

Module Interface Specification

Group 13

The Box Group

Gurpartap Kaler, kalerg1

Sagar Thomas, thomas12

Freddie Yan, yanz20

November 9, 2018

Contents

1 Module Hierarchy

M1: Hardware-Hiding Module

M2: Behaviour-Hiding Module: SceneManager Module

M3: Behaviour-Hiding Module: Scene Module

M4: Behaviour-Hiding Module: Character Module

M5: Behaviour-Hiding Module: Box Module

M6: Behaviour-Hiding Module: Game Module

Level 1	Level 2
Hardware-Hiding Module	
Behaviour-Hiding Module	SceneManager Module Scene Module Character Module Box Module Game Module
Software Decision Module	N/A (No generic modules)

Table 1: Module Hierarchy

2 MIS of Character Module

2.1 Interface Syntax

2.1.1 Exported Types

KeyT = {UP,DOWN,LEFT,RIGHT}

2.1.2 Exported Access Programs

Name	In	Out	Exceptions
init	integer,integer,integer	-	invalid input
move	KeyT	-	-
getX	KeyT, integer	integer	-
getY	KeyT, integer	integer	-

2.2 Interface Semantics

2.2.1 State Variables

x:integer//x-coordinate of the character
y:integer//y-coordinate of the character
speed:integer//length of a cell in maze

2.2.2 Environmental Variables

None

2.2.3 Assumptions

The constructor Character is called for each object instance before any other access routine is called for that object. The constructor cannot be called on an existing object.

2.2.4 Access Program Semantics

init(i,j,s):
transition:x := i*s, y := j*s, speed := s.
exception:raise invalid_argument if any of the inputs is not integer.
move(k):
transition:if k = UP: y:= y-speed, if k = DOWN: y:= y+speed,
if k = LEFT: x:= x-speed, if k = RIGHT: x:= x+speed.
exception:None.
getX(k,i):
Output:if k = UP: out:= x, if k = DOWN: out:= x,
if k = LEFT: out:=x-i*speed, if k = RIGHT: out:= x+i*speed.
exception:None.
getY(k,i):
Output:if k = UP: out:= y-i*speed, if k = DOWN: out:= y+i*speed,
if k = LEFT: out:=y, if k = RIGHT: out:= y.
exception:None.

3 MIS of Box Module

3.1 Interface Syntax

3.1.1 Exported Types

KeyT = {UP,DOWN,LEFT,RIGHT}

3.1.2 Exported Access Programs

Name	In	Out	Exceptions
init	integer,integer,integer	-	invalid_input
move	KeyT	-	-
getX		integer	-
getY		integer	-

3.2 Interface Semantics

3.2.1 State Variables

x:integer//x-coordinate of the Box
y:integer//y-coordinate of the Box
speed:integer//length of a cell in maze

3.2.2 Environmental Variables

None

3.2.3 Assumptions

The constructor Box is called for each object instance before any other access routine is called for that object. The constructor cannot be called on an existing object.

3.2.4 Access Program Semantics

init(i,j,s):
transition:x := i*s, y := j*s, speed := s.
exception:raise invalid_argument if any of the inputs is not integer.
move(k):
transition:if k = UP: y:= y-speed, if k = DOWN: y:= y+speed,
if k = LEFT: x:= x-speed, if k = RIGHT: x:= x+speed.
exception:None.
getX():
Output:out:= x.
exception:None.
getY():
Output:out:= y.
exception:None.

4 MIS of Scene Module

4.1 Interface Syntax

4.1.1 Exported Types

None

4.1.2 Exported Access Programs

Name	In	Out	Exceptions
init	Character,Sequence of Box, Sequence of integer	-	

4.2 Interface Semantics

4.2.1 State Variables

Char: Character// Boxs:Sequence of Box// maze:Sequence of integer

4.2.2 Environmental Variables

None

4.2.3 Assumptions

The constructor Scene is called for each object instance before any other access routine is called for that object. The constructor cannot be called on an existing object.

4.2.4 Access Program Semantics

init(c,b,s):

transition: Char:= c, Boxs:= b, maze:= s.

exception:None

5 MIS of SceneManager Module

5.1 Interface Syntax

5.1.1 Exported Access Programs

Name	In	Out	Exceptions
init	-	-	-
setup_scenes	Scene,Scene	-	-
update	Pygame.KeyEvent	-	-
switch_scene	-	-	-
get_event	-	-	-

5.2 Interface Semantics

5.2.1 State Variables

scenes: Sequence of scenes // Contains all scenes in the game
done: Boolean // Contains the current running status of the game
scene_name: String // Name of the current running scene
scene: Scene //The current running Scene
now: Integer // Current run-time of the game

5.2.2 Environmental Variables

Pygame.KeyEvent

5.2.3 Assumptions

The state variables will be assigned values before any scene management functions, such as switch_scene are called

5.2.4 Access Program Semantics

SceneManager():

Input: none

Transition: Load up all the game scenes

Output: Assign initial values to state variables

Exceptions: None

 setup_scenes(scenes, start_scene):

Input: scenes, start_scene

Transition: Set the current scene to the start_scene and store the rest of the scenes

Output: None

Exceptions: None

 update(keys, now):

Input: Pygame.KeyEvent, now

Transition: Checks for key events that quit the game. If not, proceed to update the current scene.

Output: None

Exceptions: None

 switch_scene():

Input: None

Transition: Changes the current scene to the next planned scene.

Output: New Scene

Exceptions: None

 get_event(event):

Input: None

Transition: Receives any event passed from the main game loop. Passes it to the current scene.

Output: None

Exceptions: None

6 MIS of Game Module

6.1 Interface Syntax

6.1.1 Exported Access Programs

Name	In	Out	Exceptions
init	-	-	-
update	-	-	-
event_loop	-	-	-
run	-	-	-

6.2 Interface Semantics

6.2.1 State Variables

now: Integer // The current run-time of the game

done: Boolean //Status of whether the game is complete or not

clock: Pygame.Clock //Clock object keeping track of fps

keys: Pygame.KeyEvent // Top level key events to be passed down to scenes

scene_manager: SceneManager // Instance of the SceneManager module to manage all the games of the scene

screen: Pygame.Surface // Used for rendering game elements to the screen

6.2.2 Environmental Variables

Pygame.Clock

Pygame.KeyEvent

Pygame.Surface

6.2.3 Assumptions

None.

6.2.4 Access Program Semantics

Game():

Input: None Transition: Initializes all state variables to their default

Output: None

Exceptions: None

update():

Input: None Transition: Updates the current run-time of the game

Output: None

Exceptions: None

event_loop():

Input: None Transition: Passes the Pygame events into scene_manager

Output: None

Exceptions: None

run():

Input: None Transition: Main game loop. Calls event_loop and update repeatedly.

Output: None

Exceptions: None

7 Major Revision History

November 9, 2018 - Revision 0