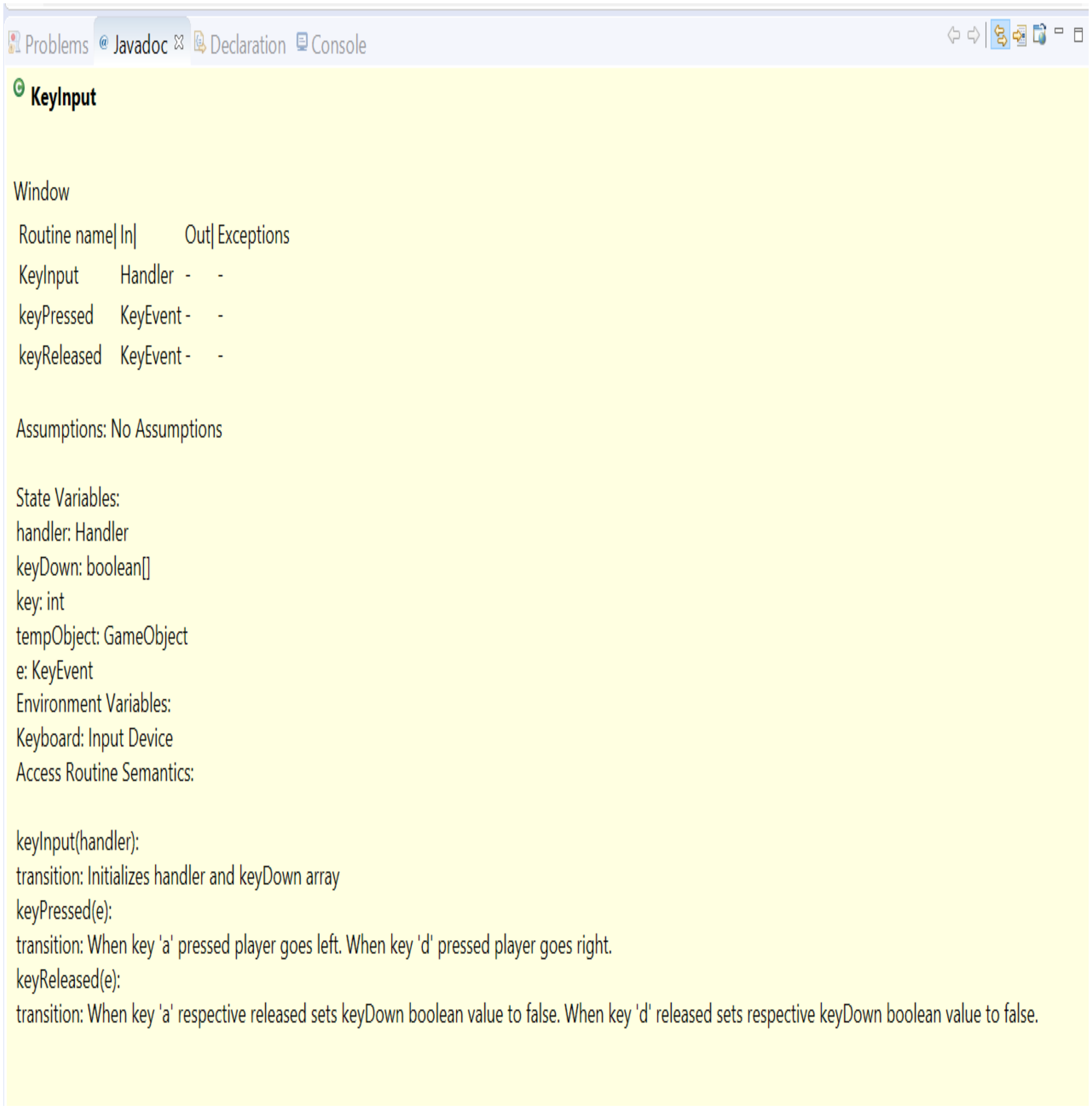


KeyInput

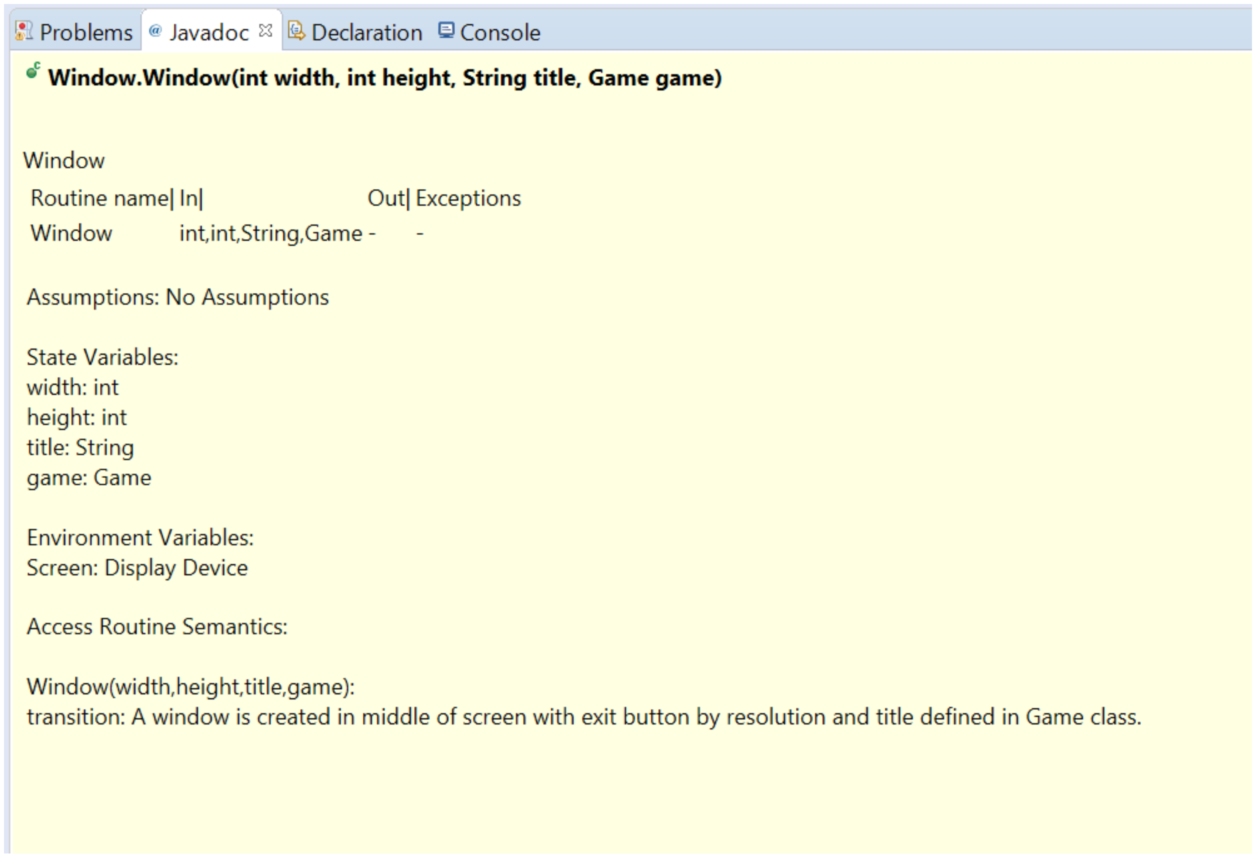
Window

Routine name	In	Out	Exceptions
KeyInput	Handler	-	-
keyPressed	KeyEvent	-	-
keyReleased	KeyEvent	-	-

Assumptions: No Assumptions

State Variables:
handler: Handler
keyDown: boolean[]
key: int
tempObject: GameObject
e: KeyEvent
Environment Variables:
Keyboard: Input Device
Access Routine Semantics:

keyInput(handler):
transition: Initializes handler and keyDown array
keyPressed(e):
transition: When key 'a' pressed player goes left. When key 'd' pressed player goes right.
keyReleased(e):
transition: When key 'a' respective released sets keyDown boolean value to false. When key 'd' released sets respective keyDown boolean value to false.


Problems | Javadoc | Declaration | Console

Window.Window(int width, int height, String title, Game game)

Window

```
Routine name| In|                  Out| Exceptions
Window       int,int,String,Game -      -
```

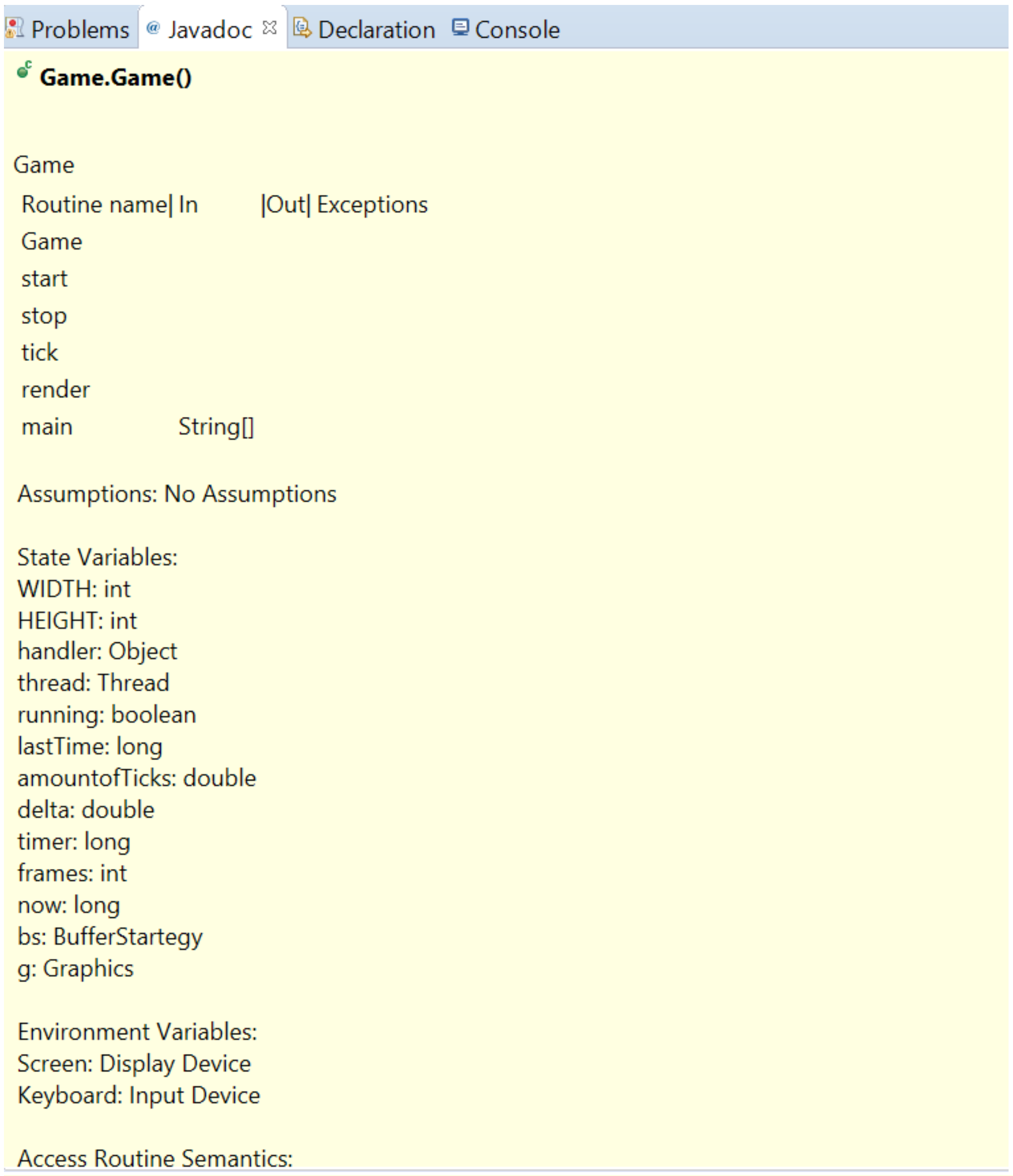
Assumptions: No Assumptions

State Variables:
width: int
height: int
title: String
game: Game

Environment Variables:
Screen: Display Device

Access Routine Semantics:

Window(width,height,title,game):
transition: A window is created in middle of screen with exit button by resolution and title defined in Game class.



Game.Game()

Game

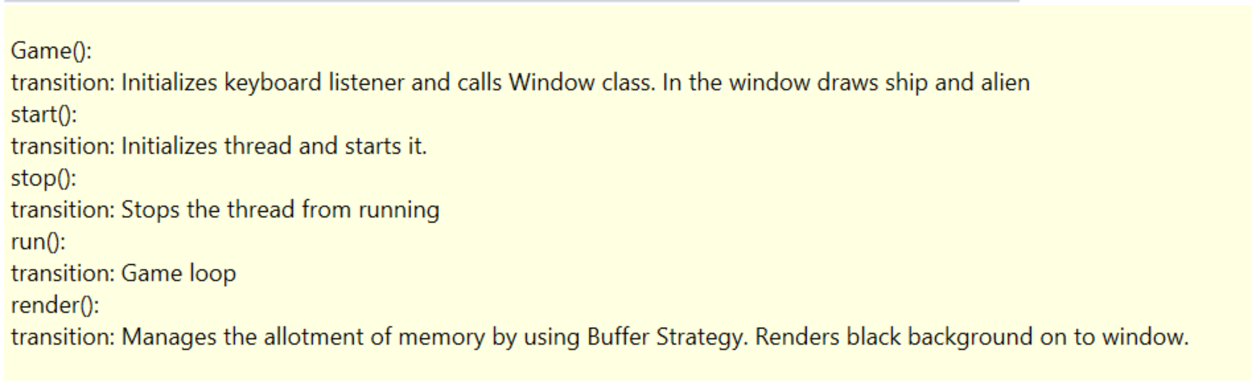
Routine name	In	Out	Exceptions
Game			
start			
stop			
tick			
render			
main	String[]		

Assumptions: No Assumptions

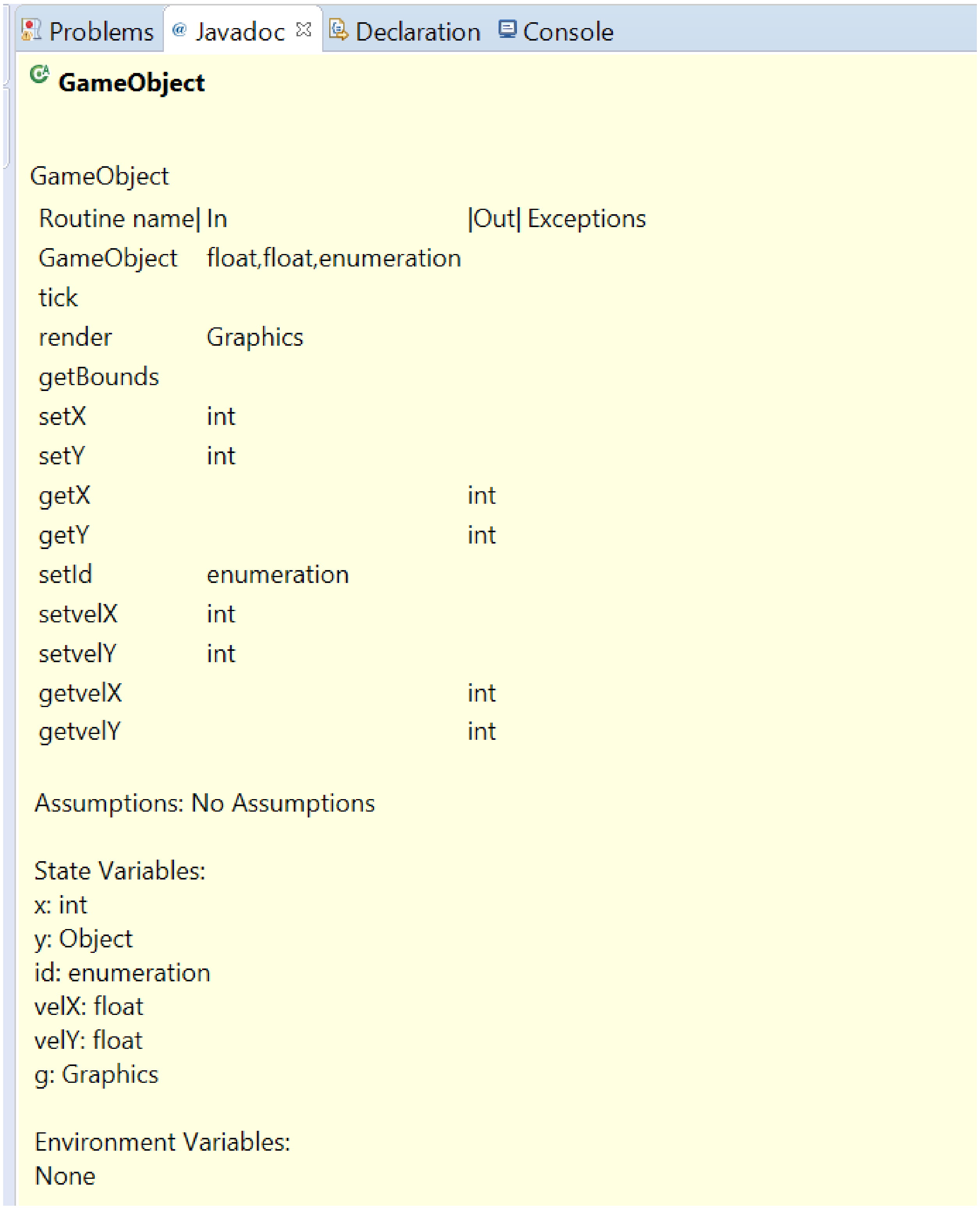
State Variables:
WIDTH: int
HEIGHT: int
handler: Object
thread: Thread
running: boolean
lastTime: long
amountofTicks: double
delta: double
timer: long
frames: int
now: long
bs: BufferStartegy
g: Graphics

Environment Variables:
Screen: Display Device
Keyboard: Input Device

Access Routine Semantics:



Game():
transition: Initializes keyboard listener and calls Window class. In the window draws ship and alien
start():
transition: Initializes thread and starts it.
stop():
transition: Stops the thread from running
run():
transition: Game loop
render():
transition: Manages the allotment of memory by using Buffer Strategy. Renders black background on to window.



GameObject

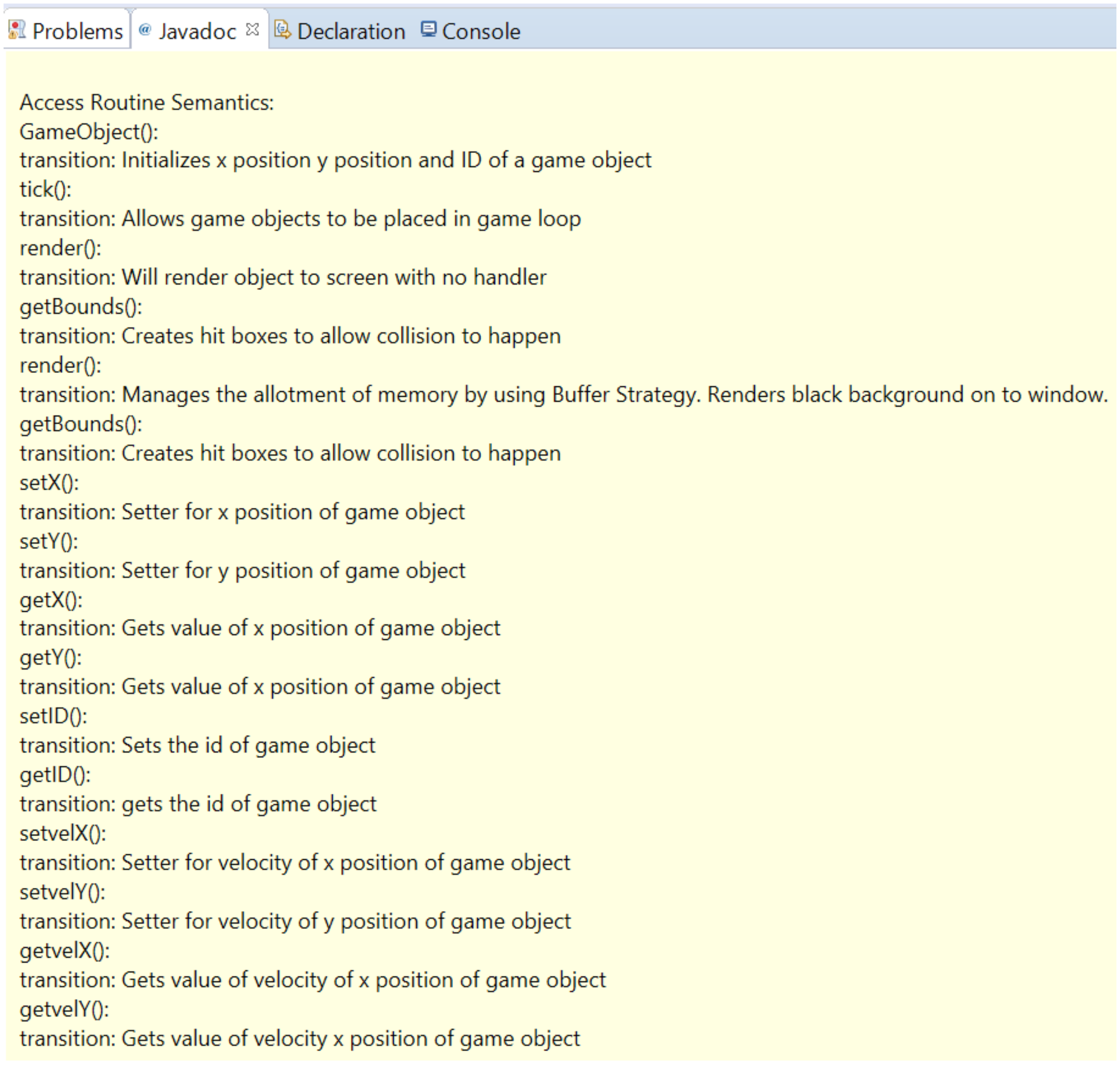
GameObject

Routine name	In	Out	Exceptions
GameObject	float,float,enumeration		
tick			
render	Graphics		
getBounds			
setX	int		
setY	int		
getX		int	
getY		int	
setId	enumeration		
setvelX	int		
setvelY	int		
getvelX		int	
getvelY		int	

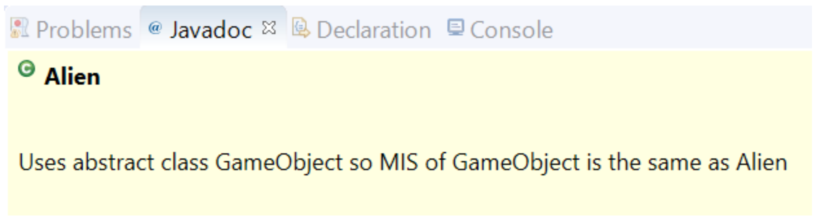
Assumptions: No Assumptions

State Variables:
x: int
y: Object
id: enumeration
velX: float
velY: float
g: Graphics

Environment Variables:
None

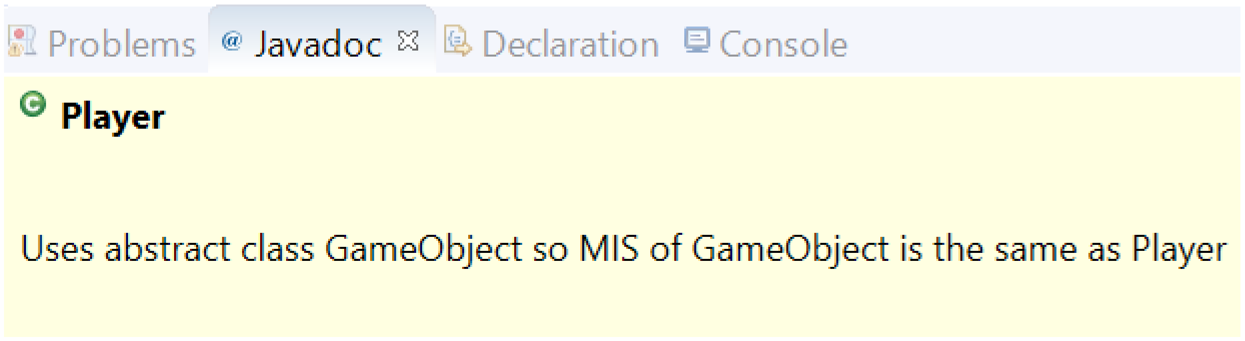


Access Routine Semantics:
GameObject():
transition: Initializes x position y position and ID of a game object
tick():
transition: Allows game objects to be placed in game loop
render():
transition: Will render object to screen with no handler
getBounds():
transition: Creates hit boxes to allow collision to happen
render():
transition: Manages the allotment of memory by using Buffer Strategy. Renders black background on to window.
getBounds():
transition: Creates hit boxes to allow collision to happen
setX():
transition: Setter for x position of game object
setY():
transition: Setter for y position of game object
getX():
transition: Gets value of x position of game object
getY():
transition: Gets value of x position of game object
setID():
transition: Sets the id of game object
getID():
transition: gets the id of game object
setvelX():
transition: Setter for velocity of x position of game object
setvelY():
transition: Setter for velocity of y position of game object
getvelX():
transition: Gets value of velocity of x position of game object
getvelY():
transition: Gets value of velocity x position of game object



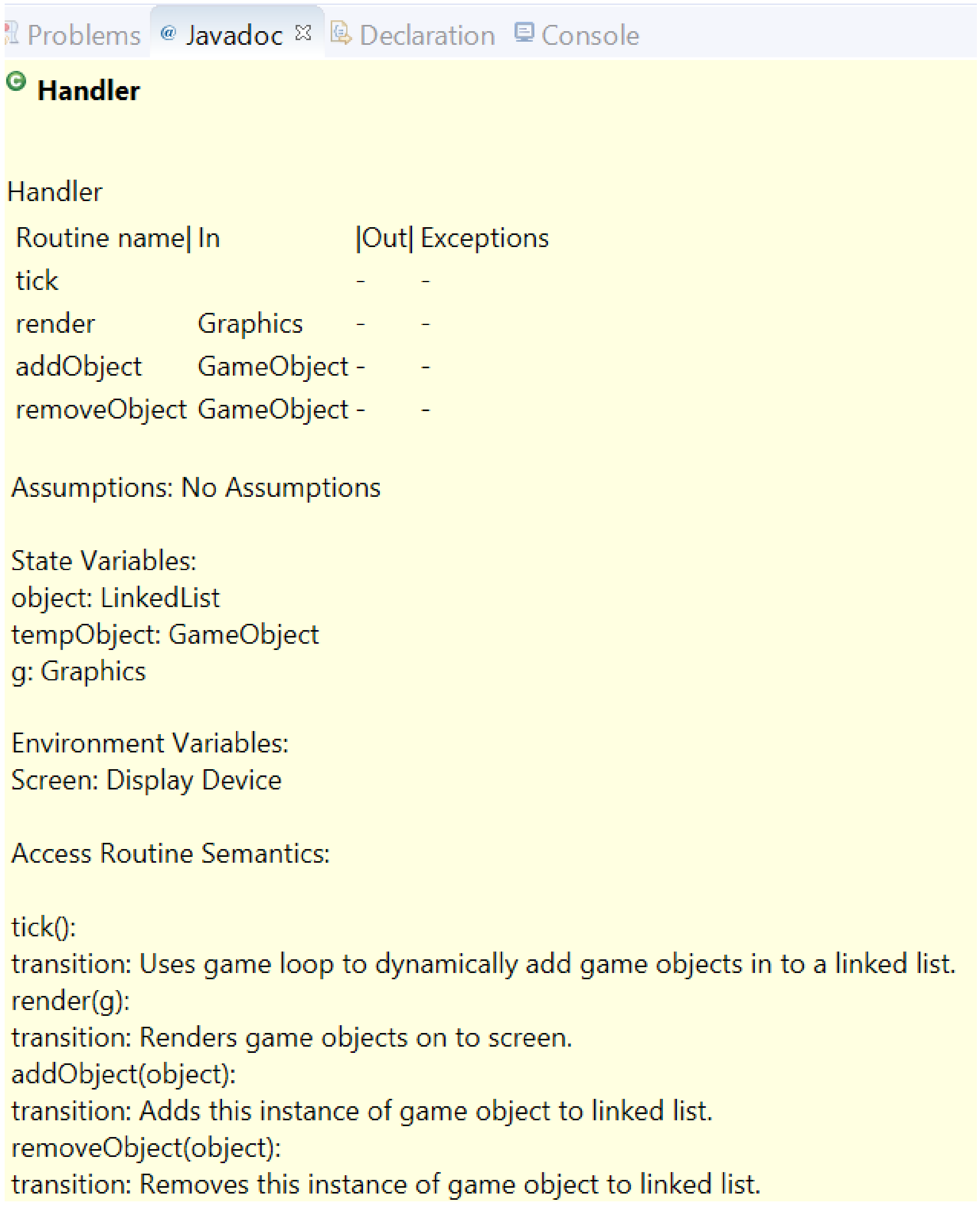
Alien

Uses abstract class GameObject so MIS of GameObject is the same as Alien



Player

Uses abstract class GameObject so MIS of GameObject is the same as Player



Handler

Handler

Routine name	In	Out	Exceptions
tick		-	-
render	Graphics	-	-
addObject	GameObject	-	-
removeObject	GameObject	-	-

Assumptions: No Assumptions

State Variables:
object: LinkedList
tempObject: GameObject
g: Graphics

Environment Variables:
Screen: Display Device

Access Routine Semantics:

tick():
transition: Uses game loop to dynamically add game objects in to a linked list.
render(g):
transition: Renders game objects on to screen.
addObject(object):
transition: Adds this instance of game object to linked list.
removeObject(object):
transition: Removes this instance of game object to linked list.