

Below are the pseudocodes and algorithms for the main classes in the program.

Program Structure:

START METHOD IN GAME CLASS --> STARTS THE START THREAD IN GAME CLASS --> CALLS RUN METHOD IN GAME CLASS --> CALLS INIT METHOD IN GAME CLASS --> CALLS DISPLAY CLASS

GameState Class:

Imports

Class header

Declare objects and variables

- Player1 object
- Player2 object
- Controller objects
- Bullet objects
- Firing variable p1
- firingTimer variable p1
- firingDelay variable p1
- Firing variable p2
- firingTimer variable p2
- firingDelay variable p2

Initialize variables and objects in the constructor

tick() method

- Tick world
- Tick player1
- Tick player2
- Tick Controllers

Player 1 Bullets Algorithm

If firing is true,

Elapsed = System timer in nanoseconds - firingTimer/1000000

If elapsed>firingDelay and player1's motionState = right,

Add a right-facing bullet, moving to the right

Set firingTimer to the System timer in nanoseconds

If elapsed>firingDelay and player1's motionState = down,

Add a down-facing bullet, moving down

Set firingTimer to the System timer in nanoseconds

If elapsed>firingDelay and player1's motionState = left,

Add a left-facing bullet, moving to the left

Set firingTimer to the System timer in nanoseconds

If elapsed>firingDelay and player1's motionState = up,

Add a up-facing bullet, moving up

Set firingTimer to the System timer in nanoseconds

Repeat the process for Player 2's bullets

render() method

- Render player1
- Render player2
- Render Controllers
- Render world

Player1 and Player2 classes (similar to each other):

Imports

Class header

Declare objects and variables

- Declare motionState variable, which keeps track of the player's direction so that they fire in the right direction

Initialize variables and objects in the constructor

tick() method

Movement Algorithm

If the up key is pressed and none of the other keys are pressed,

Move player upwards

If the down key is pressed and none of the other keys are pressed,

Move player downwards

If the left key is pressed and none of the other keys are pressed,
Move player leftwards

If the right key is pressed and none of the other keys are pressed,
Move player rightwards

Obstacle Detection Algorithm

For Horizontal Obstacles

For downward direction

If player's x position \geq __ and x position \leq __ and y position \geq __ and y position \leq __,
Set y to the first y value

For upward direction

If player's x position \geq __ and x position \leq __ and y position \geq __ and y position \leq __,
Set y to the second y value

For Vertical Obstacles

For rightward direction

If player's x position \geq __ and x position \leq __ and y position \geq __ and y position \leq __,
Set x to the first x value

For leftward direction

If player's x position \geq __ and x position \leq __ and y position \geq __ and y position \leq __,
Set x to the second x value

Repeat for every obstacle

render() method

If up key is pressed,
Set motionState = up
Draw tank facing up to the screen

Else If down key is pressed,
Set motionState = down
Draw tank facing down to the screen

Else If left key is pressed,
Set motionState = left
Draw tank facing left to the screen

Else If right key is pressed,
Set motionState = right
Draw tank facing right to the screen

If up key is pressed and none other is,
Set motionState = up
Draw tank facing up to the screen

Else If down key is pressed and none other is,
Set motionState = down
Draw tank facing down to the screen

Else If left key is pressed and none other is,
Set motionState = left
Draw tank facing left to the screen

Else If right key is pressed and none other is,
Set motionState = right
Draw tank facing right to the screen

getX() method
Return player's x position

getY() method
Return player's y position

getMotionState() method
Return player's motionState