structure-ca1(s)

course structure The *we create identity* course takes place in the first month, and allows for only a limited number of seesions. A first indication sessions included may look as follows:

course structure

1. introduction of tool(s) and platform(s)
2. topics in *creative technology* (1)
3. teamwork – *identity as a group*
4. topics in *creative technology* (2)
5. *identity* – individual portfolio(s)
6. topics in *creative technology* (3)
7. presentation(s) – individual & group work
8. evaluation – reflection(s) on *we create identity*

Apart from the sessions and viewings indicated above, we plan a number of excursions to places of interest:

excursion(s)

- visit(s) to regional initiative(s) – www.creatievefabriek.nl / www.gogbot.nl
- lecture(s) at PICNIC Academy – www.picnicnetwork.org/search?q_mm=academy
- visit(s) around campus – www.t-xchange.nl

These excursions should provide the students with ideas about what to expect later in the study and motivate them to set their individual targets and aspirations for follow-up courses.

structure-ca3(s)

course structure
Despite the self-organizing nature of the course, a sequence of seesions is needed to guarantuee continuity of work, and to avoid the risk that the *creative aaplication* diverges into one of the many pitfalls that haunt such projects.

session(s)

1. introduction topic(s) & challenge(s)
2. exploration of platform & production requirements
3. planning – concept & application development
4. establishment of business plan
5. contract negotiation & realization
6. basic media and communication theory
7. business and societal context of the creative application(s)
8. delivery and presentation of final application(s)

Dependent on the size of the group, addtional mechanisms of **monitoring progress** may be needed, including logs, periodic accounts, and regular feedback or possibly even **shout-out sessions**.

structure-ca4(s)

course structure
To guarantee continuity and monitor progress we will have a sequence of sessiuons, as outlined below.

session(s)

1. introduction topic(s) & challenge(s)
2. brain storm(s) – concept development
3. planning – concept & application development

4. development scenario(s) and storyboard(s)
5. workflow(s) – asset development
6. basic media and communication theory
7. societal context of creative application(s)
8. delivery and presentation of final application(s)

Dependent on the size of the group, addtional mechanisms of **monitoring progress** may be needed, including logs, periodic accounts, and regular feedback or possibly even **shout-out sessions**.

structure-ca5(s)

course structure
To guarantee continuity and monitor progress we will have a sequence of sessiuons, as outlined below.

session(s)

1. introduction topic(s) & challenge(s)
2. brain storm(s) – concept development
3. planning – concept & application development
4. technical issues – sensor deplyment and data management
5. workflow(s) – configuration(s) and application context
6. theory – communication & logistics
7. societal context of creative application(s)
8. delivery and presentation of final application(s)

Dependent on the size of the group, addtional mechanisms of **monitoring progress** may be needed, including logs, periodic accounts, and regular feedback or possibly even **shout-out sessions**.

structure-ce(s)

course structure
The course does require active participation of the student(s), not only in exploring the technolgy by making the assignments, but also by presenting **solutions and problems** in class.

session(s)

1. introduction
2. background(s): basic exercises
3. exploration(s): architecture & art
4. presentation(s) of concept(s)
5. miscellanous topic(s) in art & technology
6. theoretical reflection(s): art & culture
7. presentation of final assignment(s)

structure-de7(s)

course structure

session(s)

1.

structure-ethic(s)

ethic(s) – payoff(s) / challenge(s) / game(s) / math(s) / plan(s)

course structure
The course does require active participation of the student(s), not only in exploring the technolgy by making the assignments, but also by presenting **solutions and problems** in class.

session(s)

1. introduction – game design & development
2. brain storm(s) – **what is the name of the game?**
3. case study I – applying game design patterns
4. concept presentation(s) – ideas & plan(s)
5. game engine(s) – comparative evaluation(s)
6. case study II – applying game design patterns
7. urban game(s) – interactive (smart) space(s)
8. presentation of final assignment(s)

The *game development* course may considered to be a follow-up on the *virtual environments* course, shifting focus from **representational aspect(s)** to issues of **(narrative-drive) game mechanics** and **game play**. Neverteheless, at some point a choice for one or more technologies will be made, The choice of a **game engine** and **game platform** will be dependent on, among others, available **resources** and students' **(programming) skills**.

structure-iv(s)

course structure
The course does require active participation of the student(s), not only in exploring the technolgy by making the assignments, but also by presenting **solutions and problems** in class.

session(s)

1. introduction of (rich) media platform(s)
2. basic assignment(s) – animation (1)
3. essentials of animation and visualisation
4. basic assignment(s) – visualization (2)
5. principles of interactive information presentation
6. basic assignment(s) – interaction (3)
7. elements of data-driven information systems
8. presentation of final assignment(s)

At this stage it is not clear whether to enforce the use of one particular technology, **flex/as3** or **processing**, to use multiple technologies, or to allow students a choice of technologies, which would then also include **javascript**, **Ch** and **C++**.

structure-math(s)

structure – payoff(s) / challenge(s) / game(s) / math(s) / plan(s)
education(s) / question(s) / moral(s)
reference(s)
norm(s) / utopia / rule(s) / ethic(s) / answer(s) / failure(s)

structure-nm1(s)

course structure – session(s)
Although the course is essentially **project-based**, and to a great extent relies on the students' activity in completing the assignments, there will be a number of lectures, to assist the student in the assignments, and in understanding both the technical and societal context of web applications.

1. introduction of language(s), tool(s) & technology
2. scripting – basic assignment(s)
3. web standard(s) – client-side
4. styling – basic assignment(s)
5. web standard(s) – server-side
6. interaction – basic assignemnt(s)
7. advanced topic(s) – ajax, dhtml, plugin(s), addon(s)
8. presentation of final assignment(s)

The structure presented here is only indicative, and may differs from the actual sequence of topics treated in the lectures. In particular topics such as privacy and security, and societal issues, will be dealt with as items in the lecture, also dependent on actual news items and developments.

structure-nm2(s)

course structure
The course does require active participation of the student(s), not only in exploring the technolgy by making the assignments, but also by presenting **solutions and problems** in class.

session(s)

1. introduction of (rich) media platform(s)
2. basic assignment(s) – animation (1)
3. essentials of animation and visualisation
4. basic assignment(s) – visualization (2)
5. principles of interactive information presentation
6. basic assignment(s) – interaction (3)
7. elements of data-driven information systems
8. presentation of final assignment(s)

At this stage it is not clear whether to enforce the use of one particular technology, **flex/as3** or **processing**, to use multiple technologies, or to allow students a choice of technologies, which would then also include **javascript**, **Ch** and **C++**.

structure-nm3(s)

course structure
The course does require active participation of the student(s), not only in exploring the technolgy by making the assignments, but also by presenting **solutions and problems** in class.

session(s)

1. introduction of mashup technologies
2. basic assignment(s) – map(s), flow(s) & diagram(s)
3. server technologies – the (W)AMP stack
4. technical issues – services & protocols
5. student presentation of concept final assignment(s)
6. services – flickr, amazon, google, etc.

7. interaction – model(s) of immersion
8. presentation of final assignment(s)

The course will take a **technology-agnostic** approach, favoring concept(s) over implementation(s) or language(s). Neverteheless, most of the examples presented are based on flex/as3.

structure-nm4(s)

course structure
The course does require active participation of the student(s), not only in exploring the technolgy by making the assignments, but also by presenting **solutions and problems** in class.

session(s)

1. introduction of VR applications
2. basic assignment(s) – product demo(s)
3. VR technologies – input(s) & display(s)
4. experience design – user interface issues
5. student presentation of concept final assignment(s)
6. VR cases studies – visit to (TXChange)
7. mixed/augmented reality in game(s)
8. presentation of final assignment(s)

To the extent possible, the course will take a technology-agnostic approach, favoring **concept(s)** over implementation(s) or language(s). Neverteheless, at some point a choice for one or more technologies will be made, dependent on available **resources**, as well as **(programming) skills** of the students.

structure-nm5(s)

course structure
The course does require active participation of the student(s), not only in exploring the technolgy by making the assignments, but also by presenting **solutions and problems** in class.

session(s)

1. introduction – game design & development
2. brain storm(s) – **what is the name of the game?**
3. case study I – applying game design patterns
4. concept presentation(s) – ideas & plan(s)
5. game engine(s) – comparative evaluation(s)
6. case study II – applying game design patterns
7. urban game(s) – interactive (smart) space(s)
8. presentation of final assignment(s)

The *game development* course may considered to be a follow-up on the *virtual environments* course, shifting focus from **representational aspect(s)** to issues of **(narrative-drive) game mechanics** and **game play**. Neverteheless, at some point a choice for one or more technologies will be made, The choice of a **game engine** and **game platform** will be dependent on, among others, available **resources** and students' **(programming) skills**.

structure-sg(s)

course structure
The course does require active participation of the student(s), not only in exploring the technolgy by making the assignments, but also by presenting **solutions and problems** in class.

session(s)

1. introduction – game design & development
2. brain storm(s) – **what is the name of the game?**
3. case study I – applying game design patterns
4. concept presentation(s) – ideas & plan(s)
5. game engine(s) – comparative evaluation(s)
6. case study II – applying game design patterns
7. urban game(s) – interactive (smart) space(s)
8. presentation of final assignment(s)

The *game development* course may considered to be a follow-up on the *virtual environments* course, shifting focus from **representational aspect(s)** to issues of **(narrative-drive) game mechanics** and **game play**. Neverteheless, at some point a choice for one or more technologies will be made, The choice of a **game engine** and **game platform** will be dependent on, among others, available **resources** and students' **(programming) skills**.