

This is a working document for us to chart our intended learning journeys in our portfolio of 2020 courses. For the RPG syllabus see [here](#).

GameDev.tv 2020 Beginner Course Syllabus

[Read on our forum for more details and discussion.](#)

TABLE OF CONTENTS

[Designer Track \(Ben / Rick\)](#)

[KEYBOARD WARRIOR - Mental & Physical Fitness for the Computer Worker](#)

[Getting Started](#)

[Getting Things Finished](#)

[Sharing What You Finish](#)

[GAME PRODUCER / DESIGNER / MARKETER - How to Design, Tune and Make Games Fun. Also Selling, Marketing, Releasing and Supporting Games](#)

[Starting to design and develop your game - Setup and early development](#)

[Using Agile Sprint Planning](#)

[Developer Track \(Ben / Sam\)](#)

[GAME DEVELOPER - Learn To Code By Making Games in Unity with C#](#)

[Basic Coding](#)

[Coding for 2D Games](#)

[Collaboration](#)

[GAME ENGINE ARCHITECT- Using, Creating and Modifying Unreal](#)

[TECHNICAL ARTIST - Shader Programming & Procedural Generation](#)

[Artist Track \(Ben / Mikey\)](#)

[2D GAME ARTIST - Using Gimp / Photoshop TBC](#)

[3D GAME ARTIST - Using Blender & Unity](#)

[GAME AUDIO MASTER - SFX, Music and Voice](#)

Designer Track (Ben / Rick)

KEYBOARD WARRIOR - Mental & Physical Fitness for the Computer Worker

Getting Started

Getting Things Finished

Sharing What You Finish

GAME PRODUCER / DESIGNER / MARKETER - How to Design, Tune and Make Games Fun. Also Selling, Marketing, Releasing and Supporting Games

1. Users, Ownership Files & Permissions
2. Cloud File Storage Solutions inc Drive

This is a working document for us to chart our intended learning journeys in our portfolio of 2020 courses. For the RPG syllabus see [here](#).

3. Gameplay Screen UI Discussion and Feedback
4. How To Give Feedback To Artists

Starting to design and develop your game - Setup and early development

1. Managing the balance between documenting and doing - what percentage to do on each.
2. How to prototype your game mechanics- what are you trying to achieve and how to create prototypes
3. What high level decisions you need to make before starting to build things
4. When you need to lock down specific design decisions
5. How to create and share your GDD
6. How to give input into the ADD
7. How to brainstorm as a team
8. How to find teammates
9. How and when to use pre-created assets
10. How and when to outsource aspects of development
11. Why your game needs to be more than fun, it needs to be commercially viable
12. How to get early “playtesting” feedback on your game concept
13. Creating an initial Work Breakdown Structure (ie. listing out all known tasks)
14. Creating a simple 1-page business plan
15. Creating high level milestone dates and a solid ship date
16. Competitive analysis and research - what you should be doing and how to use that information
17. How To Set Priorities as a Team¹
18. Who are you making your game for - understanding your target audience and also

Using Agile Sprint Planning

1. An Overview of Agile and Lean
2. Agile Planning Tools Available Today

Developer Track (Ben / Sam)

GAME DEVELOPER - Learn To Code By Making Games in Unity with C#

This course will be an updated version of [the existing Unity Course](#).

Basic Coding

1. **Indentation & spacing:** how the compiler reads it, what it means.
2. Using Enums
3. Using [Compiler Explorer](#) to Explore Optimisation²

¹ Remember the water can go in first, then the big rocks. Then they're in by the end of the week.

² See <https://youtu.be/zBkNBP00wJE?t=2026> and

This is a working document for us to chart our intended learning journeys in our portfolio of 2020 courses. For the RPG syllabus see [here](#).

Coding for 2D Games

1. Introducing Transform, Rotation & Scale (TRS)
2. The Limitations of 32 Bit Floats
 - a. Epsilon
 - b. Range / precision
 - c. Why crash after 7 days?
3. 3D Vectors
 - a. Understanding vector addition and subtraction.
 - b. Planes and projecting onto them.
 - c. `Vector3.Angle()` as intro to rotations and vectors.³
4. 3D Raycasting
 - a. `viewCamera.ScreenPointToRay`
 - b. Layer masks.
5. 3D Rotations
 - a. Non-commutative
 - b. `Quaternion.FromToRotation`
6. Using projectors for decals and 3D cursors.

Collaboration

1. Pros and cons of various SCM.⁴
2. DVCS Workflows
3. Refactoring and Architecture - What and When
- 4.

GAME ENGINE ARCHITECT- Using, Creating and Modifying Unreal

TECHNICAL ARTIST - Shader Programming & Procedural Generation

1. Play with [ShaderToy](#)
2. What is shader code?
3. A colored screen

Artist Track (Ben / Mikey)

2D GAME ARTIST - Using Gimp / Photoshop TBC

1. Overall Project Theme
 - a. Gathering Lots Of Reference Material
 - b. Leave detail alone
2. Colour and Emotion
 - a. Creating a Storyboard
 - b. Matching colours to

³ Figuring out if something has fallen over or is flat is a great usecase for this.

⁴ See our trello evaluation from the RPG <https://trello.com/c/ISGTyQNO>

This is a working document for us to chart our intended learning journeys in our portfolio of 2020 courses. For the RPG syllabus see [here](#).

3. Picking An Art Style
 - a. Take a broad look at various other art styles
 - b. Complementary Art Styles
 - c. Produce 3 Finalists
4. Blocking Out Key Environments
 - a. Setting the Scene
 - b. Designing Your Key Set Items
5. Designing Memorable Characters
 - a. Block Work Sketching
 - b. Movement characteristics
6. UI Designs and UX Impact
7. Sprite Sheets
- 8.

3D GAME ARTIST - Using Blender & Unity

1. The package interface
2. Simple block modelling
 - a. Verti, edge, face
3. Procedural Texture Creation
 - a. Make anything mathematically.
 - b. Make an infinite amount of variations.
 - c. Baking your texture.
4. Simple UV Unwrapping
5. Applying a texture to your model.

GAME AUDIO MASTER - SFX, Music and Voice