Sunny Shaban Ali 7J K224149 Design Docs

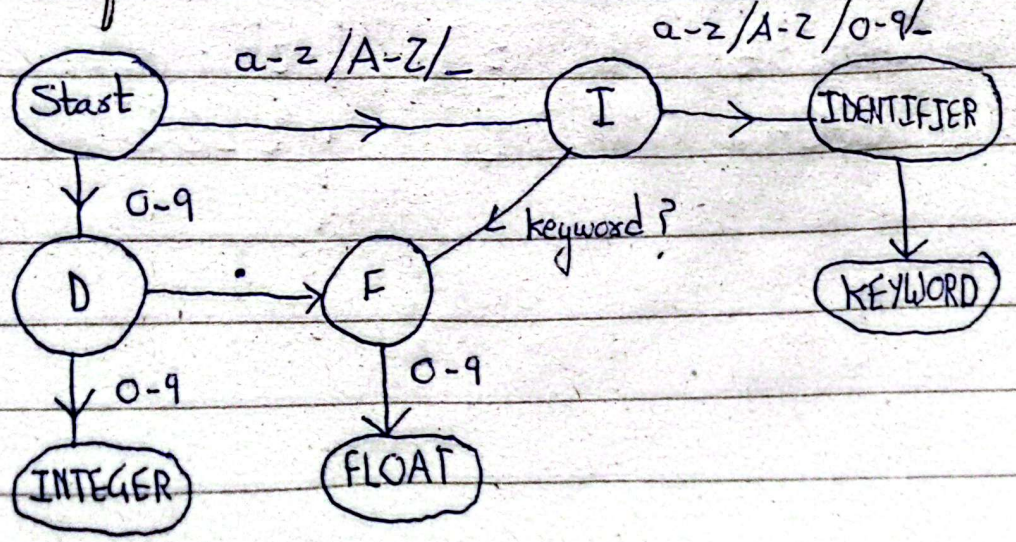
Mini Tower Defense Language

1. lexical Analysis

- Regex for token patterns

Token	Regex
KEYWORDS	map | enemy | tower | wave | spawn | place | at | size | path | count | start | interval
IDENTIFIER	[a-zA-Z_] [a-zA-Z0-9_]+
INTEGER	[0-9]+
FLOAT	[0-9]+ \. [0-9]+
OPERATORS	\, \; \(\) \[\] \{ \}
WHITESPACE	[\t\r\n]+
COMMENT	// [^/n]*

- DFA for MTDL Lexer



- Token Streams

```
1   map BasicMap {
        size = (10, 10);
    }
```

```
[KW_MAP, "map", 1]
[IDENT, "BasicMap", 1]
[LBRACE, "{", 1]
[IDENT, "size", 2]
[EQUAL, "=", 2]
[LPAREN, "(", 2]
[INT, "10", 2]
[COMMA, ",", 2]
[INT, "10", 2]
[RPAREN, ")", 2]
[SEMICOLON, ";", 2]
[RBRACE, "}", 3]
[EOF, "", 3]
```

```
2   enemy Goblin {
        hp = 50;
        speed = 1.5;
    }
```

```
[KW_ENEMY, "enemy", 1]
[IDENT, "Goblin", 1]
[LBRACE, "{", 1]
[IDENT, "hp", 2]
[EQUAL, "=", 2]
[INT, "50", 2]
[SEMICOLON, ";", 2]
[IDENT, "speed", 3]
[EQUAL, "=", 3]
[FLOAT, "1.5", 3]
[SEMICOLON, ";", 3]
[RBRACE, "}", 4]
[EOF, "", 4]
```

CamScanner

3 tower Cannon {
range = 5;
}

[KW_TOWER, "tower", 1]
[IDENT, "Cannon", 1]
[LBRACE, "{", 1]
[IDENT, "range", 2]
[EQUAL, "=", 2]
[INT, "5", 2]
[SEMICOLON, ";", 2]
[RBRACE, "}", 3]
[EOF, "", 3]

4 map x { }

[KW_MAP, "map", 1]
[IDENT, "x", 1]
[LBRACE, "{", 1]
[RBRACE, "}", 1]
[EOF, ""]

5 enemy E { }

[KW_ENEMY, "enemy", 1]
[IDENT, "E", 1]
[LBRACE, "{", 1]
[RBRACE, "}", 1]
[EOF, "", 1]

CamScanner

2. Syntax Analysis

CFG for MTDL

Program -> Declarations
Declarations -> Declaration ; Declarations | ε
Declaration -> Map Decl | Enemy Decl | Tower Decl | Wave Decl |
Place Stmt
Map Decl -> KW_MAP IDENT LBRACE Size Decl
Path Decl RBRACE
Size Decl -> IDENT EQUAL LPAREN INT COMMA INT
RPAREN SEMICOLON
Path Decl -> IDENT EQUAL LBRACKET Coord List RBRACKE
SEMICOLON
Coord List -> Coordinate | Coordinate COMMA Coord List
Coordinate -> LPAREN INT COMMA INT RPAREN

Enemy Decl -> KW_ENEMY IDENT LBRACE Attribute List RBRACE
Tower Decl -> KW_TOWER IDENT LBRACE Attribute List RBRACE
Attribute List -> Attribute | Attribute Attribute List
Attribute -> IDENT EQUAL Value SEMICOLON
Value -> INT | FLOAT

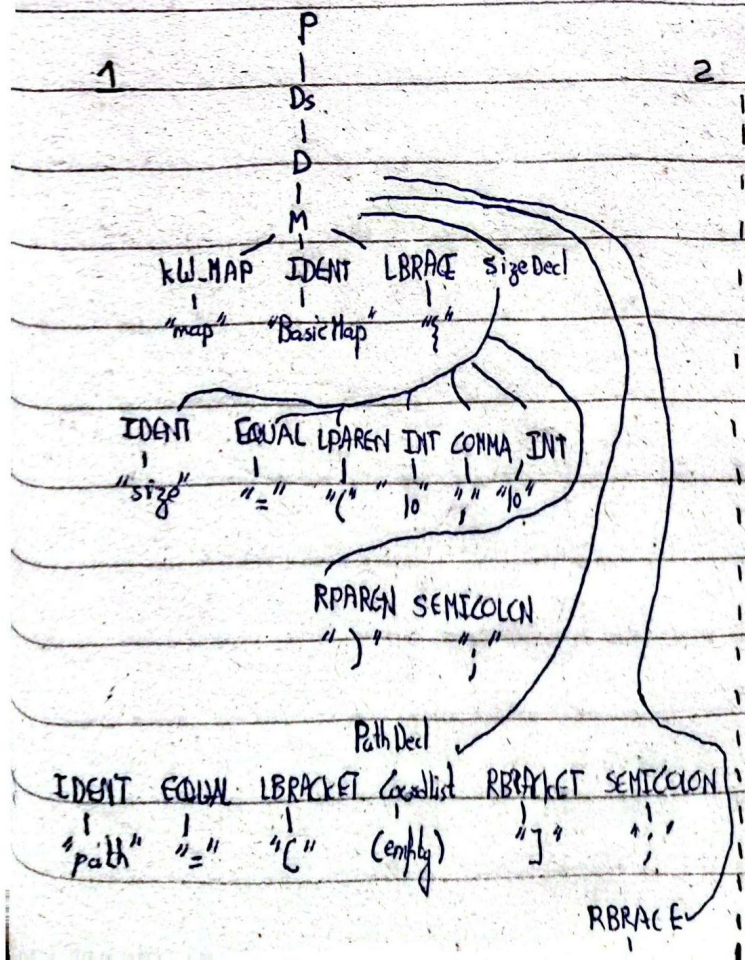
CamScanner

WaveDecl → KW_WAVE IDENT LBRACE SpawnList RBRACE
SpawnList → SpawnStmt | SpawnStmt SpawnList
SpawnStmt → KW_SPAWN LPAREN IDENT COMMA Count
COMMA Start COMMA Interval RPAREN SEMICOLON
Count → IDENT EQUAL INT
Start → IDENT EQUAL INT
Interval → IDENT EQUAL INT

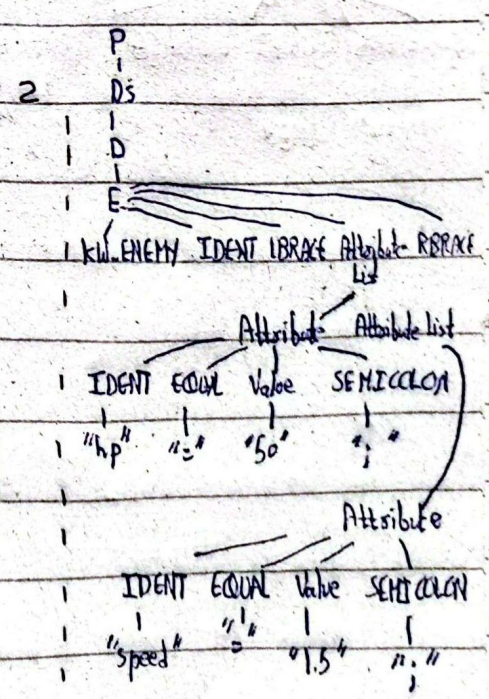
PlaceStmt → KW_PLACE IDENT KW_AT LPAREN
INT COMMA INT RPAREN SEMICOLON

- Parse Tree for Test cases.

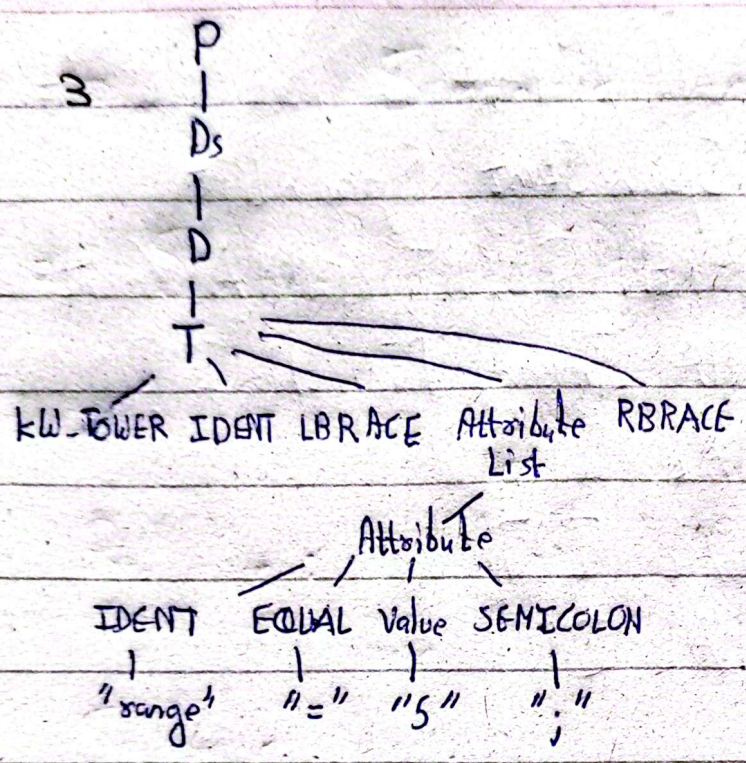
1



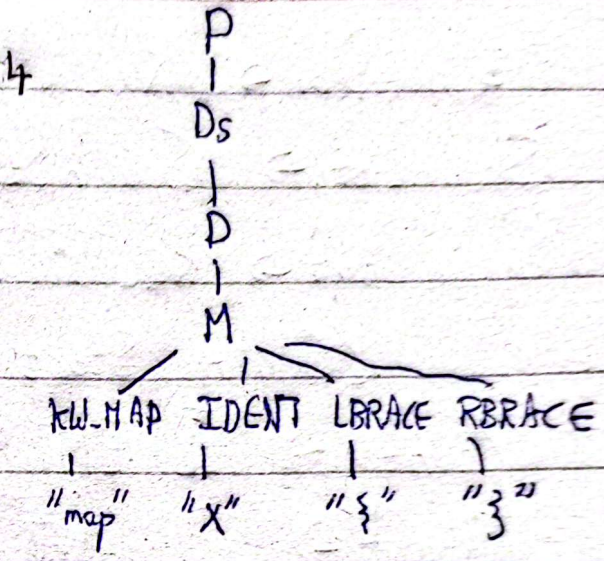
2



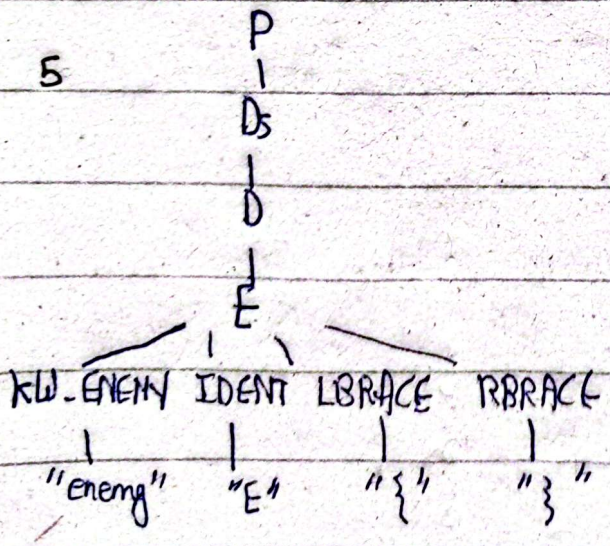
3


P
Ds
D
T
KW_TOWER
IDENT
LBRACE
Attribute List
RBRACE
Attribute
IDENT
EQUAL
Value
SEMICOLON
"range"
"="
"5"
";"

4


P
Ds
D
M
KW_MAP
IDENT
LBRACE
RBRACE
"map"
"X"
"{"
"}"

5


P
Ds
D
E
KW_ENEMY
IDENT
LBRACE
RBRACE
"enemy"
"E"
"{"
"}"

Parsing Table LL (1)

Non-Terminal	Alias	KW_MAP	KW_ENEMY	KW_TOWER	KW_WAVE	KW_PLACE	IDENT	LBRACE	RBRACE	EOF
Program	P	D	D	D	D	D	–	–	–	$
Declaration	D	M	E	T	W	P	–	–	–	–
Map Decl	M	1	–	–	–	–	–	–	–	–
Enemy Decl	E	–	2	–	–	–	–	–	–	–
Tower Decl	T	–	–	3	–	–	–	–	–	–
Wave Decl	W	–	–	–	4	–		–	–	–
Place Stmt	Ps	–	–	–	–	5	–	–	–	–

LL (1) Parsing Table

3. Semantic Analysis

Syntax-Directed Definitions:

L-Attributed Rules.

Production : MapDecl → KW_MAP IDENT LBRACE SizeDecl PathDecl RBRACE.

Semantic Rules:

```
MapDecl.name = IDENT.lexeme.
MapDecl.width = SizeDecl.width (inh)
MapDecl.height = SizeDecl.height (inh)
MapDecl.path = PathDecl.coordinates (inh).
// Check Duplicates
if (symbolTable.lookup(MapDecl.name) != null))
        error("(Duplicate): Map Definition" + MapDecl.name)
symbolTable.insert(MapDecl.name, MapDecl)
currentMap = MapDecl.
```

Production : KW_ENEMY IDENT LBRACE AttributeList RBRACE

Semantic Rules:

```
EnemyDecl.name = IDENT.lexeme
EnemyDecl.hp = AttributeList.hp
EnemyDecl.speed = AttributeList.speed
EnemyDecl.reward = AttributeList.reward.
```

// Validate attributes

if (Enemy Decl.hp <= 0) error ("Hp must be positive")

if (Enemy Decl.speed <= 0) error ("Speed must be positive")

if (Enemy Decl.reward < 0) error ("Reward cannot be negative")

// Check duplicates

if (enemy Table.lookup (Enemy Decl.name) != null)

error ("(Duplicate): Enemy " + Enemy Decl.name)

enemy Table.insert (Enemy Decl.name, Enemy Decl).

S-Attributed Rules

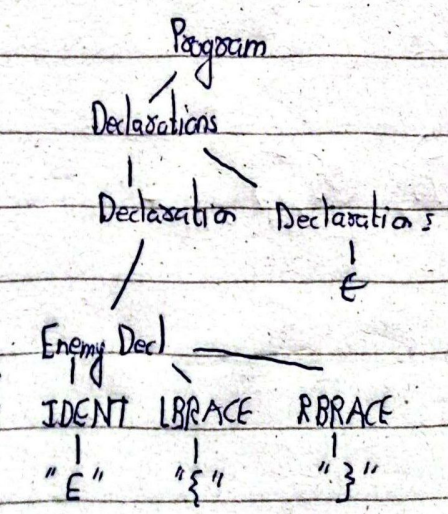
Production: Attribute → IDENT EQUAL Value SEMICOLON

Semantic Rules:

Attribute.name = IDENT.value

Attribute.value = Value.value

Annotated Parse Tree with attributes



name = "E" → Enemy Decl.name (rule 1)

▫ Global Scope Symbol Table.

Maps:

"GameMap1" → MapDecl (width = 20, height = 15)

Enemies:

"Goblin" → EnemyDecl (hp = 50, speed = 1.5)

"Orc" → EnemyDecl (hp = 150, speed = 1.0)

Towers:

"Arrows" → TowerDecl (range = 3, damage = 15)

"Cannon" → TowerDecl (range = 5, damage = 50).

Waves:

"Wave1" → WaveDecl (spawns = [...]).

▫ Current Content Tracking.

Current Map: "GameMap1"

- width: 20
- height: 15
- path: [(0,7), (5,7), ...]

CamScanner

4. Intermediate Code Generation:

tower Arrow { range = 3; damage = 15; fire_rate = 2.0; }

- 3AC

```
t1 = new_tower ("Arrow")
set_range (t1, 3)
set_damage (t1, 15)
set_fire_rate (t1, 2.0)
set_dps (t1, 30.0) // Optimized: 15 * 2.0 = 30.0
```

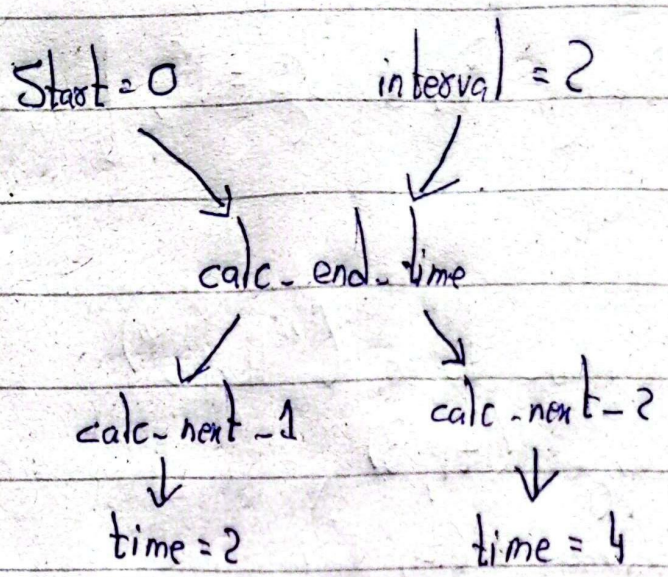
- Quadruples

```
(1) (new_tower, "Arrow", __, t1)
(2) (set_attr, t1, "range", 3)
(3) (set_attr, t1, "damage", 15)
(4) (set_attr, t1, "fire_rate", 2.0)
(5) (set_attr, t1, "dps", 30.0) // folded
        Constant.
```

- RPN

```
"Arrow" new_tower
3 "range" set_attr
15 "damage" set_attr
2.0 "fire_rate" set_attr
30.0 "dps" set_attr // Pre-calculated
```

- DAG for multiple Spawn.



5. Optimization Phase.

- Constant folding.

Before

```
tower Arrow {
    damage = 15;
    fire_rate = 2.0;
    // Game calculates at
    run time : dps = 15 * 2.0
}
```

After.

```
tower Arrow {
    damage = 15;
    fire_rate = 2.0;
    dps = 30.0;
}
```

▫ Dead Code Elimination.

IR befor DCE	IR After DCE
DEFINE_ENEMY, "Unused Enemy", hp=100	1. DEFINE_ENEMY, "Used Enemy", hp=50
DEFINE_ENEMY, "Used Enemy", hp=50	2. DEFINE_WAVE, "WAVE1", _, _
DEFINE_WAVE, "WAVE 1", _, _	3. SPAWN_ENEMY, "WAVE1", "Used Enemy"
SPAWN_ENEMY, "WAVE1", "Used Enemy"	// Unused enemy removed.

▫ Redundant Spawn Merging.

Before	After.
wave Wave 1 {	wave Wave 1 {
spawn (Goblin, count=5, start=0, interval=2)	spawn (Goblin, count=8, start=0, interval=2)
spawn (Goblin, count=3, start=0, interval=2)	}
}	

▫ Duplicate Definition Removal.

Before	After.
tower Arrow {range=3, damage=15}	tower Arrow {range=3, damage=15}.
tower Arrow {range=3, damage=15}	

6. Final Target code Generation:

1
```
{
  "game Config" : {
    "map" : {
      "name" : "BasicMap",
      "width" : 20
      "height" : 15
      "path" : [ ]
    }
  }
}
```

2
```
{
  "game Config" : {
    "enemies" : [
      {
        "name" : "Goblin"
        "hp" : 50
        "speed" : 1.50
        "reward" : 0
      }
    ]
  }
}
```

3
```
{
  "game Config" : {
    "towers" : [
      {
        "name" : "Cannon"
        "range" : 5 ,
        "damage" : 6 ,
        "fire Rate" : 0.06 ,
        "cost" : 0
      }
```

4
```
{
  "game Config" : {
    "enemies" : [
      {
        "name" : "E" ,
        "hp" : 0 ,
        "speed" : 0.00 ,
        "reward" : 0
      }
    ]
```

```
]                                   }
}                                   }
}
}
  "gameConfig" : {
      "map" : {
          "name" : "X"
          "width" : 0,
          "height" : 0,
          "path" : []
      }
  }
}
```