PROTEGE
CHEMICALS

Date : / /
Mo Tu We Th Fr Sa Su

PART 1. Define your own mini-language

Chosen example : Simple game scripting

- SYNTAX (BNF / EBNF Grammar)

<program> ::= <entity>* <statement>*

<entity> ::= "player" IDENTIFIER "{" <assign>* "}"
| "enemy" IDENTIFIER "{" <assign>* "}"

<assign> ::= IDENTIFIER "=" <expr> ";"

<statement> ::= <move_stmt> | <set_stmt> | <if_stmt> | <print_stmt>

<move_stmt> ::= "move" IDENTIFIER <direction> <expr> ";"

<direction> ::= "up" | "down" | "left" | "right"

<set_stmt> ::= "set" IDENTIFIER "." IDENTIFIER "=" <expr> ";"

<if_stmt> ::= "if" <condition> "{" <statement>* "}"

<print_stmt> ::= "print" STRING ";"

<condition> ::= <expr> <relop> <expr>

<relop> ::= "==" | "!=" | ">" | "<"

<expr> ::= <term> (("+" | "-") <term>)*

<term> ::= <factor> (("*" | "/") <factor>)*

<factor> ::= INTEGER | IDENTIFIER "." IDENTIFIER | "(" <expr> ")"

PROTEGE
CHEMICALS

Date : / /
Mo Tu We Th Fr Sa Su

- LEXICAL RULES (tokens, identifiers, keywords)

Keywords

player	set	end	left
enemy	if	up	right
move	print	down	

Operators

=	*	!=
+	/	>
-	==	<

Delimiters

{	(	;
}	)	.
		,

Token Definitions

INTEGER	::=	[0-9]+
STRING	::=	" [any character except "]* "
IDENTIFIER	::=	[a-z][a-z0-9]*
COMMENT	::=	// [any character]* newline
WHITESPACE	::=	[\t\r\n]+

PROTEGE
CHEMICALS

Date : / /
Mo Tu We Th Fr Sa Su

- SEMANTIC RULES AND TYPE SYSTEM

Type System

Type : int (only type in TinyGame)

All variables are integers
All entity properties are integers
All expressions evaluate to integers
Boolean conditions evaluate to 0 (false) or 1 (true)

Scope Rules

Global Scope:

Entity declarations (player, enemy)

Entity Scope :

Properties within entity { } block
Properties belonging to specific entity
Access via: entity_name . property_name

Semantic Constraints

Entity Declarations

Must declare entity before use
No duplicate entity names
Entity names must be identifiers

Property Declarations

Properties must be declared in entity block
No duplicate properties within same entity
Property names must be identifiers

PROTEGE
CHEMICALS

Date : / /
Mo Tu We Th Fr Sa Su

Semantic Constraints continued...

Property Access
- Format: entity_name.property_name
- Entity must exist
- Property must belong to that entity

Type Checking
- All expressions must evaluate to int
- Arithmetic and comparison operators require int operands
- Move amount must be int
- Assignment values must be int

Statement Validity
- move: Entity must exist
- set: Entity and property must exist
- if: Condition must be valid comparison
- print: String literal only

Semantic Error Examples

`move ghost right 5;`	ghost not declared
`set hero.power = 10;`	power property not declared
`if x > 5 { ... }`	standalone identifier not allowed
`set hero.x = "hello";`	string not allowed, must be int
`player hero { hero = 5; }`	cannot redeclare entity name

Date : / /
Mo Tu We Th Fr Sa Su

- EXAMPLE INPUT AND EXPECTED OUTPUT

Example 1 : Basic Movement

Input:
```
player hero {
    x = 0;
    y = 0;
}
move hero right 5;
move hero up 3;
print "Hero moved!";
```

Expected Output: hero moved right by 5 to position (5,0)
hero moved up by 3 to position (5,3)
OUTPUT: Hero moved!

Final State: player hero: x=5, y=3

Example 2 : Collision Detection

Input:
```
player hero {
    x = 0;
    y = 0;
    health = 100;
}
enemy monster {
    x = 5;
    y = 5;
move hero right 5;
move hero up 5;
if hero.x == monster.x {
    if hero.y == monster.y {
        set hero.health = hero.health - 10;
        print "Hit by monster!";
    }
}
```

PROTEGE
CHEMICALS

Date : / /
Mo Tu We Th Fr Sa Su

Example 2 continued...

Expected Output: hero moved right by 5 to position (5,0)
hero moved up by 5 to position (5,5)
OUTPUT: Hit by monster!

Final State: player hero: x = 5, y = 5, health = 90
enemy monster: x = 5, y = 5

Example 3: Score System

Input:
```
player hero {
    x = 0;
    y = 0;
    score = 0;
}
move hero right 10;
if hero.x > 5 {
    set hero.score = hero.score + 100;
    print "Bonus points earned!"
}
```

Expected Output: hero moved right by 10 to position (10,0)
OUTPUT: Bonus points earned!

Final State: player hero: x = 10, y = 0, score = 100

PROTEGE
CHEMICALS

Date : / /
Mo Tu We Th Fr Sa Su

Summary Reference Table

Aspect	TinyGame Specification
File Extension	.tg
Total Keywords	11
Total Operators	9
Data Types	int only
Scope Levels	2 (global entities + entity properties)
Control Structures	if (no else, no loops)
Entity Types	player, enemy
Directions	up, down, left, right
Comments	// (single line)
Case Sensitive	Yes (lowercase only for identifiers)