

6) `void Game::processMouseDown(sf::Event t_event)`
`{`
`m_mouseDown.x = static_cast<float>(t_event.mouseButton.x);`
`m_mouseDown.y = static_cast<float>(t_event.mouseButton.y);`
`}`

↓ This line assigns the x coordinate value of the point where the mouse button click event (`t_event.mouseButton`) occurred to the `m_mouseDown.x` variable. `static_cast<float>` is used to explicitly convert the x coordinate value to float type.

В этой строке происходит явное преобразование значения координаты x в тип float. В которой программно создано название функции `static_cast`. После чего значение x будет преобразовано в float.

11) displacement - `неперемещение`

`R * 180,0`
`D + 90,0`

`if (Left == t_event.mouseButton.Button) Velocity 100.0`
`if (Right == t_event.mouseButton.Button) Velocity 80.0`

`checkCollisionDistance(sf::Vector2f t_pos1, float t_rad1, sf::Vector2f t_pos2, float t_rad2)`
`float distance;`
`float minimumSafeDistance;`
`Velocity 100.0`
`Velocity 80.0`

`#define _USE_MATH_DEFINES` `использование констант` `определений матем. констант. (IEEE 754)`

`checkCollisionDistance` (^{big} location, ^{small} radius)

`Vector2f` ^{big} 0,0
 ^{small} 0,0

`Bounds ?` height width

`m_bigRadius = m_bigPlaneSprite.getLocalBounds().height/2.0f`

`else` — " — width/2.0f

Exploding
 Explosion

`int frame;`
`const int Rows = 6`
`const int Cols = 8`
`const int Size = 100`
`int col;`
`int row;`


```
m_expFrameTimer += m_expIncrement;  
frame = static_cast<int>(m_expFrameTimer);
```

```
if (frame > 47)
```

$\Rightarrow 48, 0f$

```
frame = 0
```

false

```
if (frame != m_expFrame)  
col = frame % 8;  
row = frame / 6;
```

$(col * SIZE, row * SIZE, SIZE, SIZE)$

(14)

```
float m_expIncrement {0.6f};  
float m_expFrameTimer {0.0f};  
float  
int m_expFrame {0};
```