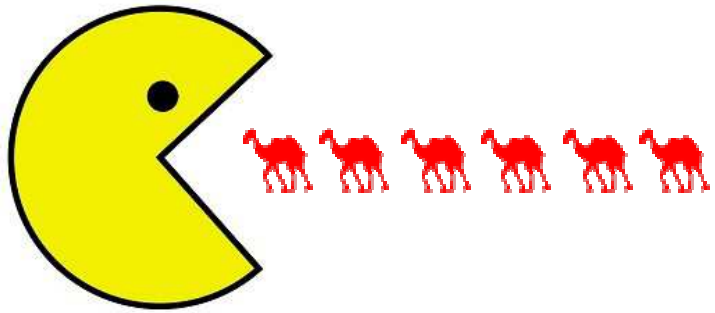
Pacaml

Pac-Man Game Programming Language

Chun-Kang Chen/Hui-Hsiang Kuo/Wenxin Zhu/Shuwei Cao

Overview

- PaCaml = Pac-Man + Ocaml
- A game programming language facilitating the design of elements in PAC-MAN scene
- Simple – people with little experience in programming can do the work / a good inspiration for children
- Interesting – be the God in the world of Pac-Man



Motivation

- Pac-Man: an arcade game immensely popular since its original release
- Pac-Man has made a great impact on a generation of people and is still appealing to the public for today
- Recall ourselves of the best memory in childhood

Game-specified Types

Name	Field	Value
Map	…	Add contain by build-in functions
Point	x	Integer
	y	Integer
Player	p_point	Point
Item	type	_GHOST _BARRIER _GIFT
	i_point	Point
	level	GHOST: _EASY, _NORMAL,_HARD GIFT: _GIFT_SPEEDUP _GIFT_SLOWDOWN _GIFT_KILLER _GIFT_SCORE
	duration	Integer

Built-in Functions

Return	Function Name
bool	setPlayer(player pacman)
bool	addGhost (item ghost)
bool	addBarrier (item barrier)
bool	addGift (item gift)
int	getMapWidth ()
int	getMapHeight ()
string	getMapItemName (point p)
bool	isPointAvailable (point p)
bool	getAvailablePoint ()

Tutorial

```
item createItem( int type, int x, int y)
{
    item i;
    point p;
    p.x = x;
    p.y = y;
    i.i_type = type;
    i.i_point = p;
    return i;
}
void main()
{
    player pacman;
    item ghost1;
    item barrier1;
    item gift1;
```

Type: item

main function

Type: player

```
    // initiate a pacman
pacman.p_point = getAvailablePoint();
setPlayer(pacman);
    // initiate a ghost
ghost1 = createItem(_GHOST, 5, 5);
addGhost(ghost1);
    // initiate a barrier
barrier1 = createItem(_BARRIER, 8 , 8);
addBarrier(barrier1);
    // initiate a gift
gift1 = createItem(_GIFT, 8 , 9 );
addGift(gift1);
play();
}
```

Place the player

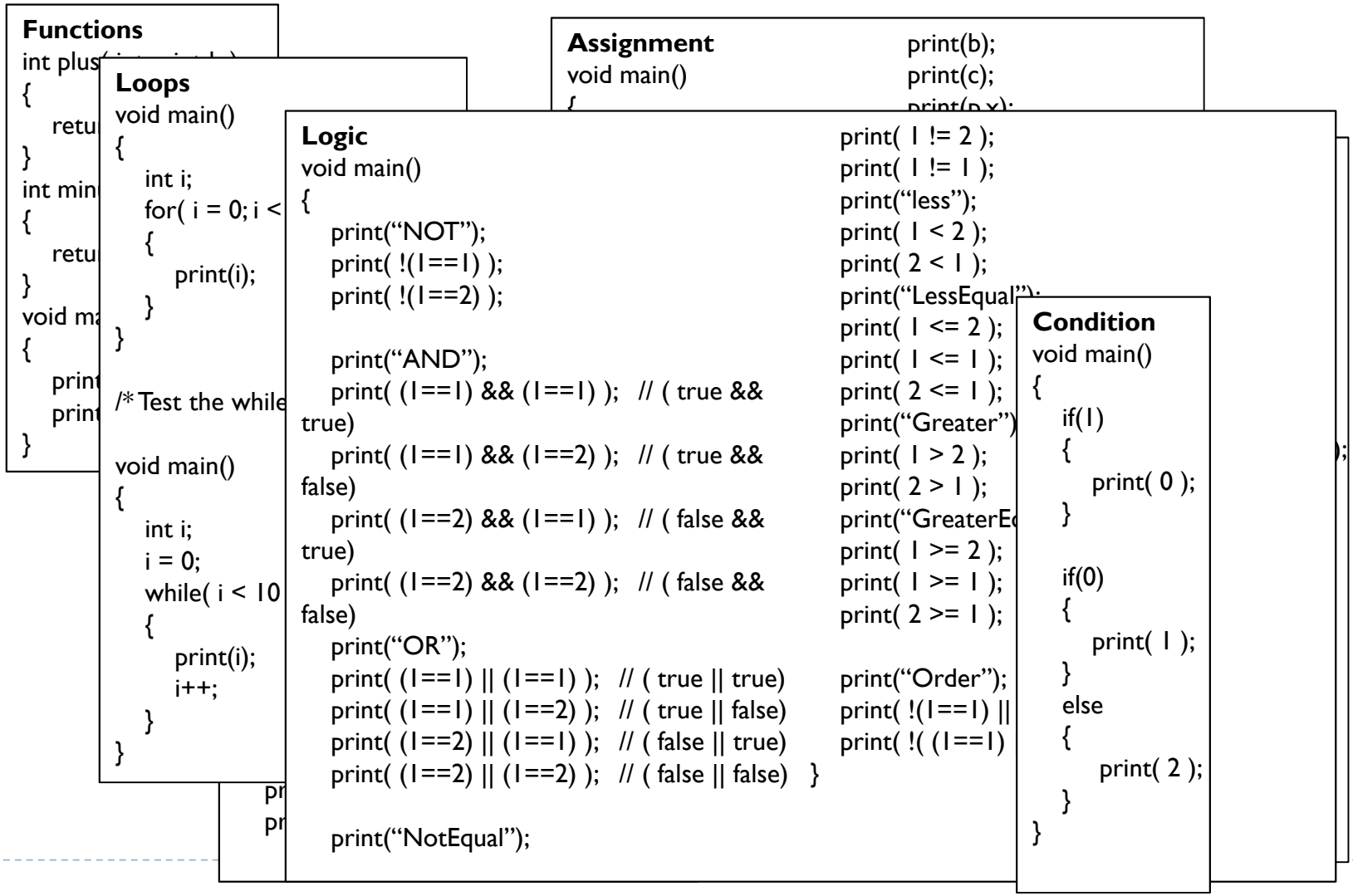
Place the ghost

Place the barrier

Place the gift

Play the game

Tutorial



Functions

```
int plus(
{
    retu
}
int min
{
    retu
}
void ma
{
    print
    print
}
```

Loops

```
void main()
{
    int i;
    for( i = 0; i <
    {
        print(i);
    }
}

/* Test the while

void main()
{
    int i;
    i = 0;
    while( i < 10
    {
        print(i);
        i++;
    }
}
```

Assignment

```
void main()
{
                                print(b);
                                print(c);
                                print(p.x);
```

Logic

```
void main()
{
    print("NOT");
    print( !(1==1) );
    print( !(1==2) );

    print("AND");
    print( (1==1) && (1==1) );   // ( true &&
true)
    print( (1==1) && (1==2) );   // ( true &&
false)
    print( (1==2) && (1==1) );   // ( false &&
true)
    print( (1==2) && (1==2) );   // ( false &&
false)
    print("OR");
    print( (1==1) || (1==1) );   // ( true || true)
    print( (1==1) || (1==2) );   // ( true || false)
    print( (1==2) || (1==1) );   // ( false || true)
    print( (1==2) || (1==2) );   // ( false || false)

    print("NotEqual");
    print( 1 != 2 );
    print( 1 != 1 );
    print("less");
    print( 1 < 2 );
    print( 2 < 1 );
    print("LessEqual");
    print( 1 <= 2 );
    print( 1 <= 1 );
    print( 2 <= 1 );
    print("Greater")
    print( 1 > 2 );
    print( 2 > 1 );
    print("GreaterE
    print( 1 >= 2 );
    print( 1 >= 1 );
    print( 2 >= 1 );

    print("Order");
    print( !(1==1) ||
    print( !( (1==1)
}
```

Condition

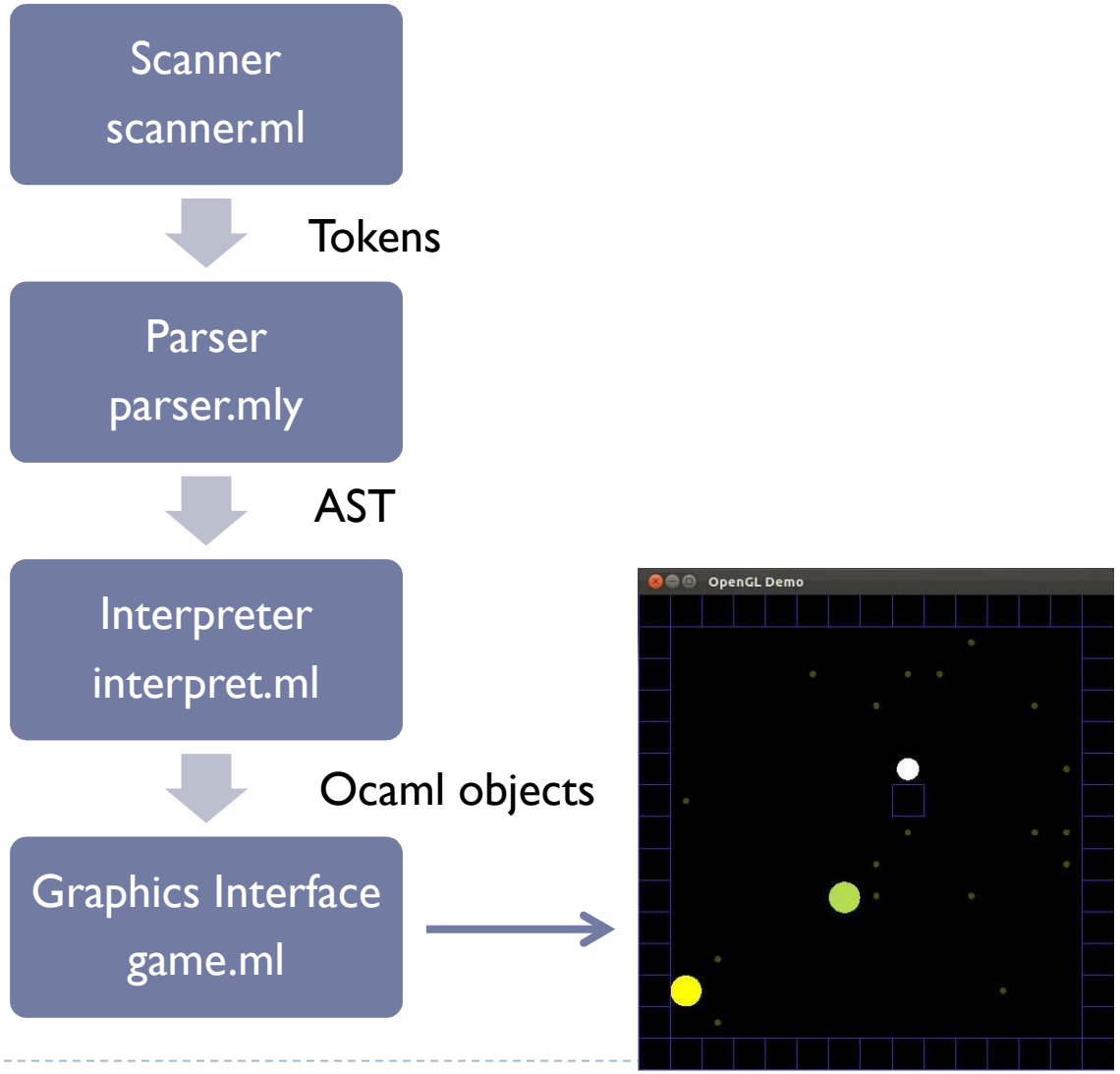
```
void main()
{
    if(1)
    {
        print( 0 );
    }

    if(0)
    {
        print( 1 );
    }
    else
    {
        print( 2 );
    }
}
```

Implementation

example.pacaml

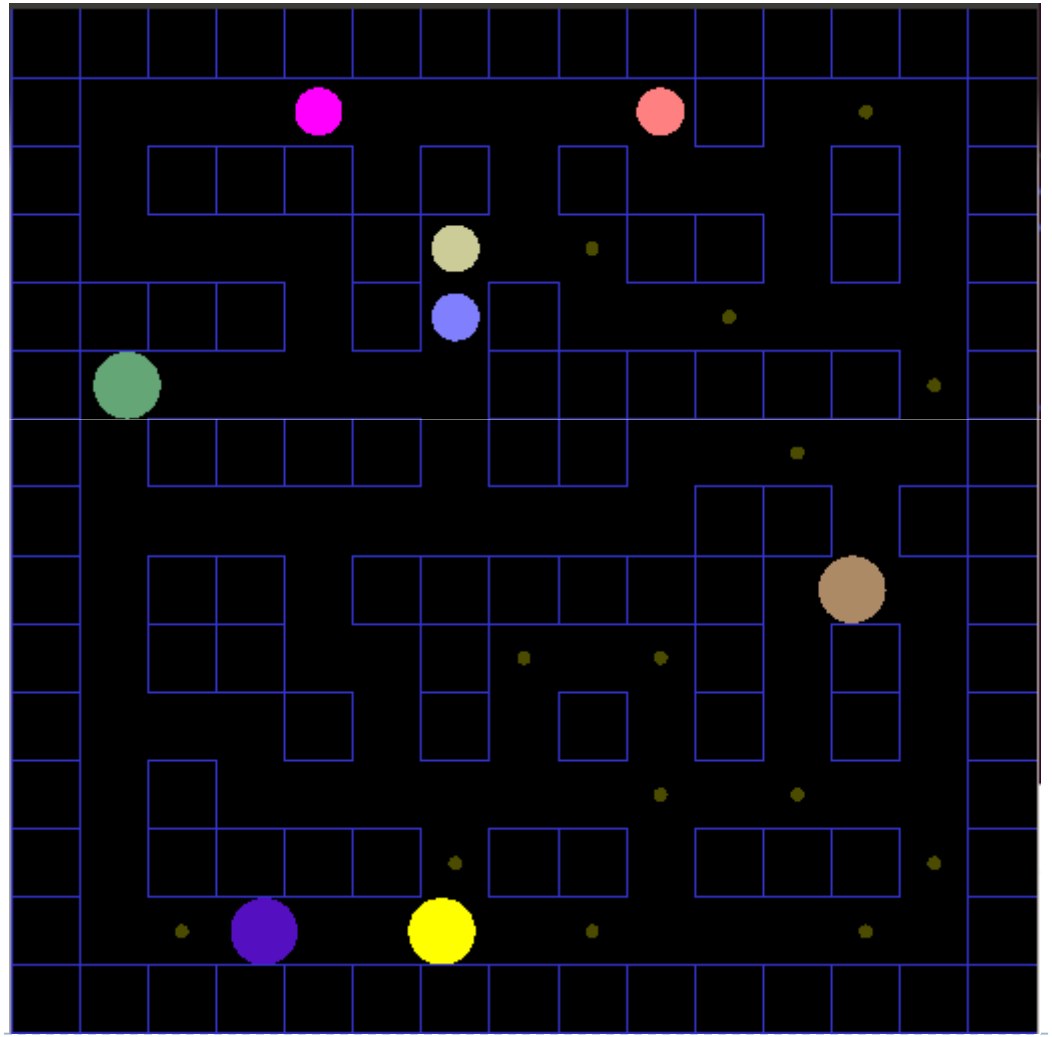
```
/*      Pacaml     */
void main()
{
    player pacman;
    item ghost1;
    item barrier1;
    item gift1;
    pacman.p_point = getAvailablePoint();
    setPlayer(pacman);
                        .
                        .
                        .
    play();
}
```



Lessons Learned

- Start early, otherwise you will stay up late
- A good plan is a half success
- Ocaml is hard to get familiar with, so practice makes perfect
- Debug is not easy. It creates more when you fix some.
- More testing, less errors

Demo Time



Pa aml