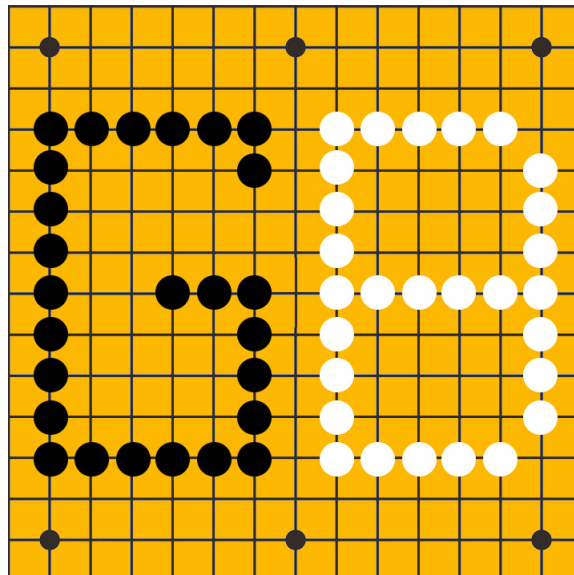

GBL: Game Building Language

Yiqing Cui(yc3121) Sihao Zhang(sz2558) Ye Cao(yc3113) Shengtong Zhang(sz2539)
May 11, 2016



CONTENTS

1	Introduction	4
2	Language Tutorial	4
3	Language Manual	6
3.1	Syntax Notations	6
3.2	Lexical Conventions	6
3.2.1	Comments	6
3.2.2	Identifiers	6
3.2.3	Keywords	7
3.2.4	Operators	7
3.2.5	Separators	8
3.2.6	Semicolons	8
3.3	Types	8
3.3.1	Primitive Types	8
3.3.2	Class	9
3.3.3	Game Class	9
3.3.4	AI Class	10
3.3.5	Main Class	10
3.4	Expressions and Operators	10
3.4.1	Expressions	10
3.4.2	Arithmetic operators	10
3.4.3	Array Access operators	11
3.4.4	Comparison operators	11
3.4.5	Logical operators	11
3.4.6	Assignment operators	11
3.4.7	Member Access operators	12
3.4.8	Operator Precedence and Associative Property	12
3.5	Statements	12
3.5.1	overview	12
3.5.2	assignment	12
3.5.3	print	13
3.5.4	if	13
3.5.5	while	14
3.5.6	for	14
3.5.7	continue	14
3.5.8	break	15
3.5.9	return	15
3.6	Functions	15
3.6.1	Function definition	16
3.6.2	Function calls	16
3.6.3	Recursion	16

3.7	Program Structure and Scope	17
3.7.1	Program Structure	17
3.7.2	Scope	17
4	Project Plan	19
4.1	processes	19
4.1.1	Planning:	19
4.1.2	Specification:	19
4.1.3	Development:	19
4.1.4	Testing:	19
4.2	Programming style guide	20
4.3	Project Timeline	20
4.4	Roles and Responsibility	20
4.5	Software development environment and tools	21
4.6	Project Log	21
5	Architectural Design	61
5.1	Architectural Diagram	61
5.2	Lexical Analysis	61
5.3	Syntactic Analysis	61
5.4	Semantic Analysis	61
5.5	Code Generation	61
6	Learned from course	62
7	Test Plan	62
7.1	Unit Test	63
8	Appendix	66

1 INTRODUCTION

Most of us grows up with games, but it is extremely difficult to create a game from zero. We would like to provide such a language which could help game developers to generate their games easily and fast. So here comes Game Building Language (GBL). GBL is a user-friendly programming language for game building, and our compiler will compile it to LLVM. Game developers can use GBL to create any games based on coordinates easily, from Gobang, Chess to Tactical War games (e.g. Fire Emblem), even Role-playing Game (Dragon Quest). The only thing that game developers need to do is providing the game rule, the map information, the picture of items and related settings. Besides, there are lots of build-in function and data type in GBL which could save lots of time for users. We are seeking to develop an intuitive and efficient language so users could focus more time on the design of games not on the technical process.

With our proposed language, we aim to let users:

1. create the coordinate-based game easily.
2. intuitively learn and develop.
3. spend the least time doing most work.
4. have a wonderful time developing experience.

2 LANGUAGE TUTORIAL

In the tutorial, we will walk you through creating your first GBL program.

Users could refer to our program "initial.gbl", which provide the user an architect of a board game they may use.

We divide a game into five parts:

- (1) Initialize: the initial value of a game.
- (2) Win condition: in what condition which player will win.
- (3) Is the sprite legal: whether the operation of a player makes is right?
- (4) Update the game: when the player place a sprite, what will change in the board.
- (5) AI (optional): users could write the AI of themselves.

Thus, the initial GBL file will include the content as "initial.gbl".

Then we move to how to create a GBL program.

Step 1: Creating your .gbl source file

Create a file ends with .gbl.

Step 2: Add Game information.

Users can refer to "initial.gbl" and add the five parts in the .gbl file. There are three build-in classes named Game, AI and Main. Users should create their class extend from these three classes first, then add Initialize information in Main-child class, add Win condition, Islegal, Update information in Game-child class, add AI in AI-child class. We explain every parts as follows:

2.1 Initialize. In this part, users may write the initial information of the game, e.g. the size of map, the number of players, the name of players, if user fight with an AI and so on.

2.2 Win condition: In this part, users may add the win condition, give the information saying that in what condition, which player will win. For example, in game Gobang, the sprites of any player give a line, of which the number of sprites is larger than or equal to five, then the player win. The return of Win function is the Winner ID.

2.3 Islegal: In this part, users may give the result whether the operation of a player makes is right. For example, in game Gobang, a legal operation is that a player put a sprite in a vacant grid. The return of Islegal function is a bool variable indicating whether it is legal.

2.4 Update: In this part, users may give the information that when a player make a right operation, what will happened in the board. For example, in game Gobang, the Update is that a sprite is placed in a vacant grid. But in other games, some sprites may be delete from the board. The return of Update function is the coordinates of sprites which will be deleted.

2.5 AI: In this part, users could write their smart AI to fight with players. The return of AI function is the sprite position AI give.

Step 3: Create .py file.

Run: `cat <your file name>.gbl | ./microc.native > gbloutput.py`

Step 4: Run gbloutput.py

Run: `python gbloutput.py`

Step 5: Play your game!

The player could choose where they want to put a sprite by click on a grid. Enjoy!

3 LANGUAGE MANUAL

3.1 SYNTAX NOTATIONS

Some special syntax notations may be used to state the lexical and syntactic rules in this document:

1. Brackets [] enclose optional items.
2. Parenthesis () means the notation.
3. Asterisks * indicate that items could repeat zero or lots of times.
4. Dash - means an area from the former to the latter.

3.2 LEXICAL CONVENTIONS

This chapter presents the lexical conventions of GBL, describing which tokens are valid, including comments, naming convention of identifiers, reserved keywords, operators, separators, new line and braces, and give the developers a right start to use GBL to avoid lexical errors.

3.2.1 COMMENTS

Our language only supports single line comment. It is made with a leading # in the line:

Example:

```
# This is a comment
```

3.2.2 IDENTIFIERS

An identifier of GBL is a case-sensitive string which starts with a letter or an underscore. You could add an optionally characters which consist of letters, underscores and numbers. An identifier could not be any Keywords (see next subsection). The length of an identifier is from 1 to 256.

We could write a regular expression of an identifier as follows:

```
[a-z or A-Z] [a-z or A-Z or 0-9 or _](optional) (length of the  
identifier should be 1 to 256)
```

Here are some legal examples:

```
a a1 ab a_ abc1
```

Here are some illegal examples:

```
1a %a a@ for int
```

3.2.3 KEYWORDS

Our keywords are divided into six types:

1. Basic Data Keywords

```
int double bool string
```

2. Composite Data Keywords

```
int[] double[] bool[] AI game
```

3. Basic Flow Control Keywords

```
for if else break continue while
```

4. Basic Logic Keywords

```
and or not
```

5. Basic Compare Keywords

```
> < >= <= ==
```

6. Other keywords

```
class void printi printd printb prints printlni printlnd printlnb  
printlns
```

3.2.4 OPERATORS

An operator is a token which leads to an operation on one, two or three operands. We divide operators into six groups. You can refer them in Chapter 5 Expressions and Operators.

1. Arithmetic Operators

```
+ - * / %
```

2. Assignment Operators

```
=
```

3. Domain Operators

```
@
```

4. Parenthesis Operators

```
[ ] ( ) { }
```

5. Comment Operators

```
# /**/
```

3.2.5 SEPARATORS

We use separators to separate tokens. The separators includes space, () and [].

3.2.6 SEMICOLONS

We use semicolons to separator different subsections. Example:

```
if (a == 1)
{
    b = 1;
}
else
{
    b = 0;
}
```

3.3 TYPES

3.3.1 PRIMITIVE TYPES

3.3.1.1 INT The *int* data type is signed 32 bit type integral number which is expressed in decimal. And it ranges from -2^{31} to $2^{31} - 1$.

Example:

```
int x = 12;
int y = -40;
```

3.3.1.2 DOUBLE The *double* data type is signed 64 bit type floating number which is expressed in decimal.

Example:

```
double x = 1.23;
double y = -0.25;
double z = 1.0;
```

3.3.1.3 BOOL The *bool* data type is 1 bit type logical value which is either true or false.

Example:

```
bool x = true;
bool y = false;
```


3.3.1.4 STRING The *string* data type is sequence of 0 or more characters. And each character is stored by char type.

Example:

```
string x = "";  
string y = "reference manual";
```

3.3.1.5 ARRAY The *array* data type is sequence of 0 or more elements. The data type of each element should be declared in advance.

example:

```
int x[10];  
string s[10];  
x[0] = 10;  
s[1] = "test";
```

3.3.2 CLASS

Programmer can extend their own class with keyword *class* and *extends*. Classes must be inherited from our base classes, and user can define their own functions and variables:

```
class identifier extends {base_class}{type-declare-list function-declare-list}
```

We can define new structure as following:

```
class mygame extends Game  
{  
int a;  
int b;  
void test()  
{  
return 0;  
}  
}  
  
#define a new class "mygame", which has attributes a, b and function  
test
```

and then use the new class as following:

```
mygame newgame;  
a@newgame = 10;  
test@newgame;
```

3.3.3 GAME CLASS

The game object, which has attributes such as player number, player list and sprite list. And it controls the status of the whole game. And it has following build-in functions:

1. initialize() : Programmer can use this function to initialize other variables in the game

Example:

```
initialize()@Game;  
#initialize the game object
```

2. Win() : Programmer can use this function to judge if any player win the game
3. Islegal() : Programmer can use this function to set how to judge if the action of sprite is legal
4. Update() : Programmer can use this function to set what to do in the game

3.3.4 AI CLASS

The AI object, user can define their own AI function in AI class.

3.3.5 MAIN CLASS

The Main class, user must define main function in Main class as the entrance of program.

3.4 EXPRESSIONS AND OPERATORS

This subsection describes the expression and operators in GBL.

3.4.1 EXPRESSIONS

Expression is a combination of at least one operand and zero or more operators, in which operand can be constant, variables, objects and so on, operators can be one of +, -, *, /, % or other function call. Some expressions examples are listed as follow

```
120  
34 * 56 + 9  
12.5 - c  
add(37, a)
```

Also, expressions can be grouped by parentheses, such as

```
(5+7) / (4+6)
```

Note that in above expression, the meaning of parentheses is different of which in function call.

3.4.2 ARITHMETIC OPERATORS

GBL supports standard arithmetic operators, including addition, subtraction, multiplication, division, module and negation. Note that for a specific binary operator, the two operands should be the same size. Here are some examples:

```
5 - 3           # subtraction, result is 2  
y * 5         # multiplication, result is 5y, in which y must be a  
              integer  
9 / 4         # division, returns the quotient part, which is 2  
2 + - 1.2     # error, the first operand is an integer while the  
              second one is float number
```

```
25 % 7           # module, result is 4
```

3.4.3 ARRAY ACCESS OPERATORS

If we want to randomly access an element among an array, we can use [] operators. Note that for an array, the index starts from 0. For example

```
A[9]           # returns the 10th element of array A, whose index is 9.
```

3.4.4 COMPARISON OPERATORS

GBL also supports different comparison operators, including >=, <=, >, <, ==, !=. These operators are derived from the corresponding English words, which is very easy to understand. In fact, comparison is a kind of special binary operators in that the return value is always Boolean.

```
a > b           # returns true if a is greater than b
a < b           # returns true if a is less than b
a >= b         # returns true if a is greater than or equal to b
a <= b         # returns true if a is less than or equal to b
a == b         # returns true if a is equal to b
a != b         # returns true if a is not equal to b
```

3.4.5 LOGICAL OPERATORS

GBL supports logical operators including and, or, not. They test the value of multiple conditions and returns a Boolean value.

```
a > b and b > c      # returns true is a>b and b>c holds at the same
                    time
a>b or b>c          # returns true if a>b or b>c
not a<b            # returns true if a<b does not hold
```

3.4.6 ASSIGNMENT OPERATORS

Like other programming languages, assignment operator stores a particular value into a variable. Also, assignment operator has a return value, which is the assigned value itself. This enables the feature of 'consecutive assignment'. Here are some examples

```
A = 5;           # stores 5 into variable A
d = foo(12);     # stores the return value of foo(12) into variable d
Z = y = 9;       # stores 9 into y, stores y into Z
```

3.4.7 MEMBER ACCESS OPERATORS

In GBL, the member access operator is denoted by symbol '@'. The usage of this symbol is quite intuitive. Here are some examples

```
fstVal@Object = 12;
```

3.4.8 OPERATOR PRECEDENCE AND ASSOCIATIVE PROPERTY

When a complicated expression being evaluated, the different operator precedence and associative property will lead to significant different results. Therefore, the precedence and associative property must be well-defined to rule out all ambiguity. Following is the list of precedence, small number means higher precedence. Same precedence shares the same number. All evaluation within the same precedence are occurred from left to right unless

1. Function call
2. Member access operator
3. parentheses
4. Unary operator
5. Multiplication; division; module
6. Addition; subtraction
7. Comparison operators
8. Logical and; logical or
9. Logical not
10. Assignment operator

3.5 STATEMENTS

3.5.1 OVERVIEW

In programming language, a statement is the smallest standalone element that express some action to be carried out. There are two kinds of statement, simple statement (usually in one line) or a compound statement (may contain statements as components).

3.5.2 ASSIGNMENT

Assignment statement is to assign a left value or right value to a left value, assign value to a right value is forbidden cause right value doesn't have a space in memory, the syntax is:

```
|| Lvalue = RValue;  
|| LValue = LValue;
```

Example:

```
int i = 3;
#Assign 3 to a integer value i
int j = i;

#Assign variable i's value to j
3 = i;
#Error, try to assign a right value
```

3.5.3 PRINT

print evaluates the expression and writes the resulting object to standard output as a string, and automatically create a new line, the syntax is:

```
|| printi/printd/prints/printb/printlni/printld/printlns/printlnb/ (
    expression)
```

Example:

```
prints ("Hello World");
#Output: Hello World
a=1;
printi (a);
#Output: 1
b=1;
printi (a+b);
#Output: 2
```

3.5.4 IF

If-Elif-Else statement is a conditional statement. It is expected to take one or multiple Boolean conditions, controlled by single If, multiple Elif and one Else. Then it evaluates each condition in sequence until it finds one satisfied condition or exits. The syntax is:

```
|| if (condition)
    {statements}
    (else
    {statements})?
```

Example:

```
int i=4;
if (i)
{
    printi (i);
    i = i - 1;
}
#Output: 4321
```

3.5.5 WHILE

While statement is a conditional statement. It allows code to be executed repeatedly based on a given boolean condition. It evaluates the condition repeatedly until it finds one can't meet the condition. The syntax is:

```
|| while (condition)
   | {statement}
```

Example:

```
int i = 3;
while(i > 1)
{
i = i - 1;
}
```

3.5.6 FOR

The for statement provides a compact way to iterate over a range of values, it repeatedly loops until a particular condition is satisfied. The syntax is:

```
|| for (expression; expression; expression)
   | {statements}
```

Example:

```
int i = 3;
for (; i < 1; i = i + 1)
{
println(i);
}
```

3.5.7 CONTINUE

The continue statement causes a jump to the end of current loop, including for and while loop.

Example:

```
#In while loop:
int i=9;
while (i>=0)
{
  if (i==5)
  {
    continue;
  }
  i = i - 1;
  print(i);
}
#Output: 987643210
```

3.5.8 BREAK

The continue statement causes a jump to the end of current loop, including for and while loop.

Example:

```
#In while loop:
int i=9;
while (i>=0)
{
    if (i==5)
    {
        break;
    }
    i = i - 1;
    printi (i);
}
#Output: 9876
```

3.5.9 RETURN

A return statement causes execution to leave the current subroutine and resume at the point in the code immediately after where the subroutine was called, known as its return address. A function is allowed to specify a return value to be passed back to the code that called the function.

```
|| return true;
|| return 1;
```

Example:

```
int func1()
{
    return 1;
}

void func2()
{
    return 2;
}

int main()
{
    printi (func1());
    #Output: 1
    printi (func2());
    #Error: return type doesn't match the function's return type
    return 0;
}
```

3.6 FUNCTIONS

3.6.1 FUNCTION DEFINITION

Function can be defined with its name and return type. The format of function definition is as following:

return-type function-name (type-declare-list) {function-body}

return-type should in assist with the return type of *function-body*. For function with no return value, programmer can use key word *func* to define it.

Here are examples of function definition:

```
void printwords(string v1, string v2)
{
    prints (v1);
    prints (v2);
}
#define function to print two strings

int add(int v1, int v2)
{
    return (v1 + v2);
}
#define function to add two integers
```

3.6.2 FUNCTION CALLS

Functions can be called with its name and appropriate parameters. The format of function calling is as following:

function-name(parameter-1, parameter-2, parameter-3,...)

Also for function which has return value, we can use it as a sub expression and the format is as following:

variable = function-name(parameter-1, parameter-2, parameter-3,...)

Here are examples of function calling:

```
printwords("start", "end");
#print string start and end
res = add(1, 2);
#add integer 1 and 2
```

3.6.3 RECURSION

Function can call itself to do some complex things. The example for function recursion is as following:

```
int fact(int val)
{
    if(val == 1)
```



```
{
  return 1;
}
return val * fact(val - 1);
}
```

3.7 PROGRAM STRUCTURE AND SCOPE

3.7.1 PROGRAM STRUCTURE

GBL's program must exist in a single source file, with a ".gbl" extension. It consists of a number of functions declarations and exactly one main function as the program entrance.

The position of function declarations in the source code doesn't matter, that means a function can be called first and defined later in the source code.

Example:

```
int func1()
{
  return 1;
}

string func2()
{
  return (": Hello World");
}

int main(){
  printi (func1());
  prints (func2());
  return 0;
}
#Output: 1: Hello World
```

3.7.2 SCOPE

Variables in GBL declared within a block is accessible only within that block and blocks enclosed by it, and only after the point of declaration. A block is defined by braces "", including function block, conditional block, for-loop block, while-loop block. A global variable is defined in main scope which is out of all braces.

Example:

```
int func1()
{
  int i=5;
  int j=10;
  return (i+j);
}
int i=10;

int main(){
```

```
printi (func1());  
#Output: 15  
printi (i);  
#Output: 10  
printi (j);  
#Error: variable 'j' not found  
return 0;  
}
```

4 PROJECT PLAN

4.1 PROCESSES

4.1.1 PLANNING:

The planning process is not a short term for us. We spent 2 weeks discussing on what kind of field we want to step in. We considered many possible applications, like role-play game, music editing, video editing and so on. We finally decided to create a new language which focuses on building chess board game, since there are so many different kinds of board games and we figured out that a universal board game-focused language will make the whole developing process very convenient. Then we decided to spend the next few weeks to set the specific features of this language.

4.1.2 SPECIFICATION:

After we decided the topic, we spent no more than 2 weeks on the specific features of the language. Since all of us is familiar with C++, besides, C++ is very flexible to configure any complex rules for a board game, so we agreed on creating a C++-like language as the source language of GBL. One point worths mentioning is that we use the sign '@' as the domain operator because the '@' sign resembles a chess in board game. Besides, we also specified some features and usage of this language. Namely, we built several base classes for this game and let the users extend their own class from these base classes. However, to build a board game, developers have to implement some interfaces(just like virtual functions in C++ but not the same) of these base classes. For example, the user has to specify the game rule to check whether a move is legal, specify the winning conditions to terminate a game, specify the updating rules to update the status of the game. From this process on, we met with TA Julie Chien every Thursday afternoon to discuss what we have done for the past weeks, and what kind of improvement we can make for the next step.

4.1.3 DEVELOPMENT:

After we settle these features down, all of us started to get hands on these specific coding part. Shengtong Zhag started to focus on parser, Sihao Zhang focus on semantic check, Ye Cao focus on code generation and Yiqing Cui focus on parser and semantic check. We divided this project into several milestones, which is shown afterwards. Then for every week, we set the goals for the next week and would had a short discussion on the current week's work. We would put our problems together and solved them together.

4.1.4 TESTING:

For the scanner, we test every symbol once and make sure that the scanner will recognize every legal characters and report illegal ones. For the parser and AST, we would write the tokens sequence manually and feed it to menhir tool to check the correctness of AST. Since the menhir can print out verbose structures of the AST, so it would not be hard for us to do unit test for the parser and AST part. For the semantic checking part, every time we implemented a

new kind of check, we would write a set of test cases to pass the semantic check and failed the check to make sure that the checking part really takes effect. For the code generation part, every time we added a new translating part, we wrote the corresponding source code to translate. Therefore, for each part, every time we did some change, we can make sure that this change is correct and wouldn't cause further problems.

4.2 PROGRAMMING STYLE GUIDE

Since our GBL is based on C++, its style conventions were mostly inherited from C++

- Every line of code should not contain more than 80 characters.
- Variables are using CamelCase naming style.
- Though not forced, we strongly recommend developers to indent their code for better readability.
- Every code blocks should be encased in a pair of curly bracket

4.3 PROJECT TIMELINE

Feb 3	first group meeting, chose fields
Feb 7	decided the topic of project
Feb 10	finished the project proposal
Feb 17	started to discuss on the specific features of the language
Feb 22	determine the features of the language
Mar 7	finished the language reference manual
Mar 9	started to implement basic language features like type, binary operators, print
Mar 28	finished hello world part, implemented basic language features
Apr 6	Hello world demo
Apr 11	finished test cases for basic language features and pass them
Apr 18	started to work on advanced language features like class and inheritance
Apr 25	wrote the first demo case, GoBang in source code
Apr 28	finished these advanced language features preliminary according to the demo case GoBang
May 1	wrote the second demo case, simple black white in source code
May 5	fixed some bugs according to second demo case
May 9	finished final project report, ppt
May 11	finished the project

Table 4.1: Project Timeline

4.4 ROLES AND RESPONSIBILITY

Team Member	Role	Responsibility
Yiqing Cui	Manager	Controls the timeline of the project. Held discussions on problems we met in development. Main developer of scanner. Developer of semantic checking part and parser part. Contributed to designing the entire framework of the project.
Shengtong Zhang	Language Guru	Tackle problems related to language design. Main developer of abstract syntax tree part. Main developer of parser part. Contributed to designing the entire framework of the project.
Ye Cao	System Architect	Tackle problems related to code generation part. Contributed to designing the entire framework of the project. Main developer of code generation part.
Sihao Zhang	Tester	Tackle problems related to semantic check part. Contributed to designing the language grammar Main developer of semantic check part Build test cases Main developer of GUI library.

Table 4.2: Roles and Responsibilities

4.5 SOFTWARE DEVELOPMENT ENVIRONMENT AND TOOLS

- Code editor: Sublime
- Compiler implementation : Ocaml
- GUI Library implementation: Python
- Object code: Python
- Version control system: git
- Code hosting service: GitHub
- Report writing: Latex
- Testing environment : Shell script

4.6 PROJECT LOG

```
commit 9c34d97ba9f85f1ee852b1e5334968b6970cdba1
Merge: 807013f fa033d5
Date: Tue May 10 22:30:44 2016 -0400
```

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 807013febd7bdbde10e3d295158dba4c57fe5bf9
Date: Tue May 10 22:30:41 2016 -0400

add AI

commit fa033d5c03687294d682c6b5c2a4fae5ac2e6431
Date: Tue May 10 21:50:53 2016 -0400

update

commit b1f5533f2edb7e25f022ad97793900dbe5fbd95d
Date: Tue May 10 21:47:29 2016 -0400

add

commit adb847ec06ad356ce936b316928167f9b29e5c3b
Merge: d400cb9 b051cc1
Date: Tue May 10 21:42:34 2016 -0400

add

commit d400cb9a43a46c44a86591d833969c65b3128027
Date: Tue May 10 21:42:08 2016 -0400

add

commit 74330c2ecadeeee008cd13228182972244409e508
Merge: 044444f b051cc1
Date: Tue May 10 21:39:32 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit b051cc165202b30a8813a3b821e4151bbee21a73
Merge: 5b23978 13ac22c
Date: Tue May 10 21:26:23 2016 -0400

Merge branch 'master' of [github.com:full-fool/GBL](https://github.com/full-fool/GBL)

commit 5b2397883334ba24f830f03ac3ef50d2c223f509
Date: Tue May 10 21:26:13 2016 -0400

testall

commit 13ac22c6a9a48e9c35a6590ca0ebd42b440a727e
Date: Tue May 10 21:23:01 2016 -0400

gbl update

commit ff86fbde07e00f734d12bf78b0259a648a473cfd
Date: Tue May 10 21:21:45 2016 -0400

gbl update

commit 142396e3c8612c98acc83374e3aac4409e4a1a04
Merge: 2c0515c 9619a47
Date: Tue May 10 21:17:57 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit 2c0515c0a977d34f11e4f751e4c24bfdd6f0f274
Date: Tue May 10 21:17:55 2016 -0400

testall.sh

commit 044444fb7d7e159cfff5d7175604e26d2399a45a
Merge: 6615f0a 9619a47
Date: Tue May 10 21:14:41 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 9619a47e48d2d222eed2d7eb69f21fea0aae2996
Date: Tue May 10 21:14:19 2016 -0400

rename eat game

commit 7356bcc66a00dd999c98a3d8b247e49724029d85
Date: Tue May 10 21:13:11 2016 -0400

Update Eat.gbl

commit 68fdf6b66ee38b47b9311b6f136e2e07b3230c81
Date: Tue May 10 21:12:18 2016 -0400

Update GoBang.gbl

commit 6615f0ab99135a0d0f00671f08024a748c06fa47
Merge: 9d87714 cfb0fb9

Date: Tue May 10 21:11:29 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit cfb0fb92757740d68f763949370180f15b0b4a4d

Date: Tue May 10 21:10:48 2016 -0400

rearrange the structure of the project

commit 9d8771465201a31743074c54c0b9a3d19b203063

Date: Tue May 10 20:45:25 2016 -0400

update

commit e188f483a341bed6306fb5304ab51d710c643a9e

Merge: 20a2711 088242a

Date: Tue May 10 08:52:26 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 20a2711540bbe5b825f303e209bd8390c3879f3b

Date: Tue May 10 08:52:22 2016 -0400

add logo

commit 088242ab914471b651f147a23e0e3b5817846404

Date: Tue May 10 00:53:05 2016 -0400

testall.sh

commit db02c69b63e3485b7e35e205b8e396c5cd81b91a

Date: Mon May 9 20:09:02 2016 -0400

add initalize.gbl

commit b0428aab314aac60b4f6a88cdcb9ebb6315a7e26

Merge: c063c78 49f27e5

Date: Mon May 9 20:04:24 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit c063c780f920830f8febe33475f6bf6091cadec4

Date: Mon May 9 20:04:21 2016 -0400

update

commit 49f27e53d792608a30391df838f3a22718392bbe
Date: Mon May 9 18:14:49 2016 -0400

run all testcases

commit d4f60d965e3080af69f7732685758fadb3a12e24
Date: Mon May 9 18:09:40 2016 -0400

run all testcases

commit fa6463e9f909ae3e11b8a8480d8d91de0921d42e
Merge: 2cfa267 05161e8
Date: Mon May 9 18:00:12 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 2cfa267b8adb05b9581023e17ae9074958c02c8c
Date: Mon May 9 18:00:06 2016 -0400

print generation

commit 05161e81181dd4de52908f33da758f97505d4a7e
Date: Mon May 9 17:37:39 2016 -0400

semantic check change

commit d15939aa4f63ca2ac60c81aab70cb10bf68471bb
Merge: 4aa9fea 34a2a4f
Date: Mon May 9 17:36:42 2016 -0400

Merge branch 'master' of [github.com:full-fool/GBL](https://github.com/full-fool/GBL)

commit 4aa9fea98e61c6a32654a019064b6eacae4c3e7a
Date: Mon May 9 17:36:38 2016 -0400

semantic check change

commit 34a2a4fa9c4989c84631df4b458049fbd6a263d7
Merge: df79cef 6eff725
Date: Mon May 9 17:34:34 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit df79cef8133c086fc4a19a0a314f96e210af5cad
Date: Mon May 9 17:34:30 2016 -0400

fix bug: class function calling

commit 6eff725fe945fd8e8b6b8760194ed079204445e4
Date: Mon May 9 17:33:43 2016 -0400

semantic check change

commit a4a1297e03fe095c24bc38e059ee68de3a471c57
Date: Mon May 9 16:32:18 2016 -0400

testcases

commit fdf1b1bd7f8b2749943c9e856c42a9c8687e12b9
Date: Mon May 9 04:22:45 2016 -0400

testcases

commit 1d8b4ab2e1445632f4246a7ae13792ad42de7e82
Date: Mon May 9 00:12:54 2016 -0400

Semantic

commit e5d25ff80fd9fb7c66cdc06efac8a432f6e0f49f
Merge: 303b418 f357721
Date: Mon May 9 00:09:24 2016 -0400

Semantic

commit 303b4189667bb8bde36cb2b61817b9d50d6dcf0f
Date: Mon May 9 00:06:00 2016 -0400

semantic

commit f357721fa3f88a4d1a13b66b85b3ab5febbbd64
Date: Mon May 9 00:04:49 2016 -0400

fix eat.gbl

commit 99b937bacb7925a90bef8461072b172ac86460b2
Merge: 48c302c e0e8402

Date: Sun May 8 23:52:27 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit 48c302cb97be8b527a1eb98f400a03d1317f0077

Date: Sun May 8 23:52:24 2016 -0400

semantic

commit e0e840218a1b7d6d0a4f7191660886050bc951fd

Date: Sun May 8 23:47:50 2016 -0400

fix sample code

commit ebf6d6e72ed27e8cda9b7d7d68b6598c063ef737

Merge: 5b59b87 1fbf03c

Date: Sun May 8 23:41:38 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit 5b59b87b8b9e5d7b1afff6907a13bcddd9bf39c7

Date: Sun May 8 23:41:31 2016 -0400

semantic

commit 1fbf03cf71a22409b551a4be509ae4ac37e6ecea

Merge: b07d7d7 5853acb

Date: Sun May 8 23:41:06 2016 -0400

Merge branch 'master' of https://github.com/full-fool/GBL

commit b07d7d77bd447771875db08514d388121be61fca

Date: Sun May 8 23:41:02 2016 -0400

fix sample code

commit 5853acbf9fae71dd97e15f0a3c6e32a08468edce

Merge: 3b522cb 3b2776e

Date: Sun May 8 23:26:21 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit 3b522cba5873695da77f78860ed27f52b24b909b

Date: Sun May 8 23:26:14 2016 -0400

semantic

commit 3b2776e869d74354a253cddb1881d6849bdc9168
Merge: e0dc409 7d1549b
Date: Sun May 8 18:52:55 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit e0dc409581922582c305519911a7ad8433f86de7
Date: Sun May 8 18:52:36 2016 -0400

eat.gbl and eat.py

commit 7d1549b64187a72f59af8c56bb653f64b2d91b40
Merge: e56fc85 acb2f92
Date: Sun May 8 18:47:01 2016 -0400

fix sample code and object finder

commit e56fc85305890ad1b0b4d2ba15c41950f47555e6
Date: Sun May 8 18:46:42 2016 -0400

fix sample code and object finder

commit acb2f92fb7d7cbd82ec12af1410a2094c25a4b50
Merge: 21f55ef 2e30206
Date: Sun May 8 18:17:18 2016 -0400

Merge branch 'master' of [github.com:full-fool/GBL](https://github.com/full-fool/GBL)

commit 21f55ef78f17cac54249055761ebc4c3ddd106ec
Date: Sun May 8 18:17:09 2016 -0400

semantic

commit 2e30206efdea98ae2cdbe43107776e74ac596398
Merge: dc89d32 08476f6
Date: Sun May 8 18:16:58 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit dc89d3234f432463ffecb2a72b512ec2520bebae
Date: Sun May 8 18:16:55 2016 -0400

parser fix

commit ef3f83c8376734e092f104a4d086f3e97eb6b8f1
Merge: 11eac84 08476f6
Date: Sun May 8 17:57:22 2016 -0400

semantic

commit 11eac84fc9cdd19d1b5e31498b5b861b8574ff8f
Date: Sun May 8 17:56:36 2016 -0400

semantic

commit 08476f630de93915aa80965dace76f1b6af27711
Date: Sun May 8 17:56:27 2016 -0400

update semant

commit e47d06013c0bfce41e8bef47a9321698e893110c
Merge: 7e602ae d71af8d
Date: Sun May 8 17:20:56 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 7e602aea2695ad11ebcb82a60d6dc8bd90d120bf
Merge: ff7c6f9 040a43c
Date: Sun May 8 17:20:49 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit d71af8d6eaca1ab8a05a752a56eccc9cdbc11b88
Date: Sun May 8 17:20:48 2016 -0400

update ast

commit ff7c6f9264e90b517856b663efa6701b28e4a8aa
Date: Sun May 8 17:20:46 2016 -0400

hard code

commit 040a43c2a339872110a52a5b6e43e6bfdad76392
Date: Sun May 8 17:00:25 2016 -0400

semantic

commit 60136a24257f4cc4298f04baf691c11aa8088483
Merge: 1e30770 614f09a
Date: Sun May 8 16:53:54 2016 -0400

semantic

commit 1e30770afd4c339c03c3e20a7c13f882f07ae322
Date: Sun May 8 16:52:45 2016 -0400

semantic

commit 614f09a7fa8e35b8f8b436004ed08eaa354eb13b
Merge: dd96a5e 9746ada
Date: Sun May 8 16:51:56 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit dd96a5e73e5f0ffaf11cbabbcf5266a2ea4e183a
Date: Sun May 8 16:51:54 2016 -0400

fix formal parser

commit 385dc62af0d08054a04937be039350c2d4069934
Merge: 259a50c 9746ada
Date: Sun May 8 16:32:20 2016 -0400

Merge branch 'master' of [github.com:full-fool/GBL](https://github.com/full-fool/GBL)

commit 259a50cba2ac9c9dbddd3477e0312da35ef9bd2e
Date: Sun May 8 16:32:17 2016 -0400

semant

commit 9746ada5479bb82fb30acdeb147da3c0ebafe57e
Date: Sun May 8 15:54:57 2016 -0400

row and col

commit be94bd1afc27f6714f0920b14d09961165699b6b
Merge: e6fe08c 6765828
Date: Sun May 8 15:53:54 2016 -0400

hard code

commit e6fe08c41c1a02c5c271ceaf6513f22a8e7e4e0b
Date: Sun May 8 15:53:08 2016 -0400

hard code

commit 6765828aae04701d16660f515dd00134c7a9981f
Date: Sun May 8 15:50:10 2016 -0400

add formalvora

commit 7320ec7d28667fe839a959dbb476d7ad3c8982ff
Merge: 2b2d5c9 e74467b
Date: Sun May 8 14:50:14 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 2b2d5c9bf8ce0939670e8a3f09d1b6ef06eb305f
Date: Sun May 8 14:50:09 2016 -0400

add eat.gbl

commit b83858fd7d4599d1888b539b7cdedfbb4a3a1108
Merge: 51155d4 e74467b
Date: Sun May 8 01:57:51 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 51155d44fb88dab89b491ec30696503e820d0c40
Date: Sun May 8 01:57:48 2016 -0400

find game and ai object

commit e74467b70c00b25ce3225527e13ccafc55141988
Merge: 6b41efe 2ca5192
Date: Sun May 8 01:56:37 2016 -0400

Merge branch 'master' of [github.com:full-fool/GBL](https://github.com/full-fool/GBL)

commit 6b41efe59ece4d2ae2d60e1bb63493f4eb1c62f1
Date: Sun May 8 01:56:34 2016 -0400

semantic

commit 2ca5192e88a6e028757546817eb8d53ee72edc9a
Date: Sun May 8 01:50:47 2016 -0400

change win from void to int

commit e345988a2d475547ac564b4a55a4c3e3b3c6066a
Merge: 07a9889 e919641
Date: Sun May 8 01:31:08 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 07a988916112239c0661dfab2b0422b89dcc8cd8
Date: Sun May 8 01:31:04 2016 -0400

add delete sprite

commit e9196413ac6d04edfb3cc9ddb185a30c8b81cca5
Merge: 7205a54 7d6c492
Date: Sun May 8 01:26:16 2016 -0400

semant

commit 7205a54aa7a2923999cc141a8670f3bc4063ad1d
Date: Sun May 8 01:24:31 2016 -0400

semant

commit 7d6c49254c51f20e782f517613575849c14390a6
Merge: 885f34b f9ee5de
Date: Sun May 8 01:23:29 2016 -0400

uptae semant check

commit 885f34b80fab4be1d11203148ecb60c9725726ca
Date: Sun May 8 01:23:03 2016 -0400

update semant check

commit f9ee5dec7608cc9be4a97ab28ee8145776d70f12
Date: Sun May 8 00:55:39 2016 -0400

add AI

commit 4ed8925cbd364a93ce1977aa0c61308bb060177a
Date: Sun May 8 00:55:03 2016 -0400

add AI

commit 17bf62beb033c07680b32554d17835d9b4814d55
Merge: 4683308 24a2dce
Date: Sat May 7 23:48:17 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 468330836d1c94ce0d0da08e9ffda4a070d06830
Date: Sat May 7 23:48:13 2016 -0400

add gobang.py

commit 24a2dceec7263e071fb560133054daa8df21198d
Date: Sat May 7 22:51:58 2016 -0400

semantic

commit 44d4401e4303969758dd7f052d917509d243cd43
Date: Sat May 7 21:35:03 2016 -0400

semantic

commit fc57987e27d03a0f2516c1ab6880c65ece666c9a
Date: Sat May 7 21:33:28 2016 -0400

semantic

commit 188ca0f77f367f2ee158ca4a9a5beb48fb66d861
Merge: 277c7cb 954db3b
Date: Sat May 7 21:30:22 2016 -0400

semantic

commit 277c7cb981340f2f721fa521cdd2c8a8dd9b5267
Date: Sat May 7 21:29:52 2016 -0400

semantic

commit 954db3b686c2e774feb60003f788b9af4c470491
Date: Sat May 7 21:06:59 2016 -0400

fix sample bug

commit 64d3fc7d448416a30d7e91dc5fc6198619f71986
Date: Sat May 7 21:03:11 2016 -0400

fix sample bug

commit 355d78633d7f0110d896d4d91fd662993f3f50c0
Merge: d9077b0 4adc573
Date: Sat May 7 21:01:10 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit d9077b03efd523d5693485fb6a5e56a00c687f33
Date: Sat May 7 21:01:06 2016 -0400

update

commit 4adc573cd7549f2ae3270b148e5eca972b5ac2a0
Date: Sat May 7 20:59:28 2016 -0400

fix

commit 53cf6e48cdb9d1a41f01380838e6b5655c1937dc
Merge: 4229c00 2f76242
Date: Sat May 7 20:55:15 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 4229c00c58ebfbba724724f6b39357c8253d68cf
Date: Sat May 7 20:55:13 2016 -0400

update

commit 23b3f6dc5286f789bcaff95515063390492a00b
Date: Sat May 7 20:51:38 2016 -0400

just update

commit 2f76242ae622ab736149796226f629d1d134acf8
Merge: 7a678c7 48bf6dd
Date: Sat May 7 20:49:20 2016 -0400

hard code

commit 7a678c742c6f9f734cfdaaf1c12f3131d829e1de
Date: Sat May 7 20:49:09 2016 -0400

hard code

commit 48bf6dd0049232ab342e1a5f79965e64af1efe88
Date: Sat May 7 19:57:19 2016 -0400

WithAI

commit a9d3c5e8aaa517c04bb150b4a6f9abe6f3b4b8e2
Date: Sat May 7 17:51:05 2016 -0400

fix sample.gbl

commit 38d3258feeb418a75e53994d7488729ce840b8a4
Merge: 48c10d5 3d7bb6c
Date: Sat May 7 17:32:40 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 48c10d5dbaf83a0678c12f7b86db653b0d4a0e39
Date: Sat May 7 17:32:36 2016 -0400

get new gbl

commit 3d7bb6cdd287e860672e1e7b2dce8ca2cdb4a371
Date: Sat May 7 17:32:09 2016 -0400

update sample code

commit 253c3788f748012b527fb2fa341ac05f0a1857c4
Date: Sat May 7 17:14:24 2016 -0400

gbloutput.py

commit cb4d83b4b5f7908be7a3c596450f8fa6879d8e81
Date: Sat May 7 17:09:10 2016 -0400

variable in class

commit f96dcd3f371a64395bcb93f6a2b5193405110791

Date: Sat May 7 16:54:28 2016 -0400

add vandadecls in class body

commit 5c4a9ce55f13cd0b517a757d472183eafffb3f2b

Date: Sat May 7 16:47:57 2016 -0400

add a[1]@class

commit b21778c7d12c8e8994f88f52eb16a81319259a60

Merge: 659a3b1 c0b0e7d

Date: Sat May 7 16:44:25 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 659a3b1ba99359d42e31510eeb5ca8a752e4b2a6

Date: Sat May 7 16:44:17 2016 -0400

add a[1]@class

commit c0b0e7d4ac31d42ed34fbd0deaf1e545c8120142

Merge: 1adfea8 aa25764

Date: Sat May 7 16:37:35 2016 -0400

gui color change

commit 1adfea81059de318d91652d3a1c60f97170c4294

Date: Sat May 7 16:37:26 2016 -0400

gui color change

commit aa257641536e34cd35b934d73e65f1bf3ef39c22

Date: Sat May 7 16:26:25 2016 -0400

delete main in scanner

commit d751fb7cb3c5ef68a8dc7c34c81af91fc5fa84cb

Merge: c6d90dc c017a87

Date: Sat May 7 16:20:34 2016 -0400

Merge branch 'master' of [github.com:full-fool/GBL](https://github.com/full-fool/GBL)

commit c017a8782675cbc4203670f438241f403afc4ee2

Date: Sat May 7 16:20:22 2016 -0400

fix ast bug

commit c6d90dcf415139f19a02a8e726205ae4556a5f7e
Merge: a17f30f 8dd1814
Date: Sat May 7 16:18:12 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit a17f30f9c94eebd30bdec8549aec561b767a0de0
Date: Sat May 7 16:18:07 2016 -0400

ast

commit 8dd181479d4abbacedb7497075ea4bf8dd20dae
Date: Sat May 7 16:17:14 2016 -0400

delete ignore

commit 04015ca67b588834fabbb1f2493d5cb92bb141da
Date: Sat May 7 16:13:19 2016 -0400

add string_of_typ... in ast

commit 997ef82b618963edd0d31e1c6f3981f70f712110
Date: Sat May 7 15:42:22 2016 -0400

negative

commit 3cd37491d4483eb293c3c1b63761f88910962dec
Date: Sat May 7 15:40:07 2016 -0400

Update parser.mly

commit 4d8bc4e347ca1ea008c4fbec31ad40856dab999a
Date: Sat May 7 15:39:41 2016 -0400

Update ast.ml

commit de84a277f860efc5b8239c4909031a655dd72c92
Merge: 879e6f2 53260b6
Date: Sat May 7 15:31:10 2016 -0400

merge

commit 879e6f26e02e7a6c747a4851e55d0c0d1b17e442
Date: Sat May 7 15:30:56 2016 -0400

delete some useless files

commit 53260b6319d835b6141b03356bc4fee8fefb27fb
Merge: 34676bd c4c5fdd
Date: Sat May 7 15:23:35 2016 -0400

fix is

commit 34676bd76c6658178d1dc694148fb9a360eb66b5
Date: Sat May 7 15:23:24 2016 -0400

fix is

commit c4c5fdd3e016af25d13735cd75b55bfccf0f9d6f
Date: Sat May 7 15:09:35 2016 -0400

Update parser.mly

commit e3c1b2895f27227384eec9e7873dadca116d5090
Merge: 2619a08 da52ae0
Date: Sat May 7 15:07:25 2016 -0400

mod

commit 2619a08adb9f7320d734df5bc5ffd84dc97193ca
Date: Sat May 7 15:06:53 2016 -0400

mod

commit 53a4b4d79d8e212ca9d9be69c88abd11960e0ab3
Date: Sat May 7 15:06:47 2016 -0400

mod

commit da52ae0637739c0fa78de999d3a0e59af7270d5f
Merge: 3b8c872 6878b14
Date: Sat May 7 15:06:25 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 3b8c8724aaaf62bf4e30b29b7a4ad03f31ec4ebc
Date: Sat May 7 15:06:21 2016 -0400

add %

commit 6878b1431591ce3ae2393643247db37b015a1c7f
Merge: 2022767 9a0c42e
Date: Sat May 7 15:06:12 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit 2022767cd6634072ee3c7a98f2cdd15c819f5ff9
Date: Sat May 7 15:06:08 2016 -0400

pythonGui"

commit 9a0c42e7bf62afac6e552d488bda07fccc35695d
Date: Sat May 7 14:59:22 2016 -0400

fix bugs

commit 93ef5892a60f2c93b3cc5d417ebd968136cacdbe
Date: Sat May 7 14:50:14 2016 -0400

print bug

commit 485eddc137873f9451cfac05fdf73661804262a7
Date: Sat May 7 14:47:03 2016 -0400

fix string and call domain

commit e788ba856dac427d63ff0d53f9d93e3820dbc847
Date: Sat May 7 14:36:07 2016 -0400

add main

commit 72514ec28c3c56fee592b9d49c04ca2ddbdc4409
Merge: c3f3516 7207d16
Date: Sat May 7 14:21:13 2016 -0400

add symbol table

commit c3f3516ee0b5377e13fbb30c582086535c3bf958
Date: Sat May 7 14:20:58 2016 -0400

add symbol table

commit ddb337cc302fdaccaa99fb312916d7884a1348d1
Merge: 31b017c 7207d16
Date: Sat May 7 14:02:00 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit 31b017c50865c07f03dcf3f99ffe6b1e1fa42903
Date: Sat May 7 14:01:57 2016 -0400

semantic

commit 7207d16ffa995e7d0ac7d9b640054a261eceedc5
Merge: d70549b 994d8a0
Date: Sat May 7 12:32:58 2016 -0400

Merge branch 'master' of https://github.com/full-fool/GBL

commit d70549b5fa0425ad606cd32e364a3cf29d70ed31
Date: Sat May 7 12:32:54 2016 -0400

add ID ID SEMI gobang mygame;

commit 994d8a0ee4fcd7ef4af715849c1ee444fab637f6
Merge: 8133b97 5090eb9
Date: Sat May 7 12:11:19 2016 -0400

fix self variable

commit 8133b972fc54c2abb28e0994dd5ece03998429c9
Date: Sat May 7 12:10:51 2016 -0400

fix self vriable

commit 5090eb9f31d31457d615718f75c497f29ea1a8a2
Date: Sat May 7 12:04:42 2016 -0400

Update test-var1.mc

commit 6925529141b5b077cb9023e4e51789d547d34242
Merge: 3489bc7 18aef4f
Date: Sat May 7 04:08:46 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit 3489bc7e106dbc013719bd953f3abd8ea673ccf1
Date: Sat May 7 04:08:41 2016 -0400

semantic

commit 18aef4f767a22a650afc48cae526085ef835c30b
Merge: ae07d48 66fa3ad
Date: Sat May 7 02:35:53 2016 -0400

class

commit ae07d4848ffb3fbec4a8676624cbf9b49fe3ef13
Date: Sat May 7 02:35:34 2016 -0400

class

commit 298d956fef0193d8df1700cbd26971da42294e91
Date: Sat May 7 02:35:12 2016 -0400

class

commit 66fa3adc1ab572a458096c7463e4d269499938ce
Date: Fri May 6 23:57:32 2016 -0400

delete ++

commit 989d1f8b300ffc1c3b15a273770f282aee0e82bc
Date: Fri May 6 23:53:46 2016 -0400

add AI class

commit 64bd93e6a9025289d35250b8c5c67e98ed0a2ec5
Merge: 3a963a0 e3821cb
Date: Fri May 6 23:43:07 2016 -0400

Merge branch 'master' of https://github.com/full-fool/GBL

commit 3a963a0b6da9f5c9390ce25531e53be345aed658
Date: Fri May 6 23:43:03 2016 -0400

new sample code

commit e3821cbdc4dc21682ff17fe3cb0dbe6913e29599
Date: Fri May 6 23:36:12 2016 -0400

fix

commit f3fab41a00eb522982c9637d41d07a801cd0c31f
Merge: 699f061 58ecc78
Date: Fri May 6 23:24:30 2016 -0400

array

commit 58ecc78068e108c1b128c23211eb619227025f11
Date: Fri May 6 23:24:03 2016 -0400

semantic

commit 699f06112ccd86ceadbc1592030d4377552b3010
Date: Fri May 6 23:23:14 2016 -0400

add array

commit f334e834aae6aa498dddb28209d976c6ec8c5279
Merge: 1c4bad1 19f34ae
Date: Fri May 6 23:13:00 2016 -0400

semant

commit 1c4bad1b7d7183de5c636a33ae5260b77bd3862c
Date: Fri May 6 23:12:38 2016 -0400

semant

commit 19f34aebdd1061b123c07547ec13aec6c7a5e8e1
Merge: 7c110d6 ca49104
Date: Fri May 6 23:11:20 2016 -0400

merge

commit 7c110d6432c59b381f84327108e04485375806b3
Date: Fri May 6 23:09:49 2016 -0400

code generation

commit ca491041e3289f596effc6760400897e23be2bd6
Date: Fri May 6 23:00:27 2016 -0400

delete array<>

commit 651083e14e68cd3e8d378308dbbda99c718fd8e5
Date: Fri May 6 22:58:57 2016 -0400

adjust if

commit a70ef12b8f48839dacf8cb3918964eaf584b1fd1
Merge: 7de618d 7e1e2f7
Date: Fri May 6 22:36:58 2016 -0400

codegeneration

commit 7de618d113fb90c217a2c9e40f8f98c45c3f8951
Date: Fri May 6 22:34:39 2016 -0400

code generation

commit 478880b227ee3cad7c5dc5f0bafbe86fd4daaea6
Merge: 871cc56 7e1e2f7
Date: Fri May 6 22:19:22 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 871cc56602d4480cf66078ff9dbabeb3b9d80311
Date: Fri May 6 22:19:20 2016 -0400

update

commit 7e1e2f736134f77931db0af9b2abc7aff8a09808
Merge: 9d7fd72 9a8cc72
Date: Fri May 6 21:26:18 2016 -0400

update sample

commit 9d7fd721dc971115be7612f025bc2968dc2aaf47
Date: Fri May 6 21:25:51 2016 -0400

update sample code

commit 9a8cc72466c4df2a796253d250c478a9bf6de41e

Merge: d99b4c4 c26975d
Date: Fri May 6 21:19:46 2016 -0400

code generation

commit d99b4c45f765e2a0ef9acb0f15cce1ed9e79162b
Date: Fri May 6 21:18:12 2016 -0400

code generation

commit c26975dab9cad44e7dba3c912c3f7c82073291b2
Merge: 161b146 c89bead
Date: Fri May 6 21:18:05 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 161b1468d4c1fcd6a551ca761ad148a29be638a4
Date: Fri May 6 21:18:00 2016 -0400

delete op in @

commit c89beadb05d8bea221b7c89de94bf59ea43f7d33
Date: Fri May 6 21:15:35 2016 -0400

update sample code

commit 5ed7d9af749b8512ec178fa87e30eb8863c9dd8d
Date: Fri May 6 20:43:15 2016 -0400

add @

commit 8c790c086daf875c4de8decdbd457a3f23647ac8
Date: Fri May 6 18:49:45 2016 -0400

add @

commit 3cf1f76cba9665e9ead2e8f3537ff4eade1c6e55
Date: Fri May 6 18:43:33 2016 -0400

add @

commit 600c511e252d30ea99432a9c78f5650b606de611
Merge: ce440c0 19b77f7
Date: Fri May 6 18:33:35 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit ce440c06f1022c7d754d09d61b11e86d2a7a5eb6
Merge: 1cf8404 05240a3
Date: Fri May 6 18:33:26 2016 -0400

just pull

commit 19b77f74f162a2416383824b6501436846359fc0
Date: Fri May 6 18:29:54 2016 -0400

code generation

commit 1cf84049b31de746414ada836c025f8a633506cc
Date: Fri May 6 18:21:01 2016 -0400

change int to expr

commit 05240a3d5ce9b6eb0d7a9090b00faf1fae030d3e
Date: Fri May 6 18:20:23 2016 -0400

Update ast.ml

commit 42fa8fee0d833082f820482b06c768a26e4d4026
Date: Fri May 6 18:19:02 2016 -0400

Update ast.ml

commit 620ecc8377aba1e8b575ee18a96a815380763d8b
Date: Fri May 6 18:18:18 2016 -0400

Update ast.ml

commit b593dc7b457f4ba1d28b3d22306caaba9666b61b
Date: Fri May 6 18:16:25 2016 -0400

add int a = 1 in class

commit f54ba1846bdf5ba82a39033f28e82cba341d63c9
Merge: 66a8f02 3d88905
Date: Fri May 6 18:03:51 2016 -0400

Merge branch 'master' of <https://github.com/full-fool/GBL>

commit 66a8f02e8faf6ac5028ec02e45335d0535b24826
Date: Fri May 6 18:03:32 2016 -0400

add int a = 1; in class

commit 3d8890511271abb696a6009dc67fe93563e1ba33
Merge: bd9d234 563a2c7
Date: Fri May 6 17:57:58 2016 -0400

code generation

commit bd9d234807eddab68ac1cb98881fb47922f0352d
Date: Fri May 6 17:57:17 2016 -0400

code generation

commit 563a2c7e5631158cca7ccd17a33acaa3239cad17
Merge: f01f47b a9f0482
Date: Fri May 6 17:47:31 2016 -0400

gui

commit f01f47b8f9c435c874e07da233c627e32c5068fd
Merge: 3140fca f04bed9
Date: Fri May 6 17:47:10 2016 -0400

gui

commit ce0accb827902fcaf70388c6436cb9561a8cb306
Merge: 4baefb6 a9f0482
Date: Fri May 6 16:45:38 2016 -0400

code generation

commit 4baefb6998d8dced84b5292c2a22e26478ef9693
Date: Fri May 6 16:43:29 2016 -0400

code generation

commit 26e9a21185f3bb6c5cc194326525fd5b1845910e
Date: Fri May 6 16:42:46 2016 -0400

code generation

commit a9f0482f5dc50e84c8fb5966b43e0b79d382e0a7
Date: Fri May 6 16:42:19 2016 -0400

Update parser.mly

commit 9733d6531f18f52e000d46ad1e3ef0093ab99a31
Date: Fri May 6 16:26:33 2016 -0400

Update ast.ml

commit 190181efb5847bb3384b1187c7e667b0dc9773dc
Merge: e5d65b1 f04bed9
Date: Fri May 6 16:22:18 2016 -0400

code generation

commit e5d65b1e36d457876402850b6db71850e8133188
Date: Fri May 6 16:21:03 2016 -0400

code generation

commit 3140fca7e0cbf29b82e7edcae9eb9b4f321573b8
Date: Fri May 6 16:20:13 2016 -0400

ast

commit f04bed96bb0223149956a2dcbaa02d572b9308f2
Date: Fri May 6 16:20:02 2016 -0400

Update ast.ml

commit 321bd7f118b293d42fa1b227c30fbb0ef5efc610
Date: Fri May 6 16:16:55 2016 -0400

update smaple code

commit a18cc7efa914edfa7c334350ad042159c6208814
Merge: 5e8b1bf 70f38b2
Date: Fri May 6 16:15:31 2016 -0400

add class

commit 5e8b1bf28e9949abeb9af15aece7171ae79e34f3

Date: Fri May 6 16:09:49 2016 -0400

add class to scanner/ast/parser

commit 70f38b2f0bc75e44df8d3f97c68b2ba492acb346

Merge: 85499b6 3d8342b

Date: Fri May 6 16:05:21 2016 -0400

code generation

commit 85499b6b5ef3f45b6b9e8f18e6bc847efcd4921d

Date: Fri May 6 15:57:38 2016 -0400

fix

commit 3d8342ba379b428f848ed52cedf3e5f112c48c55

Date: Fri May 6 12:56:18 2016 -0400

update sement check structure

commit a89b3a33fb8a287ba70800818b68fd7fde3ed192

Date: Fri May 6 03:23:02 2016 -0400

gobangGUI

commit 82b96123f7645b2751e719411e2e5480a1959ac1

Date: Thu May 5 17:27:17 2016 -0400

add array and variable initialization

commit 88367b7e04448e051c539b80dbb096fcc13fc3cf

Date: Thu May 5 03:09:46 2016 -0400

add support for array(cannot test, but can make)

commit 414bea8698dd836a53c89ffb639ca3cb5d741654

Merge: fa770d1 87a3e70

Date: Wed May 4 23:21:07 2016 -0400

add arrayele and init

commit fa770d1bba7bbd049ddb5aa17111239d4bdfa4d

Date: Wed May 4 23:19:44 2016 -0400

add arrayele and init

commit 87a3e70f351e66854cff278072f6d761058deb7a
Date: Wed May 4 23:19:00 2016 -0400

change

commit 95f4f766a5a75cd41dd90a242f4ed2e9e095c86e
Merge: 12cd7a5 969f5ad
Date: Wed May 4 20:59:29 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit 969f5ad9cf0d767fbf90cdbc627f01a166de9d1c
Date: Wed May 4 20:58:47 2016 -0400

update comment

commit 12cd7a5def2ea1d661ac86b5589e83317ef12ec9
Merge: ce34978 1a6bc18
Date: Wed May 4 20:40:29 2016 -0400

Merge branch 'master' of github.com:full-fool/GBL

commit ce34978e6265d932c4f883db7661c4b2265813bc
Date: Wed May 4 20:40:11 2016 -0400

sement

commit 1a6bc18a832f7ff52a64cd2c468f5b68ba7a16ad
Date: Wed May 4 20:39:26 2016 -0400

fix the comment in scanner

commit 1ebfe51f38bb7cce75c1db080389bc37997e9609
Merge: f7cd8ef 0ea1ce2
Date: Wed May 4 17:28:53 2016 -0400

semant

commit 70771051dd8ebc7a4d16800c77220587762bf5b3
Date: Wed May 4 17:12:22 2016 -0400

Update README.md

commit f7cd8efcc6e77a9a40cede73eeec72be2e5e9e36
Date: Wed May 4 16:53:08 2016 -0400

semant

commit 0ea1ce2c1064ffa54b3acd9b33952159bd6c0dca
Date: Wed May 4 16:42:40 2016 -0400

fix

commit 98e58a0923ff132ca22421fb0444da1f3da8513d
Date: Wed May 4 16:42:25 2016 -0400

fix

commit 317cb9ac74dab10358bf7c02afb9f58f12cef7c5
Date: Wed May 4 16:13:42 2016 -0400

fix

commit a32fd79e7440aba6de07eb9521fef313c81b29d8
Merge: 14a59e9 819307b
Date: Wed May 4 16:12:27 2016 -0400

add string in scanner

commit 14a59e9336d505093f3db848a673db1754ee5e6b
Merge: faffdbc def95bc
Date: Wed May 4 16:10:57 2016 -0400

add scanner about string

commit 819307bbcff059cb5ab3fe0be01100242b9463c0
Merge: c1b4deb def95bc
Date: Wed May 4 16:08:55 2016 -0400

fix

commit faffdbcc30c90dddfda614d45b6590b8c7156165
Date: Wed May 4 16:07:51 2016 -0400

add scanner about string

commit c1b4debf1d302e74a1ad45a0f7f14da240306bdd
Date: Wed May 4 16:07:28 2016 -0400

fix

commit 0610350997ff6a799498f91f9d0ff2e5d52375d1
Date: Wed May 4 16:07:12 2016 -0400

fix

commit def95bc056bef70200f3343af8172a357ee9d4e2
Date: Wed May 4 15:57:17 2016 -0400

Update ast.ml

commit 286034ce4bcf500abda857ff365e48f726f6973e
Date: Wed May 4 14:47:00 2016 -0400

Update parser.mly

commit ce944e2dda9458dae8e477b7f2d9c0bdc8fe029e
Date: Wed May 4 11:16:35 2016 -0400

Update parser.mly

commit cfd562856dd16224ce5a18079dc7e1b0f69e9715
Date: Tue May 3 22:54:45 2016 -0400

fix

commit 00403cc950f209365f6e670ea265845a2c690f92
Date: Tue May 3 20:49:25 2016 -0400

Update scanner.mll

commit f090119f8cd2b33c1ea9860b5fdcfbb3ced76053
Date: Tue May 3 20:36:12 2016 -0400

update todo in readme

commit c102350fccec433dfa2ea6a478ed96acf0b5dd5c
Merge: 6332eb4 4f13d49
Date: Mon May 2 16:10:40 2016 -0400

add sample code

commit 6332eb4f3dfc897e5e2d1a4440ade8d548f35f7a
Date: Mon May 2 16:10:17 2016 -0400

new sample code

commit 4f13d49e48e8ade142ca5153f77b00bb2a07fe9f
Merge: f6a7d9e 1871c4b
Date: Sat Apr 30 21:59:54 2016 -0400

testcase

commit f6a7d9ea59ce3f867ed76dced9091f79e9a91389
Date: Sat Apr 30 21:58:49 2016 -0400

testcase

commit 1871c4b3c3f26022d1602793bab572690e30ced0
Date: Thu Apr 7 19:14:09 2016 -0400

rm microc.native

commit a3f082acb6e0cd6a45a86b0dc42e6b178c56bef7
Date: Mon Apr 4 21:20:32 2016 -0400

first version hello world

commit c2cc6e368a4fd618128ce24e308bac7c91c3b4c8
Date: Mon Apr 4 21:16:44 2016 -0400

fix

commit 17a591b593d2a013734663b74c6eed66151239ee
Date: Mon Apr 4 21:10:37 2016 -0400

add semi

commit f72fdfa423553001361acdda06312f55507c65a1
Date: Mon Apr 4 20:55:47 2016 -0400

Update microc.ml

commit 8dc1700bd4a3d9447b3ea75cf2c33595e55e8cd2

Date: Mon Apr 4 20:43:44 2016 -0400

Update ast.ml

commit 226bebb8363da22e66b3eef203098ac02525dc64

Date: Mon Apr 4 19:22:28 2016 -0400

Added files via upload

commit 41bc282d502ed79cc2920b07d1ca4a2714d614b0

Date: Mon Apr 4 19:20:44 2016 -0400

fix

commit d9786f8f251b26e25bd5798e0dd8b0187faa7d0b

Date: Mon Apr 4 19:19:43 2016 -0400

fix

commit c5876a4ae1164f01794d465294741abfc8e4a51a

Date: Mon Apr 4 19:18:14 2016 -0400

Added files via upload

commit cd53c552f757e3dcad3ca00e9972a8c0aca6c886

Date: Mon Apr 4 19:17:11 2016 -0400

delete array

commit e96229212fec3d83c5305953f6ce409a37a14dbf

Date: Mon Apr 4 19:16:02 2016 -0400

Added files via upload

commit 019da088e967d9b4cd5b9f657d6c30c36d41059d

Date: Mon Apr 4 19:09:00 2016 -0400

update scanner

commit delfec62f309c2298dc9c0478b9ed5f589116ed5

Date: Mon Apr 4 19:07:49 2016 -0400

Added files via upload

commit 601ba05075a8a2c5891ab84451496ebc5ceaf216
Date: Mon Apr 4 19:07:37 2016 -0400

Update ast.ml

commit dc159fb6308ebd0dbdc235248449d0820dbda5a4
Date: Mon Apr 4 19:05:16 2016 -0400

Added files via upload

commit 72445945559221344531f2a8393e07670aa8bccd
Date: Mon Apr 4 19:03:20 2016 -0400

Update ast.ml

commit 28ab848a86686ddd6412c09cb6aeb397227f882e
Date: Mon Apr 4 19:01:58 2016 -0400

test

commit cbaa8e31034aaa0a31fbe3e940212611885987e3
Date: Mon Apr 4 18:56:46 2016 -0400

Update ast.ml

commit d1073cb55602b2acd38d88701de577239f92f09a
Date: Mon Apr 4 18:52:10 2016 -0400

Added files via upload

commit 40396377eea5f6fbd375af1315644ddb9324a087
Date: Mon Apr 4 18:47:21 2016 -0400

Added files via upload

commit 90e5e24f260806562ff03d0d23fbfaa0f4aaab50
Date: Mon Apr 4 18:46:44 2016 -0400

fix

commit a336e2a57294bfb01e5aaede0441f21a9bbff470
Date: Mon Apr 4 18:44:51 2016 -0400

test

commit 399c953d8ab08d190757d3c1e6b4eed66873d3d
Date: Mon Apr 4 18:43:32 2016 -0400

Added files via upload

commit e4d5f562d794b53e8a9c593be3f56aa5da35551d
Date: Mon Apr 4 18:42:37 2016 -0400

fix

commit aeec19c743f62cfbe6b909a3ebfb86a1f2eb4133
Date: Mon Apr 4 18:40:29 2016 -0400

add float and string constant

commit 26f0bc26c6ae31be29eb895954cd56a137ff6ee9
Date: Mon Apr 4 18:38:20 2016 -0400

fix

commit a2324b25ab7a48ee76da8839106b5edd8c18208f
Date: Mon Apr 4 18:37:14 2016 -0400

fix bug

commit 43b6c12b666ca4a42f6cfa67cb7ae584b6f4f5f2
Date: Mon Apr 4 18:36:32 2016 -0400

Update ast.ml

commit 08a2618802060499c629fe1b2fe144ea3c4f5afc
Date: Mon Apr 4 18:34:01 2016 -0400

fix bug

commit cbd5ec73f7fdd633ea529cd049c21e477bb77e15
Date: Mon Apr 4 18:32:46 2016 -0400

add string constant'

commit 21fe0a3cbeb3aa6bedc42c141483e924a5257afc
Date: Mon Apr 4 18:30:29 2016 -0400

string sub

commit e1dcfc29819f68ed05e7791023a5ae8c314042c6
Date: Mon Apr 4 18:21:18 2016 -0400

Update ast.ml

commit b191110d81e139801846aa04ba28ec4352244994
Date: Mon Apr 4 18:13:03 2016 -0400

Added files via upload

commit a2585ad8f05440ecd6acf45fb4d94159a67a6bc5
Date: Mon Apr 4 18:09:31 2016 -0400

fix bug

commit ca08fb90b2ae980992cfed2d6c34745a69ec6946
Date: Mon Apr 4 18:08:06 2016 -0400

update array in scanner

commit 5b0cbf2961ca225387cc8d15076798ed48807336
Date: Mon Apr 4 18:07:42 2016 -0400

Update ast.ml

commit 0858f4df70bcef6965cb90b831145060fb99589a
Date: Mon Apr 4 18:07:26 2016 -0400

Added files via upload

commit 7a47977a0fda463c64700e04f27c3bda0716d240
Date: Mon Apr 4 18:06:04 2016 -0400

Added files via upload

commit e180e78dcc2004a321c56fc8a1e320691b36a127
Date: Mon Apr 4 18:04:14 2016 -0400

Added files via upload

commit 6873a7563999b061689682c318e1126c0ef75ad6
Date: Mon Apr 4 18:03:06 2016 -0400

Added files via upload

commit 417b82e6ca6180cb24ed1686c79b1351e18c5eaf
Date: Mon Apr 4 18:01:08 2016 -0400

change equal to is

commit a24439f4adcbb038c47605f978670bac1a4261d4
Date: Mon Apr 4 18:00:27 2016 -0400

add float

commit 068954101a1832538d09ad34e6eed06a3f3faba1
Date: Mon Apr 4 17:55:55 2016 -0400

delete symbols

commit 2f4ed1e8da480b8364b58e0f9e92d4c0128997a3
Date: Mon Apr 4 17:55:14 2016 -0400

Update ast.ml

commit 26cece2b8749c001384aee83f1f1dcfae6414f66
Date: Mon Apr 4 17:52:07 2016 -0400

Added files via upload

commit 115689904b8fa8a52a48236c7c2205223765e296
Date: Mon Apr 4 17:51:21 2016 -0400

Added files via upload

commit c51ad42fed55feba9fe16b5e6b7334d5a4d04861
Date: Mon Apr 4 17:50:28 2016 -0400

Update ast.ml

commit 9180961d7c7ad55a2b1ec2a9609198820c994b5a
Date: Mon Apr 4 17:47:22 2016 -0400

Added files via upload

commit bd7abaa9af66d314ebbc65d9c09b53cc604b7348

Date: Mon Apr 4 17:47:10 2016 -0400

add void true false in scanner

commit a5e2db20c135316a5bc0ae1b5ba84654ebbce3b5

Date: Mon Apr 4 17:32:18 2016 -0400

update scanner

commit 7ad0a3c4cbec0a4a9cb3744d50c104277746a020

Date: Mon Apr 4 17:24:19 2016 -0400

Update scanner.mll

commit 42eecbc9bdd208d6c3a7c88cf8410b71cab772d5

Date: Mon Apr 4 17:23:35 2016 -0400

Update scanner.mll

commit f480dd4d5bd9c2c0d6f13f4d11115a717318c8ef

Date: Mon Apr 4 17:22:58 2016 -0400

Update scanner.mll

commit 563ddcee63a634f8a390e952f3a9c3910e6c0d4d

Date: Mon Apr 4 17:21:16 2016 -0400

Update scanner.mll

commit 33ab0be2cfbc0fc6448dd4be25ac054b965d2b4d

Date: Mon Apr 4 17:17:57 2016 -0400

add

commit d89dbcc27275584310f672ef8db3b85d2e811104

Date: Mon Apr 4 17:10:57 2016 -0400

Added files via upload

commit 3526dcb633e5f765a9e5ee084c298109c82c2b50

Date: Mon Apr 4 17:09:15 2016 -0400

Added files via upload

commit 2adf7b4222d7866e0a6c40e185ea1096a06c24c4
Date: Mon Apr 4 17:08:31 2016 -0400

Added files via upload

commit e38441fe0e24901a7b2f886cd987dc41dc62bd67
Date: Mon Apr 4 17:03:34 2016 -0400

Added files via upload

commit fc8452691bc440c8bbdd52c6e158b9066b0fa8bf
Date: Mon Apr 4 17:02:16 2016 -0400

Added files via upload

commit b1476712e47fd2f8e3f58882afc6c540cfdc81a1
Date: Mon Apr 4 16:50:31 2016 -0400

Update ast.ml

commit 928f8216e623165ceed42cacf6e6e56a38576f65
Date: Mon Apr 4 16:45:03 2016 -0400

Added files via upload

commit bf6c7d84e8746bfb17d7cfddc2176923b9a35d33
Date: Mon Apr 4 16:43:34 2016 -0400

Added files via upload

commit 6bd69fd533cecd44b1223aded8e0787b10d727ed
Date: Mon Apr 4 16:42:09 2016 -0400

Added files via upload

commit 4987e793f57d2380d6b67e203227e5191ec8d145
Date: Mon Apr 4 16:39:10 2016 -0400

Added files via upload

commit c5aa748b92efaefb279efe518ee687d44f6fd53a
Date: Mon Apr 4 16:18:55 2016 -0400

Added files via upload

commit 3afb429f563401a4c904a81df007c27c9737a629
Date: Mon Apr 4 12:28:15 2016 -0400

Create parser.mly

parser

commit 0fa10979c7123f90c2a6de3a51232874fb7a5336
Date: Mon Apr 4 00:06:37 2016 -0400

add string type and put function

commit 85381cc84e62cbd0be53ee5074d2a191f59cb52b
Date: Sun Apr 3 16:20:03 2016 -0400

ast

commit 7b81d27de8dc39f65f7c0451eb6cc968d680c924
Date: Sun Apr 3 11:03:21 2016 -0400

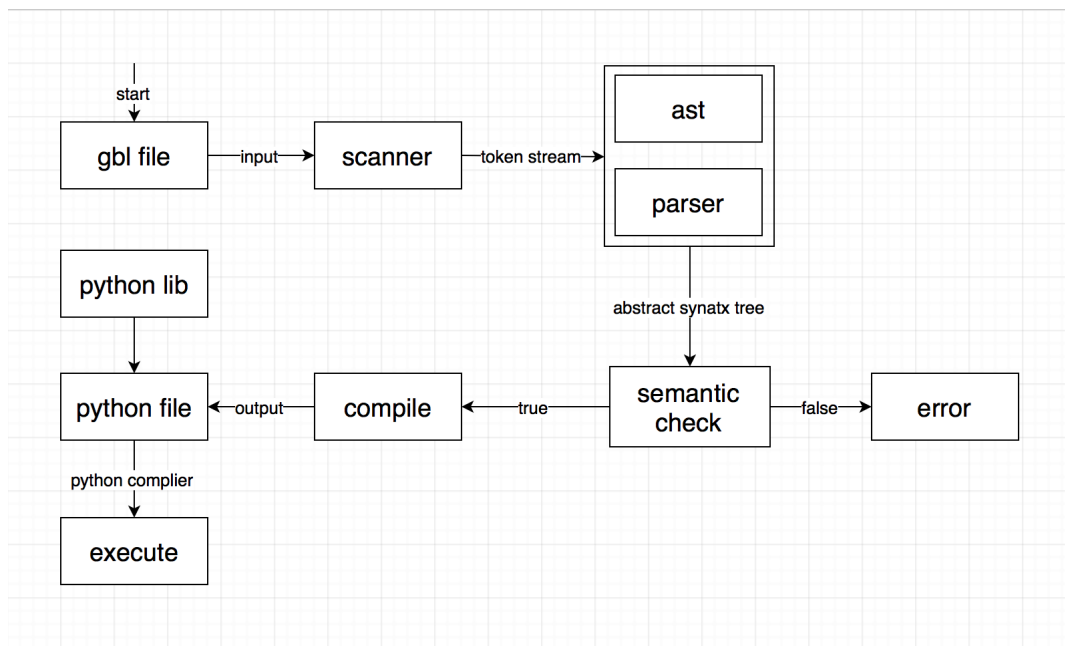
scanner

commit bce39540516b5d254ed901616c6cda9757a15a36
Date: Mon Mar 28 17:00:25 2016 -0400

Initial commit

5 ARCHITECTURAL DESIGN

5.1 ARCHITECTURAL DIAGRAM



5.2 LEXICAL ANALYSIS

The scanner use gbl file as input and transform it into stream of tokens. Also comments and white spaces are discarded through the scanner.

5.3 SYNTACTIC ANALYSIS

The ast and parser build abstract syntax tree of classes, functions statements and expressions which is based on stream of tokens. And program with invalid syntax will be detected.

5.4 SEMANTIC ANALYSIS

The semantic check will take the abstract syntax tree as input and check if the program is semantically valid. Symbol table for each statement will be generated and legal evaluation will be done to each snippet of code.

5.5 CODE GENERATION

The compile part will take the abstract syntax tree as input and match all classes, functions, statements and expressions with python code. Also our compiler will check if the program includes gaming building code and generate python GUI code for the game. The output will be executable python code.

6 LEARNED FROM COURSE

Sihao Zhang:

During the course, I've learned a lot about Compiler, including lexical analysis, semantic analysis, built a general concept about the compiler pipeline. In the meantime, the knowledge of compiler helps me think deeper when using a programming language. Furthermore, this course guides me to the door of functional programming, thinking in type brings a different view of programming, which I benefit from it a lot.

Advice: Compile to LLVM requires deep understanding of this language, so if you plan to compile to LLVM, prepare for it as early as possible and spend a lot of time in understanding LLVM.

Ye Cao:

In this class I learned how to analysis the language and change it to abstract syntax tree. Also I learned how to use Ocaml to write a complete complier. I realized that syntax of programming language is reasonable and the main point of high-level language is to compile it to low-level language and execute on computer. Also what I have learn can be applied to other language processing problems such as text transfer. Through the final project I gained the experience of teamwork and version control.

Advice:

It's hard to compile a high-level language to low-level language such as llvm, especially when the high-level language has many special features.

Shengtong Zhang:

In the course, what the most important I learn is not the Ocaml language, but a thinking about compiling. PLT and also the project tell us how a computer understand the language our human use, like C/C++, python or some other programming language. The course give me a new view to see our language. I can see in some day, our computer could fully understand I am talking now.

As for the advice, I think the team could start the project early and make a clear goal they want to realize.

Yiqing Cui:

I have a better understanding of functional programming language, especially Ocaml.

I also had deep understanding of the entire process of translating source language into target language, including how the source language is scanned for tokens, how these tokens are parsed into abstract syntax tree, how to check the correctness of each node in AST.

A better design is better than 1000 lines of code. Before we get hands on the project, we have to take as many aspects as possible into consideration. That will save a lot of time later, which is way much better than changing designs as we proceed the project.

7 TEST PLAN

As a test-driven developed project, we select 83 basic test cases in the beginning, which are consistent over the whole development process and cover all the possible branches.

7.1 UNIT TEST

Test process is divided by front-end, semantic check, code generation. In the front-end part, we print the abstract syntax tree after parser phase. Over the 83 basic test cases, 39 of them have semantic error and are designed for semantic check. Our semant.ml found all the errors. The other 44 test cases are designed for code generation. They covered all the statement and expressions.

Success test case:

GCD:

```
class mymain extends Main{
int gcd(int a, int b) {
  while (a != b) {
    if (a > b) a = a - b;
    else b = b - a;
  }
  return a;
}

int main()
{
  int a;

  a = gcd(3,15);

  printi(a);
  return 0;
}
}
```

Result: 3

While Loop:

```
class mymain extends Main{
int foo(int a)
{
  int j;

  j = 0;
  while (a > 0) {
    j = j + 2;
    a = a - 1;
    if(a == 5) {
      continue;
    }
  }
  return j;
}
}
```

```
int main()
{
    println(foo(7));
    return 0;
}
```

Result: 14

Print:

```
class mymain extends Main{
int main()
{
    string stringArray [5];
    stringArray [1]="hello";
    stringArray [2]="world";
    int i;
    for(i = 0; i < 5; i = i + 1){
        println(stringArray[i]);
    }
    return 0;
}
}
```

Result:

hello

world

Failed test cases:

Class test:

```
class mygame extends Game{
int c;
int d=5;
void fob(int dd, int ff){
    int cc = 10;
    int ss;
}
}

class myai extends AI{
int e;
int f;
```



```

}
class myMain extends Main{
int a=5.2;
bool b;

void foo(int c, bool d)
{
    mygame myg;
    int d=5.9;
    int k;
    float ss=false;
    d=10;
}

int main()
{
    foo(1, false);
    return 0;
}
}

```

Result: Fatal error: exception Failure("illegal init int = float in declaration int a = 5.2")

continue out of for block:

```

class mymain extends Main{
int main()
{
    int i;
    continue;
    for (i = 0; i < 10 ; i = i + 1) {
        println(i);
    }

    return 0;
}
}

```

Result: Fatal error: exception Failure("continue is out of for or while loop")

Assign:

```

class mymain extends Main{
int main()
{
    int intArray[10];
    string stringArray[10];
    intArray[1] = stringArray[1]; #Fail: assigning a string to a int
    return 0;
}
}

```



```

| "false"      {FALSE}
| "int"        {INT}
| "float"      {FLOAT}
| "bool"       {BOOL}
| "string"     {STRING}
| "struct"     {STRUCT}
| "class"      {CLASS}
| "extends"    {EXTENDS}
| "game"       {GAME}
| "player"     {PLAYER}
| "sprite"     {SPRITE}
| "map"        {MAP}
| "void"       {VOID}
| "for"        {FOR}
| "if"         {IF}
| "else"       {ELSE}
| "elif"       {ELIF}
| "break"      {BREAK}
| "continue"   {CONTINUE}
| "while"      {WHILE}
| "return"     {RETURN}

| ['0'-'9']+ as lxm { LITERAL(int_of_string lxm) }
| ['0'-'9']+['.']* as lxm {FLOATCONSTANT(float_of_string lxm)}
| str as lxm {STRINGCONSTANT(strip(Lexing.lexeme lexbuf))}
| ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_']* as lxm { ID(lxm) } |
  eof { EOF }
| _ as char { raise (Failure("illegal character " ^ Char.escaped char))
  }

and comment = parse
  "*/"      { token lexbuf }
| _        { comment lexbuf }

and line_comment = parse
  ['\n' '\r'] { token lexbuf }
| _          { line_comment lexbuf }

```

ast.ml

```

%\begin{lstlisting}[language=Haskell]

(* Author: Shengtong Zhang(sz2539) *)
type op = Add | Sub | Mult | Div | Mod | AddEqual | SubEqual |
  MultEqual | DivEqual | ModEqual | Neq | Less | Leq | Greater | Geq |
  And | Or | Is | At
type uop = Neg | Not
type typ = Int | Bool | Void | Float | String | Game | Player | Sprite
  | Map | Main | Ai
type bind = typ * string

```

```

type expr = Literal of int          | BoolLit of bool
          | FloatLit of float      | StringLit of string
          | Id of string           | Noexpr
          | Binop of expr * op * expr | Unop of uop * expr
          | Assign of string * expr | Call of string * expr list
          | ArrayElement of string * expr | ArrayElementAssign of
            string * expr * expr
          | IdInClass of string * string
          | ArrayInClass of string * expr * string
          | CallDomain of string * expr list * string
          | Negative of op * expr

type init = typ * string * expr
type arraybind = typ * string * expr

type stmt = Block of stmt list      | Expr of expr
          | Ifnoelse of expr * stmt
          | Ifelse of expr * stmt * stmt
          | For of expr * expr * expr * stmt
          | While of expr * stmt      | Return of expr
          | Break                     | Continue | Init of typ * string
            * expr
          | Bind of bind              (* | Init of init *)
          | ArrayBind of arraybind
          | Classdecl of string * string

type global = Bind of bind | ArrayBind of arraybind | Init of typ *
  string * expr

(* type formalvora = Bindinf of typ * string | ArrayBindinf of typ *
  string * expr *)

type func_decl = {
  typ      : typ;
  fname    : string;
  formals  : bind list;
  body     : stmt list;
}

type cbody = {
  vandadecls : global list;
  methods    : func_decl list;
}

type class_decl = {
  cname : string;
  extends : string;
  cbody : cbody;
}

```

```

type program = global list * class_decl list

(* type program = Program of bind list * func_decl list * class_decl
   list *)
(* type program = bind list * func_decl list *)
(* type program = Program of decl_stmt *)
(* Pretty-printing functions *)

let string_of_op = function
  Add -> "+"
  | Sub -> "-"
  | Mult -> "*"
  | Div -> "/"
  | Mod -> "%"
  | Is -> "=="
  | AddEqual -> "+="
  | SubEqual -> "-="
  | MultEqual -> "*="
  | DivEqual -> "/="
  | ModEqual -> "%="
  | Neq -> "!="
  | Less -> "<"
  | Leq -> "<="
  | Greater -> ">"
  | Geq -> ">="
  | And -> "&&"
  | Or -> "||"
  | Is -> "=="
  | At -> "@"

let string_of_uop = function
  Neg -> "-"
  | Not -> "!"

let rec string_of_expr = function
  Literal(l) -> string_of_int l
  | BoolLit(true) -> "true"
  | BoolLit(false) -> "false"
  | FloatLit(f) -> string_of_float f
  | StringLit(s) -> s
  | Id(s) -> s
  | Noexpr -> ""
  | Binop(e1, o, e2) ->
    string_of_expr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_expr
    e2
  | Unop(o, e) -> string_of_uop o ^ string_of_expr e
  | Assign(v, e) -> v ^ " = " ^ string_of_expr e
  | Call(f, el) -> f ^ "(" ^ String.concat ", " (List.map
    string_of_expr el) ^ ")"
  | ArrayElement(v, e) -> v ^ "[" ^ string_of_expr e ^ "]"
  | ArrayElementAssign(v, e1, e2) -> v ^ "[" ^ string_of_expr e1 ^ "]"
    ^ " = " ^ string_of_expr e2

```

```

| IdInClass(s1, s2) -> s1 ^ "@" ^ s2
| ArrayInClass(s1, e, s2) -> s1 ^ "[" ^ string_of_expr e ^ "]"@ ^ s2
| CallDomain(f, el, s) -> f ^ "(" ^ String.concat ", " (List.map
  string_of_expr el) ^ ")"@ ^ s
| Negative(o, e) -> string_of_op o ^ string_of_expr e

let string_of_typ = function
  Int -> "int"
| Bool -> "bool"
| Void -> "void"
| Float -> "float"
| String -> "string"
| Game -> "game"
| Player -> "player"
| Sprite -> "sprite"
| Map -> "map"
| Main -> "main"
| Ai -> "ai"

let typ_of_string = function
  "int" -> Int
| "bool" -> Bool
| "void" -> Void
| "float" -> Float
| "string" -> String
| "game" -> Game
| "player" -> Player
| "sprite" -> Sprite
| "map" -> Map
| "main" -> Main
| "ai" -> Ai
(* let rec string_of_stmt = function
  Block(stmts) ->
    "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^ "}\n"
| Expr(expr) -> string_of_expr expr ^ ";\n"
| Return(expr) -> "return " ^ string_of_expr expr ^ ";\n"
| Ifnoelse() -> "if ("
| If(e, s, Block([])) -> "if (" ^ string_of_expr e ^ ")\n" ^
  string_of_stmt s
| If(e, s1, s2) -> "if (" ^ string_of_expr e ^ ")\n" ^
  string_of_stmt s1 ^ "else\n" ^ string_of_stmt s2
| For(e1, e2, e3, s) ->
  "for (" ^ string_of_expr e1 ^ " ; " ^ string_of_expr e2 ^ " ; "
  ^
  string_of_expr e3 ^ ") " ^ string_of_stmt s
| While(e, s) -> "while (" ^ string_of_expr e ^ ") " ^ string_of_stmt
  s
| Continue -> "continue"
| Break -> "break"
| Bind(b) -> "bind"
| ArrayBind(ab) -> "Arraybind"
| Init(e, s, t) -> "init"
let string_of_vdecl (t, id) = string_of_typ t ^ " " ^ id ^ ";\n"

```

```

let string_of_fdecl fdecl =
  string_of_typ fdecl.typ ^ " " ^
  fdecl.fname ^ "(" ^ String.concat ", " (List.map snd fdecl.formals) ^
  ")\n{\n" ^
  (* String.concat "" (List.map string_of_vdecl fdecl.locals) ^ *)
  String.concat "" (List.map string_of_stmt fdecl.body) ^
  "}\n"
let string_of_program (vars, funcs) =
  String.concat "" (List.map string_of_vdecl vars) ^ "\n" ^
  String.concat "\n" (List.map string_of_fdecl funcs)*

```

parser.mly

```

/* Author: Shengtong Zhang (sz2539), Yiqing Cui (yc3121) */
%{ open Ast %}

%token COMMENT SEMI LPAREN RPAREN LBRACK RBRACK LBRACE RBRACE LANGLE
  RANGLE COMMA
%token PLUS MINUS TIMES DIVIDE MODULE ASSIGN NOT
%token EQ PLUSEQ MINUSEQ TIMESEQ DIVIDEEQ MODULEEQ NEQ
%token LEQ GEQ
%token AND OR
%token RIGHTSHIFT LEFTSHIFT DOMAINOP
%token BITAND BITOR BITXOR BITNEG
%token NEWLINE
%token FOR IF ELSE ELIF BREAK CONTINUE WHILE RETURN END
%token INT BOOL FLOAT STRING GAME PLAYER SPRITE MAP MAIN AI
%token VOID TRUE FALSE
%token GT LT

%token STRUCT CLASS EXTENDS

%token <int> LITERAL
%token <float> FLOATCONSTANT
%token <string> STRINGCONSTANT
%token <string> ID
%token EOF

%nonassoc NOELSE
%nonassoc ELSE
%right ASSIGN
%left OR
%left AND
%left EQ NEQ
%left LANGLE RANGLE
%left LEQ GEQ
%left PLUS MINUS
%left TIMES DIVIDE MODULE
%right NOT

%start program
%type <Ast.program> program

%%

```

```

program:
  decls EOF { $1 }

decls:
  /* nothing */ { [], [] }
  | decls vandadecl { ($2 :: fst $1), snd $1 }
  | decls cdecl { fst $1, ($2 :: snd $1) }

vandadecl:
  typ ID ASSIGN expr SEMI {Init($1, $2, $4)}
  | vdecl {Bind $1}
  | array_decl {ArrayBind $1}

vdecl:
  typ ID SEMI { ($1, $2) }

array_decl:
  typ ID LBRACK expr RBRACK SEMI {($1, $2, $4)}

/***** function *****/

fdecl:
  typ ID LPAREN formals_opt RPAREN LBRACE stmt_list RBRACE
  { { typ = $1;
    fname = $2;
    formals = $4;
    body = List.rev $7 } }

formals_opt:
  /* nothing */ { [] }
  | formal_list { List.rev $1 }

formal_list:
  typ ID { [($1,$2)] }
  | typ ID LBRACK RBRACK { [($1,$2)] }
  | formal_list COMMA typ ID LBRACK RBRACK { ($3, $4) :: $1 }
  | formal_list COMMA typ ID {($3, $4) :: $1 }

/***** class *****/

cdecl:
  CLASS ID EXTENDS ID LBRACE cbody RBRACE {{
    cname = $2;
    extends = $4;
    cbody = $6;
  }}

cbody:
  { {
    vandadecls = [];
    methods = [];
  } }

```



```

|   cbody vandadecl { {
|     vandadecls = $2 :: $1.vandadecls;
|     methods = $1.methods;
|   } }
|   cbody fdecl { {
|     vandadecls = $1.vandadecls;
|     methods = $2 :: $1.methods;
|   } }

/***** other type *****/

typ:
  INT      { Int      }
| BOOL     { Bool     }
| FLOAT    { Float    }
| STRING   { String   }
| VOID     { Void     }
| GAME     { Game     }
| PLAYER   { Player   }
| SPRITE   { Sprite   }
| MAP      { Map      }
| MAIN     { Main     }
| AI       { Ai       }

/***** statement *****/

stmt_list:
  /* nothing */ { [] }
| stmt_list stmt { $2 :: $1 }

stmt:
  expr SEMI { Expr $1 }
| RETURN SEMI { Return Noexpr }
| RETURN expr SEMI { Return $2 }
| BREAK SEMI {Break}
| CONTINUE SEMI {Continue}
| LBRACE stmt_list RBRACE { Block(List.rev $2) }
| IF LPAREN expr RPAREN stmt { Ifnoelse($3, $5) }
| IF LPAREN expr RPAREN stmt ELSE stmt { Ifelse($3, $5, $7) }
| FOR LPAREN expr_opt SEMI expr SEMI expr_opt RPAREN stmt
  { For($3, $5, $7, $9) }
| WHILE LPAREN expr RPAREN stmt { While($3, $5) }
| typ ID ASSIGN expr SEMI {Init($1, $2, $4)}
| vdecl {Bind $1}
| array_decl {ArrayBind $1}
| ID ID SEMI { Classdecl($1, $2) }

/***** expr *****/

```

```

expr_opt:
  /* nothing */ { Noexpr }
  | expr        { $1 }

expr:
  LITERAL          { Literal($1) }
  | FLOATCONSTANT  { FloatLit($1) }
  | STRINGCONSTANT { StringLit($1) }
  | TRUE           { BoolLit(true) }
  | FALSE          { BoolLit(false) }
  | ID             { Id($1) }
  | ID LBRACK expr RBRACK {ArrayElement($1, $3)}
  | ID DOMAINOP ID   { IdInClass($1, $3)}
  | ID LBRACK expr RBRACK DOMAINOP ID { ArrayInClass($1, $3, $6) }
  | expr PLUS expr   { Binop($1, Add, $3) }
  | expr MINUS expr  { Binop($1, Sub, $3) }
  | MINUS expr       { Negative(Sub, $2) }
  | expr TIMES expr  { Binop($1, Mult, $3) }
  | expr DIVIDE expr { Binop($1, Div, $3) }
  | expr MODULE expr { Binop($1, Mod, $3) }
  | expr EQ expr     { Binop($1, Is, $3) }
  | expr NEQ expr    { Binop($1, Neq, $3) }
  | expr LT expr     { Binop($1, Less, $3) }
  | expr LEQ expr    { Binop($1, Leq, $3) }
  | expr GT expr     { Binop($1, Greater, $3) }
  | expr GEQ expr    { Binop($1, Geq, $3) }
  | expr AND expr    { Binop($1, And, $3) }
  | expr OR expr     { Binop($1, Or, $3) }
  | NOT expr         { Unop(Not, $2) }
  | ID ASSIGN expr   { Assign($1, $3) }
  | ID LBRACK expr RBRACK ASSIGN expr {ArrayElementAssign($1, $3, $6)}
  | ID LPAREN actuals_opt RPAREN { Call($1, $3) }
  | ID LPAREN actuals_opt RPAREN DOMAINOP ID { CallDomain($1, $3, $6) }
  | LPAREN expr RPAREN { $2 }

init:
  typ ID ASSIGN expr {($1, $2, $4)}

actuals_opt:
  /* nothing */ { [] }
  | actuals_list { List.rev $1 }

actuals_list:
  expr          { [$1] }
  | actuals_list COMMA expr { $3 :: $1 }

```

codegen.ml

```

(* Author: Ye Cao(yc3113) *)
module A = Ast

module StringMap = Map.Make(String)

```

```

let translate (globals, classes) =

  let load_lib f =
    let ic = open_in f in
    let n = in_channel_length ic in
    let s = String.create n in
    really_input ic s 0 n; close_in ic; (s)
  in

  let gen_game_var_code = load_lib "lib/game_vars.py" in

  let gen_game_init_code = load_lib "lib/game_init.py" in

  let gen_game_gui_code = load_lib "lib/game_gui.py" in

  let find_var vname m = function
    A.Classdecl (t, s) -> if t = vname then (StringMap.add s "" m)
      else m
    | _ as s -> m
  in

  let find_fun vname m = function
    A.Expr e -> (match e with
      A.CallDomain (f, act, s) -> if f = vname then (
        StringMap.add s "" m) else m
      | _ -> m)
    | _ as s -> m
  in

  let find_in_fun fname vname ty m fdecl =
    if ty = "var" then (
      if fdecl.A.fname = fname then
        (List.fold_left (find_var vname) m fdecl.A.body)
      else m )
    else
      (if fdecl.A.fname = fname then
        (List.fold_left (find_fun vname) m fdecl.A.body)
      else m )
  in

  let find_in_class fname cname vname ty m cdecl =
    if cdecl.A.extends = cname then
      (List.fold_left (find_in_fun fname vname ty) m cdecl.A.cbody.A.
        methods)
    else m
  in

  let find_class name cdecl =
    if cdecl.A.extends = name then cdecl.A.cname else ""
  in

  let main_class = String.concat "" (List.map (find_class "Main")
    classes) in

```

```

let game_class = String.concat "" (List.map (find_class "Game")
  classes) in
let ai_class = String.concat "" (List.map (find_class "AI") classes)
  in

let ai_vars = StringMap.bindings (if ai_class = "" then StringMap.
  empty
  else List.fold_left (find_in_class "main" "Main"
    ai_class "var") StringMap.empty classes) in
let game_vars = List.fold_left (find_in_class "main" "Main"
  game_class "var") StringMap.empty classes in
let init_vars = List.fold_left (find_in_class "main" "Main" "
  initialize" "fun") StringMap.empty classes in
let game_object_vars = StringMap.bindings (StringMap.merge (fun k xo
  yo -> match xo, yo with

```

```

Some x, Some y -> xo | _ -> None) game_vars init_vars) in
let game_gui_code = let ai_name = if (List.length ai_vars) = 1 then (
  match ai_vars with
    (k, v) :: tl -> k
    | _ -> "None") else "None"
  in
  if (List.length game_object_vars) = 1 then
    let game_object_name = (match game_object_vars with
      (k, v) :: tl -> k
      | _ -> "None")
      in
      "
      _root = tk.Tk()
      _root.title(\"GBL\")
      tk.Canvas.create_circle = _create_circle
      _board = GameBoard(_root, " ^ game_object_name ^" , " ^ ai_name
        ^ ")
      _board.pack(side=\"top\", fill=\"both\", expand=\"true\", padx
        =4, pady=4)
      _root.mainloop()\n"
    else ""
  in

let gen_main_class_code =
  "if __name__ == \"__main__\":\n" ^
  "    _main = " ^ main_class ^ "()\n" ^
  "    _main.main()\n"
  in

let map_to_list m = function _ as s-> StringMap.add s "self." m in

let game_var_map =
  List.fold_left map_to_list StringMap.empty
  ["WithAI";"NextSpriteID";"NextPlayerID";"FormerPosition";"FormerId"
    ;"SpriteOwnerId";
  "SpriteId";"PlayerName";"PlayerId";"PlayerNumber";"GridNum";"
    MapSize";"InputPosition"]

```

```

in

(*compleie function parameters*)
let comp_param = function
  (t, n) -> n
  | _ -> ""
in

(*compleie symbols*)
let comp_sym = function
  A.Add      -> " + "
  | A.Sub     -> " - "
  | A.Mult    -> " * "
  | A.Div     -> " / "
  | A.And     -> " and "
  | A.Is      -> " == "
  | A.Or      -> " or "
  | A.Neq     -> " != "
  | A.Less    -> " < "
  | A.Leq     -> " <= "
  | A.Greater -> " > "
  | A.Geq     -> " >= "
  | A.Mod     -> " % "
  | _        -> " "
in

(*compleie expressions*)
let rec comp_global_expr = function
  A.Literal i -> string_of_int i
  | A.StringLit s -> s
  | A.FloatLit f -> string_of_float f
  | A.BoolLit b -> ( match b with
    true   -> "True"
    | false -> "False" )
  | A.Noexpr -> ""
  | A.Id s -> s
  | A.ArrayElement (s, i) -> s ^ "[" ^ comp_global_expr i ^ "]"
  | A.ArrayElementAssign (s, i, v) -> s ^ "[" ^ comp_global_expr i ^
    "]" ^ "=" ^ comp_global_expr v
  | A.Binop (e1, op, e2) ->
    let e1' = comp_global_expr e1
    and e2' = comp_global_expr e2 in
    "(" ^ e1' ^ ")" ^ comp_sym op ^ "(" ^ e2' ^ ")"
  | A.Negative (op, e) ->
    let e' = comp_global_expr e in
    comp_sym op ^ "(" ^ e' ^ ")"
  | A.Unop(op, e) -> "not (" ^ comp_global_expr e ^ ")"
  | A.Assign (s, e) -> s ^ " = " ^ comp_global_expr e
  | A.Call (f, act) -> f ^ "(" ^ String.concat ", " (List.map
    comp_global_expr act) ^ ")"
in

(*compleie global variables*)
let comp_global_var = function

```

```

    A.Bind e -> (match e with
      (t, n) -> n ^ " = None\n"
    | _ -> "")
  | A.ArrayBind e -> (match e with
    (t, n, a) -> n ^ " = [ None ] * " ^ comp_global_expr
      a ^ "\n"
    | _ -> "")
  | A.Init (t, n, v) -> n ^ " = " ^ comp_global_expr v ^ "\n"
in

```

```

(*compile class body*)
let comp_cbody extends cbody =

  let init_var m vdecl =
    StringMap.add (match vdecl with
      A.Bind e -> (match e with (t, n) -> n | _ -> "")
    | A.ArrayBind e -> (match e with (t, n, a) -> n | _
      -> "")
    | A.Init (t, n, v) -> n) "self." m
  in

  let init_fun m fdecl =
    StringMap.add fdecl.A.fname "self." m
  in

  let local_vars = List.fold_left init_var StringMap.empty cbody.A.
    vandadecls in
  let local_funs = List.fold_left init_fun StringMap.empty cbody.A.
    methods in

  let lookup_fun f = try StringMap.find f local_funs
    with Not_found -> ""
  in

  let rec comp_local_expr lookup = function
    A.Literal i -> string_of_int i
  | A.StringLit s -> "\"" ^ s ^ "\""
  | A.FloatLit f -> string_of_float f
  | A.BoolLit b -> ( match b with
    true -> "True"
  | false -> "False" )
  | A.Noexpr -> ""
  | A.Id s -> lookup s ^ s
  | A.ArrayElement (s, i) -> lookup s ^ s ^ "[" ^ comp_local_expr
    lookup i ^ "]"
  | A.ArrayElementAssign (s, i, v) -> lookup s ^ s ^ "[" ^
    comp_local_expr lookup i ^ "]" ^ "=" ^ comp_local_expr lookup
    v
  | A.Binop (e1, op, e2) ->
    let e1' = comp_local_expr lookup e1
    and e2' = comp_local_expr lookup e2 in
    "(" ^ e1' ^ ")" ^ comp_sym op ^ "(" ^ e2' ^ ")"
  | A.Negative (op, e) ->
    let e' = comp_global_expr e in

```

```

    comp_sym op ^ "(" ^ e' ^ ")"
  | A.IdInClass (e, s) -> lookup s ^ s ^ "." ^ e
  | A.ArrayInClass (e, i, s) -> lookup s ^ s ^ "." ^ e ^ "[" ^
    comp_local_expr lookup i ^ "]"
  | A.Unop(op, e) -> "not (" ^ comp_local_expr lookup e ^ ")"
  | A.Assign (s, e) -> lookup s ^ s ^ " = " ^ comp_local_expr
    lookup e
  | A.Call ("printi", [e]) | A.Call ("printb", [e]) | A.Call ("
    printf", [e]) | A.Call ("prints", [e]) ->
    "sys.stdout.write " ^ "(str(" ^ comp_local_expr lookup e ^ "))"
  | A.Call ("println", [e]) | A.Call ("printlnb", [e]) | A.Call ("
    printlnf", [e]) | A.Call ("printlns", [e]) ->
    "print " ^ "(" ^ comp_local_expr lookup e ^ ")"
  | A.Call (f, act) -> lookup_fun f ^ f ^ "(" ^ String.concat ", "
    (List.map (
comp_local_expr lookup) act) ^ ")"
  | A.CallDomain (f, act, s) -> lookup s ^ s ^ "." ^ f ^ "(" ^
    String.concat ", " (List.map (comp_local_expr lookup) act) ^ "
    )"
  | _ as s -> ""
in

(*compile global variables*)
let comp_local_var pos lookup = function
  A.Bind e -> (match e with
    (t, n) -> (String.make (pos * 4) ' ') ^ lookup n ^
      n ^ " = None\n"
    | _ -> "")
  | A.ArrayBind e -> (match e with
    (t, n, a) -> (String.make (pos * 4) ' ') ^ lookup
      n ^ n ^ " = [ None ] * " ^ comp_local_expr
      lookup a ^ "\n"
    | _ -> "")
  | A.Init (t, n, v) -> (String.make (pos * 4) ' ') ^ lookup n ^ n
    ^ " = " ^ comp_local_expr lookup v ^ "\n"
in

(*compile statements*)
let rec comp_stmt pos lookup = function
  A.Block sl -> String.concat "" (List.map (comp_stmt pos lookup)
    sl)
  | A.Expr e -> (String.make (pos * 4) ' ') ^ comp_local_expr
    lookup e ^ "\n"
  | A.Return A.Noexpr -> (String.make (pos * 4) ' ') ^ "return\n"
  | A.Break -> (String.make (pos * 4) ' ') ^ "break\n"
  | A.Continue -> (String.make (pos * 4) ' ') ^ "continue\n"
  | A.Return e -> (String.make (pos * 4) ' ') ^ "return " ^
    comp_local_expr lookup e ^ "\n"
  | A.Bind e -> (match e with
    (t, n) -> (String.make (pos * 4) ' ') ^ lookup n ^
      n ^ " = None"
    | _ -> "") ^ "\n"
  | A.ArrayBind e -> (match e with
    (t, n, a) -> (String.make (pos * 4) ' ') ^ lookup

```

```

        n ^ n ^ " = [ None ] * " ^ comp_local_expr
        lookup a
        | _ -> "" ^ "\n"
| A.Init (t, n, v) -> (String.make (pos * 4) ' ') ^ lookup n ^ n
  ^ " = " ^ comp_local_expr lookup v ^ "\n"
| A.Classdecl (t, s) -> (String.make (pos * 4) ' ') ^ lookup s ^
  s ^ " = " ^ t ^ "()\n"
| A.Ifelse (predicate, then_stmt, else_stmt) -> (String.make (pos
  * 4) ' ') ^ "if (" ^ comp_local_expr lookup predicate ^ "):\n"
  " ^
  comp_stmt (pos + 1)
  lookup then_stmt ^
  (String.make (pos
  * 4) ' ') ^ "else
  :\n" ^
  comp_stmt (pos + 1)
  lookup else_stmt
| A.Ifnoelse (predicate, then_stmt) -> (String.make (pos * 4) ' '
  ) ^ "if (" ^ comp_local_expr lookup predicate ^ "):\n" ^
  comp_stmt (pos + 1)
  lookup then_stmt
| A.For (e1, e2, e3, body) -> (match e1 with
  A.Noexpr -> ""
  | _ -> (String.make (pos * 4) ' ')
    ^ comp_local_expr lookup e1 ^ "\n"
    ^
    (String.make (pos * 4) ' ') ^
    "while (" ^ comp_local_expr lookup
    e2 ^ "):\n" ^ comp_stmt (pos +
    1) lookup body ^
    (match e3 with
    A.Noexpr -> ""
    | _ -> (String.make ((pos + 1) * 4)
      ' ') ^ comp_local_expr lookup
      e3 ^ "\n")
| A.While (predicate, body) -> (String.make (pos * 4) ' ') ^ "
  while (" ^ comp_local_expr lookup predicate ^ "):\n" ^
  comp_stmt (pos + 1) lookup body
| _ as s -> ""
in

let comp_class_var pos vdecls =
  (String.make (pos * 4) ' ') ^ "def __init__(self):\n" ^
  String.concat "" (List.map (comp_local_var (pos + 1)
    (fun n -> try StringMap.find n local_vars
      with Not_found -> "")) vdecls) ^
  (if extends = "Game" then (gen_game_var_code ^ "\n") else "") ^
  (String.make ((pos + 1) * 4) ' ') ^ "pass" ^ "\n\n"
in

let comp_function pos fdecl =
  (String.make (pos * 4) ' ') ^ "def " ^ fdecl.A.fname ^ "(" ^
  String.concat "," ("self" :: (
    if extends = "Game" && (fdecl.A.fname = "win" || fdecl.A.fname

```



```

        = "isLegal" || fdecl.A.fname = "update")
    then []
    else List.map comp_param fdecl.A.formals)) ^
    "):\n" ^ String.concat "" (List.map (comp_stmt (pos + 1)
      (if extends = "Game" && (fdecl.A.fname = "win" || fdecl.A.
        fname = "isLegal" || fdecl.A.fname = "update")
      then
        (fun n -> try StringMap.find n local_vars
          with Not_found -> try StringMap.find n game_var_map
            with Not_found -> "")
        else
          (fun n -> try StringMap.find n local_vars
            with Not_found -> ""))
      ) fdecl.A.body) ^
    (if extends = "Main" && fdecl.A.fname = "main" then game_gui_code
      else "") ^
    (String.make ((pos + 1) * 4) ' ') ^ "pass" ^ "\n\n"
  in
    comp_class_var 1 cbody.A.vandadecls ^ (if extends = "Game" then (
      gen_game_init_code ^ "\n") else "") ^
    String.concat "" (List.map (comp_function 1) cbody.A.methods)
  in
    let comp_class cdecl =
      (*comp_cbody (cdecl.A.cname ^ "_" ^ cdecl.A.extends ^ "_") cdecl.A.
        cbody*)
      "class " ^ cdecl.A.cname ^ ":\n" ^ comp_cbody cdecl.A.extends cdecl.
        A.cbody ^ "\n"
    in
      "import sys\n" ^ String.concat "" (List.map comp_global_var globals)
      ^
      (if (String.length game_gui_code) > 0 then gen_game_gui_code else "")
      ^
      String.concat "" (List.map comp_class classes) ^
      gen_main_class_code ^ "\n"

```

semant.ml

```

% \begin{lstlisting}[language=Haskell]
(* Author: Sihao Zhang(sz2558), Yiqing Cui(yc3121) *)
(* Semantic checking for the GBL compiler *)

open Ast

module StringMap = Map.Make(String)

(* Semantic checking of a program. Returns void if successful,
  throws an exception if something is wrong.
  Check each global variable, then check each function *)
let check (vandadecls, revcdecls) =
  let cdecls = List.rev revcdecls in
  (*let print_cdecl cdecl =

```

```

    print_string (cdecl.cname ^ "@" ^ cdecl.extends)
in
List.iter print_cdecl cdecls;*)
(* Raise an exception if the given list has a duplicate *)
let built_in_decls =
List.fold_left (fun map (key, value) ->
    StringMap.add key value map
) StringMap.empty [("printi", { typ = Void; fname = "printi"; formals
    = [(Int, "x")];
body = [] }); ("printb", { typ = Void; fname = "printb"; formals = [(
    Bool, "x")];
body = [] }); ("printlni", { typ = Void; fname = "printlni"; formals
    = [(Int, "x")];
body = [] }); ("printlnb", { typ = Void; fname = "printlnb";
    formals = [(Bool, "x")];
body = [] }); ("printf", { typ = Void; fname = "printf"; formals =
    [(Float, "x")];
body = [] }); ("prints", { typ = Void; fname = "prints"; formals =
    [(String, "x")];
body = [] }); ("printlnf", { typ = Void; fname = "printlnf";
    formals = [(Float, "x")];
body = [] }); ("printlns", { typ = Void; fname = "printlns";
    formals = [(String, "x")];
body = [] }); ("Game@initialize", { typ = Void; fname = "
    initialize"; formals = [(Int, "x");(Int, "x");(Int, "x");(
    String, "x");(Bool, "x")];
body = [] }))]

in

let add_function_of_class crr_func_list cdecl =
    let class_name = cdecl.extends in
    let tmpAddFunction m fd = StringMap.add (class_name ^ "@" ^ fd.
        fname) fd m in
    List.fold_left tmpAddFunction crr_func_list cdecl.cbody.methods

in

let function_decls = List.fold_left add_function_of_class
    built_in_decls cdecls

in

let print_funcdecl key value =
    print_string ((string_of_typ value.typ) ^ " " ^ key ^ "\n")

in

let mapChildtoParentClass classmap cdecl =
    StringMap.add cdecl.cname cdecl.extends classmap

in

let class_decls =
    List.fold_left mapChildtoParentClass StringMap.empty cdecls

in

let class_decl child = try StringMap.find child class_decls

```

```

        with Not_found -> raise (Failure ("unrecognized class " ^ child
        ))
    in
    (*StringMap.iter print_funcdecl function_decls;*)
    let function_decl scope s = try StringMap.find s function_decls
        with Not_found -> try StringMap.find (scope ^ "@" ^ s)
            function_decls
            with Not_found -> raise (Failure ("unrecognized function " ^ s
            ))
    in

```

```

let report_duplicate exceptf list =
    let rec helper = function
        n1 :: n2 :: _ when n1 = n2 -> raise (Failure (exceptf n1))
        | _ :: t -> helper t
        | [] -> ()
    in helper (List.sort compare list)
in

(* Raise an exception if a given binding is to a void type *)
let check_not_void exceptf = function
    (Void, n) -> raise (Failure (exceptf n))
    | _ -> ()
in

let print_symbolList key value =
    print_string(key ^ " " ^ (string_of_ttyp value) ^ "\n")
in
(* Raise an exception of the given rvalue type cannot be assigned to
the given lvalue type *)
let check_assign lvaluet rvaluet err =
    if lvaluet == rvaluet then lvaluet else raise err
in

let check_exist s symbol_list =
    if StringMap.mem s symbol_list then raise (Failure ("identifier "
        ^ s ^ " exists"))
    else ()
in

let check_valid_extend s =
    try List.mem s ["Game";"AI";"Main"]
        with Not_found -> raise(Failure ("invalid base class " ^ s))
in

let type_of_identifier s symbol_list =
    try StringMap.find s symbol_list
        with Not_found -> raise (Failure ("undeclared identifier " ^ s))
in

let type_of_object s object_list =
    try StringMap.find s object_list

```

```

    with Not_found -> raise (Failure ("undeclared object " ^ s))
in
let print_object key value =
  print_string(key ^ " " ^ value ^ "\n")
in
let rec check_expr (expr_symbol_list, className) = function
  Literal _ -> Int
| BoolLit _ -> Bool
| FloatLit _ -> Float
| StringLit _ -> String
| Id s -> type_of_identifier (className ^ "@" ^ s) expr_symbol_list

| Binop(e1, op, e2) as e -> let t1 = check_expr (expr_symbol_list,
  className) e1
                                and t2 = check_expr (expr_symbol_list,
  className) e2 in
(match op with
  Add | Sub | Mult | Div | AddEqual | SubEqual | MultEqual |
  DivEqual when t1 = Int && t2 = Int -> Int
| Add | Sub | Mult | Div | AddEqual | SubEqual | MultEqual |
  DivEqual when t1 = Float && t2 = Int -> Float
| Add | Sub | Mult | Div | AddEqual | SubEqual | MultEqual |
  DivEqual when t1 = Int && t2 = Float -> Float
| Add | Sub | Mult | Div | AddEqual | SubEqual | MultEqual |
  DivEqual when t1 = Float && t2 = Float -> Float
| Mod | ModEqual when t1 = Int && t2 = Int -> Int
| Is | Neq when t1 = t2 -> Bool
| Less | Leq | Greater | Geq when t1 = Int && t2 = Int -> Bool
| Less | Leq | Greater | Geq when t1 = Int && t2 = Float ->
  Bool
| Less | Leq | Greater | Geq when t1 = Float && t2 = Int ->
  Bool
| Less | Leq | Greater | Geq when t1 = Float && t2 = Float ->
  Bool
| And | Or when t1 = Bool && t2 = Bool -> Bool
| _ -> raise (Failure ("illegal binary operator " ^
  string_of_typ t1 ^ " " ^ string_of_op op ^ " " ^
  string_of_typ t2 ^ " in " ^ string_of_expr e)))

| Unop(op, e) as ex -> let t = check_expr (expr_symbol_list,
  className) e in
(match op with
  Neg when t = Int -> Int
| Neg when t = Float -> Float
| Not when t = Bool -> Bool
| _ -> raise (Failure ("illegal unary operator " ^ string_of_uop
  op ^
  string_of_typ t ^ " in " ^ string_of_expr ex)))

| ArrayElement(var, e) ->
  let lt = type_of_identifier (className ^ "@" ^ var)
  expr_symbol_list
  and rt = check_expr (expr_symbol_list, className)

```

```

        e in
        (match rt with
         Int -> lt
         | _ -> raise (Failure("array subscript is not
                               integer in " ^ var)))
| _ -> raise (Failure("Declaration not valid"))

in

```

```

let check_vandadecl (sybollist, classname) = function
  Bind(b) -> (*print_string ((string_of_typ (fst b)) ^ " " ^ (snd
    b) ^ "\n");*)
    check_not_void (fun n -> "illegal void " ^ n) b;
    check_exist (classname ^ "@" ^ (snd b)) sybollist;
    let typstring = (fst b) and idstring = (classname ^ "
      @" ^ (snd b)) in
    let new_symbol_list = StringMap.add idstring
      typstring sybollist in
    (*ignore(StringMap.iter print_sybollist
      new_symbol_list);*)
      (new_symbol_list, classname)

| Init(t, s, e) -> let rt = check_expr (sybollist, classname) e in
  check_assign t rt (
    Failure ("illegal
             init " ^
             string_of_typ t ^
             " = " ^ string_of_typ rt
             ^ " in declaration "
             ^ (string_of_typ t)
             ^ " " ^ s ^ " = " ^
             string_of_expr e));
  ignore(check_not_void (fun n -> "illegal void "
    ^ n) (t, s));
  ignore(check_exist (classname ^ "@" ^ s)
    sybollist);
  ignore(check_expr (sybollist, classname) e);
  let new_symbol_list = StringMap.add (classname ^
    @" ^ s) t sybollist in (new_symbol_list,
    classname)

| ArrayBind((t, s, e)) -> ignore(check_not_void (fun n -> "illegal
  void" ^ n) (t, s));
  ignore(check_exist (classname ^ "@" ^ s)
    sybollist);
  let ty = check_expr (sybollist,
    classname) e
  in let _ =
    (match ty with Int -> ()
     | _ -> raise (Failure("
       array subscript is
       not integer in " ^ s
       )))

in

```

```

                                let new_symbol_list = StringMap.add
                                    (classname ^ "@" ^ s) t
                                    symbollist in (new_symbol_list,
                                        classname)

in

let (symbols, _) = List.fold_left check_vandadecl (StringMap.empty,
    "Global") vandadecls
in

let check_cdecl symbol_list cdecl =
    let _ = check_valid_extend cdecl.extends in
    let check_class (classSymbolList, className, globals, functions
        ) =

        let (symbol_table, _) = List.fold_left check_vandadecl (
            classSymbolList, className) globals in

        (**** Checking Functions ****)
        report_duplicate (fun n -> "duplicate function " ^ n) (List
            .map (fun fd -> fd.fname) functions);
        (* Function declaration for a named function *)

        ignore(function_decl "Main" "main"); (* Ensure "main" is
            defined *)
        let symbol_table_temp = symbol_table in
        let check_function func =

            List.iter (check_not_void (fun n -> "illegal void formal
                " ^ n ^ " in " ^ func.fname)) func.formals;

            report_duplicate (fun n -> "duplicate formal " ^ n ^ " in
                " ^ func.fname)
            (List.map snd func.formals);

            let addToMap stringmap (key, value) =
                StringMap.add (classname ^ "@" ^ value) key stringmap
            in

```

```

let formal_symbol_list = List.fold_left addToMap symbol_table_temp func
    .formals in

    (* Type of each variable (global, formal, or local *)
    let rec expr (expr_symbol_list, expr_object_list) =
        function
            Literal _ -> Int
          | BoolLit _ -> Bool
          | FloatLit _ -> Float
          | StringLit _ -> String
          | Id s -> type_of_identifier (classname ^ "@" ^ s)
              expr_symbol_list
          | Binop(e1, op, e2) as e -> let t1 = expr (

```

```

expr_symbol_list, expr_object_list) e1
                                and t2 = expr (
                                    expr_symbol_list,
                                    expr_object_list) e2
                                in
(match op with
  Add | Sub | Mult | Div | AddEqual | SubEqual |
  MultEqual | DivEqual when t1 = Int && t2 = Int
  -> Int
| Add | Sub | Mult | Div | AddEqual | SubEqual |
  MultEqual | DivEqual when t1 = Float && t2 = Int
  -> Float
| Add | Sub | Mult | Div | AddEqual | SubEqual |
  MultEqual | DivEqual when t1 = Int && t2 = Float
  -> Float
| Add | Sub | Mult | Div | AddEqual | SubEqual |
  MultEqual | DivEqual when t1 = Float && t2 =
  Float -> Float
| Mod | ModEqual when t1 = Int && t2 = Int -> Int
| Is | Neq when t1 = t2 -> Bool
| Less | Leq | Greater | Geq when t1 = Int && t2 =
  Int -> Bool
| Less | Leq | Greater | Geq when t1 = Int && t2 =
  Float -> Bool
| Less | Leq | Greater | Geq when t1 = Float && t2
  = Int -> Bool
| Less | Leq | Greater | Geq when t1 = Float && t2
  = Float -> Bool
| And | Or when t1 = Bool && t2 = Bool -> Bool
| _ -> raise (Failure ("illegal binary operator " ^
  string_of_typ t1 ^ " " ^ string_of_op op ^ "
  " ^
  string_of_typ t2 ^ " in " ^ string_of_expr e
  )))

| Unop(op, e) as ex -> let t = expr (expr_symbol_list,
  expr_object_list) e in
  (match op with
    Neg when t = Int -> Int
  | Neg when t = Float -> Float
  | Not when t = Bool -> Bool
  | _ -> raise (Failure ("illegal unary operator " ^
    string_of_uop op ^
    string_of_typ t ^ " in " ^ string_of_expr ex
    )))

| Noexpr -> Void
| Assign(var, e) as ex -> let lt = type_of_identifier (className ^ "@"
  ^ var) expr_symbol_list
                                and rt = expr (
                                    expr_symbol_list,
                                    expr_object_list) e
                                in check_assign lt rt
                                    (Failure ("illegal

```

```

assignment " ^
string_of_typ lt ^
" = " ^ string_of_typ rt
^ " in " ^
string_of_expr ex))
| Call(fname, actuals) as call -> (*print_string (
className ^ "@" ^ fname); *)
let fd = function_decl
className fname in
if List.length actuals != List.length fd.formals then
raise (Failure ("expecting " ^ string_of_int
(List.length fd.formals) ^ " arguments in " ^
string_of_expr call))
else
List.iter2 (fun (ft, _) e -> let et = expr (
expr_symbol_list, expr_object_list) e in
ignore (check_assign ft et
(Failure ("illegal actual argument found "
^ string_of_typ et ^
" expected " ^ string_of_typ ft ^
" in " ^ string_of_expr e))))
fd.formals actuals;
fd.typ

```

```

| ArrayElement(var, e) -> let lt = type_of_identifier (className ^ "@"
^ var) expr_symbol_list
and rt = expr (
expr_symbol_list,
expr_object_list) e in
(match rt with
Int -> lt
| _ -> raise (Failure("array subscript is not
integer in " ^ var)))
| ArrayElementAssign(var, e1, e2) as aea -> let lt =
type_of_identifier (className ^ "@" ^ var)
expr_symbol_list
and rt = expr (expr_symbol_list, expr_object_list) e1 in
(match rt with Int -> let rrt = expr (
expr_symbol_list, expr_object_list) e2 in
check_assign lt rrt (Failure("
illegal assignment " ^
string_of_typ lt ^
" = " ^ string_of_typ rrt ^ " in " ^
string_of_expr aea)) | _ -> raise (Failure("array subscript is not
integer in " ^ var)))

```

```

| IdInClass(vid, objectname) -> let class_name = type_of_object
objectname expr_object_list in
type_of_identifier
(class_name ^

```



```

                                "@" ^ vid)
                                expr_symbol_list

    | ArrayInClass(vid, subexpr, objectname) -> let subtype
      = expr (expr_symbol_list, expr_object_list) subexpr
      in if subtype = Int then let class_name =
          type_of_object objectname expr_object_list in
          type_of_identifier (class_name ^ "@" ^ vid)
          expr_symbol_list else raise (Failure("array
          subscript is not integer in " ^ vid))
| CallDomain(f_name, acts_opt, objectname) as calld-> let class_name =
  type_of_object objectname expr_object_list in let fd = function_decl
  class_name f_name in if List.length acts_opt != List.length fd.
  formals then raise (Failure ("expecting " ^ string_of_int (List.
  length fd.formals) ^ " arguments in " ^ string_of_expr calld)) else
  List.iter2 (fun (ft, _) e -> let et = expr (expr_symbol_list,
  expr_object_list) e in

```

```

ignore (check_assign ft et (Failure ("illegal actual argument found " ^
  string_of_typ et ^ " expected " ^ string_of_typ ft ^ " in " ^
  string_of_expr e)))) fd.formals acts_opt; fd.typ
| Negative(_, negexpr) -> expr (expr_symbol_list, expr_object_list)
  negexpr in
let check_bool_expr (symbol_list, object_list) e = if expr (symbol_list
, object_list) e != Bool then raise (Failure ("expected Boolean
expression in " ^ string_of_expr e)) else () in
let rec stmt (stmt_symbol_list, stmt_object_list, for_while_flag) =
function Block sl -> let rec check_block (block_symbol_list,
block_object_list) = function [Return _ as s] -> stmt (
block_symbol_list, block_object_list, for_while_flag) s | Return _
:: _ -> raise (Failure "nothing may follow a return") | Block sl ::
ss -> ignore(check_block (block_symbol_list, block_object_list) sl);
  check_block (block_symbol_list, block_object_list) ss;
      | s :: ss -> let (rstsymbol_list,
        rstobject_list) = stmt (
          block_symbol_list,
          block_object_list, for_while_flag)
        s

```

```

in check_block (rstsymbol_list, rstobject_list) ss;
    | [] -> (block_symbol_list,
      block_object_list)

      in check_block (stmt_symbol_list,
        stmt_object_list) sl

    | Expr e -> (*print_string (string_of_expr e); *)
      ignore (expr (stmt_symbol_list,
        stmt_object_list) e);
      (stmt_symbol_list, stmt_object_list)

```

```

| Return Noexpr -> if Void = func.typ then (
    stmt_symbol_list, stmt_object_list) else
    raise (Failure ("return gives void
                    expected " ^
                    string_of_ttyp func.typ ^ " in return
                    "))

```

```

| Return e -> let t = expr (stmt_symbol_list,
    stmt_object_list) e
    in
    if t = func.typ then (
        stmt_symbol_list, stmt_object_list
    ) else
    raise (Failure ("return gives " ^
                    string_of_ttyp t ^ " expected " ^
                    string_of_ttyp func.
                    typ ^ " in " ^
                    string_of_expr e
                ))

| Ifnoelse(p, b) -> check_bool_expr (stmt_symbol_list,
    stmt_object_list) p;
    ignore(stmt (stmt_symbol_list,
        stmt_object_list,
        for_while_flag) b);
    (stmt_symbol_list,
    stmt_object_list)

| Ifelse(p, b1, b2) -> ignore(check_bool_expr (
    stmt_symbol_list, stmt_object_list) p);
    ignore(stmt (stmt_symbol_list,
        stmt_object_list,
        for_while_flag) b1);
    ignore(stmt (stmt_symbol_list,
        stmt_object_list,
        for_while_flag) b2);
    (stmt_symbol_list,
    stmt_object_list)

| For(e1, e2, e3, st) -> ignore (expr (stmt_symbol_list
    , stmt_object_list) e1);
    check_bool_expr (
        stmt_symbol_list,
        stmt_object_list) e2;
    ignore (expr (stmt_symbol_list
        , stmt_object_list) e3);
    stmt (stmt_symbol_list,
        stmt_object_list, true) st

| While(p, s) -> check_bool_expr (stmt_symbol_list,
    stmt_object_list) p;
    stmt (stmt_symbol_list,

```

```

                                stmt_object_list, true) s
| Bind(b) -> let typstring = fst b and idstring = (
              className ^ "@" ^ (snd b)) in
              (*print_string (string_of_typ typstring
                             );
              print_string idstring;*)
              let new_symbol_list = StringMap.add
                idstring typstring stmt_symbol_list
              in
              (*ignore(StringMap.iter
                        print_symbolList new_symbol_list);*)
              (new_symbol_list, stmt_object_list)

| Init(t, s, e) -> let rt = expr (stmt_symbol_list, stmt_object_list) e
                  in
                  check_assign t rt (
                    Failure ("illegal
                              init " ^
                              string_of_typ t ^
                              " = " ^ string_of_typ rt
                              ^ " in " ^
                              string_of_expr e));
                  ignore(expr (stmt_symbol_list,
                               stmt_object_list) e);

                  let new_symbol_list = StringMap.add
                    (className ^ "@" ^ s) t
                    stmt_symbol_list
                  in (*ignore(StringMap.iter
                              print_symbolList
                              new_symbol_list);*)
                    (new_symbol_list, stmt_object_list)

| ArrayBind((t, s, e)) -> let ty = expr (
                             stmt_symbol_list, stmt_object_list) e
                           in let _ =
                             (match ty with Int -> ()
                              | _ -> raise (Failure("
                                                    array subscript is
                                                    not integer in " ^ s
                                                    )))
                           in
                           let new_symbol_list = StringMap.
                             add (className ^ "@" ^ s) t
                             stmt_symbol_list
                           in (new_symbol_list,
                               stmt_object_list)

| Break -> if for_while_flag then (stmt_symbol_list,
                                   stmt_object_list)
           else raise (Failure ("break is out of for
                                 or while loop"))

```

```

        | Continue -> if for_while_flag then (stmt_symbol_list,
            stmt_object_list)
            else raise (Failure ("continue is out of
                for or while loop"))

        | Classdecl(class_name, objectname) -> (*print_string (
            class_decl class_name); print_string objectname;*)
            let new_object_list = StringMap.add objectname (
                class_decl class_name) stmt_object_list in (*
                StringMap.iter print_object new_object_list; *) (
                stmt_symbol_list, new_object_list)
in let (stmt_result_symbol_list, _) = stmt (formal_symbol_list,
    StringMap.empty, false) (Block func.body) in () in List.iter
    check_function (List.rev functions); symbol_table; in
    check_class (symbol_list, cdecl.extends, cdecl.cbody.vandadecls,
        cdecl.cbody.methods);
in
List.fold_left check_cdecl symbols cdecls

```

initial.gbl

```

# Author: Shengtong Zhang (sz2539)
### gloabl variables including array, initialization
class Gobang extends Game
{
    int win(int MapSize[], int GridNum, int PlayerNumber, int PlayerId
        [], string playerName[], int SpriteId[], int SpriteOwnerId[],
        int FormerId, int FormerPosition[], int NextPlayerID, int
        NextSpriteID, bool WithAI, int InputPosition[])
    {
        return -1;
    }

    #check if the move is legal
    bool isLegal(int MapSize[], int GridNum, int PlayerNumber, int
        PlayerId[], string playerName[], int SpriteId[], int
        SpriteOwnerId[], int FormerId, int FormerPosition[], int
        NextPlayerID, int NextSpriteID, bool WithAI, int InputPosition
        [])
    {
        return true;
    }

    int update(int MapSize[], int GridNum, int PlayerNumber, int
        PlayerId[], string playerName[], int SpriteId[], int
        SpriteOwnerId[], int FormerId, int FormerPosition[], int
        NextPlayerID, int NextSpriteID, bool WithAI, int InputPosition
        [])
    {
        int a[2];
        a[0] = -1;
        a[1] = -1;
        return a;
    }
}

```

```

    }
}

class GobangAI extends AI{

    int returnposition()
    {
        int r[2];
        r[0] = 0;
        r[1] = 0;
        return r;
    }
}

class UserMain extends Main{
    void main()
    {
        Gobang mygame;
        #initialize
        int MapS[2];
        MapS[0] = 15;
        MapS[1] = 15;
        int GridNumber = MapS[0] * MapS[1];
        int InputPlayerNumber = 2;
        int InputPlayerId[InputPlayerNumber];
        InputPlayerId[0] = 0;
        InputPlayerId[1] = 1;
        string InputPlayerName[InputPlayerNumber];
        InputPlayerName[0] = "Alice";
        InputPlayerName[1] = "Bob";
        bool InputWithAI = false;

        initialize(MapS, InputPlayerNumber, InputPlayerId,
            InputPlayerName, InputWithAI)@mygame;

        #GobangAI gobangai;
    }
}

```

GoBang.gbl

```

# Author: Shengtong Zhang(sz2539)
### gloabl variables including array, initialization
class Gobang extends Game
{
    int win(int MapSize[], int GridNum, int PlayerNumber, int PlayerId
        [], string PlayerName[], int SpriteId[], int SpriteOwnerId[],
        int FormerId, int FormerPosition[], int NextPlayerID, int
        NextSpriteID, bool WithAI, int InputPosition[])
    {
        #string PlayerSprite = PlayerSpriteName[FormerId];
        int PlayerSprite = FormerId;
        int position[2];
    }
}

```

```

position[0] = FormerPosition[0];
position[1] = FormerPosition[1];
int up;
int down;
int left;
int right;
if((position[0] - 4) > 0) up = position[0] - 4;
else up = 0;
if((position[0] + 4) > (MapSize[0] - 1)) down = MapSize[0] - 1;
else down = position[0] + 4;
if((position[1] - 4) > 0) left = position[1] - 4;
else left = 0;
if((position[1] + 4) > (MapSize[1] - 1)) right = MapSize[1] - 1;
else right = position[1] + 4;
int i;
int j;
int count;
int ret;
#check row
int PreSprite = -1;
count = 0;

for(j = left; j <= right; j = j + 1)
{
    int curSprite = SpriteOwnerId[position[0] * MapSize[0] + j];
    #string curSprite = Text@SpriteList[position[0]][j];
    if(curSprite == PlayerSprite)
    {
        if(PreSprite != PlayerSprite)
        {
            count = 1;
        }
        else count = count + 1;
        if(count == 5)
        {
            ret = NextPlayerID - 1;
            if(ret == -1)
                ret = PlayerNumber - 1;
            return ret;
        }
    }
    else count = 0;

    PreSprite = curSprite;
}

#check column
PreSprite = -1;
count = 0;
for(i = up; i <= down; i = i + 1)
{
    int curSprite = SpriteOwnerId[i * MapSize[0] + position[1]];
    if(curSprite == PlayerSprite)
    {

```

```

    if(PreSprite != PlayerSprite)
    {
        count = 1;
    }
    else count = count + 1;
    if(count == 5)
    {
        ret = NextPlayerID - 1;
        if(ret == -1)
            ret = PlayerNumber - 1;
        return ret;
    }
}
else count = 0;

PreSprite = curSprite;
}

#check diagonal left-up to right-down
PreSprite = -1;
count = 0;
int leftup[2];
int leftupDistance;
leftupDistance = 4;
if(position[0] < 4) leftupDistance = position[0];
if(position[1] < 4) leftupDistance = position[1];
leftup[0] = position[0] - leftupDistance;
leftup[1] = position[1] - leftupDistance;

int rightdown[2];
int rightdownDistance;
rightdownDistance = 4;
if((MapSize[0] - 1 - position[0]) < 4) rightdownDistance =
    MapSize[0] - 1 - position[0];
if((MapSize[1] - 1 - position[1]) < 4) rightdownDistance =
    MapSize[1] - 1 - position[1];
rightdown[0] = position[0] + rightdownDistance;
rightdown[1] = position[1] + rightdownDistance;

int curposition[2];

for(i = 0; i <= (rightdown[0] - leftup[0]); i = i + 1)
{
    curposition[0] = leftup[0] + i;
    curposition[1] = leftup[1] + i;
    int curSprite = SpriteOwnerId[curposition[0] * MapSize[0] +
        curposition[1]];
    if(curSprite == PlayerSprite)
    {
        if(PreSprite != PlayerSprite)
        {
            count = 1;
        }
        else count = count + 1;
    }
}

```

```

    if(count == 5)
    {
        ret = NextPlayerID - 1;
        if(ret == -1)
            ret = PlayerNumber - 1;
        return ret;
    }
}
else count = 0;

PreSprite = curSprite;
}

#check diagonal left-down to right-up
PreSprite = -1;
count = 0;
int leftdown[2];
int leftdownDistance = 4;
if(position[0] < 4) leftdownDistance = position[0];
if((MapSize[1] - 1 - position[1]) < 4) leftdownDistance = MapSize
    [1] - 1 - position[1];
leftdown[0] = position[0] - leftdownDistance;
leftdown[1] = position[1] + leftdownDistance;

int rightup[2];
int rightupDistance = 4;
if((MapSize[0] - 1 - position[0]) < 4) rightupDistance = MapSize
    [0] - 1 - position[0];
if((position[1]) < 4) rightupDistance = position[1];
rightup[0] = position[0] + rightupDistance;
rightup[1] = position[1] - rightupDistance;

int curposition[2];

for(i = 0; i <= (rightup[0] - leftdown[0]); i = i + 1)
{
    curposition[0] = leftdown[0] + i;
    curposition[1] = leftdown[1] - i;
    int curSprite = SpriteOwnerId[curposition[0] * MapSize[0] +
        curposition[1]];
    if(curSprite == PlayerSprite)
    {
        if(PreSprite != PlayerSprite)
        {
            count = 1;
        }
        else count = count + 1;
        if(count == 5)
        {
            ret = NextPlayerID - 1;
            if(ret == -1)
                ret = PlayerNumber - 1;
            return ret;
        }
    }
}

```



```

    }
    else count = 0;

    PreSprite = curSprite;
}

return -1;
}

#check if the move is legal
bool isLegal(int MapSize[], int GridNum, int PlayerNumber, int
    PlayerId[], string PlayerName[], int SpriteId[], int
    SpriteOwnerId[], int FormerId, int FormerPosition[], int
    NextPlayerID, int NextSpriteID, bool WithAI, int InputPosition
    [])
{
    if(not (((InputPosition[0]>=0) and (InputPosition[0]<MapSize[0]))
        and ((InputPosition[1]>=0) and (InputPosition[1]<MapSize[1]))
        ))
        return false;
    if(SpriteOwnerId[InputPosition[0] * MapSize[0] + InputPosition
        [1]] == -1)
        return true;
    else return false;
}

int update(int MapSize[], int GridNum, int PlayerNumber, int
    PlayerId[], string PlayerName[], int SpriteId[], int
    SpriteOwnerId[], int FormerId, int FormerPosition[], int
    NextPlayerID, int NextSpriteID, bool WithAI, int InputPosition
    [])
{
    SpriteId[InputPosition[0] * MapSize[0] + InputPosition[1]] =
        NextSpriteID;
    SpriteOwnerId[InputPosition[0] * MapSize[0] + InputPosition[1]]
        = NextPlayerID;

    FormerId = NextPlayerID;
    FormerPosition[0] = InputPosition[0];
    FormerPosition[1] = InputPosition[1];
    NextSpriteID = NextSpriteID + 1;
    NextPlayerID = NextPlayerID + 1;
    if(NextPlayerID == PlayerNumber)
        NextPlayerID = 0;
    int a[2];
    a[0] = -1;
    a[1] = -1;
    return a;
}

#void printGame();
#for(int i = 0; i < MapSize[0]; i = i + 1)
#{
# for(int j = 0; j < MapSize[1]; j = j + 1)

```

```

    # {
    #     print(Text@SpriteList[i][j]@Gobang);
    # }
    # print("\n");
    #}
}

class GobangAI extends AI{

    int returnposition()
    {
    /*
        int position[2];
        int i;
        int j;
        for(i = 0; i < MapSize[0]@mygame; i = i + 1)
        {
            for(j = 0; j < MapSize[1]@mygame; j = j + 1)
            {
                if(SpriteOwnerId[i * MapSize[0] + j] == -1)
                {
                    position[0] = i;
                    position[1] = j;
                }
            }
        }
        int r;
        r = position[0]* MapSize[0]@mygame + position[1];
        return r;
    */
        int r[2];
        r[0] = 0;
        r[1] = 0;
        return r;
    }
}

class UserMain extends Main{
    void main()
    {
        Gobang mygame;
        int MapS[2];
        MapS[0] = 15;
        MapS[1] = 15;
        int GridNumber = MapS[0] * MapS[1];
        int InputPlayerNumber = 2;
        int InputPlayerId[InputPlayerNumber];
        InputPlayerId[0] = 0;
        InputPlayerId[1] = 1;
        string InputPlayerName[InputPlayerNumber];
        InputPlayerName[0] = "Alice";
        InputPlayerName[1] = "Bob";
        bool InputWithAI = false;
    }
}

```

```

        initialize(MapS, InputPlayerNumber, InputPlayerId,
            InputPlayerName, InputWithAI)@mygame;
        #printGame()@mygame;

        #GobangAI gobangai;
    }
}

```

Flew.gbl

```

# Author: Shengtong Zhang(sz2539)
### gloabl variables including array, initialization
class Gobang extends Game
{
    #variables:
    #Map:
    /*
    int MapSize[2];
    int GridNum; #GridNum = MapSize[0] * MapSize[1]
    #Player:
    int PlayerNumber;
    int PlayerId[10];
    string PlayerName[10];
    #Sprite:
    int SpriteId[1000];
    int SpriteOwnerId[1000];
    #LastMove:
    int FormerId; #the ID of former player
    int FormerPosition[2];
    #Next Turn:
    int NextPlayerID;
    int NextSpriteID;
    #AI?
    bool WithAI;
    int InputPosition[2];

    */
    int win(int MapSize[], int GridNum, int PlayerNumber, int PlayerId
        [], string PlayerName[], int SpriteId[], int SpriteOwnerId[],
        int FormerId, int FormerPosition[], int NextPlayerID, int
        NextSpriteID, bool WithAI, int InputPosition[])
    {
        return -1;
    }

    #check if the move is legal
    bool isLegal(int MapSize[], int GridNum, int PlayerNumber, int
        PlayerId[], string PlayerName[], int SpriteId[], int
        SpriteOwnerId[], int FormerId, int FormerPosition[], int
        NextPlayerID, int NextSpriteID, bool WithAI, int InputPosition
        [])
    {
        if(not (((InputPosition[0]>=0) and (InputPosition[0]<MapSize[0]))

```

```

        and ((InputPosition[1]>=0) and (InputPosition[1]<MapSize[1]))
    ))
    return false;
if(SpriteOwnerId[InputPosition[0] * MapSize[0] + InputPosition
[1]] == -1)
    return true;
else return false;
}

int update(int MapSize[], int GridNum, int PlayerNumber, int
PlayerId[], string PlayerName[], int SpriteId[], int
SpriteOwnerId[], int FormerId, int FormerPosition[], int
NextPlayerID, int NextSpriteID, bool WithAI, int InputPosition
[])
{
    int CurSpriteID;
    int CurPlayerID = NextPlayerID;
    SpriteId[InputPosition[0] * MapSize[0] + InputPosition[1]] =
        NextSpriteID;
    SpriteOwnerId[InputPosition[0] * MapSize[0] + InputPosition[1]]
        = NextPlayerID;
    FormerId = NextPlayerID;
    FormerPosition[0] = InputPosition[0];
    FormerPosition[1] = InputPosition[1];
    NextSpriteID = NextSpriteID + 1;
    NextPlayerID = NextPlayerID + 1;
    if(NextPlayerID == PlayerNumber)
        NextPlayerID = 0;

    int a[8];
    int t;
    int curr = 0;
    for(t = 0; t < 8; t = t + 1)a[t] = -1;
    if((((InputPosition[0]>=2) and (SpriteOwnerId[(InputPosition[0]
- 2) * MapSize[0] + InputPosition[1]] == CurPlayerID)) and (
        SpriteOwnerId[(InputPosition[0] - 1) * MapSize[0] +
        InputPosition[1]] == (NextPlayerID)))
    {
        SpriteOwnerId[(InputPosition[0] - 1) * MapSize[0] +
            InputPosition[1]] = -1;
        a[curr] = InputPosition[0] - 1;
        a[curr + 1] = InputPosition[1];
        curr = curr + 2;
    }

    if((((InputPosition[0]<=(MapSize[0]-3) and (SpriteOwnerId[(
        InputPosition[0] + 2) * MapSize[0] + InputPosition[1]] ==
        CurPlayerID)) and (SpriteOwnerId[(InputPosition[0] + 1) *
        MapSize[0] + InputPosition[1]] == (NextPlayerID)))
    {
        SpriteOwnerId[(InputPosition[0] + 1) * MapSize[0] +
            InputPosition[1]] = -1;
        a[curr] = InputPosition[0] + 1;
    }
}

```

```

        a[curr + 1] = InputPosition[1];
        curr = curr + 2;
    }

    if(((InputPosition[1]>=2) and (SpriteOwnerId[InputPosition[0] *
        MapSize[0] + (InputPosition[1] - 2)] == CurPlayerID)) and (
        SpriteOwnerId[InputPosition[0] * MapSize[0] + (InputPosition
            [1] - 1)] == (NextPlayerID)))
    {
        SpriteOwnerId[InputPosition[0] * MapSize[0] + (InputPosition
            [1] - 1)] = -1;
        a[curr] = InputPosition[0];
        a[curr + 1] = InputPosition[1] - 1;
        curr = curr + 2;
    }

    if(((InputPosition[1]<=(MapSize[1]-3)) and (SpriteOwnerId[
        InputPosition[0] * MapSize[0] + (InputPosition[1] + 2)] ==
        CurPlayerID)) and (SpriteOwnerId[InputPosition[0] * MapSize
            [0] + (InputPosition[1] + 1)] == (NextPlayerID)))
    {
        SpriteOwnerId[InputPosition[0] * MapSize[0] + (InputPosition
            [1] + 1)] = -1;
        a[curr] = InputPosition[0];
        a[curr + 1] = InputPosition[1] + 1;
        curr = curr + 2;
    }

    return a;
}

void printGame();
for(int i = 0; i < MapSize[0]; i = i + 1)
#{
    # for(int j = 0; j < MapSize[1]; j = j + 1)
    # {
    #     print(Text@SpriteList[i][j]@Gobang);
    # }
    # print("\n");
#}
}

class GobangAI extends AI{

    int returnposition()
    {
        /*
        int position[2];
        int i;
        int j;
        for(i = 0; i < MapSize[0]@mygame; i = i + 1)
        {

```

```

        for(j = 0; j < MapSize[1]@mygame; j = j + 1)
        {
            if(SpriteOwnerId[i * MapSize[0] + j] == -1)
            {
                position[0] = i;
                position[1] = j;
            }
        }
    }
    int r;
    r = position[0]* MapSize[0]@mygame + position[1];
    return r;
}
*/
int r[2];
r[0] = 0;
r[1] = 0;
return r;
}
}

#assume the input stream is stdin>>...
class UserMain extends Main{
    void main()
    {
        Gobang mygame;
        int MapS[2];
        MapS[0] = 15;
        MapS[1] = 15;
        int GridNumber = MapS[0] * MapS[1];
        int InputPlayerNumber = 2;
        int InputPlayerId[InputPlayerNumber];
        InputPlayerId[0] = 0;
        InputPlayerId[1] = 1;
        string InputPlayerName[InputPlayerNumber];
        InputPlayerName[0] = "Alice";
        InputPlayerName[1] = "Bob";
        bool InputWithAI = false;
        GobangAI gobangai;

        initialize(MapS, InputPlayerNumber, InputPlayerId,
            InputPlayerName, InputWithAI)@mygame;
    }
}

```

game_init.py

```

# Automatic generated
def initialize(self, MapS, InputPlayerNum, InputPlayerID,
    InputPlayerName, InputWithAI):
    self.MapSize[0]=MapS[0]
    self.MapSize[1]=MapS[1]
    self.GridNum = (self.MapSize[0]) * (self.MapSize[1])

```

```

self.PlayerNumber = InputPlayerNum
self.WithAI = InputWithAI
i = None
i = 0
while ((i) < (self.PlayerNumber)):
    self.PlayerId[i]=InputPlayerID[i]
    self.PlayerName[i]=InputPlayerName[i]
    i = (i) + (1)
i = None
j = None
i = 0
while ((i) < (self.MapSize[0])):
    j = 0
    while ((j) < (self.MapSize[1])):
        self.SpriteId[((i) * (self.MapSize[0])) + (j)]= - (1)
        self.SpriteOwnerId[((i) * (self.MapSize[0])) + (j)]= -
            (1)
        j = (j) + (1)
    i = (i) + (1)
self.FormerId = - (1)
self.FormerPosition[0]= - (1)
self.FormerPosition[1]= - (1)
self.NextPlayerID = 0
self.NextSpriteID = 0
pass

```

game_gui.py

```

# Author: Sihao Zhang (sz2558)
import Tkinter as tk

def _create_circle(self, x, y, r, **kwargs):
    return self.create_oval(x - r, y - r, x + r, y + r, **kwargs)

class GameBoard(tk.Frame):
    def __init__(self, parent, mygame, gobangai):
        '''size is the size of a square, in pixels'''
        self.mygame = mygame
        self.gobangai = gobangai
        self.rows = mygame.MapSize[1]
        self.columns = mygame.MapSize[0]
        self.size = 32
        self.colors = ["black", "white", "blue", "yellow", "grey", "pink"]
        self.color = "bisque" # board color
        self.turns = 0 # current player
        self.pastPieces = [] #sprite list
        self.numPlayers = mygame.PlayerNumber
        self.enable = True # win or tie makes it false
        self.AI = mygame.WithAI
        canvas_width = (self.columns+5) * self.size
        canvas_height = self.rows * self.size

        tk.Frame.__init__(self, parent)
        self.canvas = tk.Canvas(self, borderwidth=0, highlightthickness
            =0,

```

```

        width=canvas_width, height=
            canvas_height, background="bisque")
self.canvas.pack(side="top", fill="both", expand=True, padx=2,
    pady=2)

# this binding will cause a refresh if the user interactively
# changes the window size
self.canvas.bind("<Configure>", self.refresh)
self.canvas.bind("<Button-1>", self.callback)
self.addavatar()

#When mouse clicks
def callback(self, event):
    if self.mygame.NextPlayerID != 1 or not self.mygame.WithAI:
        if event.x < self.size*self.rows and event.y < self.size*
            self.columns and self.enable: # inside the board and
                hasn't finished the game yet
                    #if (event.y/self.size, event.x/self.size) not in self.
                        pastPieces:
                            #isLegal()
                                self.mygame.InputPosition[0] = event.y/self.size
                                self.mygame.InputPosition[1] = event.x/self.size
                                if(self.mygame.isLegal()):
                                    self.addpiece(str((event.y/self.size, event.x/self.
                                        size)), event.y/self.size, event.x/self.size)
                                    self.pastPieces.append((event.y/self.size, event.x/
                                        self.size))

                                dele = self.mygame.update()
                                l = len(dele) / 2
                                for i in range(l):
                                    delex = dele[2 * i]
                                    deley = dele[2 * i + 1]
                                    if delex != -1 or deley != -1:
                                        Spritename = "Sprite" + str(delex*self.
                                            mygame.MapSize[0] + deley)
                                        self.canvas.delete(Spritename)

                                winner = self.mygame.win();
                                if(winner != -1):
                                    self.win(winner)
                                #if isTie():
                                    #tie()
                                #turn to next player
                                self.turns += 1
                                if self.turns >= self.numPlayers:
                                    self.turns = 0

                                self.addavatar()
                                print "clicked at", self.mygame.InputPosition[0], self.
                                    mygame.InputPosition[1]
                    else:
                        position = self.gobangai.returnposition(self.mygame.MapSize

```



```

        , self.mygame.SpriteOwnerId)
self.mygame.InputPosition[0] = position[0]
self.mygame.InputPosition[1] = position[1]

self.addpiece(str((position[0], position[1])), position[0],
              position[1])
self.pastPieces.append((position[0], position[1]))

dele = self.mygame.update()
l = len(dele) / 2
for i in range(l):
    delex = dele[2 * i]
    deley = dele[2 * i + 1]
    if delex != -1 or deley != -1:
        Spritename = "Sprite" + str(delex*self.mygame.
                                     MapSize[0] + deley)
        self.canvas.delete(Spritename)

winner = self.mygame.win();
if(winner != -1):
    self.win(winner)
    #if isTie():
    #tie()
    #turn to next player
self.turns += 1
if self.turns >= self.numPlayers:
    self.turns = 0

self.addavatar()

#update()
def addpiece(self, name, row=0, column=0, curr=-1):
    '''Add a piece to the playing board'''
    x0 = (column * self.size) + int(self.size/2)
    y0 = (row * self.size) + int(self.size/2)
    Sname = "Sprite" + str(row * self.mygame.MapSize[0] + column)
    if curr==-1:
        self.canvas.create_circle(x0, y0, self.size/3, fill=self.
                                 colors[self.turns], tag = Sname)
    else:
        self.canvas.create_circle(x0, y0, self.size/3, fill=self.
                                 colors[curr], tag = Sname)
    self.canvas.coords(name, x0, y0)

#draw avatars
def addavatar(self):
    '''Add a piece to the playing board'''
    self.canvas.delete("avatar")
    for i in range(self.numPlayers):
        name = "Player "+str(i)
        column = self.columns + 1.5
        row = self.rows * i / self.numPlayers + 1
        x0 = column * self.size
        y0 = row * self.size

```

```

        if i==self.turns:
            self.canvas.create_circle(x0, y0, self.size*3/4, fill=
                self.colors[i], tags="avatar")
        else:
            self.canvas.create_circle(x0, y0, self.size*2/3, fill=
                self.colors[i], tags="avatar")
        self.canvas.coords(name, x0, y0)
        self.canvas.create_text((self.columns+3.5)*self.size, row*
            self.size, text=name, tags="avatar")

# redraw whole window
def redraw(self):
    self.canvas.delete("all")
    for row in range(self.rows):
        for col in range(self.columns):
            x1 = (col * self.size)
            y1 = (row * self.size)
            x2 = x1 + self.size
            y2 = y1 + self.size
            self.canvas.create_rectangle(x1, y1, x2, y2, outline="
                black", fill=self.color, tags="square")

    i = 0
    for name in self.pastPieces:
        self.addpiece(str(name), name[0], name[1], curr=i)
        i += 1
        if i >= self.numPlayers:
            i = 0

    self.addavatar()
    self.canvas.tag_raise("piece")
    self.canvas.tag_lower("square")

def refresh(self, event):
    '''Redraw the board, possibly in response to window being
        resized'''
    xsize = int((event.width-1) / self.columns)
    ysize = int((event.height-1) / self.rows)
    self.size = min(xsize, ysize)
    self.redraw()

def clear(self):
    self.enable = True
    self.turns = 0
    self.pastPieces = set([])
    self.redraw()

def win(self, i):
    self.enable = False
    winStr = "Player "+str(i) + " WIN"
    self.canvas.create_text(self.columns*self.size/2, self.rows*
        self.size/2, text = winStr, font=("Helvetica",32), fill = "
            red")

def tie(self):

```

```
self.enable = False
self.canvas.create_text(self.columns*self.size/2, self.rows*
    self.size/2, text = "TIE", font=("Helvetica",50), fill = "
red")
```

game_vars.py

```
# Auto generated
self.WithAI = None
self.NextSpriteID = None
self.NextPlayerID = None
self.FormerPosition = [ None ] * 2
self.FormerId = None
self.SpriteOwnerId = [ None ] * 1000
self.SpriteId = [ None ] * 1000
self.PlayerName = [ None ] * 10
self.PlayerId = [ None ] * 10
self.PlayerNumber = None
self.GridNum = None
self.MapSize = [ None ] * 2
self.InputPosition = [ None ] * 2
```