

SLOVENSKÁ TECHINCKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Semestrálne zadanie

VOID game engine

VÝVOJ APLIKÁCIÍ S VIACVRSTVOVOU ARCHITEKTÚROU

Samuel Bubán

ID: 102879

25.4.2021

	1
GAME CORE	2
GameMain	2
GameController	2
GameWindow	2
KeyController	3
GameObject	4
Vector2	4
PREVIEW	4
EditorMain	4
GameObject	4
MOŽNÉ VYLEPŠENIE	4
GameEngine	4
Component system	4
Collision detection	5
Physics	5
Dashboard	5
ZÁVER	5
ZDROJE	6

GAME CORE

GameMain

Celá časť spustiteľnej aplikácie sa nachádza v balíčku gameCore. Obsahuje hlavnú triedu GameMain.java, v ktorej bežia dva loop-y. Jeden je na fyzikálne simulácie, druhý je totožný s renderovaním. Oba sú obmedzené, to znamená, že sa dá nastaviť, koľkokrát za sekundu sa majú vykonať. Default hodnota je 75 Hz pre fyzikálny loop a 300 Hz pre update loop.

Po sputení triedy GameMain.java sa vytvorí nové okno, na ktorom sa nachádzajú všetky objekty, ktoré boli používateľom vytvorené. Tieto objekty môžu mať vlastnú triedu, a teda môžu mať každý špeciálne správanie. Keďže sa jedná o game engine, používateľ môže objektom priradiť ľubovoľné správanie – ovládanie pomocou klávesnice, časované správanie ...

GameController

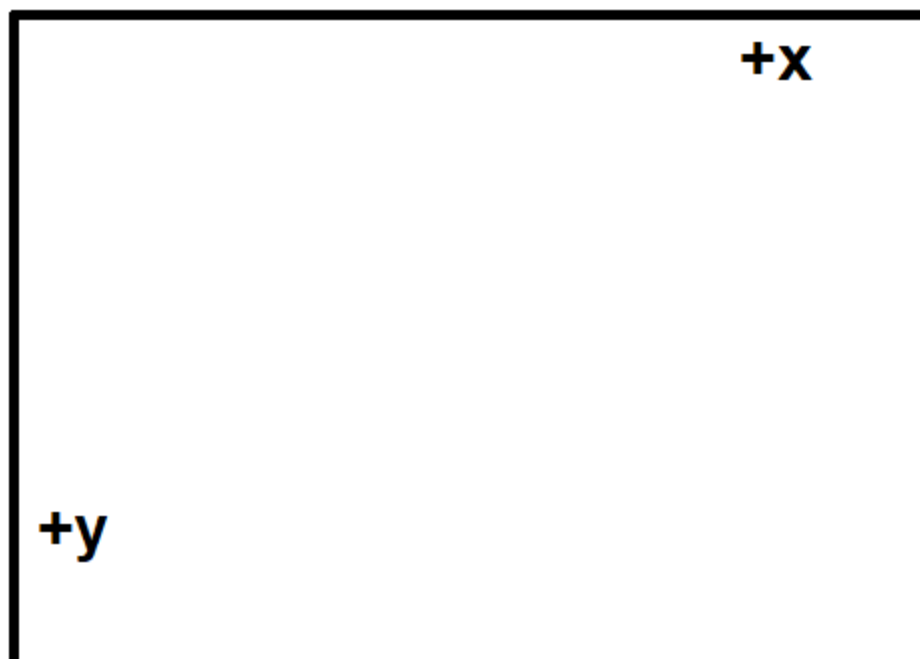
Nastaviteľné vlastnosti hry sa nachádzajú v triede GameController.java. Nachádzajú sa tu veci ako rozlíšenie obrazovky, názov okna, alebo obnovovacie frekvencie. V tejto triede sa po spustení hry nachádzajú všetky informácie pre programátora – ako dlho hra beží, ako dlho sa renderoval posledný snímok, koľko času uplynulo od posledného fyzikálneho update-u, kamera, zoznam objektov ...

Táto trieda sa ďalej stará o to, aby sa každý snímok zavolali metódy vo všetkých existujúcich objektoch, ktoré sa majú spustiť v určitom čase. Prvá metóda je Start(), ktorá sa spustí pri vytvorení objektu. Update() sa zavolá presne raz za snímok, konkrétne pred tým, ako sa obraz renderuje. LateUpdate() sa volá po vyrenderovaní obrazu. PhysicsUpdate() sa volá v presne stanovených intervaloch, a nemala by sa používať na vytváranie nových objektov, alebo náročné výpočty, pretože táto metóda zaručuje, aby sa hra správala rovnako pri každom spustení, teda sa využíva štandardne na simulácie (padanie objektov, kolízie ...).

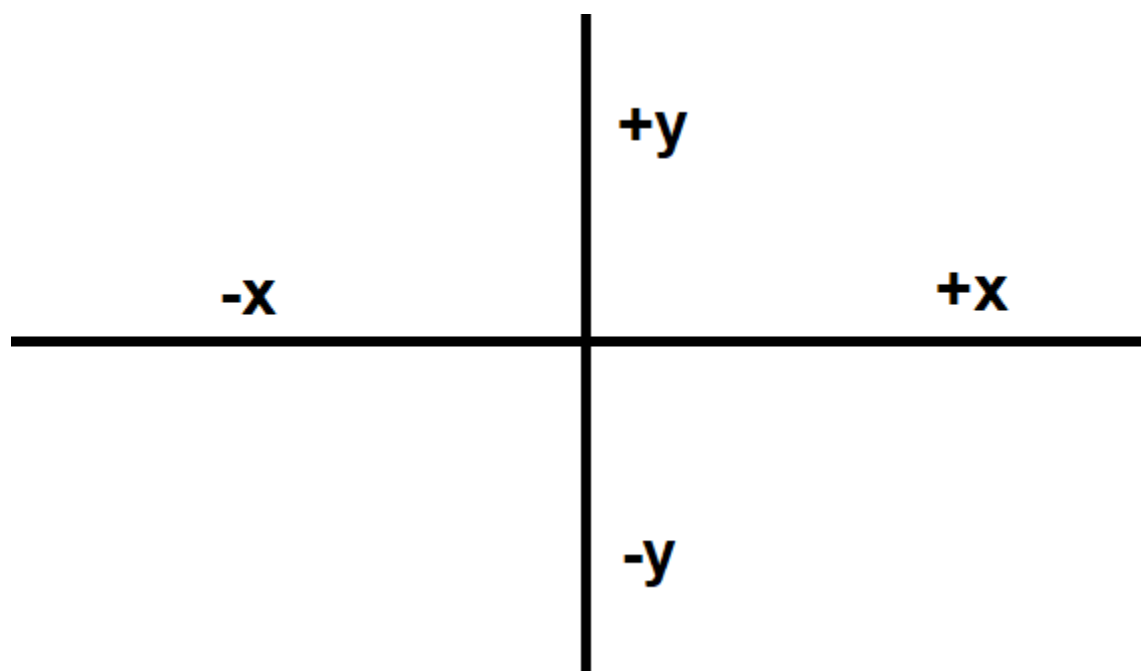
Je tu tiež výpis do konzoly, ktorá sa nachádza v editore, aktuálny frame rate – FPS (frames per second) ktorý je aktualizovaný s každým novým snímkom.

GameWindow

Táto trieda zabezpečuje okno hry. Jeho počiatočnú inicializáciu, a aj vykresľovanie objektov na obrazovku. Nakoľko štandardný JPanel pracuje s nie veľmi prirodzenou sústavou súradníc, táto trieda prevádza nové súradnice do tejto sústavy. Vykresľovanie objektov sa deje podľa ich zOrder parametra – ako sú na sebe naskladané, ktorý objekt prekrýva ktorý.



Obrázok 1 Originálny súradnicový systém



Obrázok 2 Prevedený súradnicový systém

KeyController

Táto trieda zaznamenáva, ktoré klávesy boli stlačené pri spustení hry, a teda programátor ju môže použiť na pohyb, alebo ovládanie objektov v hre.

GameObject

Trieda, od ktorej sú odvodené všetky objekty na ploche. Obsahuje pozíciu, mierku, obrázok a poradie vykreslenia. Nachádzajú sa v nej všetky metódy Start(), Update() ... ktoré programátor môže override-núť, a tým vytvoriť vlastné správanie. Nachádza sa tu aj metóda Move(), ktorá je aplikovanie pohybu na aktuálnu pozíciu.

Vector2

Dátová trieda, ktorá obsahuje x a y súradnicu. Trieda obsahuje aj metódy, ako Right(), Left(), Up(), Down(), ktoré vrátia daný smer, a dajú sa využiť na korekciu, alebo upresnenie pohybu tak, že sa prenášobí danou zložkou.

PREVIEW

Programátor nemá prístup k žiadnej z tried balíčka Preview.

EditorMain

V tejto triede sa odohráva väčšina správania editoru. Ovládanie všetkých tlačidiel, pridávanie objektov, ukladanie, načítavanie ...

GameObject

Jedná sa o modifikovanú triedu gameCore.GameObject, ktorá obsahuje nejaké informácie navyše, ako cestu k skriptu a podobne.

MOŽNÉ VYLEPŠENIE

GameEngine

Nakoľko vývoj game engine-u trvá neskutočne dlho, tento môj projekt nie je ani z ďaleka finálny. Aj svetovo najlepšie engine-y ako Unity alebo Unreal Engine sú stále vo vývoji, a pridávajú sa im nové a nové funkcionality.

Component system

Prvá vec, ktorá by sa musela zmeniť, aby bol engine lepšie rozšíriteľný, je prechod na component system. To znamená, že každý objekt by mohol obsahovať ľubovoľné množstvo komponentov (skript, kolízie, rigidbody ...).

Collision detection

Po prechode na component system by bolo logické ako prvé pridať kolízie. Pre game engine by to bol obrovský skok vpred, pretože by sa dalo okamžite vytvárať omnoho väčšie množstvo hier.

Physics

Druhý pomerne veľký skok by bolo pridanie fyziky. To znamená, že všetky objekty, ktoré majú komponent rigidbody by boli simulované – simulácia gravitácie, prenos energie pri nárazoch ...

Dashboard

Po spustení aplikácie by sa otvoril dashboard, ktorý by obsahoval posledné projekty, na ktorých používateľ pracoval, ktoré by sa dali priamo odtiaľto otvoriť. Tiež by sa dala vytvoriť vlastná koncovka pre súbor, teda po otvorení projektu by sa automaticky otvoril editor.

ZÁVER

VOID game engine nie je ani zďaleka dokončený. To, čo som vytvoril môže slúžiť akurát tak na ukážku. Ak by som sa v budúcnosti chcel tomuto znovu venovať, určite by som sa radšej rozhodol najskôr vytvoriť o dosť robustnejší game engine bez editora, a pridával by som GUI až neskôr. Teda priorita by bola pridať najprv funkcionálnu pre engine, až neskôr nejaké rozhranie.

ZDROJE

1. Samuel Bubán: VoidGameEngine 24.4.2021
<https://github.com/Mahrkeenerh/VoidGameEngine>