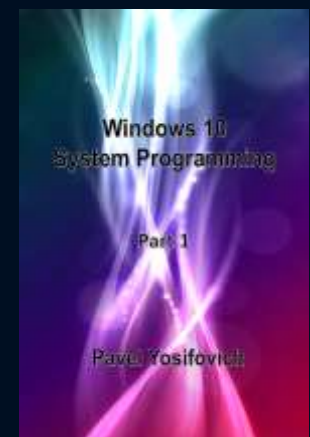
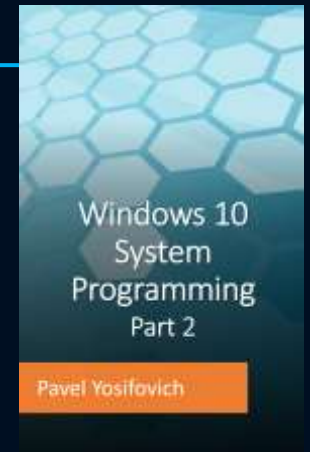
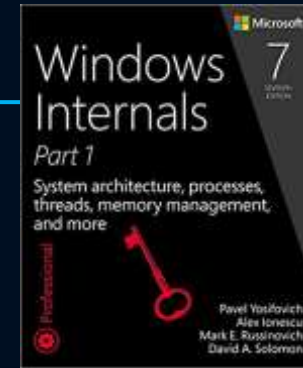


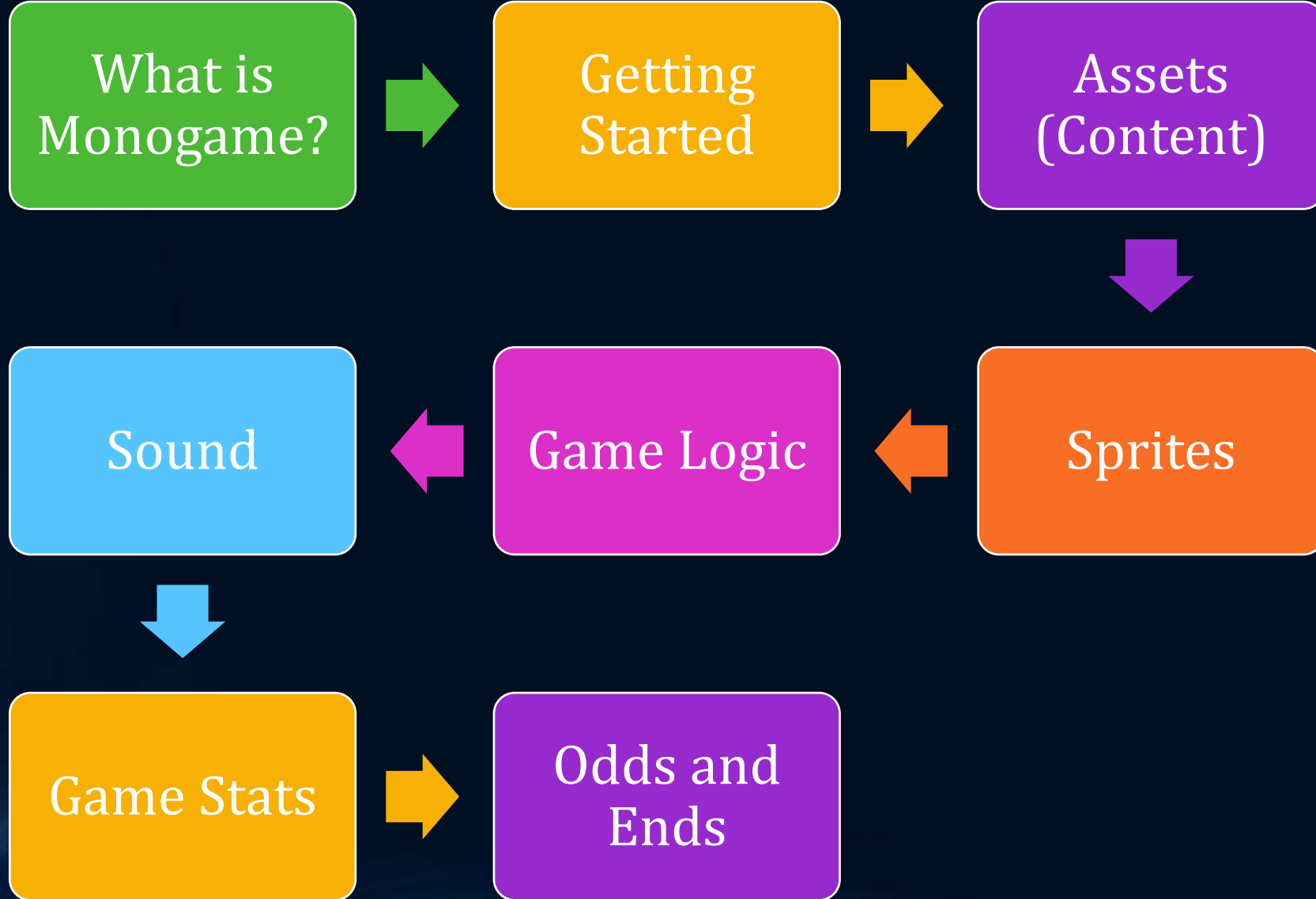
Building a Cross Platform 2D Game with MONOGAME and C#

PAVEL YOSIFOVICH
@ZODIACON

About Me

- **Developer, Trainer, Author and Speaker**
- **Book author**
 - “Windows Kernel Programming” (2019)
 - “Windows Internals 7th edition, Part 1” (co-author, 2017)
 - “Windows 10 System Programming, Part 1” (2020)
 - “Windows 10 System Programming, Part 2” (WIP)
- **Pluralsight and PentesterAcademy course author**
- **Author of several open-source tools**
(<http://github.com/zodiacon>)
- **Website:** <http://scorpiosoftware.net>





What is Monogame?

- Cross platform, .NET-based game engine
- Evolution of the classic XNA Framework
 - Same object model
- Why Monogame?
- Why not Monogame?

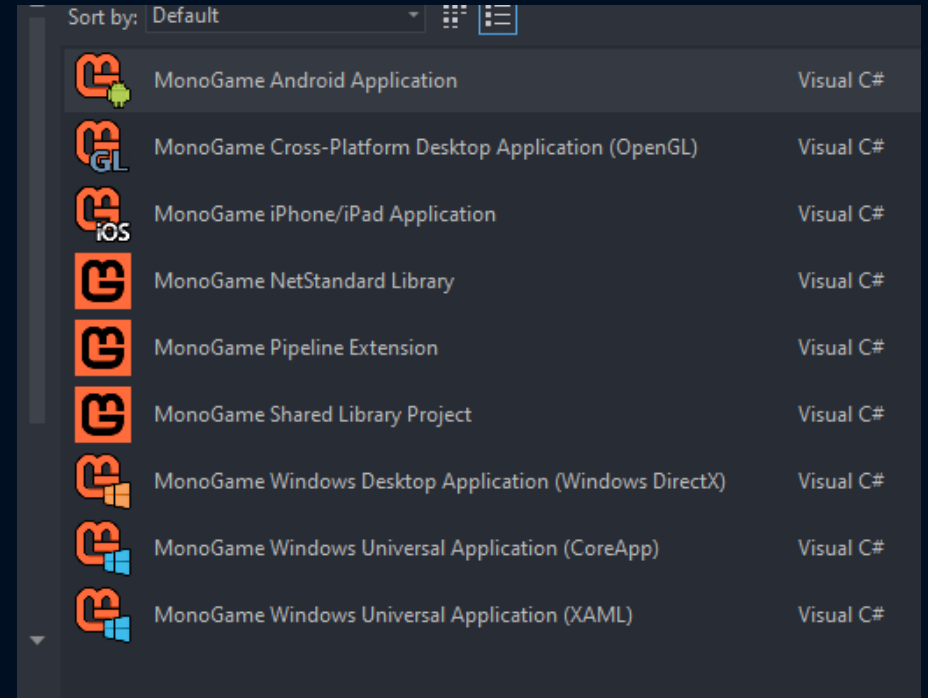
What are we going to Build

- Space.NET Game



Getting Started

- **Install Monogame**
 - Project templates via extension
 - Or use Nuget
- **Create project(s)**
 - One shared/.net standard project
 - Platform-specific project(s)



https://docs.monogame.net/articles/getting_started/0_getting_started.html

Getting Started

DEMO



The Game Loop

- The beating heart of any game
- In Monogame
 - `Game.Update` override
 - Game logic updates
 - `Game.Draw` override
 - Draw scene
 - Repeats based on the selected FPS

Game Assets

- **Media files**
 - Images (textures)
 - Sound and music
 - 3D Models (if using 3D)
- **Fonts**
- **Any other “Content” file (e.g. XML)**
- **Processed with the MonoGame Content Pipeline**
 - Result is *.xnb files

Sprites

- Game objects
 - Usually moving, animating, etc.
- Visuals built from texture objects (sprite sheet)
- Drawn with a SpriteBatch object
- Can customize position, size, rotation angle, tint color

Tint Examples



Sprite Sheets

- **A set of images used for sprite animation**
 - Stored in a single file (sprite sheet)
- **Tools exist for building sprite sheets**
 - Or you can write your own
- **Free sprite sheet packer**
 - <https://www.codeandweb.com/free-sprite-sheet-packer>



Sprites

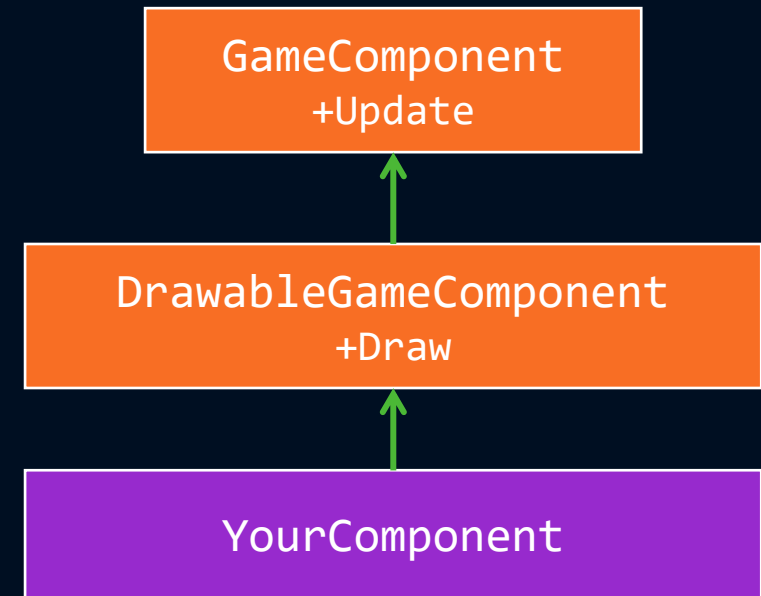
DEMO



MONOGAME

Game Components

- **Game components get calls to Update (and Draw)**
 - `Game.Components` property
 - Best for “top-level” components
- **More overridables**
 - `Initialize`, `LoadContent`, `UnloadContent`



Game Components

DEMO



Collision Detection

- Game objects collisions must be detected and handled
- A general “exact” method of collision detection is too slow to be practical
 - And not usually needed
- Common techniques
 - Bounding box intersection
 - Ellipse/circle intersection



Collision Detection

DEMO

Sound Effects and Music

- A game without sound loses a lot of its appeal
- Background music
 - The `Song` class (typically bound to an MP3 file)
- Sound effects
 - The `SoundEffect` class (usually bound to a WAV file)
 - Sound effect instances (`SoundEffectInstance` class)



Sound Effects & Music

DEMO

Game Stats

- Game information needs to be displayed
 - Level, score, lives, power, ...
- Monogame can use fonts described by a *spritefont* file
 - XML containing font description
 - Processed by the content pipeline
- `SpriteBatch.DrawString` method
- Alternatively, characters can be built as sprites

Game Stats

DEMO



Odds and Ends

- Game title & Game over
- Game design
- Game engine
- Multiplayer games

Resources

- **Monogame home page**
 - <https://www.monogame.net/>
- **Monogame documentation**
 - <https://docs.monogame.net/index.html>
- **This session slides and demos**
 - <https://github.com/zodiacon/DotNextPieter2021>

Thank You!

Q & A