DA379A Applied Software Architecture - Project

Game presentations: Monday 2019-05-27 (see Kronox).
Report submission deadline (for peer-review): Friday 2019-05-31 (Canvas)
Peer-review submission deadline: Friday 2019-06-07 (Canvas)

Project (game and report) is to be carried out in groups of 3-5 students.
Peer-reviews are conducted individually

Introduction

Modern computer games are complex, highly optimized applications, and one way to speed up development cycles is via re-use or third-party integration of common tools and systems. Game engines, broadly speaking, are complete frameworks of such tools and systems, intended to unburden developers from as many common tasks as possible while designing the game (at the cost of some control and flexibility).
Your task is simple: take on the role as a small team of indie developers and **make a game in the Unity game engine**.

Project Goals

The intent with this project is for you to experience and apply concepts of software architecture as taught during he course, along the following goals:

- Informal goal
  - Gain experience in making a well-designed game in a modern game engine.
- More formally
  - Implement an application (a game) within a system (Unity) with a given architecture.
  - Identify architectural traits and qualities within said system.
  - Evaluate how these traits and qualities translate into strengths and limitations with respect to the application (your game) - as well as variants of it.
  - Make good design decisions based on the above.
- And finally
  - Consider alternative ways of implementation - both of the application (game), and of the system (game engine) itself.

Scope

The type of game is up to you as long as the game features the following elements:

1. Character & camera controls
   The game camera should be interactive in some way, either directly or via a game character. The game character may be a first-person character, third person character or something totally novel. To make things interesting, we're assuming that the game world is larger than one screen.

2. Collisions and physics
   The game world should contain elements of collisions and/or physics. This may be for instance platforms, hit detection for projectiles, health pickups, floor triggers, moveable objects and so on.

3. Animation
   The game world should also contain elements of animation. In its simplest form this could be for instance a spinning, hovering goal marker, but it is highly encouraged that you explore "real" skeletal animation as well. Unity has tools that makes this process straightforward.

4. Some kind of artificially intelligent entity
   This could be for example a decision state-machine and/or a navigation mesh (see Unity's NavMesh component) for enemy agents. Note that intelligence doesn't necessarily have to be a applied to "hostile" entity - it could for example be a sidekick or an NPC.

5. Some kind of GUI or Heads-Up-Display.
   This doesn't have to be complex, it could be some text displaying the score or a health/progress bar. It should however be connected to the state of the game, not just static text/images.

On top of these elements, the game should be **complete** in that there exists some kind of meaningful gameplay and not just the elements in isolation. The game should be relatively simple, but it must still be a game!

It is important also that you aim for something that you will be able to finish in time. Since Unity is unfamiliar to most, it is recommended that you use an iterative process during development: start with a general idea, make tests, and adjust accordingly. Rinse and repeat.

Resources

Unity has several built-in tutorials and template projects which are excellent to follow in order to get started. There are many more available elsewhere online (too many, so pick wisely).

See Unity's Asset Store for a wide range of free textures, models, sounds etc.

Some assets are bundled in *prefabs* (collections of pre-fabricated components), such as *game character prefabs* containing meshes, colliders, animation controllers and other useful components. You are permitted to use prefabs as long as they not contain entire gameplay elements (which would defeat the idea of the project), and as long as it is clearly stated what they do and where they come from. Ask any of us if you are unsure before using prefabs.

Online resources:
- Unity tutorials and topics
https://unity3d.com/learn/tutorials
- Unity docs
https://docs.unity3d.com/Manual/index.html
- MonoBehavior scripting API (C#)
https://docs.unity3d.com/ScriptReference/MonoBehaviour.html
- Unity Asset Store
https://assetstore.unity.com/
- Sketchfab, free & paid 3D models, with extensive browser preview
https://sketchfab.com
- Adobe Mixamo, compose animated characters in the browser and export to Unity
https://mixamo.com
- Something to put you in the right mood: *Indie Game: The Movie*
https://www.imdb.com/title/tt1942884/

Documentation:

In addition to the programming each group should write a technical report which documents the game, the work process and your experiences with respect to the project goals.

Outside of presenting the game itself and how it works, discuss and motivate you decisions along the way, including any problems and setbacks. Make it clear what your initial goals were, and where you ended up.

List all external assets, prefabs and other resources you may have used, as well as clear references to sources. The report may be written in either English or Swedish.

Higher marks are given based on the extent by which the project goals are adressed. You may for instance analyse critically on how the game engine affects various quality attributes, i.e. its favourable or unfavourable qualities - and why they are favourable or unfavourable. Anther discussion topic is patterns and architectural characteristics of your code and the game engine. You are also encouraged to discuss alternative implementations of the game, and of the engine itself, for instance by making a short survey of another engine such as **Unreal Engine**.

Presentation

Details: see above.
Duration per group: *TBA*.

Each group demonstrates their game to the audience. Discuss not only the end product, but the work process, lessons learned and so on, with the project goals in mind.

Peer-review

Details: see top.

Each person will individually peer-review one project (game and report combined). Review assignments and other details on this will be published after the report submission date.