

Архитектура игрица

v1.1

ЛПРС2

РТ-РК

April 28, 2020

Саставни делови играчке платформе

- Контролни улази - контролише ток игре,
- Видео излаз - исцртавање резултата,
- Аудио излаз - баш и не радимо,
- Игрица - чува тренутно стање и повезује све ово.

Игрица је практично као један велики **FSM**.

Реализација играчке платформе

Платформа се може реализовати потпуно софтверски.

Али је уобичајено да се користи хардверска акцелерација за неке делове, пре свега графику која је најзахтевнија, онда звук (декодовање, мешање), физику (која се обично одрађује на графичком процесору).

Слојеви игрице

Scripting

Engine

HW

Scripting

Саставни делови су:

- **Gameplay** - логика, како се игра игра,
- **Мапе** - да припреми мапе,
- **AI** - вештачка интелигенција, ботови,
- **UI** - кориснички интерфејс, GUI, подешавања.

Реализација Scripting-a

Обично се пише у неким интерпретираним (скрипт) језицима (отуда назив). На скромнијим платформама (као што је E2LP, Sega) се пише у компајлираним (хард коре) језицима, као што је С. Ствар је компромиса:

Језик	Интерпретирани	Компајлирани
Развој	Бржи	Спорији
Измене	Лакше	Теже
Проширивост	Лакша	Тежа
Брзина (CPU)	Мања	Већа
Меморија	Већа	Мања

Оправдање за интерпретиране језике

На нивоу целе игрице:

Језик	Scripting	Остало
Захтевност (CPU, меморија)	Мања	Већа
Количина кода	Већа	Мања
Измене (tweaking, нова верзија)	Честе	Ређе
Раду	Шљакери	Мастер Јода

У случају мањих игрица, где неће бити много мењања и проширивања, Scripting може бити урађен у компајлираним језицима.

Scripting у нашем случају

На ЛПРС2 радите га у С-у.

Подразумева рад са периферијама (видео, аудио, контролер), али и хардверску акцелерацију (графика, аудио, физика, итд.). Практично, графика не може без хардвера.

Аудио ћемо прескочити овога пута.

Овде није искључиво хардвер у питању, већ и софтвер у облику системске подршке за хардвер (дравери, библиотеке).

Таксономија графике

Врсте графике (примитиве):

- Векторска - цртамо геометријске облике:
 - линије,
 - полигоне,
 - слова.
- Растерска - цртамо неке сличице:
 - спрајтове - (sprite),
 - текстуре - (texture).

Врсте простора боја:

- Color-mode - RGB, директан рад са компонентама боја,
- Index-mode - постоји палета боја, а пиксели у framebuffer-у су индекси.

По геометрији:

- 2D,
- 3D.

Спрајтови

- 2D слике које се исцртавају на већој сцени

Спрајтови



Сцена



Текстуре

- 2D слике која се мапира на 3D модел (ткз. UV мапирање)

Текстуре

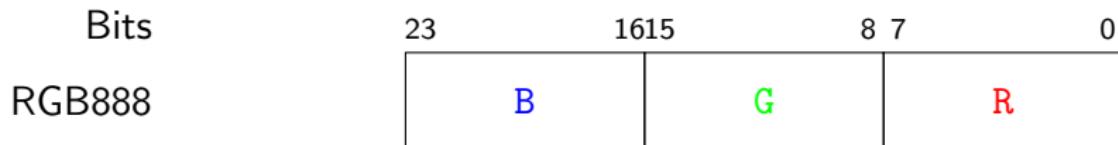


Мапиран 3D модел



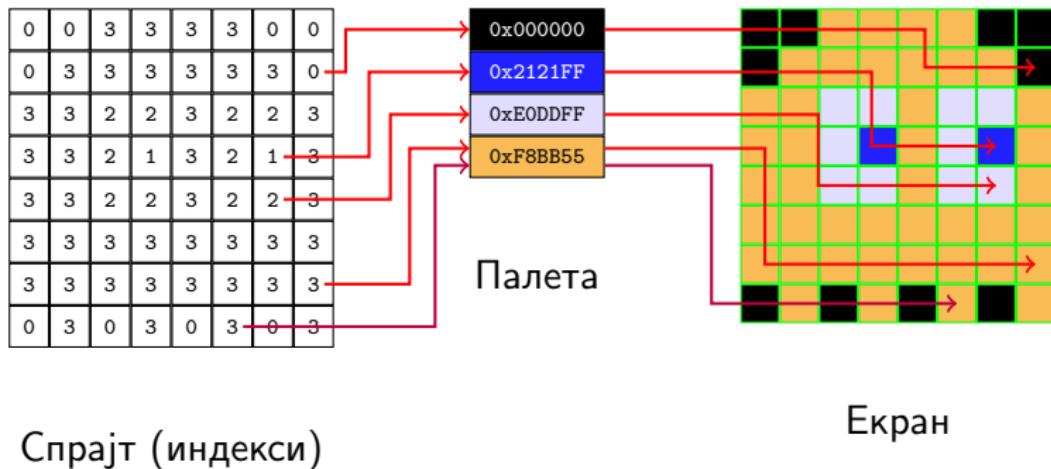
Color-mode

- Пиксел у меморији садржи вредност боје.
 - Вредност боје је тројка: R - црвена, G - зелена, B - плава.
- Типични формати: RGB888, RGB565, BGR888...
- Пример представа за RGB888 црвену боју:
 - низ бајта: 255,0,0
 - хекса у речи: 0x0000ff
 - HTML: #ff0000



Index-mode

- Пиксел у меморији садржи индекс на палету.
- У палети су color-mode вредности боја.



Color-mode vs. Index-mode

Простор боја	Color-mode	Index-mode
Обрада (мешање боја, интерполяција)	Лакше	Теже
Меморија (framebuffer)	Више	Мање
Приступ меморији	Мање	Више
Број боја	Више	Мање

Практично, индексни мод је бољи на простијим графичкима, 2D и без икаквих интерполяција и слично.

Транспарентност

- Маскирање
 - Пример на слици
- Chroma key
 - Једна боја је прозирна
 - Мање меморије и приступа истој него код маскирања
 - Типично за index-mode
- Alpha канал
 - Уз RGB додаје се и A канал (RGBA8888 типичан формат)
 - 255 - непрозирно, 0 - скроз прозирно
 - Могуће мешење, поглед кроз (обојено) стакло...

First step:



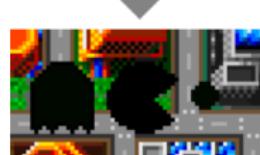
AND



Second step:

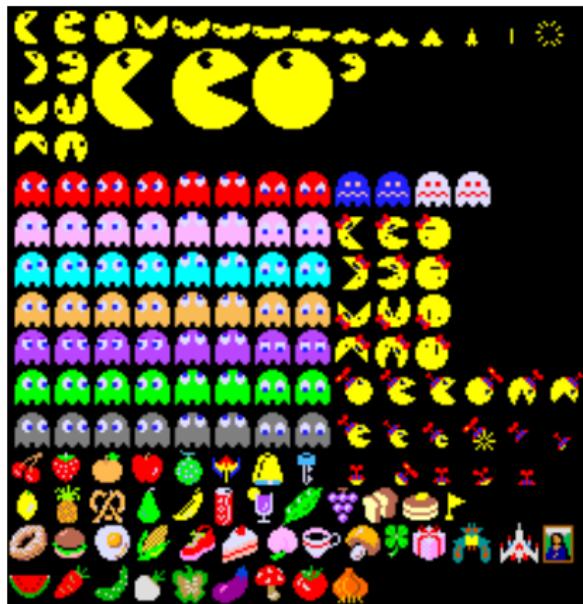


OR



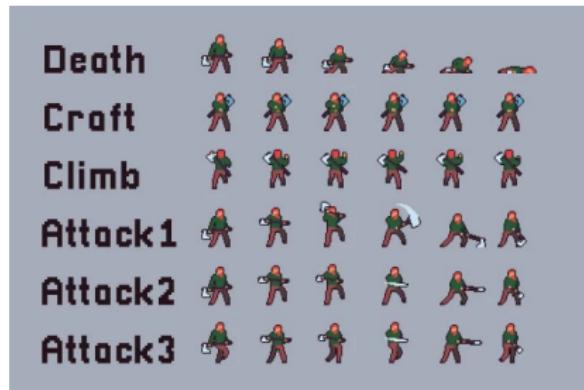
Спрајт мапе

- Велики спрајт са свим малим спрајтовима потребним за апликацију.
- Синоними: texture atlas, sprite (spread)sheet



Анимације (2D)

- Брзим наизменичним мењањем спрјатова добија се осећај покрета.



Сортирање

Практично сви алгоритми за примењени у графичком режиму, се своде на неку врсту сортирања.



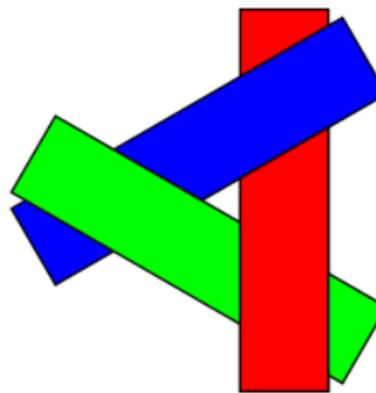
<http://ragecomic.appspot.com>

Z-оса алгоритми

- Painter's algorithm (Сликарски алгоритам) - програмер одређује у z-оси редослед исцртавања полигона.
- Z-buffering - алгоритам сам одређује који ће полигон бити преко којег.

Проблем са сликарским алгоритмом

Мало је теже цртати загуљеније 3D случајеве.



Z-buffering то успева да исхендла јер ради сортирање исцртавања на нивоу пиксела а не полигона.

Реализације

Реализација излаза може бити:

- **Framebuffer (бафер фрејма)** - сортирање преко меморије, па касније их пошаље на екран,
- **Beamchasing** - нема framebuffer-а, израчуна пиксел таман пре него што га пошаље на екран.

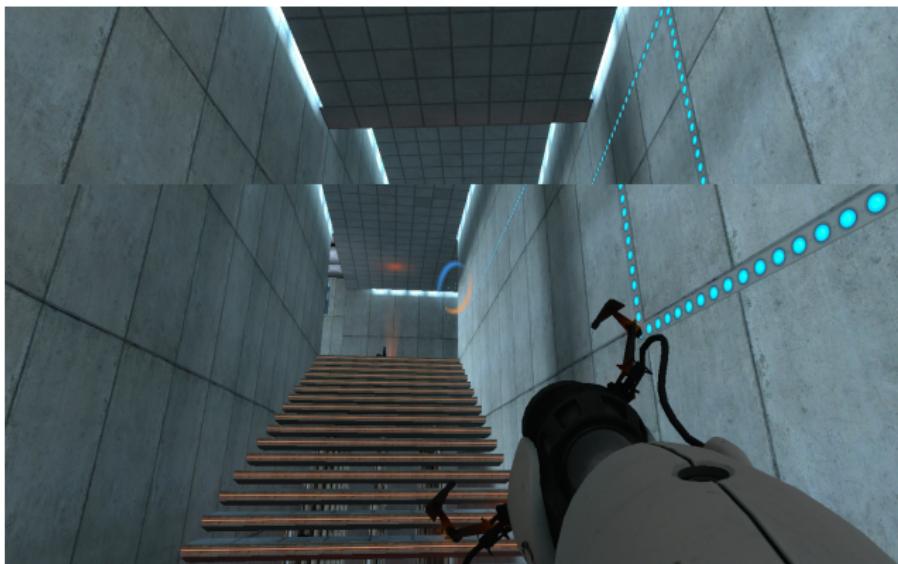
Framebuffer vs. Beamchasing

Реализација	Framebuffer	Beamchasing
Сортирање на нивоу	Пиксела	Примитива
Алгоритам	Лакши	Тежки
Паралелизљивост алгоритма	Већа	Мања
Меморија	Више	Мање
Меморијски bandwidth	Већи	Мањи

Beamchasing јест компликованији за хардвер, али је мањи bandwidth, што је већи проблем у будућности.

Double buffering

- Док се црта у један бафер фрејма, из другог се шаље слика на екран
- Ако је само један бафер, долази до мешања 2 сцене на екрану ткz. Tearing (слика доле)
- Мењање бафера диктира фреквенција освежавања екрана (VSync)



2D Операције

- **Draw Line** - векторска, исцртај линију на екрану (framebuffer-y),
- **Draw Rect** - векторска, исцртај полигон на екрану (framebuffer-y),
- **Copy-Blit** - растерска, копирај слику на екран, пиксел на пиксел.
- **Copy-Stretch** - растерска, различите димензије извора и одредишта, мора да користи интерполацију.

3D Операције

Могуће је цртање:

- тачака,
- линија,
- троуглови.

Главна операција је цртање троуглова. Сваки троугао има 3 вертекса (vertex). Могу бити исцртани као:

- линија,
- једна боја (сва три вертекса иста боја)
- интерполяција боја са сваког од три вертекса,
- мапирана текстура (texture mapping).

- **Rasterization** - израчунава фрагмент (fragment), будући пиксел, на основу троугла и извора светлости, па врши сортирање фрагмената и који је најближи постаје пиксел.
- **Ray-casting** - прати уназад зрак из пиксела, на први троугао који нађе, па се одбија па према извору светлости.

Rasterization vs Ray-casting

Алгоритам	Rasterization	Ray-casting
Сортирање на нивоу	Фрагмената (Пиксела)	Троуглова
Базични алгоритам	Лакши	Тежи
Крајњи алгоритам	Тежи	Лакши
Паралелизљивост алгоритма	Већа	Мања
Реализацију у HW	Лакша	Тежа
Меморија	Више	Мање
Меморијски bandwidth	Већи	Мањи
Сенке	Не	Да
Огледала	Не	Не

Ray-tracing

Слично као и ray-casting само се зрак може одбити више пута.

Резултат: огледало је тривијално реализовати (рефлексија), прозирне материјале исто (рефракција) и још много тога.



Произвођачи

Главни:

- AMD,
- NVidia,
- Imagination Technologies.

Ко још ради:

- Intel,
- Broadcom,
- ARM...

Слојеви у 3D системској подршци

Lib with API

Driver

HW

3D lib API-ји

- OpenGL,
- DirectX,
- Vulkan.

Слојеви у 3D системској подршци 2

Обично произвођачи подржавају све API-је.



Програмабилан део рендерског pipeline-а.

- **Geometry**,
- **Vertex**,
- **Fragment**.

Engine

Engine је софтверски радни оквир (framework) који олакшава развој игрица.

Шта онда ради Engine

- **Мапе** - да припреми мапе,
- **Gameplay** - логика, како се игра игра,
- **AI** - вештачка интелигенција, ботови,
- **Физика** - да срачуна ефекте.

Алгоритам игре

