This is not even half of all the engines!
This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines

This is not even half of all the engines!
This is not even half of all the engines!
This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines

This is not even half of all the engines!
This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines

This is not even half of all the engines!
Game engines
There is no “best” game engine

As there is no “best” programming language

But some people like to express their opinion loudly!

I consider myself to be quite unbiased. But I’m expressing my opinions based on my own experience. Your experience might differ.
You will learn

1. Fundamental knowledge to use **Unity**, **Godot** and **Unreal Engine 4 GE**.
2. Using these engines to program a game.
3. Architecture patterns of these engines.
4. Comparing these engines for solving a specific problem.
Course Organization

3 EAP = 78h of work

Practices: $16 \times 1.5h = 24h$

Individual work (homeworks): 46h

Test: 8h

Non-differentiable grading:
passed / not passed / not present

Homeworks: 60p
Test: 40p
(at least 30p from homeworks required)

points > 50 = passed

Course information: [https://courses.cs.ut.ee/2020/ge/fall](https://courses.cs.ut.ee/2020/ge/fall)
Course Organization

<table>
<thead>
<tr>
<th>Introduction</th>
<th>Unity</th>
<th>Unity</th>
<th>Unity</th>
<th>Unity</th>
<th>Godot</th>
<th>Godot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Godot</td>
<td>Godot</td>
<td>Unreal</td>
<td>Unreal</td>
<td>Unreal</td>
<td>Unreal</td>
<td>Unreal</td>
</tr>
<tr>
<td>Unreal</td>
<td>Unreal</td>
<td>...</td>
<td>Test</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Course Organization

4 Homeworks (15 points each):

- 1 in Unity
- 1 in Godot
- 2 in Unreal

Test (40p) - questions about:
- Game engine differences
- Game engine architecture patterns
- Game engine specific questions
Course Organization

Discord channel (combined with Game Dev and Design Course)

Link in courses page: https://courses.cs.ut.ee/2020/ge/fall

- Have a game dev discussion with other students
- Ask help
- Help others
Any questions about the course organization?
Early games

Most of the early games were written in Assembly language.

- Hardware specific
- Memory limited

Every byte was counted = very little could be reused

Pitfall: A Classic Game Postmortem
GDC • 45 tuh vaatamist • 4 aasta eest

In this 2011 GDC Postmortem, Pitfall developer David Crane walks attendees through the creation of the classic Atari game. GDC ...

Planet X3 for MS-DOS Development - Part 1
The 8-Bit Guy • 377 tuh vaatamist • 2 aasta eest

Support this channel on Patreon: https://www.patreon.com/8bitguy1 Pixel Artist: Renaud Heidt
http://reno-pixelaru.com/blog/

Subliitrid
Game engine purpose

Game engines provide a suit of **development tools** and **reusable software components**.

Most provide facilities such: graphics, sounds, physics and artificial-intelligence (AI)

- Initially in-house only
- during 90s the 3D first person shooter games (Doom, Quake…) gain popularity →
  - need for licensed game cores →
  - games like Unreal were designed with separated engine and content
About me

Software Engineering Assistant since 2016

Teaching:
- Computer Graphics
- Computer Game Development and Design
- Game Engines (NEW)

Main interests
- Modern technologies
- Procedural generation
- Art creation workflows
- Visual effects
My journey

Middle school and high school → During bachelors

During conscription

During masters → 2018

Now

During bachelors

Game Maker

xna & Allegro

MONOGAME

unity

UNREAL ENGINE

GODOT

Game engine
My journey

Middle school and high school

Language: GameMaker Language (GML)
My journey

Middle school and high school
My journey

Bachelor study years

- Switched to XNA engine (free engine from microsoft)
- Studied OOP programming

```csharp
protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
    spriteBatch.Begin();
    spriteBatch.Draw(background, new Rectangle(0, 0, 800, 480), Color.White);
    spriteBatch.Draw(earth, new Vector2(400, 240), Color.White);
    spriteBatch.Draw(shuttle, new Vector2(450, 240), Color.White);
    spriteBatch.End();
    base.Draw(gameTime);
}
```
My journey

Space game (inspired from Asteroids)
My journey

- Designed to be fully compatible with XNA 4.0
- Designed to be like XNA but not 100%

Gamasutra February 1, 2013
My journey

Web games using Javascript + HTML5 Canvas
My journey

Multiplayer game in C++ (Allegro + Enet)

Development was really slow
My journey

Reimplementation in Monogame with two weeks

This image illustrates an improved state after the reimplementation
My journey

Unity and Computer Graphics during Master’s study
My journey

2018 Unreal Engine - Multiplayer remake of Optimatica
My journey

Now - Godot
Types of game engines

- **Platform specific game engines**
- Game engines with Visual scripting
- Specialized game engines
- Framework like game engines
- Niche game engines

These are not mutually exclusive
Platform specific game engines

Some game engines are designed for making game on a specific platform like:

- Game Console
- Web
- PC
- Mobile
Platform specific game engines

**Web game engines** - usually extend the functionality of HTML5 canvas or WebGL

Game Dev Tycoon - packaged with chromium for PC
Types of game engines

- Platform specific game engines
- **Game engines with Visual scripting**
- Specialized game engines
- Framework like game engines
- Niche game engines

These are not mutually exclusive
Game engines with visual scripting

They simplify game programming by allowing even non-programmers to make games. **Excellent for learning.**
Game engines with visual scripting

Even modern game engines use visual scripting:
Game engines with visual scripting

Games made with GameMaker
Types of game engines

- Platform specific game engines
- Game engines with Visual scripting
- Specialized game engines
- Framework like game engines
- Niche game engines

These are not mutually exclusive
Specialized game engines

1. **RPG Maker** - for Japanese style role playing games.
2. **FPS Creator** - for first person shooter games
3. **Adventure Game Studio** - for point and click adventure games
Specialized game engines

**RPG maker** - for making Japanese style RPG games

Release date: 1992

https://www.thegamecreators.com/product/fps-creator-classic-open-source
Specialized game engines

**FPS Creator** - for first person shooter games, provides library of 800 3D objects

Release date: 2005

https://www.thegamecreators.com/product/fps-creator-classic-open-source
Specialized game engines

Adventure game studio (AGS) - for making point and click adventure games

Release date: 2009
Types of game engines

- Platform specific game engines
- Game engines with Visual scripting
- Specialized game engines
- Framework like game engines
- Niche game engines

These are not mutually exclusive
Framework like game engines

Game engines that only provide a framework of components for things like rendering, audio, collision checking...

- Ogre3D (C++)
- Allegro (C++)
- XNA / Monogame / FNA (C#)
- Lightweight Java Game Library (Java)

Sometimes they are called rendering engines.

Framework like game engines

- XNA / FNA / Monogame
Types of game engines

- Platform specific game engines
- Game engines with Visual scripting
- Specialized game engines
- Framework like game engines
- Niche game engines

These are not mutually exclusive
Types of game engines

Niche game engines

MaxPlay game engines aims to be ‘Google Docs for game devs’

### Most popular game engines today

**What are the best game engines?**

<table>
<thead>
<tr>
<th>Engine</th>
<th>Language</th>
<th>Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Godot</td>
<td>C++, library, C#, GDscript</td>
<td>Windows, OSX, Linux</td>
</tr>
<tr>
<td>Unreal Engine 4</td>
<td>C++, Blueprints (Visual Scripting)</td>
<td>Windows, Mac OS X, Linux</td>
</tr>
<tr>
<td>Unity3d</td>
<td>FREE*</td>
<td>-</td>
</tr>
<tr>
<td>Orx</td>
<td>C, C++, Objective C</td>
<td>-</td>
</tr>
<tr>
<td>libGDX</td>
<td>FREE</td>
<td>Java</td>
</tr>
</tbody>
</table>

Don’t take it as a ground truth, but it gives an indication.
Popular game engines today

GMTK 2020 game jam engines (biggest game jam in Itch.IO hisotry, 5397 entries)
Popular game engines today

GMTK 2020 game jam engines (biggest game jam in Itch.IO history, 5397 entries)
Other game engines to consider

**Pricing** (Paid)
- Has trial version
- One time purchase (each release target separately)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Trial</th>
<th>Desktop</th>
<th>Web</th>
<th>UWP</th>
<th>Mobile</th>
<th>Console</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unlimited Resources</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Expert Features</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Target Platform(s)</td>
<td>Windows TEST only</td>
<td>Windows, Mac, Ubuntu</td>
<td>HTML5</td>
<td>Microsoft UWP</td>
<td>Android, iOS</td>
<td>PS4, Xbox One</td>
</tr>
<tr>
<td>Marketplace</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Support</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>DOWNLOAD</strong></td>
<td>$99.99</td>
<td>$149.99</td>
<td>$399.99</td>
<td>$399.99</td>
<td>Coming Soon</td>
<td></td>
</tr>
</tbody>
</table>
Other game engines to consider

**Pricing** (Free to use)

- Previously pay-what-you want
- Now 5% of revenue

**Pricing** (Free to use)

- Have to use Amazon services (if you need any)
Other game engines to consider

**Pricing** (Free to use)
Steam exclusive (games must be released on Steam)
Other game engines to consider

**Pricing** (Paid)
- Has free edition with restrictions
- Personal licence with subscription

### Comparison Table

<table>
<thead>
<tr>
<th>Feature</th>
<th>Free Edition</th>
<th>Personal Edition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Events</td>
<td>25</td>
<td>∞</td>
</tr>
<tr>
<td>Get ambitious. Make unique games that stand out from the crowd.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum Layers</td>
<td>2</td>
<td>∞</td>
</tr>
<tr>
<td>Add more depth and visuals into your games.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Special Effects</td>
<td>2</td>
<td>∞</td>
</tr>
<tr>
<td>Don't just create games, create gorgeous games.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Multiplayer Games</td>
<td></td>
<td>∞</td>
</tr>
<tr>
<td>Make multiplayer games with our tools and</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Very easy to learn, even for non-programmers
Other game engines to consider

Open source game engines

- MonoGame
- C# Game
- ORX
- libGDX
- Allegro
Unity vs Godot vs Unreal Engine 4

- **Unity**
  - Largest community
  - 2D / 3D
  - C#

- **Godot**
  - Open source
  - 2D / 3D
  - Growing rapidly
  - GDscript / C#

- **Unreal Engine**
  - Based on in-house game development
  - Mostly 3D (Amazing)
  - C++ / Blueprints

**Pricing**
- **Unity** (Free to use)
  - Free personal version up to rev 100k
  - Pro and Plus with subscriptions

- **Godot** (Free to use)
  - If you make money you should support them really

- **Unreal Engine** (Free to use)
  - Completely free until first Million $
  - 5% royalty after that
Next time - introduction to Unity

- Largest community
- 2D / 3D
- C#

Pricing (Free to use)
Free personal version up to rev 100k
Pro and Plus with subscriptions

Let me know:

1. What engine are you most interested in?
2. Is there anything specific you would like to learn?
3. Any suggestions how to make this a better course for you!