*else*

*{*

*//You can only die once.*

*animation\_over = 1;*

*}*

*}*

*else*

*{*

*texture.loadFromFile("Resources/Images/Pacman" + std::to\_string(CELL\_SIZE) + ".png");*

*sprite.setTexture(texture);*

*sprite.setTextureRect(sf::IntRect(CELL\_SIZE \* frame, CELL\_SIZE \* direction, CELL\_SIZE, CELL\_SIZE));*

*i\_window.draw(sprite);*

*animation\_timer = (1 + animation\_timer) % (PACMAN\_ANIMATION\_FRAMES \* PACMAN\_ANIMATION\_SPEED); // 6\*4*

*}*

*}*

*void Pacman::update(unsigned char i\_level, std::array<std::array<Cell, MAP\_HEIGHT>, MAP\_WIDTH>& i\_map)*

*{*

*//анализ стен*

*//проверка кнопок*

*//изменение координат пакмана в зависимости от стен и направления*

*//проход по тунелю*

*//обработка таблеток и пилюли*

*//обработка таймера энергии*

*std::array<bool, 4> walls{};*

*walls[0] = map\_collision(CollectPellets::No, I\_Use\_Door::No\_, PACMAN\_SPEED + position.x, position.y, i\_map);*

*walls[1] = map\_collision(CollectPellets::No, I\_Use\_Door::No\_, position.x, position.y - PACMAN\_SPEED, i\_map);*

*walls[2] = map\_collision(CollectPellets::No, I\_Use\_Door::No\_, position.x - PACMAN\_SPEED, position.y, i\_map);*

*walls[3] = map\_collision(CollectPellets::No, I\_Use\_Door::No\_, position.x, PACMAN\_SPEED + position.y, i\_map);*

*if (1 == sf::Keyboard::isKeyPressed(sf::Keyboard::Right))*

*{*

*if (0 == walls[0]) //You can't turn in this direction if there's a wall there.*

*{*

*direction = 0;// pacman moves right for default (it implements in the pacman class)*

*}*

*}*

*if (1 == sf::Keyboard::isKeyPressed(sf::Keyboard::Up))*

*{*

*if (0 == walls[1])*

*{*

*direction = 1;*

*}*

*}*

*if (1 == sf::Keyboard::isKeyPressed(sf::Keyboard::Left))*

*{*

*if (0 == walls[2])*

*{*

*direction = 2;*

*}*

*}*

*if (1 == sf::Keyboard::isKeyPressed(sf::Keyboard::Down))*

*{*

*if (0 == walls[3])*

*{*

*direction = 3;*

*}*

*}*

*if (0 == walls[direction])//if not the button, then it's necessary check*

*{*

*switch (direction)*

*{*

*case 0:*

*{*

*position.x += PACMAN\_SPEED;*

*break;*

*}*

*case 1:*

*{*

*position.y -= PACMAN\_SPEED;*

*break;*

*}*

*case 2:*

*{*

*position.x -= PACMAN\_SPEED;*

*break;*

*}*

*case 3:*

*{*

*position.y += PACMAN\_SPEED;*

*break;*

*}*

*}*

*//}*

*//////////////////////////////////////////////////////////////////boarder/////////*

*if (-CELL\_SIZE >= position.x) // -1|0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 | 21*

*// | |*

*{*

*position.x = CELL\_SIZE \* MAP\_WIDTH - PACMAN\_SPEED;*

*}*

*else if (CELL\_SIZE \* MAP\_WIDTH <= position.x)*

*{*

*position.x = PACMAN\_SPEED - CELL\_SIZE;*

*}*

*/////////////////////////////////////////////////////////////////////////////////////*

*if (1 == map\_collision(1, 0, position.x, position.y, i\_map)) //When Pacman eats an energizer...*

*{*

*//He becomes energized!*

*energizer\_timer = static\_cast<unsigned short>(ENERGIZER\_DURATION / pow(2, i\_level));*

*}*

*else*

*{*

*energizer\_timer = std::max(0, energizer\_timer - 1);*

*//energizer\_timer = 0;*

*/\*template<class T>*

*const T& max(const T & a, const T & b)*

*{*

*return (a < b) ? b : a;*

*}\*/*

*}*

*}*

*}*