

RAPPORT JEUX PICO PARK

STRUCTURES DU JEUX

- Le jeux est divisé en 6 Class.
- MainMenu
- Scene1
- Scene2
- Scene3
- Keyboard
- WinScene

LA CLASSE KEYBOARD

LA CLASSE Keyboard

- Elle nous laisse changer l'emplacement du joueur ,en utilison le clavier de plus , il ajout aussi de l'animation.
- Cette Class est constitué tous simplement d'un seul constructeur.
- Ce constructeur prend 3 variable, la scene actuelle , le Sprite qu'on va ajouter le pouvoir de boujer et de l'animation,et un Bool qui décide si d'inclure de l'animation ou non.

La forme du constructeur keyboard

Dans le constructeur on cree un event Listener de type Keyboard,
Puis j'ai créer une fonction lambda :

```
eventListener->onKeyPressed = [anim](EventKeyboard::KeyCode  
keyCode, Event *event)
```

La forme du constructeur keyboard

Dans cette fonction il existe un switch qui dans chaque cas il cree une animation :

```
auto animation = Animation::create();  
    for (int i = 1; i <= 5; i++)  
    {  
        std::string name = StringUtils::format("res/vikingPlayer/viking  
%d.png", i);  
        animation->addSpriteFrameWithFile(name.c_str());  
    }
```

La forme du constructeur keyboard

IL cree aussi deux Action et un Spawn,il va choisir un a utilisé dependant de la valeur du Bolléan anim :

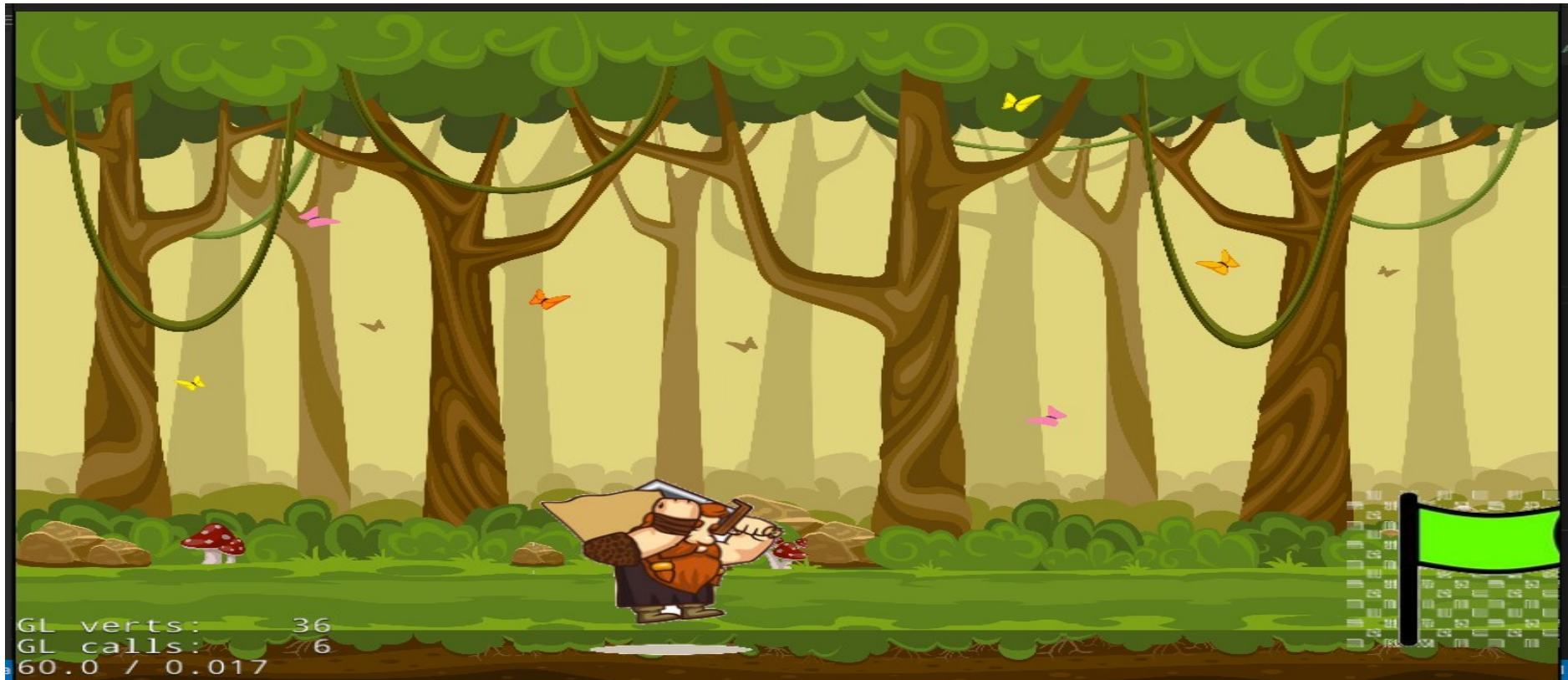
```
auto action1 = Animate::create(animation);  
auto action2 = MoveBy::create(0.5, Vec2(-10, 0));  
auto action = Spawn::create(action1, action2, nullptr);  
if(anim==true){  
    event->getCurrentTarget()->runAction(action);  
}else event->getCurrentTarget()->runAction(action2);  
}
```

LA CLASSE Scene1

Classe Scene1

- C'est le premiere niveau du jeu ,il hérite de la classe cocos2d::Scene .
- Ce niveau contient trois Sprite, une qui fonctionne comme background, et une qui service comme joueur , et finalement une flag, qui sert comme le but du du joueur.

Classe Scene1



Classe Scene1

- Pour que le joueur puisse bouger ,j'ai fais appelle a la Classe Keyboard :`keyboard(sprite,this,true);`
- Et pour que le joueur puisse aller au niveau 2, j'ai ajouter un contact Listener :
- *`auto contactListener =EventListenerPhysicsContact::create();`*
- En contact entre le but et le joueur ,la scene1 est remplacé en Scene2:
- *`Director::getInstance()->replaceScene(TransitionFade::create(1.0,Scene2::createScene()));`*

Classe Scene2



Classe Scene2

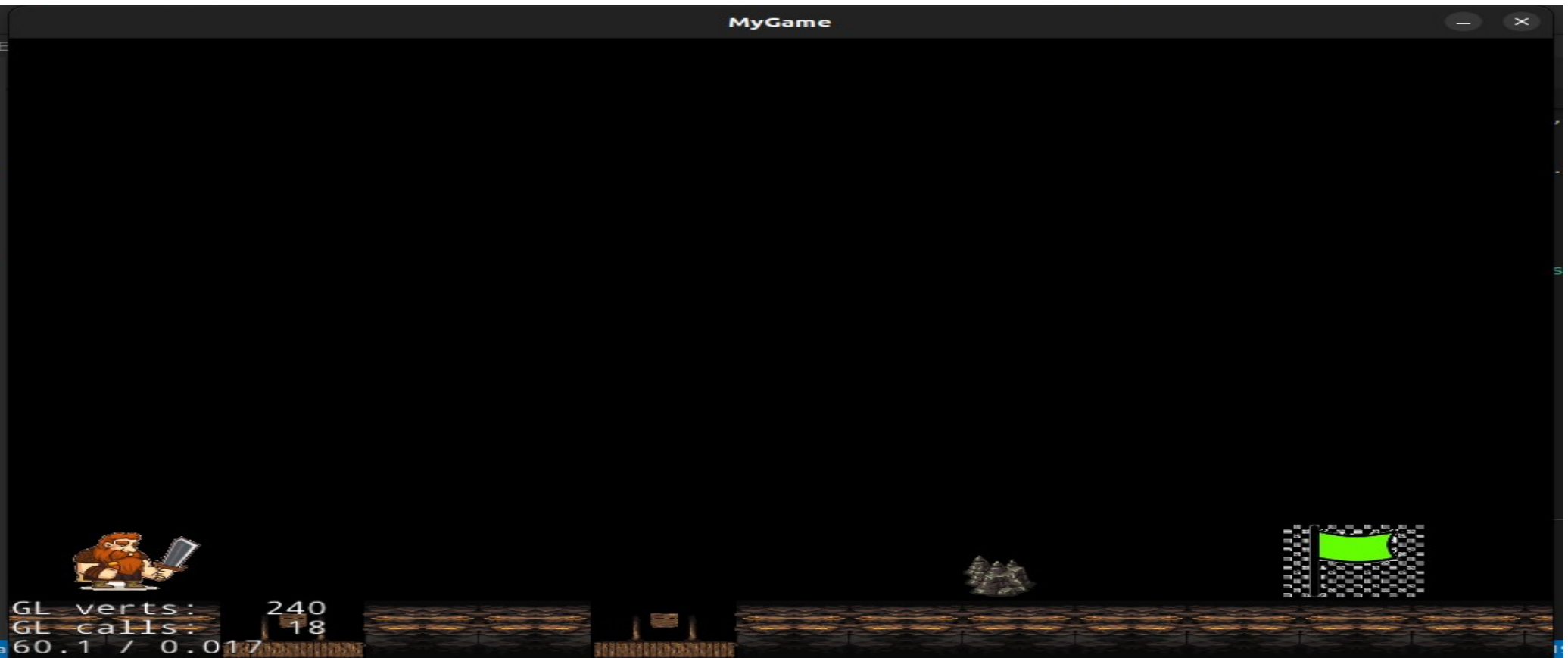
- Meme que la scene1 , le joueur doit arrivé jusqu'a le but pour aller a la scene 2 , mais il y'a un obstacle si le joueur se met en contact avec le l'animal ,il perd et il doit repété le niveau.

Classe Scene2

- Pour savoir qu'elle objet le joueur a mis contact avec,chaque sprite a un tag :
- `auto nodeA = contact.getShapeA()->getBody()->getNode();`
- `auto nodeB = contact.getShapeB()->getBody()->getNode();`
- `if (nodeA->getTag() == 10 && nodeB->getTag() == 10)`
- `{`
- `Director::getInstance()->replaceScene(MainMenu::createScene());`
- `}`
- `else`
- `{`
- `Director::getInstance()->replaceScene(Scene3::createScene());`
- `}`

LA CLASSE Scene3

Classe Scene3



CLASSE Scene3

- La terre que le joueur marche sur est un vecteur de Type Sprite* :
- `Vector<Sprite *> listTile;`
- Il ya un sprite trap, est un rock que si le joueur touche il perd le niveau .
- Comme les niveaux précédent il doit arrivé jusqu'à
- le but .

LA CLASSE MainMenu



LEVEL 1

LEVEL 2

LEVEL 3

GL verts: 48
GL calls: 8
39.9 / 0.017

Classe MainMenu

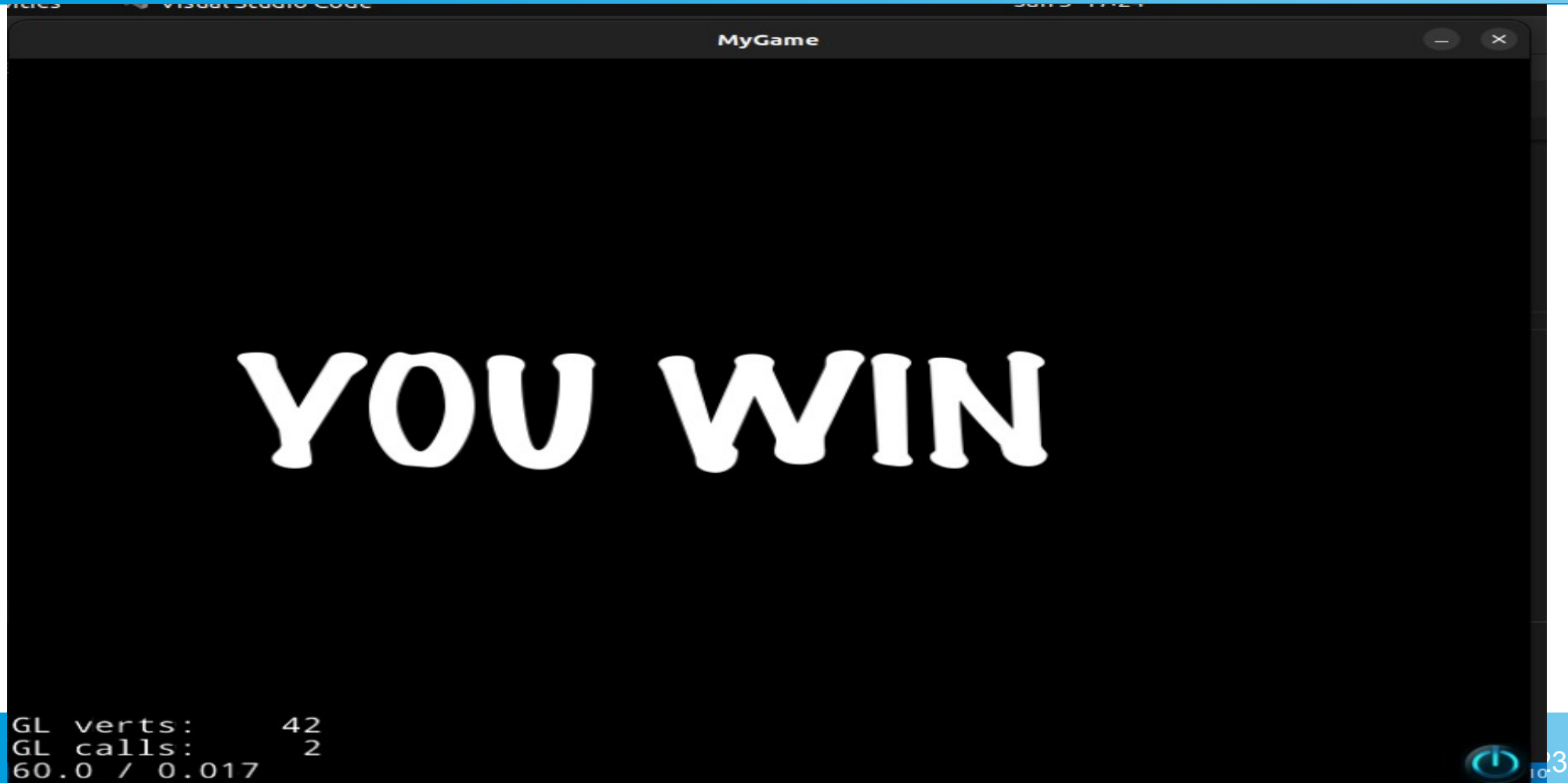
- C'est la scene qui s'ouvre l'orsque le jeux commence.
- Il est composé de trois MenuItemImage,chaque image envoie vers un niveau.
- *auto level1=MenuItemImage::create("res/level3.png", "res/level3.png", CC_CALLBACK_1(MainMenu::GoToScene3, this));*
- *auto menu = Menu::create(level1,level2,level3,nullptr);*

Classe MainMenu

```
void MainMenu::GoToScene1(cocos2d::Ref*  
pSender)  
{  
    auto scene = Scene1::createScene();  
    Director::getInstance()-  
    >replaceScene(TransitionFade::create(1.0,scene);  
}
```

LA CLASSE WinScene

Classe WinScene



Classe WinScene

- Il est composé d'un Label et d'une méthode qui ferme l'application :
- `void menuCloseCallback(cocos2d::Ref* pSender);`
- `void WinScene::menuCloseCallback(Ref* pSender)`
- `{`
- `Director::getInstance()->end();}`