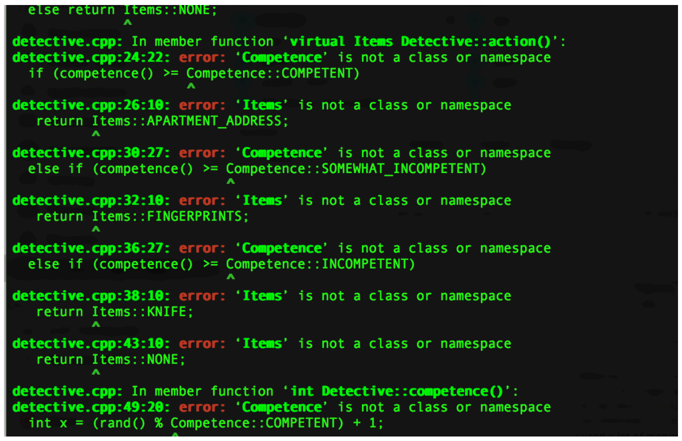
4) Jail
   execute class
   Donald the Splitsville Splitter is electrocuted in electric chair.

Test case	Input values	Driver functions	Expected outcome	Observed outcome
Interact with detective at police station	1	mindhunter, menu	Display any evidence (items). List items. Display next space menu	Everything except next space menu option
Interrogate junk yard manager	2	mindhunter, menu	Should display location and generate manager response	As expected
Go to serial killer's apt.	3	Mindhunter menu	Display location and items (evidence) found	C++ 11 error makefile updated to compile c++11
Execute serial killer	-	Mindhunters, others	Display 'serial killer has been executed via electrocution'	As expected

Reflection -------------------------

This project was fun albeit a bit ambitious on my part. The game play is not advanced but it gets the job done. Since, the scope of this project was cumulative of prior projects—I didn't feel to overwhelmed, more of just wanting to get it over with.

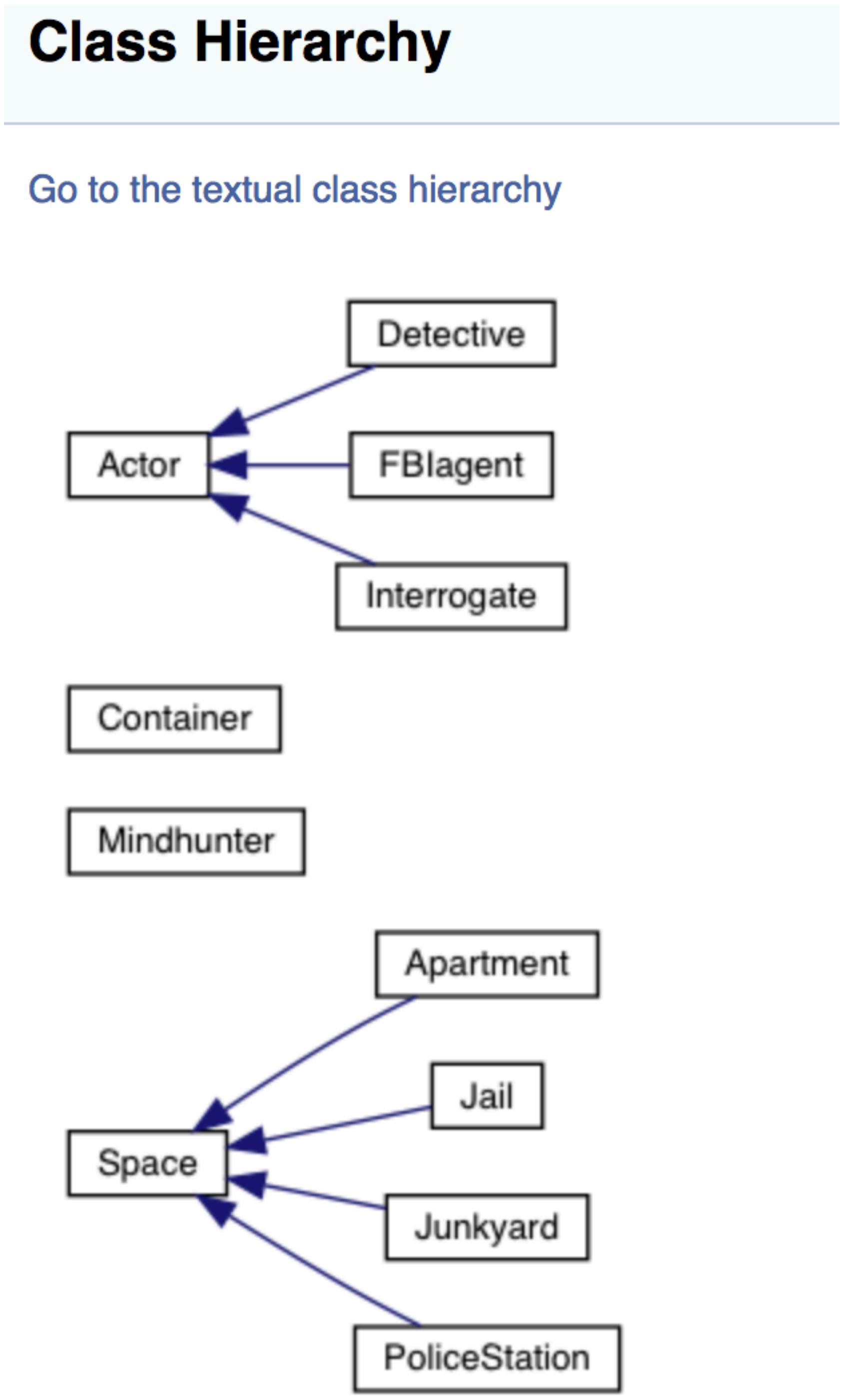
I used a lot of menus to keep track of the game play and spaces. I got some nasty errors regarding Classes/namespaces:



```
  else return Items::NONE;
              ^
detective.cpp: In member function 'virtual Items Detective::action()':
detective.cpp:24:22: error: 'Competence' is not a class or namespace
  if (competence() >= Competence::COMPETENT)
                      ^
detective.cpp:26:10: error: 'Items' is not a class or namespace
   return Items::APARTMENT_ADDRESS;
          ^
detective.cpp:30:27: error: 'Competence' is not a class or namespace
  else if (competence() >= Competence::SOMEWHAT_INCOMPETENT)
                           ^
detective.cpp:32:10: error: 'Items' is not a class or namespace
   return Items::FINGERPRINTS;
          ^
detective.cpp:36:27: error: 'Competence' is not a class or namespace
  else if (competence() >= Competence::INCOMPETENT)
                           ^
detective.cpp:38:10: error: 'Items' is not a class or namespace
   return Items::KNIFE;
          ^
detective.cpp:43:10: error: 'Items' is not a class or namespace
   return Items::NONE;
          ^
detective.cpp: In member function 'int Detective::competence()':
detective.cpp:49:20: error: 'Competence' is not a class or namespace
  int x = (rand() % Competence::COMPETENT) + 1;
```

but updating the makefile to compile using C++11 did the trick. I am not sure if this proper try to code using enums but C++11 took it.

I learned a lot in CS 162 and I feel much better going into whatever is next.

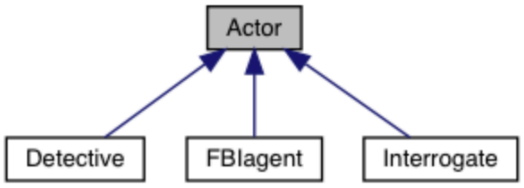


Class Hierarchy

Go to the textual class hierarchy

Actor Class Reference abstract

Inheritance diagram for Actor:



[legend]

Public Member Functions

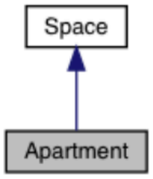
virtual Items **action** ()=0

The documentation for this class was generated from the following files:

- **actor.hpp**
- actor.cpp

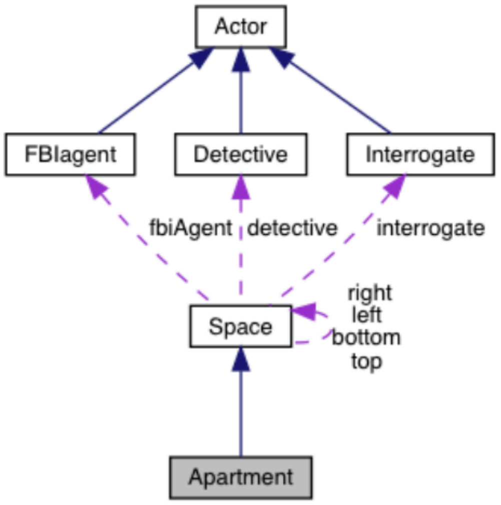
Apartment Class Reference

Inheritance diagram for Apartment:



[legend]

Collaboration diagram for Apartment:



[legend]

Container Class Reference

Public Member Functions

bool	**addItem** (Items item)
Items	**getItem** (int index)
int	**getNumItems** ()
bool	**hasItem** (Items item)

Static Public Attributes

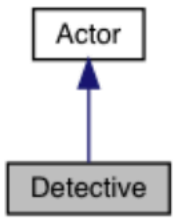
static const int	**MAX_ITEMS** = 10

The documentation for this class was generated from the following files:

- **container.hpp**
- container.cpp

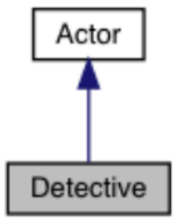
Detective Class Reference

Inheritance diagram for Detective:



[legend]

Collaboration diagram for Detective:



[legend]

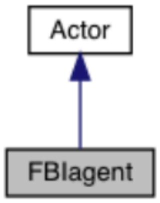
Public Member Functions

Items **action** ()

The documentation for this class was generated from the following files:

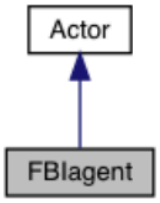
- **detective.hpp**
- detective.cpp

Inheritance diagram for FBIagent:



[legend]

Collaboration diagram for FBIagent:



[legend]

Public Member Functions

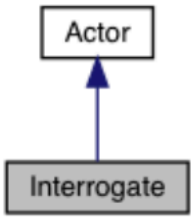
Items	**action** ()
void	**addItem** (Items item)
bool	**hasItem** (Items item)
void	**showContainer** ()

The documentation for this class was generated from the following files:

- **fbiagent.hpp**
- fbiagent.cpp

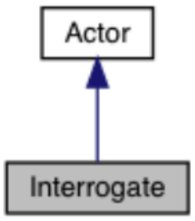
Interrogate Class Reference

Inheritance diagram for Interrogate:



[legend]

Collaboration diagram for Interrogate:



[legend]

Public Member Functions

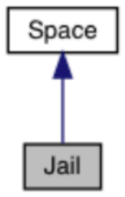
Items **action** ()

The documentation for this class was generated from the following files:

- **interrogate.hpp**
- interrogate.cpp

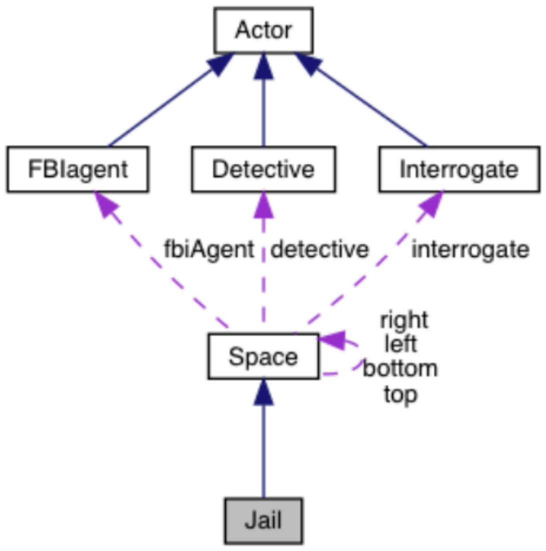
Jail Class Reference

Inheritance diagram for Jail:



[legend]

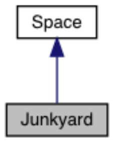
Collaboration diagram for Jail:



[legend]

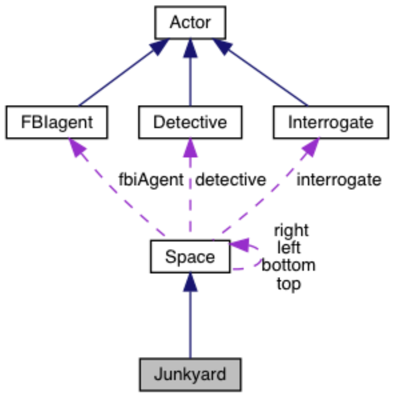
Junkyard Class Reference

Inheritance diagram for Junkyard:



[legend]

Collaboration diagram for Junkyard:



[legend]

Public Member Functions

void	**menu** ()
void	**action** ()

▸ **Public Member Functions inherited from Space**

Additional Inherited Members

▸ **Public Attributes inherited from Space**

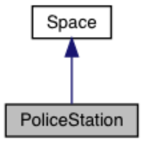
▸ **Static Protected Attributes inherited from Space**

The documentation for this class was generated from the following files:

- **junkyard.hpp**
- junkyard.cpp

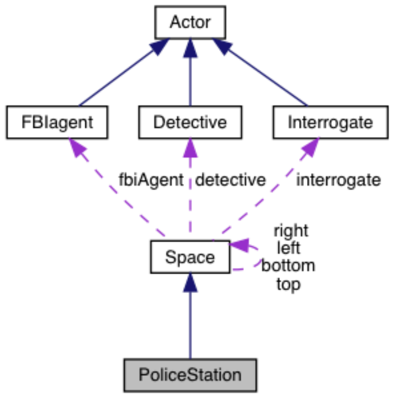
PoliceStation Class Reference

Inheritance diagram for PoliceStation:



[legend]

Collaboration diagram for PoliceStation:



[legend]

Public Member Functions

void **menu** ()

▸ **Public Member Functions inherited from Space**

Additional Inherited Members

▸ **Public Attributes inherited from Space**

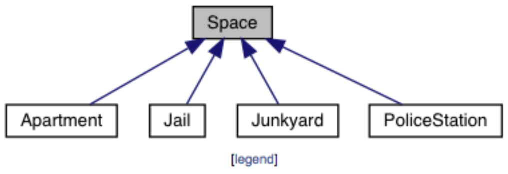
▸ **Static Protected Attributes inherited from Space**

The documentation for this class was generated from the following files:

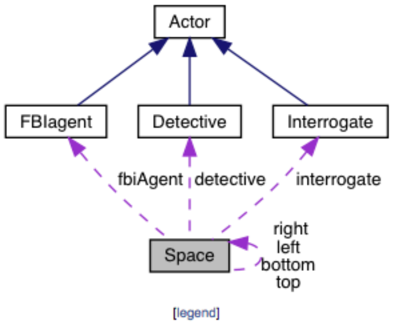
- **policestation.hpp**
- policestation.cpp

Space Class Reference abstract

Inheritance diagram for Space:



Collaboration diagram for Space:



Public Member Functions

virtual void	**menu** ()=0
void	**timeOut** ()

Public Attributes

Space *	**top**
Space *	**right**
Space *	**left**
Space *	**bottom**

Static Protected Attributes

static FBIagent	**fbiAgent**
static Detective	**detective**
static Interrogate	**interrogate**
static int	**turn** =0
static const int	**MAX_TURNS** = 24

Mindhunter Class Reference

Public Member Functions

void	**start** ()
void	**introduction** ()
void	**menu** ()

The documentation for this class was generated from the following files:

- **mindhunter.hpp**
- mindhunter.cpp