Advanced Graphics and Interaction 2015: Lecture 10

Mario Romero 2015/10/05

KTH VETENSKAP OCH KONST

VICSTHLM
VISUALISATION INTERACTION COLLABORATION

AGI15 Calendar: link

- Mon 31 aug 15:00-17:00 — Lecture 1 – Introduction
- Tue 1 sep 13:00-17:00 — Lecture 2-3: Forming Groups and Brainstorming
- Mon 7 sep 15:00-17:00 — Lecture 4: Proposals
- Thu 10 sep 10:00-12:00 — Lecture 5: Discussion based on Proposals
- Mon 14 sep 15:00-17:00 — Lecture 6: Hello World Demos
- Thu 17 sep 10:00-12:00 — Lecture 7: Discussion based on the Hello World Demos
- Tue 22 sep 10:00-12:00 — Lecture 8: Preparing ForskarFredag 2015
- Fri 25 sep 8:00-16:00 — ForskarFredag
- Mon 28 sep 15:00-17:00 — Lecture 9: Reflecting on ForskarFredag
- Mon 5 oct 15:00-17:00 — Lecture 10: Agile Development 1 towards Comic Con - Gamex 2015
- Mon 12 oct 15:00-17:00 — Lecture 11: Agile Development 2 towards Comic Con - Gamex 2015
- Fri 30 oct 9:00 – Sun 1 Nov 16:00 — Comic Con Gamex
- Mon 2 nov 15:00-17:00 — Lecture 12: Reflecting on Comic Con Gamex
- Tue 3 nov 13:00-17:00 — Lecture 13-14: Forming new groups and brainstorming project 2
- Tue 10 nov 10:00-12:00 — Lecture 15: Proposals Project 2
- Tue 17 nov 10:00-12:00 — Lecture 16: Hello World Demos for Project 2
- Tue 24 nov 10:00-12:00 — Lecture 17: Agile Development 1 for Open House
- Tue 1 dec 10:00-12:00 — Lecture 18: Agile Development 2 for Open House
- Fri 4 dec 15:00-19:00 — VIC AGI15 Open House

Agenda

1. Hopscotch demos
2. Deliverable Oct 23
3. Results from Assignment 2
4. Logos and Posters
5. Next phase development (quickly fix interaction problems and focus on advancing the graphics)
    1. Special Effects
        1. Particle Systems
    2. Rendering quality
        1. Un-Aliasing
        2. Shadows
        3. Reflections
        4. Refractions
6. Focus on your own goals
    1. Make feature that is:
        1. Non critical
        2. Independent
        3. Focused on your learning goal
        4. Easily integrated
7. FB questions
8. Next Lecture
9. Schedule individual group meeting

Hopscotch

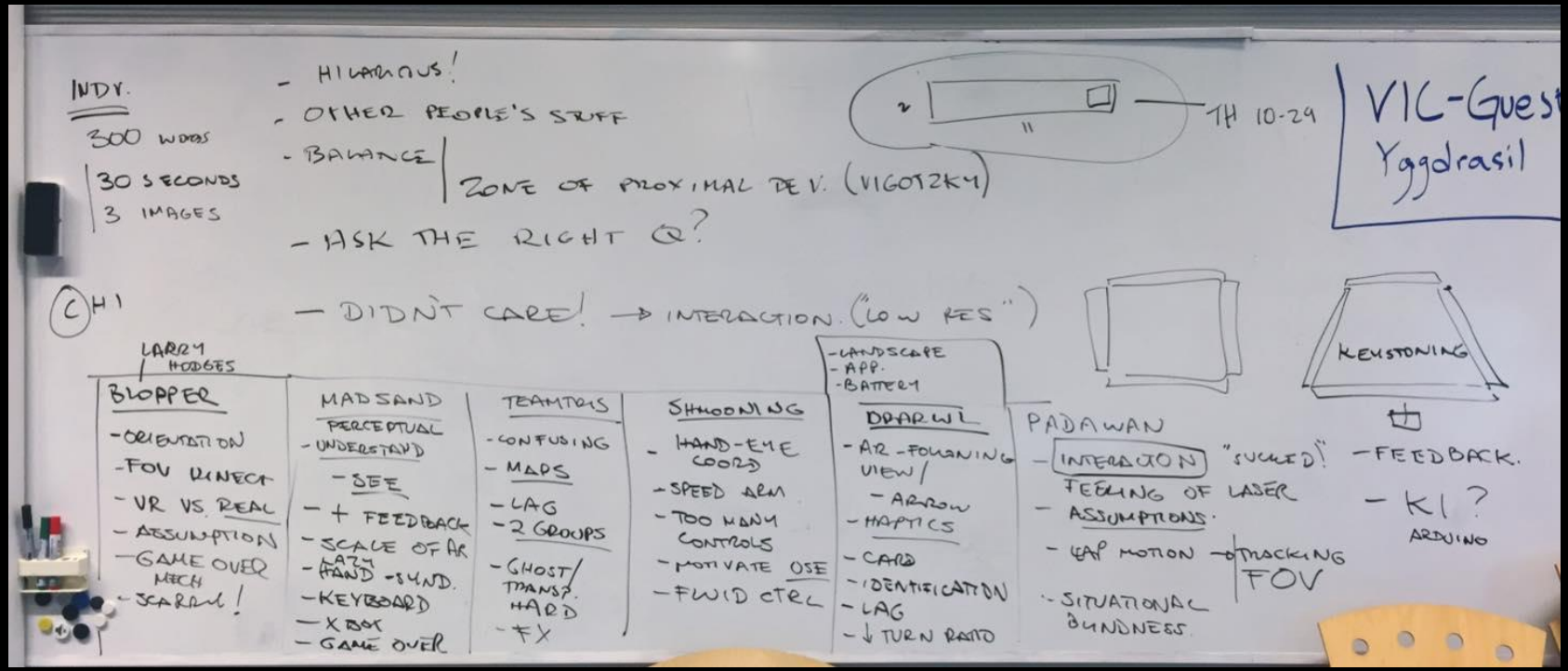
Is this AGI?

- Why?
  - Flexibility
  - Freedom of creation
  - Interaction can be gesture- and sound-based, but not experimental enough to have to deal with noise (like you do in your projects)
- Why not?
  - Fixed (limited to certain controls and graphics)
  - No cool shaders
  - Plugins?
  - Limited ability to mathematically model simulations of light interacting with complex objects (this is the core of modern advanced computer graphics).

Remember: Deliverable Oct 23

- Webpage with:
  - Description
    - Goal and motivation of the project
    - Explanation and Justification of the graphics and interaction technologies used and developed
    - Challenges
    - Obstacles
    - Related work
    - Lessons learned
  - Look at AGI14 web pages
  - Photos
  - "Making of" documentary (2 minutes)
  - Demo Reel (30 seconds)
  - Optional PR material (logo, trailer, flyers, posters, catalog)
  - User testimonials (what did people say)

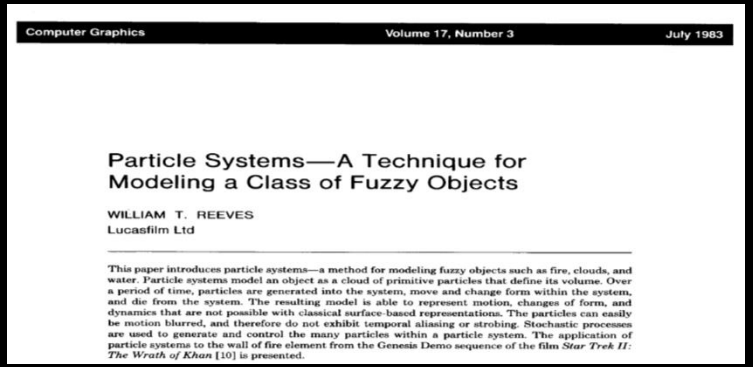
Remember


INDY.
300 WORDS
30 SECONDS
3 IMAGES
- HILARIOUS!
- OTHER PEOPLE'S STUFF
- BALANCE
ZONE OF PROXIMAL DEV. (VIGOTZKY)
- ASK THE RIGHT Q?
TH 10-29
VIC-Guest
Yggdrasil
CHI
- DIDN'T CARE! → INTERACTION ("LOW RES")
KEYSTONING
LARRY HODGES
BLOPPER
-ORIENTATION
-FOV KINECT
- VR VS. REAL
- ASSUMPTION
-GAME OVER MECH
- SCARY!
MADSAND
PERCEPTUAL
- UNDERSTAND
- SEE
- + FEEDBACK
- SCALE OF AR
- LAZY HAND - HAND.
-KEYBOARD
-XBOX
- GAME OVER
TRAMTRAS
-CONFUSING
- MAPS
- LAG
-2 GROUPS
-GHOST/TRANSP. HAND
- FX
SHMOOING
- HAND-EYE COORD
- SPEED ARM
- TOO MANY CONTROLS
- MOTIVATE USE
- FLUID CTRL
-LANDSCAPE
- APP.
-BATTERY
DRAWL
- AR - FOLLOWING VIEW
- ARROW
- HAPTICS
- CARD
- IDENTIFICATION
- LAG
- ↓ TURN RATIO
PADAWAN
- INTERACTION "SUCKED!"
FEELING OF LASER
- ASSUMPTIONS.
- LEAP MOTION - TRACKING FOV
- SITUATIONAL BUNDNESS
-FEEDBACK.
- KI?
ARDUINO

For example… particle systems

Particle Systems

- Original Paper
  - Particle Systems A Technique for Modeling a Class of Fuzzy Objects
  - ACM SIGGRAPH 1983
  - William Reeves
  - Let's read the paper



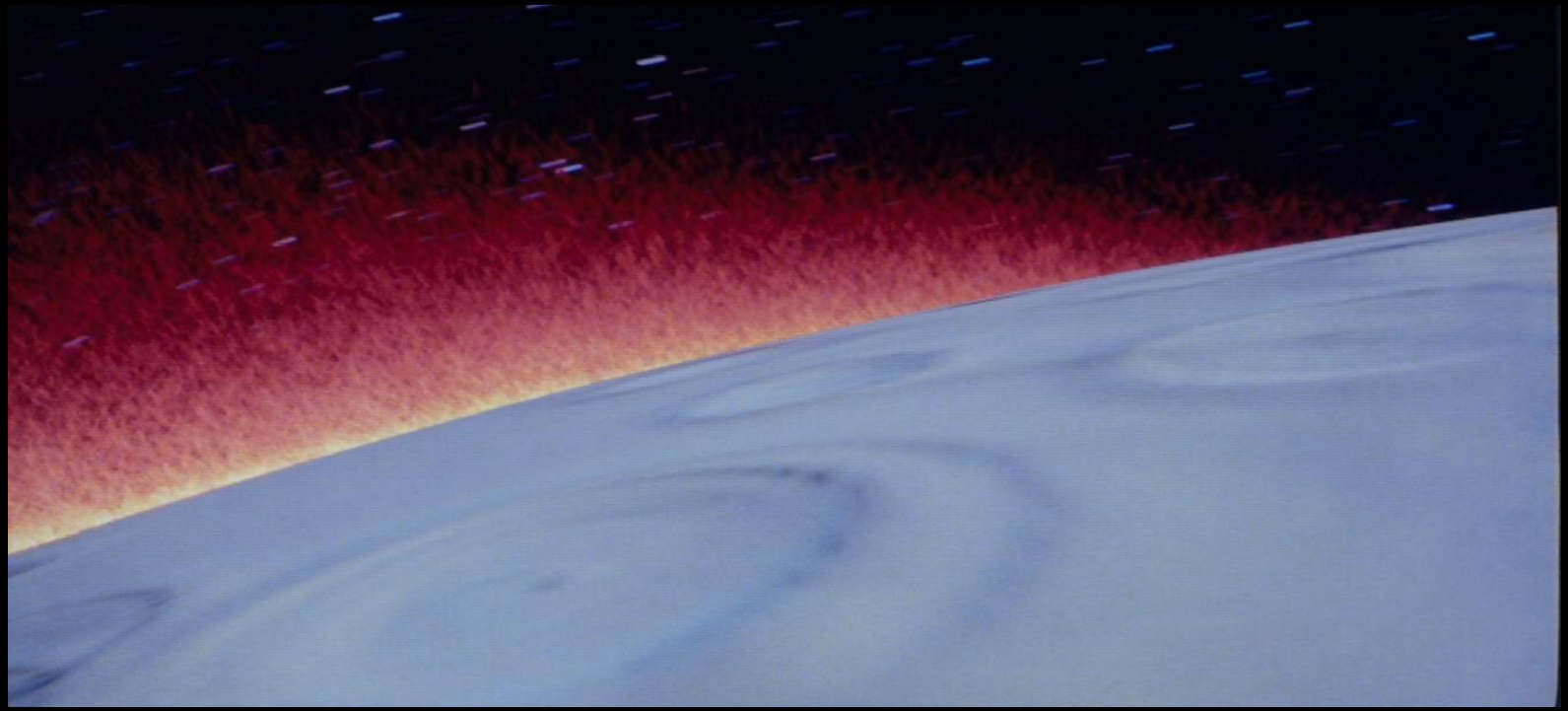
Computer Graphics Volume 17, Number 3 July 1983

Particle Systems—A Technique for Modeling a Class of Fuzzy Objects

WILLIAM T. REEVES
Lucasfilm Ltd

This paper introduces particle systems—a method for modeling fuzzy objects such as fire, clouds, and water. Particle systems model an object as a cloud of primitive particles that define its volume. Over a period of time, particles are generated into the system, move and change form within the system, and die from the system. The resulting model is able to represent motion, changes of form, and dynamics that are not possible with classical surface-based representations. The particles can easily be motion blurred, and therefore do not exhibit temporal aliasing or strobing. Stochastic processes are used to generate and control the many particles within a particle system. The application of particle systems to the wall of fire element from the Genesis Demo sequence of the film *Star Trek II: The Wrath of Khan* [10] is presented.

Demo



Paper's Abstract

- Model fuzzy objects
  - Fire
  - Clouds
  - Water
- Particles = primitives of a volume
- Shape of volume is non-deterministic (stochastic)
- In time:
  - Generated
  - Move
  - Change
  - Die

Model (Algorithm)

1. Generate new particles
2. For all particles
    1. Assign individual attributes
    2. If too old, kill
    3. Else, transform = f(**dynamics**)
3. Render image of living particles

Exercise

- Simple Particle System
  - Link
- Understand Code
- Modify to
  1. Simplify to one particle with simple trajectory
  2. Add physics like wind and gravity

Example of particle systems 2015

video

LazyFluids: Appearance Transfer for Fluid Animations

Ondřej Jamriška[1*] Jakub Fišer[1] Paul Asente[2] Jingwan Lu[2] Eli Shechtman[2] Daniel Sýkora[1]
[1]CTU in Prague, FEE [2]Adobe Research

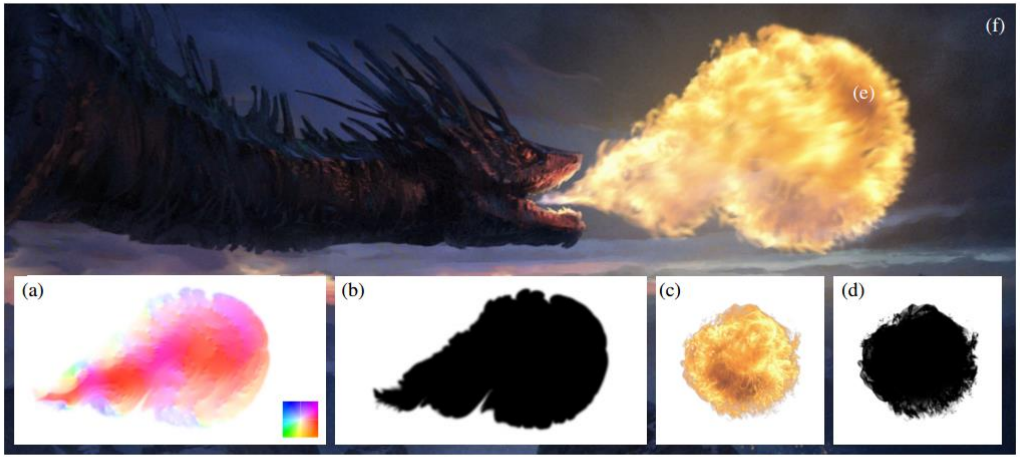


Figure 1: *LazyFluids in action—an artist first designs a target fluid animation that consists of a sequence of motion fields (a) and alpha masks (b), and then selects a video exemplar of a fluid element with desired appearance (c) and alpha mask (d). Finally, our algorithm transfers appearance of the exemplar to the target animation while respecting its motion properties and boundary effects (e). The resulting animation can then be used as a part of a more complex composition (f). All alpha masks in the paper are visualised in a way that fully opaque pixels are black and fully transparent are white. Dragon painting © Jakub Javora.*

Task

1. Next phase development (quickly fix interaction problems and focus on advancing the graphics)
    1. Special Effects
        1. Particle Systems
        2. Mathematically model simulations of light interacting with complex objects …
    2. Rendering quality
        1. Un-Aliasing
        2. Shadows
        3. Reflections
        4. Refractions
2. Focus on your own goals
    1. Make feature that is:
        1. Non-critical
        2. Independent
        3. Focused on your learning goal
        4. Easily integrated

FB questions

What to deliver?

- Freedom to fail
- Show work
  - Research
  - Paper reading and understanding
  - Development
  - Integration
- Write a short individual report (?)
  - At most
    - 1000 words
    - 3 references
    - 3 figures
    - 1 video-figure
- If working, working code

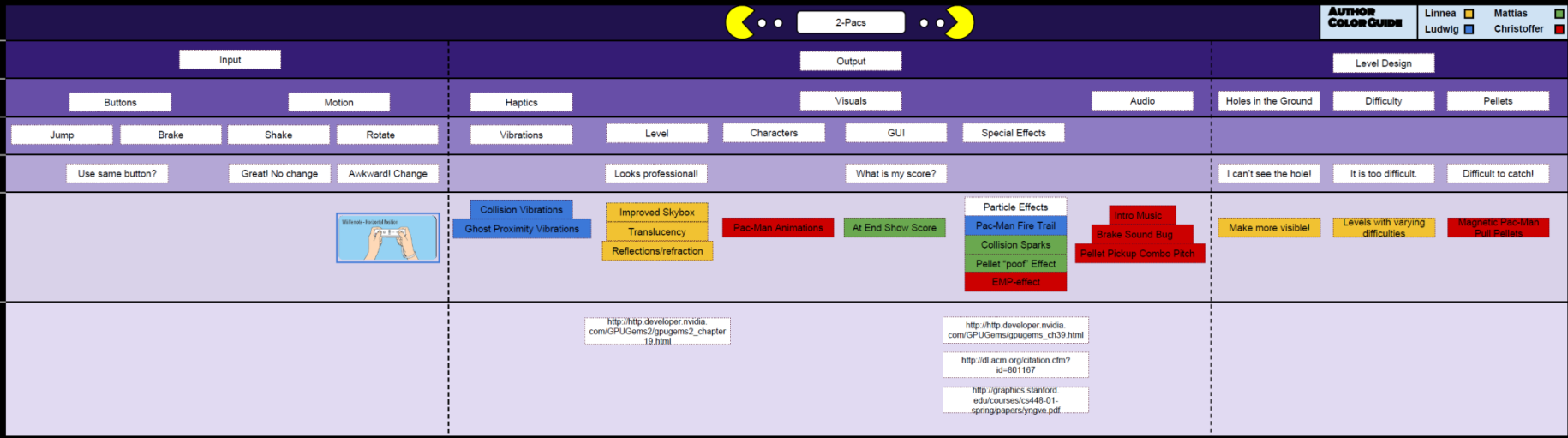
Ahead of you

- Phase 1 – fix interaction issues and work on core graphics together
- Phase 2 – research and propose a feature that works independently from the rest of the project and implement a simple version of it.

Example work plans from AGI14

- The following are plans made between ForskarFredag and Comic Con…

2Pacs


Awesomeness
2-Pacs
Author Color Guide
Linnea
Mattias
Ludwig
Christoffer
General Categories
Input
Output
Level Design
Specific Categories
Buttons
Motion
Haptics
Visuals
Audio
Holes in the Ground
Difficulty
Pellets
Subcategories
Jump
Brake
Shake
Rotate
Vibrations
Level
Characters
GUI
Special Effects
Feedback
Use same button?
Great! No change
Awkward! Change
Looks professional!
What is my score?
I can't see the hole!
It is too difficult.
Difficult to catch!
Features
Collision Vibrations
Ghost Proximity Vibrations
Improved Skybox
Translucency
Reflections/refraction
Pac-Man Animations
At End Show Score
Particle Effects
Pac-Man Fire Trail
Collision Sparks
Pellet "poof" Effect
EMP-effect
Intro Music
Brake Sound Bug
Pellet Pickup Combo Pitch
Make more visible!
Levels with varying difficulties
Magnetic Pac-Man Pull Pellets
Related Work
http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter19.html
http://http.developer.nvidia.com/GPUGems/gpugems_ch39.html
http://dl.acm.org/citation.cfm?id=801167
http://graphics.stanford.edu/courses/cs448-01-spring/papers/yngve.pdf

YA3

Gameplay

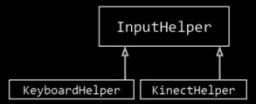
- Gameplay encapsulates tree pipeline
- search-for-player loop
- Cinematic tour of the forest while waiting

Tree Pipeline

Input Interface | Tree Data | Tree Data Simulation Data | Shader Data

Input → **Tree Generation** → **Simulation** → **Tree Construction** → **Rendering**

- **Input separated from Interaction**
- Abstract Input Interface



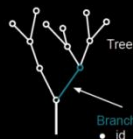
Interface:

- HandShoulderAngle()
- HandShoulderDist()
- HandShoulderDist3D()
- ElbowShoulderAngle()
- ElbowShoulderDist()
- ElbowHandAngle()
- ElbowHandDist()
- HandTorsoDist()
- ElbowTorsoDist()
- TorsoTiltAngle()
- ...

Logging, Debugging:

- log Actions
- replay sessions
- use debugging view
- behavior statistics

- Interaction is handled here
- *Abstract representation of the tree*



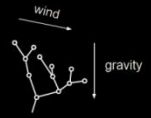
Branch Data:

- id
- parent
- children[]
- position, rotation, scale
- age, depth
- leaves, flowers, lateral buds

- **not rendering specific!**
  - no vertices, faces, normals
  - no colours
  - no shaders

- Apply physics simulations
- Store in separate SimulationData structure.

- => Wind, Gravity …
- Also other systems:
  - Leaves in air
  - Particle Systems
  - Day-night cycle



- Creates full tree model
- Ready for rendering
- Data to GPU!

- Tree ready to be rendered

- Deferred Rendering Pipeline
  - First pass: *to gBuffer* (positions, normals, materials..)
  - Second pass: fragment shader using data in gBuffer

- **NPR** - Non Photorealistic Rendering

- Lots of things to add:
  - DoF
  - Motion blur
  - Anti-aliasing
  - Volumetric light scattering
  - Bloom effects
  - Godrays
  - …

Pod Racer

- Focus on the graphics
  - Fire
  - Electricity
  - Dust
  - Pseudo-realism
  - Advanced effects
  - Avatar with arms controlling the pod
  - Visualization of engine thrust
  - Map?
  - Virtual arrows?
- Sound
  - Location?
  - Special effects
  - Sound track
    - Expert: Roberto Bresin
  - You may justify spending time on good sound by calling it ”advanced interaction” but I will not force you to do it.
- Change track
- In-game tutorial

Space Survival

- Controls
  - Hard
  - Confusing
  - Vertical thrusters
- In-game tutorial
- Self-localization
  - Map?
  - Virtual Arrows?
  - On avatar's body?!
- Space dust – speed and direction
- Look at 3D movie creation – layering
- Sound in helmet
  - Sonification
  - Early warning
  - Communication with ground control, mission command, or space craft

Next week

- Planning Comic Con Gamex (45 minutes)
  - Logistics
  - Schedules
  - Space
  - Furniture
  - Posters
- Presenting updates on projects (45 minutes)
  - Group Goals
  - Individual Goals
  - Schedule
- Set up individual meeting with me

Thank you!

marior@kth.se