Graphics Programming

Module description

This module is aimed at programmers that have learned the basics of programming using the p5.js programming frameworks through Introduction to Programming part one.

Learners will begin by looking into building their own physics engine for animating elements on the screen, using vectors and forces. Having learned the basics of such a system they will then look into an off-the-shelf physics library. After that, learners will look into the principles of procedural generation of graphics and will hear first hand accounts from a game designer and a computational artist.

In the second part of the module attention is shifted to 3D graphics at first before looking deeper into the basic elements of images, namely pixels. Once learners have grasped the basics of pixel reading and manipulation they will use these techniques in combination with others to programme computer vision applications and image processing filters. The second part of the module finishes with an interview with an accomplished artist-hacker who uses computer vision techniques for a lot of his work.

Learners get to practice all of the above in multiple, weekly, fun, programming exercises and their knowledge is tested through engaging programming assignments and quizzes.

Module goals and objectives

Upon successful completion of this module, you will be able to:

1. Manipulate numerical representations of visual media.
2. Explain the basic mathematical and theoretical principles of visual computing.
3. Write basic but complete graphics software systems
4. Apply filters on images in order to transform them for processing in computer vision scenarios.
5. Explain the fundamental tenets of computer vision and write code that processes images to make intelligent assertions about their content.
6. Select and use software libraries for software development.

Textbook and Readings

Specific essential readings for each week from the following list are included in the Readings page for each week:

During each week your instructor will post links of interest and things he would like you to explorer. There is no official textbook or readings as most of the module is very practical and video-based.

Module outline

The module consists of twenty weeks that focus on key areas of Graphics Programming.

Week 1	**Key concepts:** 2D transformations **Learning outcomes:** Explain how transformations work. Describe how classes work. Use transformations to program a basic solar system.
Week 2	**Key concepts:** Vectors, vector arithmetic, acceleration in movement **Learning outcomes:** Describe how vectors work. Apply vector arithmetic. Implement simple systems that use vectors.
Week 3	**Key concepts:**

	Forces, gravity, friction, particles, collision detection **Learning outcomes:** Explain how forces work. Use physics concepts in animation scenarios. Implement simple physics systems.
Week 4	**Key concepts:** physics engines, matter.js, bodies, constraints, composites **Learning outcomes:** Describe what physics engines are and what they do. Describe the basic elements of matter.js Implement simple physics systems using matter.js
Week 5	**Key concepts:** physics engines, matter.js, bodies, constraints, composites **Learning outcomes:** Describe what physics engines are and what they do. Describe the basic elements of matter.js Implement simple physics systems using matter.js
Week 6	**Key concepts:** generative art, randomness, noise

	Learning outcomes: Explain what generative art and design is. Identify important characteristics of generative art. Apply randomness and noise to create simple generative systems
Week 7	**Key concepts:** generative art, randomness, noise **Learning outcomes:** Explain what generative art and design is. Identify important characteristics of generative art. Apply randomness and noise to create simple generative systems
Week 8	**Key concepts:** generative art and design, polar coordinates, oscillation, additive synthesis, fractals, recursion **Learning outcomes:** Use trigonometry to make shapes. Use oscillation to code movement. Implement generative systems using additive synthesis. Implement recursive systems to make fractals.
Week 9	**Key concepts:** generative art and design, polar coordinates, oscillation, additive synthesis, fractals, recursion

	Learning outcomes: Use trigonometry to make shapes. Use oscillation to code movement. Implement generative systems using additive synthesis. Implement recursive systems to make fractals.
Week 10	**Key concepts:** procedural generation in games, generative art and design, fabrication, Andy Lomas, Alan Zucconi **Learning outcomes:** Explain what procedural generation for games is and what the pros and cons are. Identify ways Andy Lomas' work is linked to the syllabus.
Week 11	**Key concepts:** 3D renderer, lights, material, texture, camera **Learning outcomes:** Explain the difference between 2D and 3D graphics. Use 3D primitives, lights and materials to program simple animations. Implement 3D animations by manipulating the camera parameters
Week 12	**Key concepts:** 3D renderer, lights, material, texture, camera

	Learning outcomes: Explain the difference between 2D and 3D graphics. Use 3D primitives, lights and materials to program simple animations. Implement 3D animations by manipulating the camera parameters
Week 13	**Key concepts:** colors, color theory, images, webcam, pixels **Learning outcomes:** Explain the difference between bitmap and vector graphics. Use principles of color theory to generate color procedurally. Use images from the hard disk or from the webcam. Accessing and manipulating pixels directly.
Week 14	**Key concepts:** colors, color theory, images, webcam, pixels **Learning outcomes:** Explain the difference between bitmap and vector graphics. Use principles of color theory to generate color procedurally. Use images from the hard disk or from the webcam. Accessing and manipulating pixels directly.
Week 15	**Key concepts:** image processing, filters, convolution

	Learning outcomes: Explain what image processing is and what its applications are. Use simple to manipulate images. Use convolution to apply advanced filters on images. Implement an instagram-type image processing filter.
Week 16	**Key concepts:** image processing, filters, convolution **Learning outcomes:** Explain what image processing is and what its applications are. Use simple to manipulate images. Use convolution to apply advanced filters on images. Implement an instagram-type image processing filter.
Week 17	**Key concepts:** computer vision, color tracking, background subtraction, frame differencing, optical flow, face tracking **Learning outcomes:** Explain what computer, its applications and challenges are. Describe what color tracking, background subtraction is. Use frame differencing to track movement. Explore advanced techniques such as optical flow and face tracking. Use a computer vision library.
Week 18	**Key concepts:** computer vision, color tracking, background subtraction, frame differencing, optical flow, face tracking **Learning outcomes:**

	Explain what computer, its applications and challenges are. Describe what color tracking, background subtraction is. Use frame differencing to track movement. Explore advanced techniques such as optical flow and face tracking. Use a computer vision library.
Week 19	**Key concepts:** computer vision, color tracking, background subtraction, frame differencing, optical flow, face tracking **Learning outcomes:** Explain what computer, its applications and challenges are. Describe what color tracking, background subtraction is. Use frame differencing to track movement. Explore advanced techniques such as optical flow and face tracking. Use a computer vision library.
Week 20	**Key concepts:** Interview with Memo Akten **Learning outcomes:** Explain how computer vision is used creatively.

Activities of this module

The module is comprised of the following elements:

- Lecture videos. In each week the concepts you need to know will be presented through a collection of short video lectures. You may stream these videos for playback within the browser by clicking on their titles or download the videos.

- Practice Quizzes. Topics include practice quizzes, intended for you to assess your understanding of the content. You will be allowed unlimited attempts at each practice quiz. There is no time limit on how long you take to complete each attempt at the quiz. These quizzes do not contribute toward your final score in the class.
- Practice Programming exercises: Every week you will apply what you learned by completing short, engaging programming exercises. These assignments do not contribute toward your final score in the class.
- Graded Assignments. There are two graded assignments, each is worth 50% of the final module grade. Each of these assignments is comprised of multiple parts which learners work on during earlier weeks. All assignments will be graded by the project tutors.
- Discussion Prompt. Topics also include discussion prompts. You will see the discussion prompt alongside other items in the lesson. Each prompt provides a space for you to respond. After responding, you can see and comment on your peers' responses. All prompts and responses are also accessible from the general discussion forum and the topic discussion forum.
- Readings. Topics may include several suggested readings. They are good supplementary materials for you to further understand the module topics.

How to pass this module

The module has two major assessments:

- Midterm assignment (50%): This assignment is to be submitted in week 12 of the module. It comprises of multiple programming assignments worked on during earlier weeks.
- Final project submission (50%): This assignment is to be submitted in week 21 of the module. It comprises of multiple programming assignments worked on during earlier weeks. It will be assessed at the end the course in week 22.

This is a detailed breakdown of all of the marks.

Activity	**Required?**	**Deadline week**	**Estimated time per week**	**% of final grade**
Midterm submission	Yes	12	5 hours	50%
Final project submission	Yes	21	5 hours	50%