

Design Overview for Space Race

Name: Nguyen Nam Tung

Student ID: 103181157

Summary of Program

My custom program is a game called “Space Race”. It is a shooting-game, and the player ‘s mission is controlling a spaceship to take down the enemies. However, there are bosses that always spawn in the game. The boss is much tougher than the normal enemies and they can withstand multiple hits from the player. You win if you destroy all the enemies or all the bosses. You lose if you get hit by an enemy, a boss or you get out of the safe zone (which is the top-screen. You also lose when one of the bosses reach the base (the bottom-screen). However, to support your gameplay experience, special abilities have also been added. When you obtain the special ability, the player gets healed and an ally ship shows up, destroys all enemies and bosses in a horizontal path.

A sketch of the output

Required Data Types

Describe each of the records and enumerations you will create using the following table (one per record).

radius: distance from the centre of the image to the edge

Record 1: Player

Field Name	Type	Notes
:x	Integer	x coordinate of the player
:y	Integer	y coordinate of the player
:angle	Integer	angle of the ship
:image	Gosu::Image	image of the player
:velocity_x	Integer	x velocity of the player
:velocity_y	Integer	y velocity of the player
:radius	Integer	radius of the ship
:window	Gosu::Window	the space race screen

Record 2: Enemy

Field Name	Type	Notes
:x	Integer	x coordinate of the enemy
:y	Integer	y coordinate of the enemy
:radius	Integer	radius of the enemy
:image	Gosu::Image	image of the enemy

Record 3: Boss

Field Name	Type	Notes
:x	Integer	x coordinate of the boss
:y	Integer	y coordinate of the boss
:radius	Integer	radius of the boss
:image	Gosu::Image	image of the boss
:hp	Integer	health of the boss

Record 4: Bullet

Field Name	Type	Notes
:x	Integer	x coordinate of the bullet
:y	Integer	y coordinate of the bullet
:radius	Integer	radius of the bullet
:direction	Integer	direction of the bullet(base on angle of the player)
:image	Gosu::Image	image of the bullet
:window	Gosu::Window	the space race screen

Record 5: Ally

Field Name	Type	Notes
:x	Integer	x coordinate of the ally
:y	Integer	y coordinate of the ally
:radius	Integer	radius of the ally
:image	Gosu::Image	image of the ally

Record 6:Ability

Field Name	Type	Notes
:x	Integer	x coordinate of the ability
:y	Integer	y coordinate of the ability
:radius	Integer	radius of the ability
:image	Gosu::Image	image of the ability

Record 7: Explosion

Field Name	Type	Notes
:x	Integer	x coordinate of the explosion
:y	Integer	y coordinate of the explosion
:radius	Integer	radius of the explosion
:images	Gosu::Image	images of the explosion (from beginning to end)

:image_index	Integer	the starting image
:finished	Boolean	To check whether the explosion has finished to delete the explosion

Overview of Program Structure

Player function:

- turn_right_player(player): change the angle of the ship
- turn_left_player(player): change the angle of the ship
- accelerate(player): calculate the change in the coordinate of the ship
- move_player(player): change of the coordinate of player
- draw_player(player): draw the player

Ally function

- move_ally(ally): change the coordinate of the ally
- draw_ally(ally): draw the ally

Ability function

- move_ability(ability): change the coordinate of the special ability
- draw_ability(ability): draw the special ability

Boss function

- move_boss(boss): change the coordinate of the boss
- draw_boss(boss): draw the boss

Bullet function

- move_bullet(bullet): change the coordinate of the bullet
- draw_bullet(bullet): draw the bullet
- check_onscreen(bullet): check whether the bullet is in the screen. If it is not, delete the bullet

Enemy function

- move_enemy(enemy): change the coordinate of the enemy
- draw_enemy(enemy): draw the enemy

Explosion function

- draw_explosion(explosion): draw the explosion

Game function

- initialize: run the starting screen
- initialize_game: run the start of the game
- initialize_end(fate): run the end screen of the game
- draw
 - draw_start: draw the starting screen

- draw_game: draw the object in the game
- draw_end: draw the messages at the end
- update
 - update_game: control the movements and functions of the object in the game
- button_down(id): use button to control the game
 - button_down_start(id)
 - button_down_game(id)
 - button_down_end(id)

