schedule(s) – common(s) / register! / request for grading

participant(s) / lab(s) / assignment(s) / CA3

1. introduction – topic(s) & challenge(s)
2. exploration(s) – platform & production requirements
3. planning – concept & application development
4. concept(s) – mechanics, story, aesthetics, technology
5. consideration(s) – infrastructure & realization
6. background(s) – basic media and communication theory
7. context(s) – creative application(s)
8. production(s) – delivery and presentation of final application(s)

(*) no lecture at 1/3 and 28/6 (exam math), otherwise (in principle) every monday

co-instructor(s):

- Dhaval Vyas (...) – wwwhome.cs.utwente.nl/∼vyasdm
- artist(s) in residence – tentative
  - Victor Onstein – color(s)

‘

basic exercise(s) – learn your skill(s)

basic exercise(s) / CA3

- mini game(s) – in unity

criteria for grading: basic technical skills, hygiene of code, adequacy of solution(s) & overall design.

final application(s) – be creative ...

final application(s) / CA3

- **interactive space(s)** – critical game(s) & installation(s) / CTSG

criteria for grading: originality & creativity, technical & design challenge(s), overall development skill(s).

essay(s) – reflection(s) on ...

www.writingstudio.eu / tip(s) / how to write an essay? / CA3

- interactive space(s) – design, development, technology
- game design – aspirations & responsibilities
- innovation(s) – societal problems & (technological) solution(s)

criteria for grading: clarity of exposition, understanding of technology & context(s), originality of argument(s).

comment(s) & feedback: oral and/or written, (partly) based on **student presentation(s)** in class and online portfolio(s). Student **peer review(s)** may provide additional feedback. but will play no dominant role in grading.

deliverable(s) – have fun and play!

document(s) / scenario(s) / format(s)

1. concept(s) – (short) synopsis, with (optional) sketches
2. requirement(s) – with shareholders, planning, MOSCOW
3. story board – storyline(s), non-linear storygraph, assets
4. prototype(s) – partial version(s) of interactive application(s)
5. final application – full interactive application
6. accompanying website – with application and support
7. promotional clip – one/two minute trailer
8. justification – explanation of design decisions, reflection(s)
9. package – all the material with documentation

session(s): have fun and play! CA3: 1

- prepare – lab(s) / game(s)
- introducing – abbreviation(s)
  - project – cooperative artefact memory
- challenge(s) – CTSG
  - interactive space(s) – story & task(s)
- disclaimer(s) – have fun and play = work for you!

lens 02 – surprise

_ .. []: << >> / _ / .

... remind yourself to fill your game with interesting surprises:

- what will surprise players when they play my game?
- does the story in my game have surprises?
- do your rules give players ways to surprise eachother?
- do your rules give players ways to surprise themselves?

Surprise is a crucial part of entertainment – it is at the root of humor, strategy and problem solving. Our brains are hardwired to enjoy surprises.

play / social(s) / machine(s) / method(s) / cycle(s)

lens 03 – fun

_ .. []: << >> / _ / .

... fun is desirable in almost every game, although sometimes fun defies analysis.
To maximize fun ask yourself the these questions:

- what parts of my game are fun?
- why?
- what parts need more fun?

play / social(s) / machine(s) / method(s) / cycle(s)

lens 04 – curiosity

_ .. []: << >> / _ / .

... think about the player's true motivations – not just the goals (y)our game has set forth, but the reasons the player wants to achieve those goals:

- what goals does my game put into the player's mind?
- what am I doing to make them care about these questions?
- what can I do to make them invent even more questions?

For example, a maze-finding videogame might have a time-limit goal at each level. A way to make players care more is to play interesting animations when they solve each maze ...

play / social(s) / machine(s) / method(s) / cycle(s)

lens 06 – problem solving

..., think about the problems (y)our player must solve to succeed at (y)our game, for every game has problems to solve:

- what problems does my game ask the player to solve?
- are there hidden problems to solve that arise as part of gameplay?
- how can my game generate new problems so that players keep coming back?

lens 07 – mechanics / story / aesthetics / technology

... take stock of what element(s) (y)our game is truly made of:

- is my game design using elements of all types?
- could my design be improved by enhancing any element?
- are the elements in harmony, reinforcing eachother and working together towards a common theme?

Together, the **elements** are also referred to as the **elemental tetrad**.