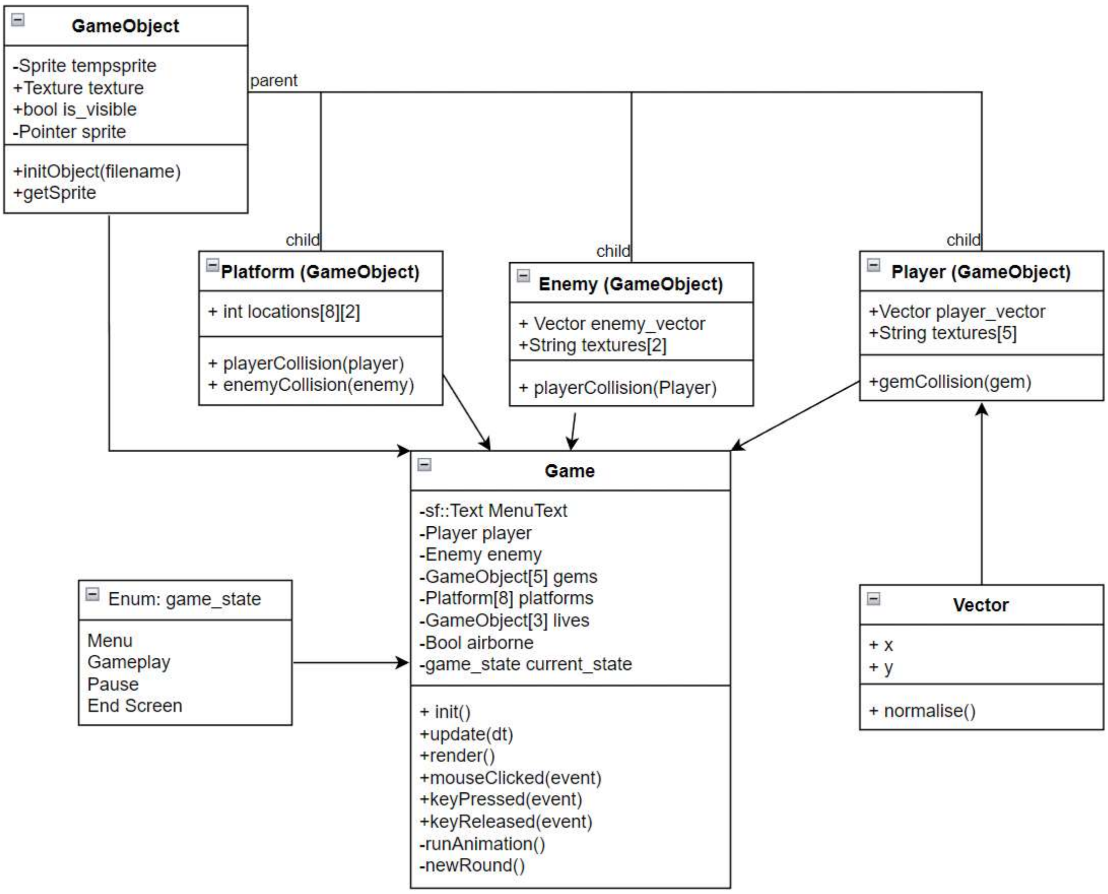
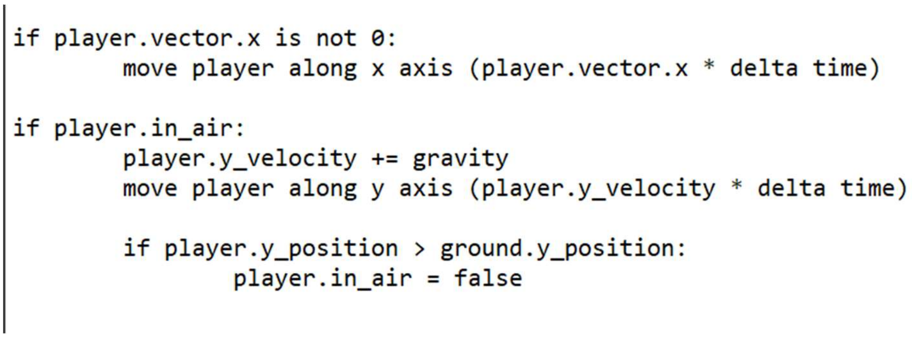
Games in C++
Assignment 2: Platformer

Initial UML Diagram



Movement Pseudocode

Update



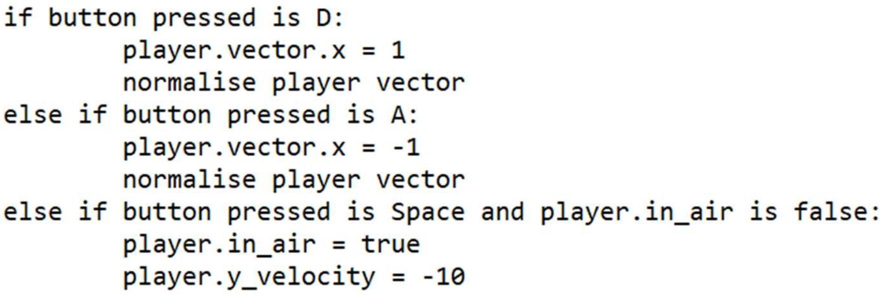
```
if player.vector.x is not 0:
        move player along x axis (player.vector.x * delta time)

if player.in_air:
        player.y_velocity += gravity
        move player along y axis (player.y_velocity * delta time)

        if player.y_position > ground.y_position:
                player.in_air = false
```

The update function first moves the player along the x axis if the x value of the vector has a value (is not equal to 0) After this it checks whether the player is in the air (jumping) and if so, uses the y_velocity variable to move the player over a curve by adding a gravity variable to it. After moving the player on the y axis, it checks to see if the player is on the ground, and if so, it sets the in air variable to false so the jump code will not run until it is true again.

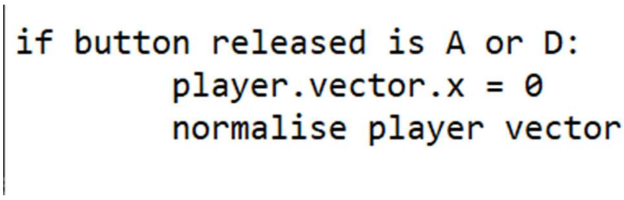
Key press



```
if button pressed is D:
        player.vector.x = 1
        normalise player vector
else if button pressed is A:
        player.vector.x = -1
        normalise player vector
else if button pressed is Space and player.in_air is false:
        player.in_air = true
        player.y_velocity = -10
```

The first part of the key press function sets the x vector to either 1 or minus 1 depending on whether the input is A or D, and then normalises the vector. If the input is Space and the player is on the ground, it sets the player to be in the air, and sets the y_velocity to –10 so the player can start its jump.

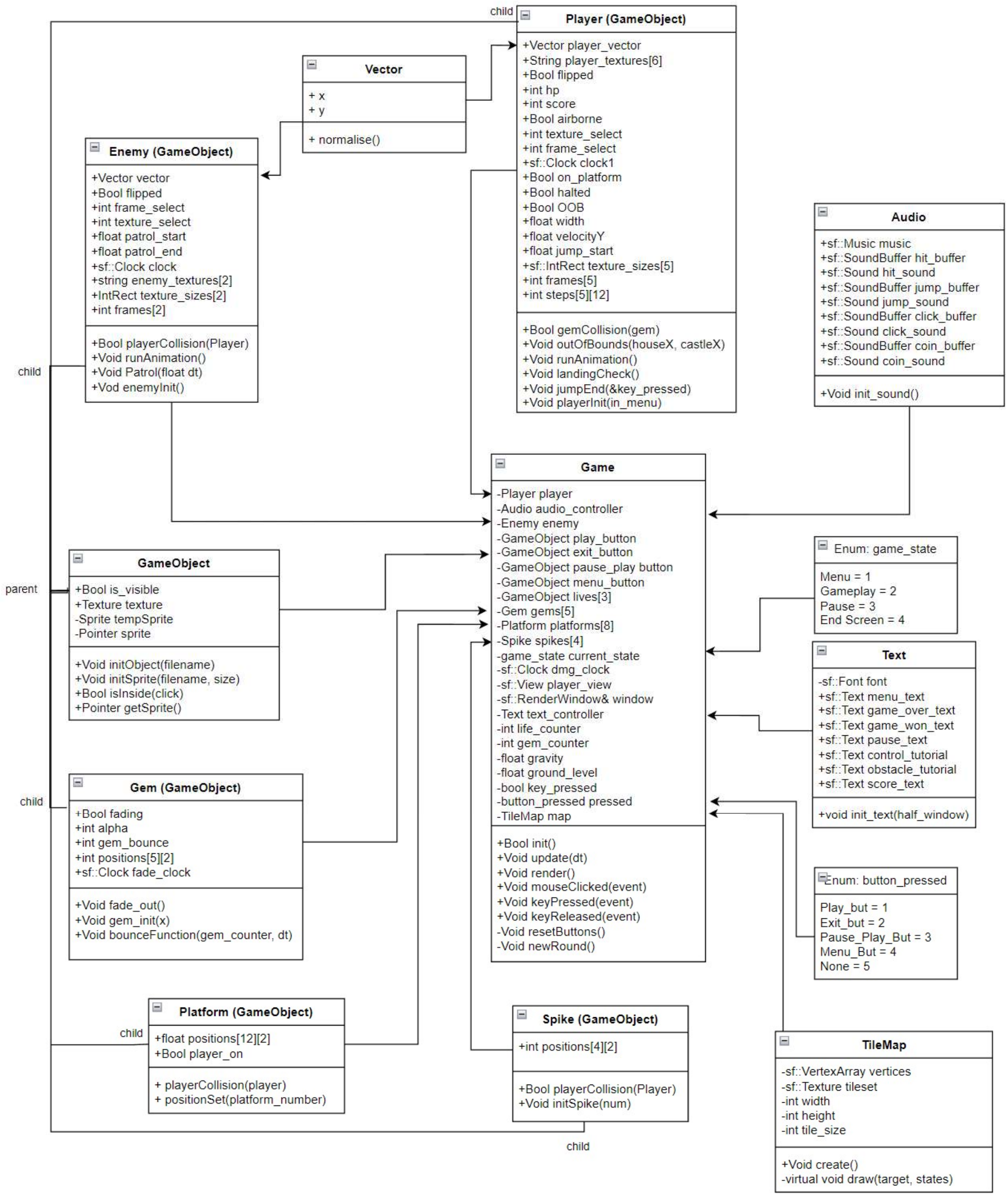
Key released



```
if button released is A or D:
        player.vector.x = 0
        normalise player vector
```

The key released function just sets the x value of the vector to 0 and normalises it so the player stops moving.

Final UML Diagram



child
Player (GameObject)
+Vector player_vector
+String player_textures[6]
+Bool flipped
+int hp
+int score
+Bool airborne
+int texture_select
+int frame_select
+sf::Clock clock1
+Bool on_platform
+Bool halted
+Bool OOB
+float width
+float velocityY
+float jump_start
+sf::IntRect texture_sizes[5]
+int frames[5]
+int steps[5][12]
+Bool gemCollision(gem)
+Void outOfBounds(houseX, castleX)
+Void runAnimation()
+Void landingCheck()
+Void jumpEnd(&key_pressed)
+Void playerInit(in_menu)
Vector
+ x
+ y
+ normalise()
Enemy (GameObject)
+Vector vector
+Bool flipped
+int frame_select
+int texture_select
+float patrol_start
+float patrol_end
+sf::Clock clock
+string enemy_textures[2]
+IntRect texture_sizes[2]
+int frames[2]
+Bool playerCollision(Player)
+Void runAnimation()
+Void Patrol(float dt)
+Vod enemyInit()
child
Audio
+sf::Music music
+sf::SoundBuffer hit_buffer
+sf::Sound hit_sound
+sf::SoundBuffer jump_buffer
+sf::Sound jump_sound
+sf::SoundBuffer click_buffer
+sf::Sound click_sound
+sf::SoundBuffer coin_buffer
+sf::Sound coin_sound
+Void init_sound()
Game
-Player player
-Audio audio_controller
-Enemy enemy
-GameObject play_button
-GameObject exit_button
-GameObject pause_play button
-GameObject menu_button
-GameObject lives[3]
-Gem gems[5]
-Platform platforms[8]
-Spike spikes[4]
-game_state current_state
-sf::Clock dmg_clock
-sf::View player_view
-sf::RenderWindow& window
-Text text_controller
-int life_counter
-int gem_counter
-float gravity
-float ground_level
-bool key_pressed
-button_pressed pressed
-TileMap map
+Bool init()
+Void update(dt)
+Void render()
+Void mouseClicked(event)
+Void keyPressed(event)
+Void keyReleased(event)
-Void resetButtons()
-Void newRound()
parent
GameObject
+Bool is_visible
+Texture texture
-Sprite tempSprite
-Pointer sprite
+Void initObject(filename)
+Void initSprite(filename, size)
+Bool isInside(click)
+Pointer getSprite()
Enum: game_state
Menu = 1
Gameplay = 2
Pause = 3
End Screen = 4
Text
-sf::Font font
+sf::Text menu_text
+sf::Text game_over_text
+sf::Text game_won_text
+sf::Text pause_text
+sf::Text control_tutorial
+sf::Text obstacle_tutorial
+sf::Text score_text
+void init_text(half_window)
child
Gem (GameObject)
+Bool fading
+int alpha
+int gem_bounce
+int positions[5][2]
+sf::Clock fade_clock
+Void fade_out()
+Void gem_init(x)
+Void bounceFunction(gem_counter, dt)
Enum: button_pressed
Play_but = 1
Exit_but = 2
Pause_Play_But = 3
Menu_But = 4
None = 5
child
Platform (GameObject)
+float positions[12][2]
+Bool player_on
+ playerCollision(player)
+ positionSet(platform_number)
Spike (GameObject)
+int positions[4][2]
+Bool playerCollision(Player)
+Void initSpike(num)
child
TileMap
-sf::VertexArray vertices
-sf::Texture tileset
-int width
-int height
-int tile_size
+Void create()
-virtual void draw(target, states)