

Coding Rush Hour

Diagrams and Details



```
char type[] [] = {{'c', 'c', 'c', 'n', 'n', 'n'},
                  {'c', 'n', 'n', 'c', 'n', 'n'},
                  {'c', 'c', 'c', 'c', 'n', 'x'},
                  {'c', 'n', 'n', 'c', 'n', 'n'},
                  {'c', 'c', 'c', 'n', 'n', 'c'},
                  {'n', 'n', 'c', 'c', 'c', 'c'}};
```

```
int car[] [] = {{8, 2, 2, 0, 0, 0},
                {8, 0, 0, 9, 0, 0},
                {8, 0, 0, 9, 0, 0},
                {6, 0, 0, 9, 0, 0},
                {6, 1, 1, 0, 0, 5},
                {0, 0, 4, 4, 4, 5}};
```

```
char slice[] [] = {{'f', 'f', 'b', 'n', 'n', 'n'},
                  {'m', 'n', 'n', 'f', 'n', 'n'},
                  {'b', 'f', 'b', 'm', 'n', 'n'},
                  {'f', 'n', 'n', 'b', 'n', 'n'},
                  {'b', 'f', 'b', 'n', 'n', 'f'},
                  {'n', 'n', 'f', 'm', 'b', 'b'}};
```

```

else
{ //code to handle the game
  int n = Integer.parseInt (e.getActionCommand ());
  int x = n / col;
  int y = n % col;
  if (type [x] [y] == 'c')
  {
    whichCar = car [x] [y];
    findFront (x, y);
    currentPic.setIcon (createImageIcon ("little" + whichCar + ".png"));
  }
}
redraw ();

```

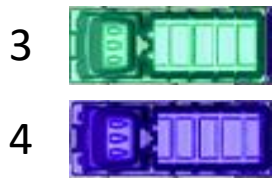
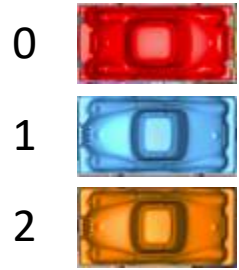


```

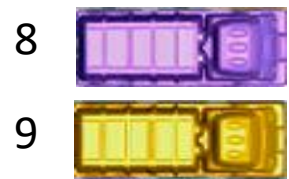
char type[] [] = {{'c', 'c', 'c', 'n', 'n', 'n'},
                  {'c', 'n', 'n', 'c', 'n', 'n'},
                  {'c', 'c', 'c', 'c', 'n', 'x'},
                  {'c', 'n', 'n', 'c', 'n', 'n'},
                  {'c', 'c', 'c', 'n', 'n', 'c'},
                  {'n', 'n', 'c', 'c', 'c', 'c'}};
int car[] [] = {{8, 2, 2, 0, 0, 0},
                {8, 0, 0, 9, 0, 0},
                {8, 0, 0, 9, 0, 0},
                {6, 0, 0, 9, 0, 0},
                {6, 1, 1, 0, 0, 5},
                {0, 0, 4, 4, 4, 5}};
//current Car info
JLabel currentPic;
int whichCar = 0;
int curX = 2;
int curY = 1;

```

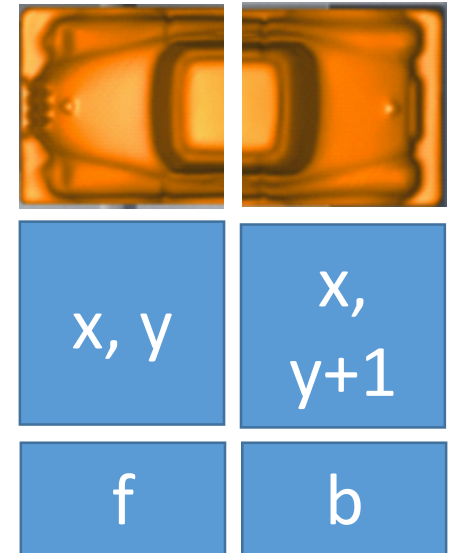
Left – Right



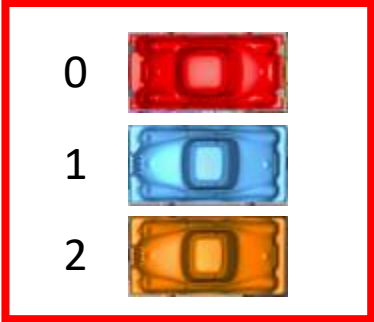
Up – Down



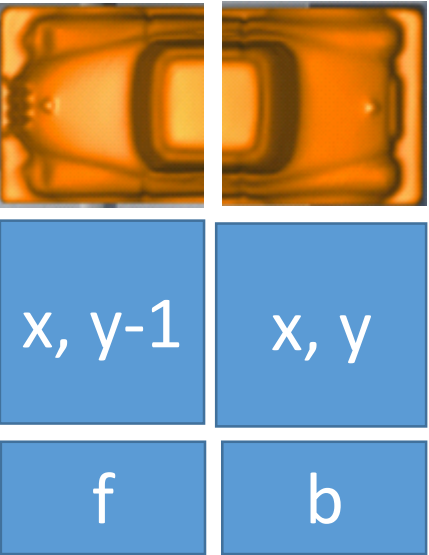
```
public void findFront (int x, int y)
    {
        if (slice [x] [y] == 'f')
        {
            curX = x;
            curY = y;
        }
    }
```



Left – Right



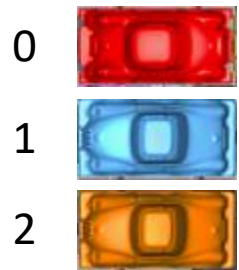
```
//2 piece cars, go left-right  
else if (whichCar <= 2 && slice [x] [y] == 'b')  
{  
    curX = x;  
    curY = y - 1;  
}
```



Up – Down



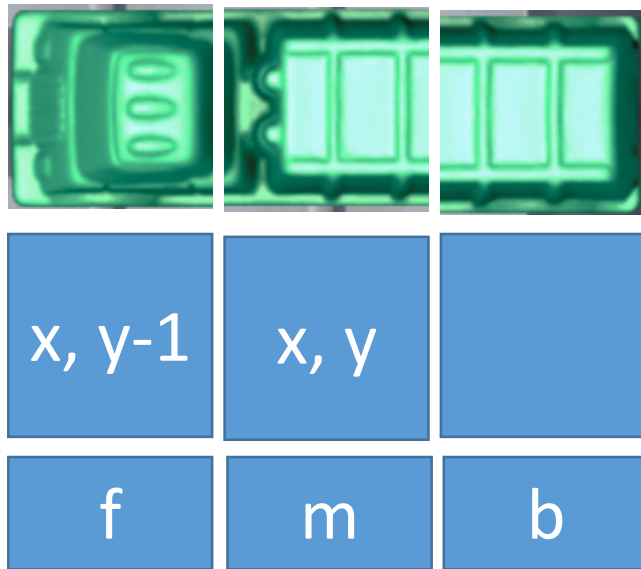
Left – Right



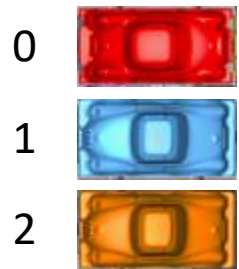
Up – Down



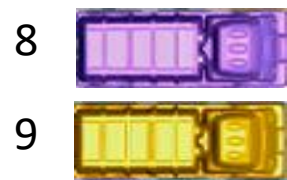
```
//3 piece trucks, go left-right  
else if (whichCar <= 4 && slice [x] [y] == 'm')  
{  
    curX = ___;  
    curY = ___;  
}
```



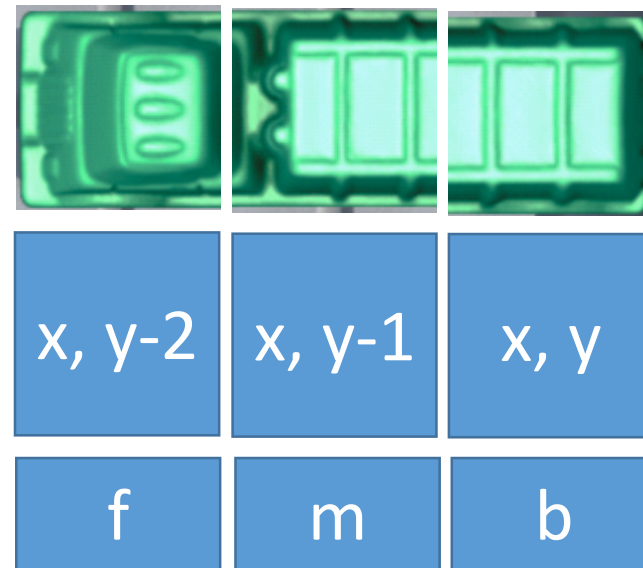
Left – Right



Up – Down



```
else if (whichCar <= 4 && slice [x] [y] == 'b')  
{  
    curX = ___;  
    curY = _____;  
}
```



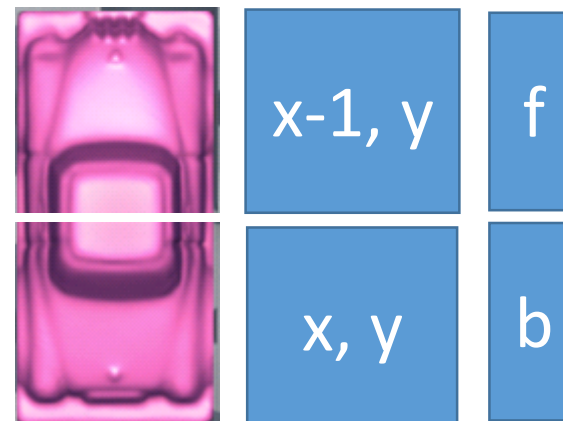
Left – Right



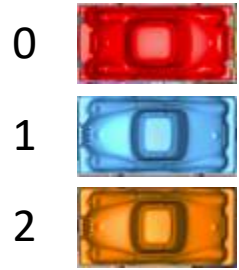
Up – Down



```
//2 piece cars, go up-down  
else if (whichCar <= 7 && slice [x] [y] == 'b')  
{  
    curX = ___;  
    curY = ___;  
}
```

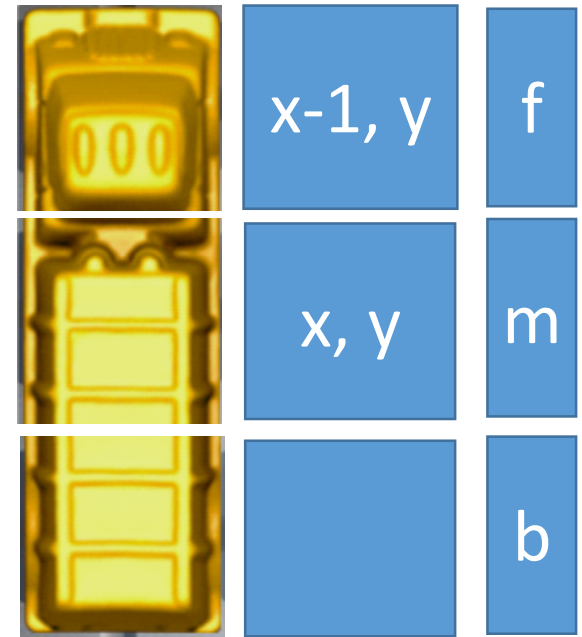
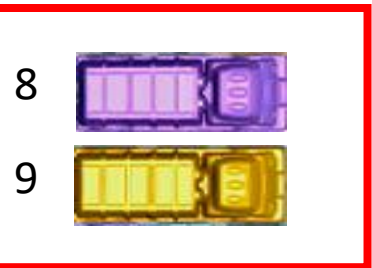


Left – Right



```
//3 piece trucks, go up-down  
else if (whichCar <= 9 && slice [x] [y] == 'm')  
{  
    curX = ___;  
    curY = ___;  
}
```

Up – Down



Left – Right

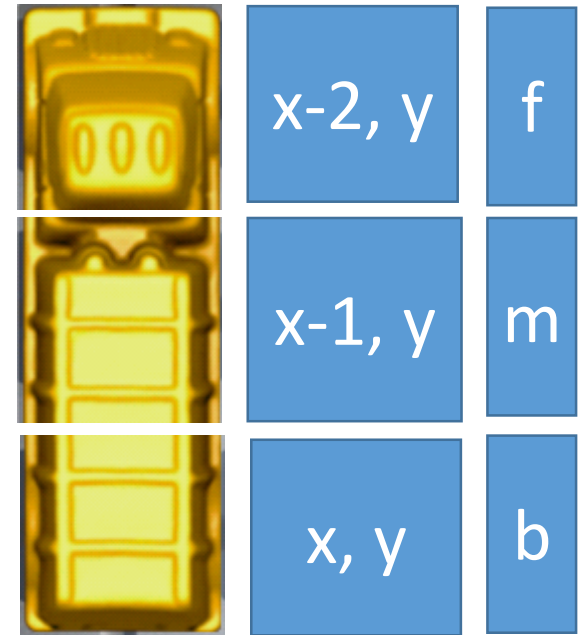


Up – Down



```
else if (whichCar <= 9 && slice [x] [y] == 'b')  
{  
    curX = ___;  
    curY = ___;  
}  
showStatus (curX + ", " + curY);  
}
```

Coding
findFront #7



```

public void actionPerformed (ActionEvent e)
{
    //moves between the screens
    if (e.getActionCommand ().equals ("s1"))
        cdLayout.show (p_card, "1");
    else if (e.getActionCommand ().equals ("s2"))
        cdLayout.show (p_card, "2");
    else if (e.getActionCommand ().equals ("s3"))
        cdLayout.show (p_card, "3");
    else if (e.getActionCommand ().equals ("s4"))
        cdLayout.show (p_card, "4");
    else if (e.getActionCommand ().equals ("s5"))
        cdLayout.show (p_card, "5");
    else if (e.getActionCommand ().equals ("s6"))
        System.exit (0);
    else if (e.getActionCommand ().equals ("up"))
        moveUp ();
    else if (e.getActionCommand ().equals ("down"))
        moveDown ();
    else if (e.getActionCommand ().equals ("left"))
        moveLeft ();
    else if (e.getActionCommand ().equals ("right"))
        moveRight ();
    else
    { //code to handle the game
        int n = Integer.parseInt (e.getActionCommand ());
        int x = n / col;
        int y = n % col;
        if (type [x] [y] == 'c')
        {
            whichCar = car [x] [y];
            findFront (x, y);
            currentPic.setIcon (createImageIcon ("little" + whichCar + ".png"));
        }
    }
    redraw ();
}
}

```



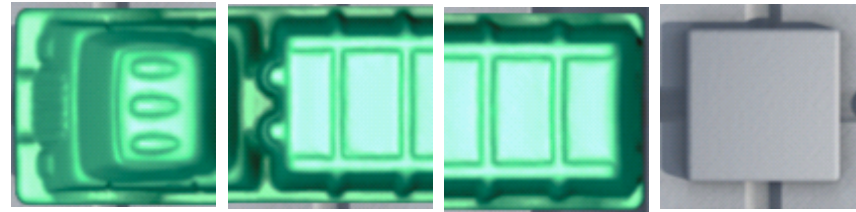
```
public void moveRight ()
{
    if (whichCar >= 5)
        showStatus ("Can't move left or right");
    //2 piece cars, go up-down
    else if (whichCar <= 2 && curY + 2 < col)
    {
        ?
    }

    //3 piece trucks, go up-down
    else if (whichCar <= 4 && curY + 3 < col)
    {
        ?
    }
}
```

3 piece
trucks

Moving
Right

Before



	curX, curY	curX, curY+1	curX, curY+2	curX, curY+3
--	---------------	-----------------	-----------------	-----------------

type

c	c	c	n
---	---	---	---

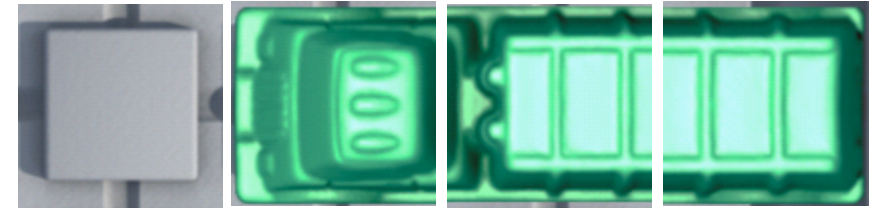
car

whichCar	whichCar	whichCar	0
----------	----------	----------	---

slice

f	m	b	n
---	---	---	---

After



curX, curY	curX, curY+1	curX, curY+2	curX, curY+3
---------------	-----------------	-----------------	-----------------

type

n	c	c	c
---	---	---	---

car

0	whichCar	whichCar	whichCar
---	----------	----------	----------

slice

n	f	m	b
---	---	---	---

```
//3 piece trucks, go right
```

```
else if (whichCar <= 4 && curY + 3 < col)
```

```
{
```

```
    type [curX] [curY] = 'n';
```

```
    type [curX] [curY + 1] = 'c';
```

```
    type [curX] [curY + 2] = 'c';
```

```
    type [curX] [curY + 3] = 'c';
```

```
    slice [curX] [curY] = 'n';
```

```
    slice [curX] [curY + 1] = 'f';
```

```
    slice [curX] [curY + 2] = 'm';
```

```
    slice [curX] [curY + 3] = 'b';
```

```
    car [curX] [curY] = 0;
```

```
    car [curX] [curY + 1] = whichCar;
```

```
    car [curX] [curY + 2] = whichCar;
```

```
    car [curX] [curY + 3] = whichCar;
```

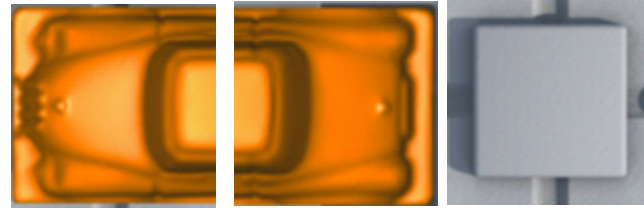
```
    curY++;
```

```
}
```

2 piece
cars

Moving
Right

Before



curX,
curY

curX,
curY+1

curX,
curY+2

type

c

c

n

car

whichCar

whichCar

0

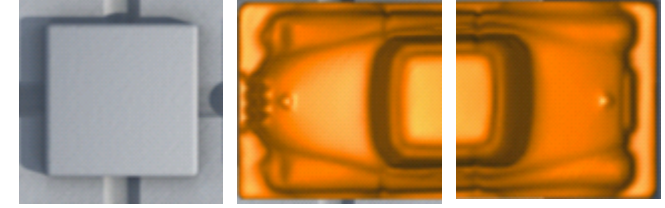
slice

f

b

n

After



curX,
curY

curX,
curY+1

curX,
curY+2

type

n

c

c

car

0

whichCar

whichCar

slice

n

f

b

```
//2 piece cars, go right
```

```
else if (whichCar <= 2 && curY + 2 < col)
```

```
{  
  type [curX] [curY] = '___';  
  type [curX] [curY + 1] = '___';  
  type [curX] [curY + 2] = '___';
```

```
  slice [curX] [curY] = '___';  
  slice [curX] [curY + 1] = '___';  
  slice [curX] [curY + 2] = '___';
```

```
  car [curX] [curY] = ___;  
  car [curX] [curY + 1] = ___;  
  car [curX] [curY + 2] = ___;  
  curY++;
```

```
}
```

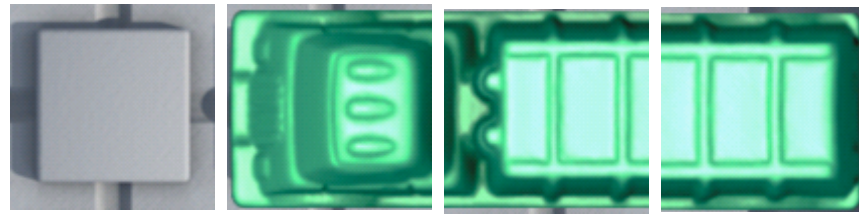
```
public void moveLeft ()
{
    if (whichCar >= 5)
        showStatus ("Can't move left or right");
    //2 piece cars, go left
    else if (whichCar <= 2 && curY - 1 >= 0)
    {
        ?
    }

    //3 piece trucks, go left
    else if (whichCar <= 4 && curY - 1 >= 0)
    {
        ?
    }
}
```

3 piece
trucks

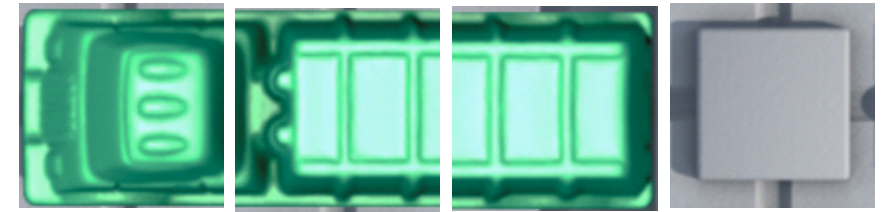
Moving
Left

Before



	curX, curY-1	curX, curY	curX, curY+1	curX, curY+2
type	n	c	c	c
car	0	whichCar	whichCar	whichCar
slice	n	f	m	b

After



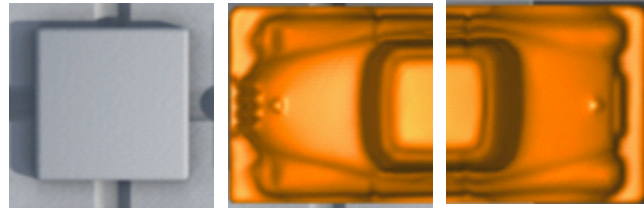
	curX, curY-1	curX, curY	curX, curY+1	curX, curY+2
type	c	c	c	n
car	whichCar	whichCar	whichCar	0
slice	f	m	b	n

```
//3 piece trucks, go left
else if (whichCar <= 4 && curY - 1 >= 0)
{
    type [curX] [curY - 1] = '__';
    type [curX] [curY] = '__';
    type [curX] [curY + 1] = '__';
    type [curX] [curY + 2] = '__';
    slice [curX] [curY - 1] = '__';
    slice [curX] [curY] = '__';
    slice [curX] [curY + 1] = '__';
    slice [curX] [curY + 2] = '__';
    car [curX] [curY - 1] = __;
    car [curX] [curY] = __;
    car [curX] [curY + 1] = __;
    car [curX] [curY + 2] = __;
    curY--;
}
```


2 piece
cars

Moving
Left

Before



curX,
curY-1

curX,
curY

curX,
curY+1

type

n

c

c

car

0

whichCar

whichCar

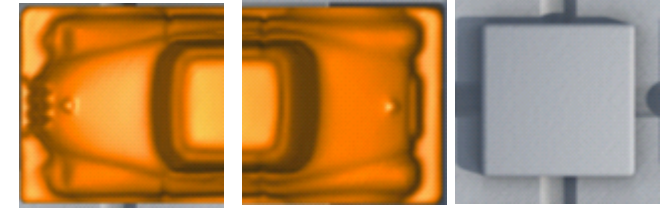
slice

n

f

b

After



curX,
curY-1

curX,
curY

curX,
curY+1

type

c

c

n

car

whichCar

whichCar

0

slice

f

b

n

```
//2 piece cars, go left
```

```
else if (whichCar <= 2 && curY - 1 >= 0)
```

```
{
```

```
    type [curX] [curY - 1] = '__';
```

```
    type [curX] [curY] = '__';
```

```
    type [curX] [curY + 1] = '__';
```

```
    slice [curX] [curY - 1] = '__';
```

```
    slice [curX] [curY] = '__';
```

```
    slice [curX] [curY + 1] = '__';
```

```
    car [curX] [curY - 1] = __;
```

```
    car [curX] [curY] = __;
```

```
    car [curX] [curY + 1] = __;
```

```
    curY--;
```

```
}
```




```
public void moveUp ()
{
    if (whichCar <= 4)
        showStatus ("Can't move up or down");
    //2 piece cars, go up-down
    else if (whichCar <= 7 && curX - 1 >= 0)
    {
        ?
    }

    //3 piece trucks, go up
    else if (whichCar <= 9 && curX - 1 >= 0)
    {
        ?
    }
}
```

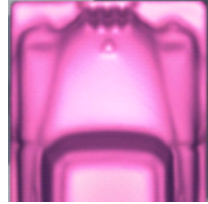


2 piece
cars

Moving
Up

Before

		type	car	slice
	curX-1, curY	n	0	n
	curX, curY	c	whichCar	f
	curX+1, curY	c	whichCar	b

After

		type	car	slice
	curX-1, curY	c	whichCar	f
	curX, curY	c	whichCar	b
	curX+1, curY	n	0	n

```
//2 piece cars, go up
```

```
else if (whichCar <= 7 && curX - 1 >= 0)
```

```
{
```

```
    type [curX - 1] [curY] = '___';
```

```
    type [curX] [curY] = '___';
```

```
    type [curX + 1] [curY] = '___';
```

```
    slice [curX - 1] [curY] = '___';
```

```
    slice [curX] [curY] = '___';
```

```
    slice [curX + 1] [curY] = '___';
```

```
    car [curX - 1] [curY] = ___;
```

```
    car [curX] [curY] = ___;
```

```
    car [curX + 1] [curY] = ___;
```





```
    curX--;
```

```
}
```





3 piece
trucks

Moving
Up

Before

		type	car	slice
	curX-1, curY	n	0	n
	curX, curY	c	whichCar	f
	curX+1, curY	c	whichCar	m
	curX+2, curY	c	whichCar	b

After

		type	car	slice
	curX-1, curY	c	whichCar	f
	curX, curY	c	whichCar	m
	curX+1, curY	c	whichCar	b
	curX+2, curY	n	0	n

```
//3 piece trucks, go up
```

```
else if (whichCar <= 9 && curX - 1 >= 0)
```

```
{
```

```
    type [curX - 1] [curY] = '___';
```

```
    type [curX] [curY] = '___';
```

```
    type [curX + 1] [curY] = '___';
```

```
    type [curX + 2] [curY] = '___';
```

```
    slice [curX - 1] [curY] = '___';
```

```
    slice [curX] [curY] = '___';
```

```
    slice [curX + 1] [curY] = '___';
```

```
    slice [curX + 2] [curY] = '___';
```

```
    car [curX - 1] [curY] = ___;
```

```
    car [curX] [curY] = ___;
```

```
    car [curX + 1] [curY] = ___;
```

```
    car [curX + 2] [curY] = ___;
```

```
    curX--;
```




```
}
```

```
public void moveDown ()
{
    if (whichCar <= 4)
        showStatus ("Can't move up or down");
    //2 piece cars, go down
    else if (whichCar <= 7 && curX + 2 < row)
    {
        ?
    }
    //3 piece trucks, go down
    else if (whichCar <= 9 && curX + 3 < row)
    {
        ?
    }
}
```




2 piece
cars

Moving
Down

Before

		type	car	slice
	curX, curY	c	whichCar	f
	curX+1, curY	c	whichCar	b
	curX+2, curY	n	0	n

After

		type	car	slice
	curX, curY	n	0	n
	curX+1, curY	c	whichCar	f
	curX+2, curY	c	whichCar	b

```
//2 piece cars, go down
```

```
else if (whichCar <= 7 && curX + 2 < row)
```

```
{
```

```
    type [curX] [curY] = '___';
```

```
    type [curX + 1] [curY] = '___';
```

```
    type [curX + 2] [curY] = '___';
```

```
    slice [curX] [curY] = '___';
```

```
    slice [curX + 1] [curY] = '___';
```

```
    slice [curX + 2] [curY] = '___';
```

```
    car [curX] [curY] = ___;
```

```
    car [curX + 1] [curY] = ___;
```





```
    car [curX + 2] [curY] = ___;
```

```
    curX++;
```





```
}
```

3 piece
trucks
Moving
Down

Before

		type	car	slice
	curX, curY	c	whichCar	f
	curX+1, curY	c	whichCar	m
	curX+2, curY	c	whichCar	b
	curX+3, curY	n	0	n

After

		type	car	slice
	curX, curY	n	0	n
	curX+1, curY	c	whichCar	f
	curX+2, curY	c	whichCar	m
	curX+3, curY	c	whichCar	b

```
//3 piece trucks, go down
else if (whichCar <= 9 && curX + 3 < row)
{
    type [curX] [curY] = '_';
    type [curX + 1] [curY] = '_';
    type [curX + 2] [curY] = '_';
    type [curX + 3] [curY] = '_';
    slice [curX] [curY] = '_';
    slice [curX + 1] [curY] = '_';
    slice [curX + 2] [curY] = '_';
    slice [curX + 3] [curY] = '_';
    car [curX] [curY] = __;
    car [curX + 1] [curY] = __;
    car [curX + 2] [curY] = __;
    car [curX + 3] [curY] = __;
    curX++;
}
```