

SpriteKit

Martin Hrubý

Seminář iOS, FIT VUT v Brně

Game knihovny

- SpriteKit — 2D hry.
 - <http://spritekitlessons.wordpress.com>
- SceneKit — 3D hry.
- Metal — profi hry v C/C++.

Schopnosti SpriteKitu

- Dynamika objektů na obrazovce.
- Fyzikální simulátor.
- Kolize.
- Výkonnost enginu není zázračná. U jednodušších her je vhodné snížit FPS kvůli nárokům na výpočetní výkon (a baterku).
 - Bohužel přechod na Metal bude značný skok.
 - Přenositelnost na jiné platformy.

Složení scény

- **SKView** — obdoba UIWindow. Vkládají se sem scény.
- Odvozeno od UIResponder:
 - **SKScene** — obdoba UIView, obsahuje SKNode. Editor scén. Viditelná je jedna scéna. Přechody mezi scénami.
 - **SKNode** — objekt ve scéně. Referencuje se typicky jménem (string). Taky obsahuje další SKNode.
 - **SKAction** — chování objektu.

Instanciacie scény

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    // Configure the view.
    SKView * skView = (SKView *)self.view;
    skView.showsFPS = YES;
    skView.showsNodeCount = YES;
    /* Sprite Kit applies additional optimizations to improve rendering
performance */
    skView.ignoresSiblingOrder = YES;

    // Create and configure the scene.
    GameScene *scene = [GameScene unarchiveFromFile:@"GameScene"];
    scene.scaleMode = SKSceneScaleModeAspectFill;

    // Present the scene.
    [skView presentScene:scene];
}
```

Scéna

- size, anchorPoint (LL corner), view
- -(void)didMoveToView:(SKView *)view
 - - (void) willMoveFromView: (SKView *) view
- -(void)touchesBegan:(NSSet *)touches
withEvent:(UIEvent *)event
- -(void)update:(CFTimeInterval)currentTime
 - Vykreslovací smyčka.

SKNode

- Typy:
 - SKLabelNode — textový objekt.
 - SKShapeNode — čtverec, kolo, čára, ...
 - SKSpriteNode — obrázky.
 - SKLightNode — zdroj světla ve scéně.
 - SKFieldNode — zdroj pole (magnetické, gravitační).
 - SKEmitterNode — emituje částice (nejsou to SK objekty).
- Referencování: názvem nebo referencí.

SKLabelNode

```
-(void)didMoveToView:(SKView *)view {
    SKLabelNode *myLabel = [SKLabelNode
labelNodeWithFontNamed:@"Chalkduster"];

    myLabel.name = @"text";
    myLabel.text = @"Hello, World!";
    myLabel.fontSize = 65;
    myLabel.position = CGPointMake(CGRectGetMidX(self.frame),
                                   CGRectGetMidY(self.frame));

    [self addChild:myLabel];

    SKNode *node = [self childNodeWithName: @"text"];
}
```


Ukázka objektu: míček

```
static const int brickCat = 1;  
static const int deskaCat = 2;  
static const int micekCat = 4;
```

```
@interface SKMicek : SKNode  
@end
```

```

-(id) init
{
    self = [super init];

    SKShapeNode *kulicka = [SKShapeNode shapeNodeWithCircleOfRadius:
MicekRadius];
    kulicka.strokeColor = [UIColor redColor];
    kulicka.fillColor = [UIColor blueColor];
    kulicka.name = @"micek-telo";
    [self addChild: kulicka];

    self.name = @"micek";
    self.physicsBody = [SKPhysicsBody bodyWithCircleOfRadius:
MicekRadius];
    self.physicsBody.affectedByGravity = NO;
    // nezpomaluje se
    self.physicsBody.linearDamping = 0;

    // kolize
    self.physicsBody.categoryBitMask = micekCat;
    self.physicsBody.contactTestBitMask = micekCat | brickCat |
deskaCat;

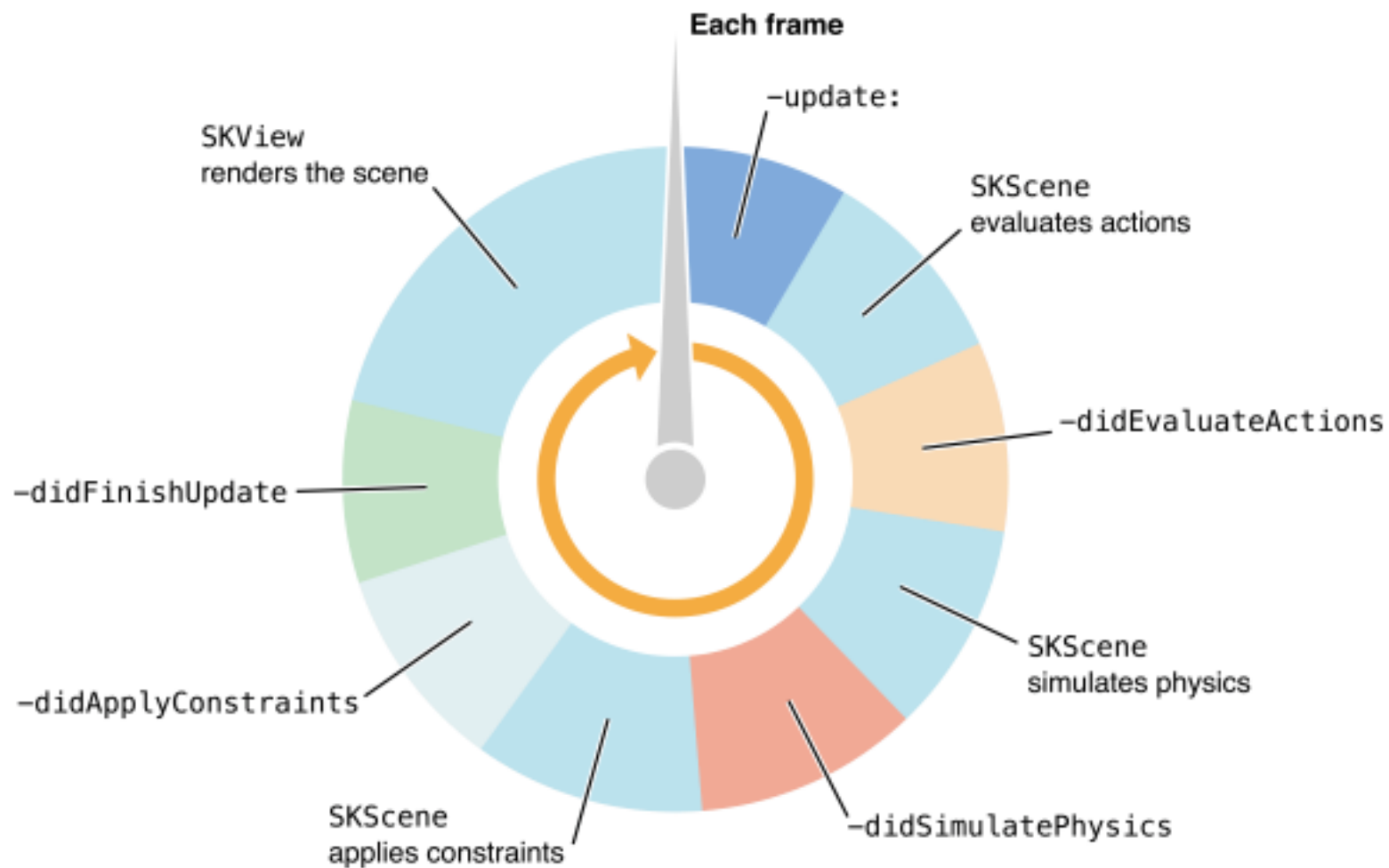
    return self;
}

```

Dynamika uzlů

- SKAction — explicitně definované chování uzlu.
- SKPhysicsBody — definice fyzického těla (geometrie tělesné schránky). Pak reaguje na pole (gravitace) a na kolize (explicitně specifikované).
- Kolize — nutno definovat vztah dvou objektů.

Činnost simulátoru



Akce

- Specifikuje se typ akce a doba trvání akce.
 - Přesun do bodu, přesun po cestě. Rotace. Zrychlení.
 - Změna geometrie. Skrytí/zobrazení. Odstranění.
 - group / sequence / repeat.
 - runBlock / performSelector — uživatelsky programované akce.
- Objekt akce se zasílá uzlu (runAction:). Akce lze zrušit (removeAllActions).

Akce nad objekty

- Objekty bez fyzikálního těla — SKAction a update:
- Objekty s fyzikálním tělem — rychlost (velocity), kolize a kontakty, pole.

Chování uzlu v čase

```
-(void) behavior  
{  
}
```

```
-(void) installBehavior  
{  
    SKAction *wait = [SKAction waitForDuration: 0.1];  
    SKAction *call = [SKAction performSelector: @selector(behavior)  
                    onTarget: self];  
    SKAction *behSeq = [SKAction sequence: @[wait, call]];  
  
    SKAction *top = [SKAction repeatActionForever: behSeq];  
    [self runAction: top];  
}
```

Reakce na touch události

```
-(void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    UITouch *touch = [touches anyObject];
    CGPoint xy = [touch locationInNode: self];
    SKNode *touchedNode = [self nodeAtPoint: xy];

    if ([touchedNode.name isEqualToString: @"tlacitko"]) {
        // akce
    }
}
```


Jednoduchá akce

```
if ([touchedNode.name isEqualToString: @"tlacitko"]) {  
    SKAction *act = [SKAction moveByX: 100 y:100 duration:3];  
    [micek runAction: act];  
}
```

Reakce na gesta

- Pozor: při odchodu ze scény zrušit rozpoznávače gest!

```
grecog = [[UISwipeGestureRecognizer alloc] initWithTarget: self action:  
@selector(actionGRecog:)];  
grecog.direction = UISwipeGestureRecognizerDirectionRight;  
[self.view addGestureRecognizer: grecog];  
  
// zruseni  
[self.view removeGestureRecognizer: grecog];
```

Přechody mezi scénami

- Nová scéna nahradí předchozí (není návratový zásobník scén).
- Návrat nutno provést explicitně. Opět se volá konstruktor scény!

```
SKAction *tm = [SKAction waitForDuration: 1.1];

[self runAction: tm completion:^(
    SKTransition *door = [SKTransition
doorsOpenHorizontalWithDuration: 2];
    SKResultScene *nscene = [[SKResultScene alloc] initWithSize:
self.scene.size];
    nscene.prevScene = self;

[self.view presentScene: nscene transition:door];
)];
```

Fyzikální simulátor

- Objekty mohou podléhat gravitaci (gravitaci nastavuje scéna).
- Udělení impulsu objektům.
- Instalace dalších polí (radialGravity, magnetic, electric).

SKPhysicsBody

- Objekt SKPhysicsBody je atribut SKNode.
- Modeluje vztah SKNode k fyzikálnímu simulátoru. (PhysSim operuje nad physBodies).
- Atributy:
 - BOOL — affectedByGravity, dynamic
 - friction, restitution
 - linearDumping — koeficient zpomalování (vektor rychlosti).
 - atributy pro kolize

Vektor rychlosti objektu

```
SKShapeNode *pokus = [SKShapeNode shapeNodeWithRect: CGRectMake(0, 0,  
100, 100)];  
pokus.position = CGPointMake(100, 300);  
pokus.physicsBody = [SKPhysicsBody bodyWithRectangleOfSize:  
CGSizeMake(100, 100)];  
pokus.physicsBody.velocity = CGVectorMake(100, 100);  
[self addChild: pokus];
```

```
pokus.physicsBody.affectedByGravity = YES;  
self.physicsWorld.gravity = CGVectorMake(0, -1.81);
```

Přitažlivost

```
SKFieldNode *grava = [SKFieldNode radialGravityField];
    SKShapeNode *ins = [SKShapeNode shapeNodeWithCircleOfRadius:10];
    ins.fillColor = [UIColor yellowColor];
    [grava addChild: ins];
    grava.strength = 0.2;
    grava.position = CGPointMake(400, 400);
    grava.physicsBody = [SKPhysicsBody bodyWithCircleOfRadius:10];
    grava.physicsBody.affectedByGravity = NO;
    grava.physicsBody.dynamic = NO;
    [self addChild: grava];
```

Konfigurace kolizí

- Kontakty a kolize. Aplikováno na `physicsBody`.
- Každý objekt má `categoryBitmask` (bitové pole).
 - `collisionBitMask` — jádro samo vypočítává reakce na kolizi.
 - `contactBitMask` — je zaslána zpráva delegátovi. Program musí vypočítat reakci.

Kontakty

- Zavedení delegáta na výpočet reakcí na kolize.
 - SKPhysicsContactDelegate
 - self.physicsWorld.contactDelegate = self;
 - -(void) didBeginContact:(SKPhysicsContact *)contact
- SKPhysicsContact: bodyA, bodyB
 - změna vektoru pohybu

Life ukázka a závěr

- Výkonnost SpriteKitu.