SE 3XA3: MIS
DinoDodger

Team 39, S.R.A Squad
Shrey Mittal, mittas1
Razan Abujarad, abujarar
Allen Yang, yangz18

9 November, 2017

Contents

1 UI Module **v**
- 1.1 Module . . . v
- 1.2 Uses . . . v
- 1.3 Syntax . . . v
  - 1.3.1 Exported Access Programs . . . v
- 1.4 Semantics . . . v
  - 1.4.1 State Variables . . . v
  - 1.4.2 State Invariant . . . vi
  - 1.4.3 Access Routine Semantics . . . vi

2 Sprite Animation Module **viii**
- 2.1 Template Module . . . viii
- 2.2 Uses . . . viii
- 2.3 Syntax . . . viii
  - 2.3.1 Exported Constants . . . viii
  - 2.3.2 Exported Types . . . viii
  - 2.3.3 Exported Access Programs . . . viii
- 2.4 Semantics . . . viii
  - 2.4.1 State Variables . . . viii
  - 2.4.2 State Invariant . . . ix
  - 2.4.3 Assumptions . . . ix
  - 2.4.4 Access Routine Semantics . . . ix

3 Character Module **xi**
- 3.1 Template Module . . . xi
- 3.2 Uses . . . xi
- 3.3 Syntax . . . xi
  - 3.3.1 Exported Constants . . . xi
  - 3.3.2 Exported Types . . . xi
  - 3.3.3 Exported Access Programs . . . xi
- 3.4 Semantics . . . xi
  - 3.4.1 State Variables . . . xi
  - 3.4.2 State Invariant . . . xi
  - 3.4.3 Assumptions . . . xi
  - 3.4.4 Access Routine Semantics . . . xii

4 Animation Module **xiii**
- 4.1 Template Module . . . xiii
- 4.2 Uses . . . xiii
- 4.3 Syntax . . . xiii
  - 4.3.1 Exported Types . . . xiii

4.3.2 Exported Access Programs . . . xiii
4.4 Semantics . . . xiii
4.4.1 State Variables . . . xiii
4.4.2 State Invariant . . . xiii
4.4.3 Assumptions . . . xiii
4.4.4 Access Routine Semantics . . . xiv

5 DinoDodger Module **xv**
5.1 Template Module . . . xv
5.2 Uses . . . xv
5.3 Syntax . . . xv
5.3.1 Exported Types . . . xv
5.3.2 Exported Access Programs . . . xv
5.4 Semantics . . . xv
5.4.1 State Variables . . . xv
5.4.2 State Invariant . . . xv
5.4.3 Assumptions . . . xv
5.4.4 Access Routine Semantics . . . xvi

6 UI Module **xvii**
6.1 Module . . . xvii
6.2 Uses . . . xvii
6.3 Syntax . . . xvii
6.3.1 Exported Access Programs . . . xvii
6.4 Semantics . . . xvii
6.4.1 State Variables . . . xvii
6.4.2 State Invariant . . . xviii
6.4.3 Access Routine Semantics . . . xviii

7 Sprite Animation Module **xx**
7.1 Template Module . . . xx
7.2 Uses . . . xx
7.3 Syntax . . . xx
7.3.1 Exported Constants . . . xx
7.3.2 Exported Types . . . xx
7.3.3 Exported Access Programs . . . xx
7.4 Semantics . . . xx
7.4.1 State Variables . . . xx
7.4.2 State Invariant . . . xxi
7.4.3 Assumptions . . . xxi
7.4.4 Access Routine Semantics . . . xxi

8 Character Module **xxiii**
8.1 Template Module . . . xxiii
8.2 Uses . . . xxiii
8.3 Syntax . . . xxiii
8.3.1 Exported Constants . . . xxiii
8.3.2 Exported Types . . . xxiii
8.3.3 Exported Access Programs . . . xxiii
8.4 Semantics . . . xxiii
8.4.1 State Variables . . . xxiii
8.4.2 State Invariant . . . xxiii
8.4.3 Assumptions . . . xxiii
8.4.4 Access Routine Semantics . . . xxiv

9 PlayScene Module **xxv**
9.1 Template Module . . . xxv
9.2 Uses . . . xxv
9.3 Syntax . . . xxv
9.3.1 Exported Types . . . xxv
9.3.2 Exported Access Programs . . . xxv
9.4 Semantics . . . xxv
9.4.1 State Variables . . . xxv
9.4.2 State Invariant . . . xxv
9.4.3 Assumptions . . . xxv
9.4.4 Access Routine Semantics . . . xxvi

10 DinoDodger Module **xxvii**
10.1 Template Module . . . xxvii
10.2 Uses . . . xxvii
10.3 Syntax . . . xxvii
10.3.1 Exported Types . . . xxvii
10.3.2 Exported Access Programs . . . xxvii
10.4 Semantics . . . xxvii
10.4.1 State Variables . . . xxvii
10.4.2 State Invariant . . . xxvii
10.4.3 Assumptions . . . xxvii
10.4.4 Access Routine Semantics . . . xxviii

11 Cactus Module **xxix**
11.1 Template Module . . . xxix
11.2 Uses . . . xxix
11.3 Syntax . . . xxix
11.3.1 Exported Constants . . . xxix
11.3.2 Exported Types . . . xxix

11.3.3 Exported Access Programs . . . xxix
11.4 Semantics . . . xxix
11.4.1 State Variables . . . xxix
11.4.2 State Invariant . . . xxix
11.4.3 Assumptions . . . xxix
11.4.4 Access Routine Semantics . . . xxx

12 Pteradactyl Module **xxxi**
12.1 Template Module . . . xxxi
12.2 Uses . . . xxxi
12.3 Syntax . . . xxxi
12.3.1 Exported Constants . . . xxxi
12.3.2 Exported Types . . . xxxi
12.3.3 Exported Access Programs . . . xxxi
12.4 Semantics . . . xxxi
12.4.1 State Variables . . . xxxi
12.4.2 State Invariant . . . xxxi
12.4.3 Assumptions . . . xxxii
12.4.4 Access Routine Semantics . . . xxxii

List of Tables

1 **Revision History** . . . iv

List of Figures

Table 1: **Revision History**

Date	Version	Notes
2017/11/9	1.0	Creating MIS
2017/12/04	2.0	MIS Revision

1 UI Module

1.1 Module

User Interface

1.2 Uses

N/A

1.3 Syntax

1.3.1 Exported Access Programs

Routine name	In	Out	Exceptions
start	Stage	Scene	none
playButtonSelected	Event	none	none
char1Selected	Event	none	none
char2Selected	Event	none	none
char3Selected	Event	none	none
landscape1Selected	Event	none	none
landscape2Selected	Event	none	none
landscape3Selected	Event	none	none
main	none	none	none

1.4 Semantics

1.4.1 State Variables

button1 := Button
button2 := Button
button3 := Button
button4 := Button
button5 := Button
button6 := Button
button7 := Button
char1 := String
char2 := String
char3 := String
char := String
landscape := String
landscape1 := String
landscape2 := String
landscape3 := String

scene1 := Scene
scene2 := Scene
scene3 := Scene

1.4.2 State Invariant

none

1.4.3 Access Routine Semantics

start(Stage):

- transition: Creation of stage(window) with scene
- exception: none

char1Selected:

- transition: char = char1
- exception: none

char1Selected:

- transition: char = char2
- exception: none

char3Selected:

- transition: char = char3
- exception: none

landscape1Selected:

- transition: landscape = landscape1
- exception: none

landscape2Selected:

- transition: landscape = landscape2
- exception: none

landscape3Selected:

- transition: landscape = landscape3

- exception: none

okayButtonSelected:

- transition: Goes from scene1 to scene2
- exception: none

2 Sprite Animation Module

2.1 Template Module

Sprite Animation

2.2 Uses

N/A

2.3 Syntax

2.3.1 Exported Constants

imageView: ImageView
COUNT: int
COLUMNS: int
OFFSET_X: int
OFFSET_Y: int
WIDTH: int
HEIGHT: int

2.3.2 Exported Types

SpriteAnimation

2.3.3 Exported Access Programs

Routine name	In	Out	Exceptions
SpriteAnimation	ImageView, Duration, int, int, int, int, int, int	SpriteAnimation	none
setOffSetX	int	none	none
setOffSetY	int	none	none
interpolate	double	none	none

2.4 Semantics

2.4.1 State Variables

imageView: ImageView
duration: Duration
count: int
columns: int
offSetX: int
offSetY: int

width: inti
height: int

2.4.2 State Invariant

none

2.4.3 Assumptions

none

2.4.4 Access Routine Semantics

SpriteAnimation(imageView, duration, count, columns, offSetX, offSetY, width, height):

- transition: imageView, setCycleDuration(duration), COUNT, COLUMNS, OFFSET_X, OFFSET_Y, WIDTH, HEIGHT := imageView, duration, count, columns, offSetX, offSetY, width, height
- output: *out* := *self*
- exception: none

setOffSetX(x):

- transition: OFFSET_X := x
- output: none
- exception: none

setOffSetY(y):

- transition: OFFSET_Y := y
- output: none
- exception: none

interpolate($frac$):

- transition: imageView is set to new viewport using a Rectangle2D object wifth values x, y, width, height, where x and y are as follows:

  index := min(floor(COUNT*frac, COUNT-1))
  x := (index mod(COLUMNS))*WIDTH+OFFSET_X
  y := (index/COLUMNS)*HEIGHT+OFFSET_Y

- output: none
- exception: none

3 Character Module

3.1 Template Module

PointT

3.2 Uses

SpriteAnimation

3.3 Syntax

3.3.1 Exported Constants

imageView: ImageView
COUNT: int
COLUMNS: int
OFFSET_X: int
OFFSET_Y: int
WIDTH: int
HEIGHT: int

3.3.2 Exported Types

Character

3.3.3 Exported Access Programs

Routine name	In	Out	Exceptions
Character	ImageView	Character	none
jump	none	Character Animated to Jump	none

3.4 Semantics

3.4.1 State Variables

animation := SpriteAnimation

3.4.2 State Invariant

none

3.4.3 Assumptions

none

3.4.4 Access Routine Semantics

Character($imageView$):

- transition: $imageView := imageView$
- output: Outputs Animation onto an imageView
- exception: none

jump:

- transition: none
- output: Character jumps
- exception: none

4 Animation Module

4.1 Template Module

Animation

4.2 Uses

Character, DinoDodger

4.3 Syntax

4.3.1 Exported Types

Animation

4.3.2 Exported Access Programs

Routine name	In	Out	Exceptions
start	Stage	Window	none
main	none	all arguments launched	none

4.4 Semantics

4.4.1 State Variables

goAnimation := SpriteAnimation
goUp := SpriteAnimation
$imageView1$:= ImageView
$imageView2$:= ImageView
$imageView3$:= ImageView
animation := SpriteAnimation

4.4.2 State Invariant

While character has not intersected obstacle, continue.

4.4.3 Assumptions

none

4.4.4 Access Routine Semantics

start($imageView$):

- transition: $imageView := imageView$
- output: Outputs Character Animation, Obstacles and Background onto Scene
- exception: none

main:

- transition: All arguments launched
- output: Gameplay mode
- exception: none

5 DinoDodger Module

5.1 Template Module

DinoDodger

5.2 Uses

User Interface Module

5.3 Syntax

5.3.1 Exported Types

none; this module is medium of communication

5.3.2 Exported Access Programs

Routine name	In	Out	Exceptions
getCharacterSelected	String	Image	none
getLandScapeSelected	String	Image	none
getScore	none	int	none
getHighScore	none	int	none

5.4 Semantics

5.4.1 State Variables

$POINTS$:= int
$HIGHSCORE$:= int
$landScape\_1$:= Image
$landScape\_2$:= Image
$landScape\_3$:= Image
$character\_1$:= Image
$character\_2$:= Image
$character\_3$:= Image

5.4.2 State Invariant

none

5.4.3 Assumptions

none

5.4.4 Access Routine Semantics

getCharacterSelected(character)

- transition: returns Image based on String equivalence
- output: character_1, character_2, character_3 based on equivalence of character
- exception: none

getLandScapeSelected(landScape)

- transition: returns Image based on String equivalence
- output: landScape_1, landScape_2, landScape_3 based on equivalence of character
- exception: none

getScore:

- transition: HIGHSCORE = POINTS
- output: POINTS
- exception: POINTS ¡ HIGHSCORE in which case transition does not occur.

getHighScore:

- transition: none
- output: HIGHSCORE
- exception: none

6 UI Module

6.1 Module

User Interface

6.2 Uses

N/A

6.3 Syntax

6.3.1 Exported Access Programs

Routine name	In	Out	Exceptions
start	Stage	Scene	none
playButtonSelected	Event	none	none
char1Selected	Event	none	none
char2Selected	Event	none	none
char3Selected	Event	none	none
landscape1Selected	Event	none	none
landscape2Selected	Event	none	none
landscape3Selected	Event	none	none
main	none	none	none

6.4 Semantics

6.4.1 State Variables

button1 := Button
button2 := Button
button3 := Button
button4 := Button
button5 := Button
button6 := Button
button7 := Button
char1 := String
char2 := String
char3 := String
char := String
landscape := String
landscape1 := String
landscape2 := String
landscape3 := String

scene1 := Scene
scene2 := Scene
scene3 := Scene

6.4.2 State Invariant

none

6.4.3 Access Routine Semantics

start(Stage):

- transition: Creation of stage(window) with scene
- exception: none

char1Selected:

- transition: char = char1
- exception: none

char1Selected:

- transition: char = char2
- exception: none

char3Selected:

- transition: char = char3
- exception: none

landscape1Selected:

- transition: landscape = landscape1
- exception: none

landscape2Selected:

- transition: landscape = landscape2
- exception: none

landscape3Selected:

- transition: landscape = landscape3

- exception: none

okayButtonSelected:

- transition: Goes from scene1 to scene2
- exception: none

7 Sprite Animation Module

7.1 Template Module

Sprite Animation

7.2 Uses

N/A

7.3 Syntax

7.3.1 Exported Constants

imageView: ImageView
COUNT: int
COLUMNS: int
OFFSET_X: int
OFFSET_Y: int
WIDTH: int
HEIGHT: int

7.3.2 Exported Types

SpriteAnimation

7.3.3 Exported Access Programs

Routine name	In	Out	Exceptions
SpriteAnimation	ImageView, Duration, int, int, int, int, int, int	SpriteAnimation	none
setOffSetX	int	none	none
setOffSetY	int	none	none
interpolate	double	none	none

7.4 Semantics

7.4.1 State Variables

imageView: ImageView
duration: Duration
count: int
columns: int
offSetX: int
offSetY: int

width: inti
height: int

7.4.2 State Invariant

none

7.4.3 Assumptions

none

7.4.4 Access Routine Semantics

SpriteAnimation(imageView, duration, count, columns, offSetX, offSetY, width, height):

- transition: imageView, setCycleDuration(duration), COUNT, COLUMNS, OFFSET_X, OFFSET_Y, WIDTH, HEIGHT := imageView, duration, count, columns, offSetX, offSetY, width, height
- output: *out* := *self*
- exception: none

setOffSetX(x):

- transition: OFFSET_X := x
- output: none
- exception: none

setOffSetY(y):

- transition: OFFSET_Y := y
- output: none
- exception: none

interpolate($frac$):

- transition: imageView is set to new viewport using a Rectangle2D object wifth values x, y, width, height, where x and y are as follows:

  index := min(floor(COUNT*frac, COUNT-1))
  x := (index mod(COLUMNS))*WIDTH+OFFSET_X
  y := (index/COLUMNS)*HEIGHT+OFFSET_Y

- output: none
- exception: none

8 Character Module

8.1 Template Module

PointT

8.2 Uses

SpriteAnimation

8.3 Syntax

8.3.1 Exported Constants

imageView: ImageView
COUNT: int
COLUMNS: int
OFFSET_X: int
OFFSET_Y: int
WIDTH: int
HEIGHT: int

8.3.2 Exported Types

Character

8.3.3 Exported Access Programs

Routine name	In	Out	Exceptions
Character	ImageView	Character	none
jump	none	Character Animated to Jump	none

8.4 Semantics

8.4.1 State Variables

animation := SpriteAnimation

8.4.2 State Invariant

none

8.4.3 Assumptions

none

8.4.4 Access Routine Semantics

Character($imageView$):

- transition: $imageView := imageView$
- output: Outputs Animation onto an imageView
- exception: none

jump:

- transition: none
- output: Character jumps
- exception: none

9 PlayScene Module

9.1 Template Module

PlayScene

9.2 Uses

Character, DinoDodger

9.3 Syntax

9.3.1 Exported Types

PlayScene

9.3.2 Exported Access Programs

Routine name	In	Out	Exceptions
start	Stage	Window	none
main	none	all arguments launched	none

9.4 Semantics

9.4.1 State Variables

goAnimation := SpriteAnimation
goUp := SpriteAnimation
*imageView*1 := ImageView
*imageView*2 := ImageView
*imageView*3 := ImageView
animation := SpriteAnimation

9.4.2 State Invariant

While character has not intersected obstacle, continue.

9.4.3 Assumptions

none

9.4.4 Access Routine Semantics

start($imageView$):

- transition: $imageView := imageView$
- output: Outputs Character Animation, Obstacles and Background onto Scene
- exception: none

main:

- transition: All arguments launched
- output: Gameplay mode
- exception: none

10 DinoDodger Module

10.1 Template Module

DinoDodger

10.2 Uses

User Interface Module

10.3 Syntax

10.3.1 Exported Types

none; this module is medium of communication

10.3.2 Exported Access Programs

Routine name	In	Out	Exceptions
getCharacterSelected	String	Image	none
getLandScapeSelected	String	Image	none
getScore	none	int	none
getHighScore	none	int	none

10.4 Semantics

10.4.1 State Variables

$POINTS$:= int
$HIGHSCORE$:= int
$landScape\_1$:= Image
$landScape\_2$:= Image
$landScape\_3$:= Image
$character\_1$:= Image
$character\_2$:= Image
$character\_3$:= Image

10.4.2 State Invariant

none

10.4.3 Assumptions

none

10.4.4 Access Routine Semantics

getCharacterSelected(character)

- transition: returns Image based on String equivalence
- output: character_1, character_2, character_3 based on equivalence of character
- exception: none

getLandScapeSelected(landScape)

- transition: returns Image based on String equivalence
- output: landScape_1, landScape_2, landScape_3 based on equivalence of character
- exception: none

getScore:

- transition: HIGHSCORE = POINTS
- output: POINTS
- exception: POINTS ¡ HIGHSCORE in which case transition does not occur.

getHighScore:

- transition: none
- output: HIGHSCORE
- exception: none

11 Cactus Module

11.1 Template Module

N/A

11.2 Uses

UI

11.3 Syntax

11.3.1 Exported Constants

cactus1 : Image
cactus2 : Image
cactus3 : Image
cactus4 : Image

11.3.2 Exported Types

Cactus

11.3.3 Exported Access Programs

Routine name	In	Out	Exceptions
Cactus	none	none	none
getRandomCactus	none	Cactus ImageView	none

11.4 Semantics

11.4.1 State Variables

N/A

11.4.2 State Invariant

none

11.4.3 Assumptions

none

11.4.4 Access Routine Semantics

Cactus

- transition: $image := image$
- output: Outputs cactus onto an image
- exception: none

getRandomCactus:

- transition: none
- output: Cactus object
- exception: none

12 Pteradactyl Module

12.1 Template Module

N/A

12.2 Uses

SpriteAnimation

12.3 Syntax

12.3.1 Exported Constants

imageView: ImageView
COUNT: int
COLUMNS: int
OFFSET_X: int
OFFSET_Y: int
WIDTH: int
HEIGHT: int

12.3.2 Exported Types

Pteradactyl

12.3.3 Exported Access Programs

Routine name	In	Out	Exceptions
Pteradactyl	ImageView	Pteradactyl	none
getRandomHeight	none	Random int	none

12.4 Semantics

12.4.1 State Variables

animation := SpriteAnimation

12.4.2 State Invariant

none

12.4.3 Assumptions

none

12.4.4 Access Routine Semantics

Pteradactyl($imageView$):

- transition: $imageView := imageView$
- output: Outputs Pteradactyl Animation onto an imageView
- exception: none

getRandomHeight:

- transition: none
- output: Random integer
- exception: none

showPoints:

- transition: none
- output: points
- exception: none

showHighScore:

- transition: none
- output: highscore
- exception: none

updateHighScore:

- transition: $highscore := points$
- output: none
- exception: none

reset:

- transition: $points, highscore := 0, 0$
- output: none
- exception: none

run:

- transition: $points := points + 1$
- output: none
- exception: none