

## Probeklausur „Multimedia-Programmierung“, SS 2017

### Aufgabe 1: Szenengraph und User Interfaces

Gegeben sei folgende Scene (Cocos2d-x):

```
1   Size visibleSize = Director::getInstance()->getVisibleSize();
2   Vec2 origin = Director::getInstance()->getVisibleOrigin();
3
4   auto welcomeL = Label::createWithTTF("Willkommen zu MMP",
5   "fonts/Marker Felt.ttf", 40);
6   welcomeL->setPosition(Vec2(origin.x + visibleSize.width/2,
7   origin.y + visibleSize.height - welcomeL->getContentSize().height));
8   this->addChild(welcomeL, 1);
9
10  auto nameL = Label::createWithTTF("Wie heisst du?",
11  "fonts/Marker Felt.ttf", 40);
12  nameL->setPosition(Vec2(origin.x + visibleSize.width/3,
13  origin.y + 2*(visibleSize.height/3)));
14  this->addChild(nameL, 2);
15
16  auto nameF = cocos2d::ui::TextField::create("dein Name",
17  "fonts/Marker Felt.ttf", 40);
18  nameF->setPosition(Vec2(origin.x + 2*(visibleSize.width/3),
19  origin.y + 2*(visibleSize.height/3)));
20  this->addChild(nameF, 2);
21
22  auto whatL = Label::createWithTTF("Was willst du hier lernen?",
23  "fonts/Marker Felt.ttf", 30);
24  whatL->setPosition(Vec2(origin.x + visibleSize.width/2,
25  origin.y + (visibleSize.height/2)));
26  this->addChild(whatL, 3);
27
28  auto startB = cocos2d::ui::Button::create("animationbuttonnormal.png",
29  "animationbuttonpressed.png");
30  startB->setTitleText("Spiele designen & programmieren");
31  startB->setTitleFontSize(15);
32  startB->setPosition(Vec2(origin.x + visibleSize.width/3,
33  (visibleSize.height/3)));
34  this->addChild(startB, 3);
35
36  auto goonB = cocos2d::ui::Button::create("animationbuttonnormal.png",
37  "animationbuttonpressed.png");
38  goonB->setTitleText("Eigentlich will ich nur ne gute Note");
39  goonB->setTitleFontSize(15);
40  goonB->setPosition(Vec2(origin.x + 2*(visibleSize.width/3),
41  (visibleSize.height/3)));
42  this->addChild(goonB, 3);
43
44  auto sprite = Sprite::create("blackbackground.png");
45  sprite->setPosition(Vec2(visibleSize.width/2 + origin.x,
46  visibleSize.height/2 + origin.y));
47  this->addChild(sprite, 0);
```

Geben Sie die wesentlichen Elemente des Scene Graphs an und erklären Sie kurz deren Zweck und Zusammenhang? Welche Vorteile bringt dieser Aufbau mit sich?

Skizzieren Sie einen Cocos2d-x konformen Szenengraph für dieses Dokument. Die Klassennamen der Elemente sind ausreichend für die Beschriftung der Knoten.

Skizzieren Sie grob das Interface, das entsteht, wenn dieses Skript ausgeführt wird. Achten Sie dabei auch auf die richtige Beschriftung der Elemente. Der Hintergrund des Spiels ist einfach schwarz gefärbt und muss nicht gezeichnet werden.

## Aufgabe 2: Grafik und Animation

Die folgenden Abbildungen skizzieren ein Rechteck in Cocos2d-x vor (links) und nach einer Transformation (rechts). Bearbeiten Sie die Transformation in folgendem Code so, dass der Zustand in Abbildung 2 (rechts) erreicht wird. Während die Rotation sofort geschieht, wird das Verschieben des Rechtecks über zwei Sekunden animiert. Es darf kein Transformationstyp zweimal vorkommen.

```
1  Scene* HelloWorld::createScene(){
2      auto scene = Scene::create();
3      auto mySprite = Sprite::create("rect.png");
4      mySprite->setPosition(Vec2(55, 55));
5
6      auto moveTo = _____;
7      mySprite->_____;
8      mySprite->_____;
9
10
12     scene->addChild(mySprite);
13     return scene;
14 }
```

Gegeben sei ein typisches Koordinatensystem für grafische Ausgaben auf Bildschirmen. Ein Objekt soll sich von Punkt (20,40) zu Punkt (10,50) bewegen. Geben Sie den Bewegungsvektor des Objekts an.

### Aufgabe 3: Gemischtes

Gegeben sei folgender Python-Programmcode. Geben Sie an, welche Ausgaben durch den Code auf der Konsole generiert werden.

```
text = "Probeklausur MMP" print text[13:]  
_____  
_____ print text[0:1]  
_____ print text[:]
```

Gegeben sei folgender Python-Programmcode und folgende Ausgaben. Geben Sie Python-Programmcode an, der folgende Ausgaben auf der Konsole erzeugt.

```
text = "Probeklausur MMP" >> 'Probe'  
_____  
_____ >> 'MMP'
```

Gegeben sei folgender Python-Programmcode. Bei der Ausführung entsteht ein ungewünschter Effekt. Nennen Sie diesen und korrigieren Sie den Code.

```
arr = [10,20,101,2] i = 0 while i < len(arr): print arr[i] i =  
i+1
```