

The GameEduc Framework



Guidelines for a constructive
game education environment



grafisch
lyceum
UTRECHT



Keilir
Institute of
Technology

With the support of the
Erasmus+ Programme
of the European Union



The intention of this framework is to act as a guide for faculty and organizations who teach or will start teaching Game Development in any capacity.

These guidelines were established by game development teachers, students, and administrators from four European schools between 2019 and 2021 as part of an Erasmus+ project with the same name.

This framework distributed under the [CC BY-NC-ND Creative Commons license](#) aims to give a shared starting point for educators in the game development field and acts as a checklist for both teachers and students.

Alongside this framework is a formal invitation to collaborate with the founding schools in the yearly International Educations Game Jam held each spring. Information about this event can be found at the tail end of this document.

These guidelines are separated into groups and additional information is available through the hyperlinks associated with each collection when applicable.

Authors and contributors:

Max Friberg (Yrgo, SE)	Sander Scholl (Grafisch Lyceum Utrecht, NL)
Robert Esbjörnsson (Yrgo, SE)	Laura Endert (Grafisch Lyceum Utrecht, NL)
Marie Kalmnäs (Yrgo, SE)	Arnbjörn Ólafsson (Keilir Academy, IS)
Linda Almström (Yrgo, SE)	Allan Schnoor (Dania Games, DK)
Susan Johansson (Yrgo, SE)	Jonatan Yde (Dania Games, DK)

Editing and Layout:
Sarah Damm (Dania Games, DK)

Vocabulary and Communication

The Game industry has a lot of terminologies, phrasings, and idiosyncratic word use. These are both technical and [memetic](#). The language of game development exists online and as such it's constantly revised and updated by a global user base. Understanding this international language is key to understanding the industry. Below are guidelines aimed to assist students and faculty in this field.

- Derive job titles and terminology from a global perspective rather than a local one to increase your student's abilities to apply for and find international work and internships.
- Aim to find reasons to connect with international schools to set up knowledge exchanges, study tours, and fostering more international cooperation both between schools and students.
- Teach your students about the industry as a whole, not just their chosen discipline, this can mean providing lectures on topics that do not seem immediately relevant. E.g holding game design workshops for programmers or basic scripting workshops for artists. These sorts of workshops will help students understand and contribute to the workflow and pipeline of their peers as well as beginning to understand the vocabulary constructed within the "walled gardens" of their respective disciplines. Fitting these workshops into the curriculum of any program can be difficult - but in our experience, it's both meriting and encouraging for students who are given the opportunity. We've found that we get this time back during project work as it speeds up the teams' communication.
- Teach students about job titles and what sets of skills and proficiencies they require - and which are not strictly necessary but highly meriting. Doing this arms our students with an understanding of where they are currently situated and helps

them prioritize their studies to accomplish their goals. The chapter linked above outlines the project's findings on job titles.

- When communicating the role of game developer outwards we're examining the impact of including the word "creator" as a way of boosting interest from groups who might consider the profession too technical for them. Yrigo, the participating school from Sweden, is currently implementing this for both their game development programs ("Game Creator Artist" as well as "Game Creator Programmer"). Our initial findings look promising and therefore we're including this guideline even though it's early in adoption.
- Give teachers and mentors time away from their regular tasks to help them research and practice the tools they are teaching as these are quickly evolving with each passing year.



Curriculums

During this project the topic of curriculum development was discussed at length, the difficulties surrounding such an important topic were exacerbated by the age differential between the students from each participating school. Below are the common guidelines born from discussing curriculums from this multifaceted starting point.

- Keep a steady line of communication between the school and the game companies around it to ensure the educations are kept relevant and up to date. As well as familiarizing the students with the local industry they will be joining.
- Focus on projects as the primary way of practicing a given skill as compared to tests and thesis work.
- Utilize the [Unity Engine](#) as a common platform to ensure that knowledge can be shared with minimal friction between collaborating schools. Note: this guideline is only relevant for as long as the Unity Engine is an industry leader.
- Rely on existing resources online to enhance learning and give our students multiple perspectives.
- Make sure that all students develop a strong understanding of teamwork and team dynamics alongside their chosen discipline, this should most likely be a combination of lectures and group projects.
- Introduce high-level concepts in an easy-to-digest way by starting the students in less complex software and toolbox before transitioning them into the industry standard.
- When introducing industry software, utilize the often available layout options to limit the on-screen distractions/clutter to help students focus on an initially small but rapidly growing set of tools.
- Teach students how their skillset in game development can be translated into other professions and surrounding industries.

Equality

The area of discussion around equality spanned a lot of subjects, ranging from inclusion, gender, prejudice, and acceptance. And focused on different areas, outreach, lecturers, physical spaces, and the industry as a whole. Below are guidelines to help ensure a more equitable teaching environment for both students and faculty.

- Ensure that appropriate policies against discrimination and other concrete countermeasures are in place and work as intended. Both for students and at an organizational level
- Make sure that everyone that's a part of the organization (including students) has been briefed on these issues so we can limit hurtful events sprung from ignorance rather than intolerance.
- Be aware of how far the game industry has come, and frame conversations on these topics with this in mind to help paint a more accurate version of the game development landscape.
- Try when possible to work towards an equal gender split for lecturers and mentors.
- Remind those we teach that the game industry is more than what's being advertised to us, both in terms of job opportunities but also in the types of stories, genres, and influences that grow within the industry.
- Understand that change is a slow process and that the only things that can change beliefs are examples, education, and kindness. We feel that if we as educators become too forceful in our efforts to change the perception of games and gaming we run a very large risk of creating a counter-reaction having the exact opposite effect. We want to ensure that when we tackle these issues we focus on nudging the conversations in a healthy direction and remember that often we're trying to introduce change (and therefore inherently some conflict) into what might be a person's passion, safe space or identity.
- Make sure that dissemination activities, printed material, website, social media, and open information meetings reflect the industry as a whole.

Outreach

The gaming industry grows each year. With this sort of development, there is always a need for new talent. To ensure that we can graduate students of all walks of life at a steady pace and high caliber potential students must find our schools and experience the towns these schools inhabit as growing game hubs. Below are a set of guidelines aimed at helping to facilitate this.

- If the school doesn't already have contact with the local game community an effort should be made to create and foster this relationship. If the city lacks a formal game scene the school will do well to attempt to act as a proxy by supplying students with tools and spaces to practice working together outside of scheduled school activities as well as post-graduation. Without these, the risk increases that students move away from post-graduation, and no long-term community can exist.
- Local incubators, focusing on games or otherwise can help tremendously in showcasing the self-employment or limited company route that some students are looking to take after finishing their education. Introducing these two parties early will encourage students to think long-term.
- We've found that with the rapid expansion of the gaming industry it's easy for a student to feel like an outsider. By organizing meet & greets, playtesting events, inviting industry professionals to [fireside chats](#) can really help make the industry feel less distant and scary. These events can be extra beneficial when they focus on interactions and have a more casual atmosphere.
- Students in game development fields are generally very technologically savvy. By noting how students are interfacing with each other online a school can drastically improve their information dissemination and response rates. Currently, we've found that the Discord platform is used by a large number of our students though this might change in the future. By communicating with students on the platforms they are already using we've seen an increase in response frequency and overall dialogue.

- Use student projects as a great way to market your school. We suggest hosting games developed by students on a school-held itch.io account. An example of this can be found on the [Yrgo itch.io page](#).
- Participate in industry events and conferences by sending students to showcase their projects, this will be a great way to reach potential students as well as a great experience for the participating students.



Screenshot from "The Lost Skywhales", a school project made at Yrgo.

Digital Learning

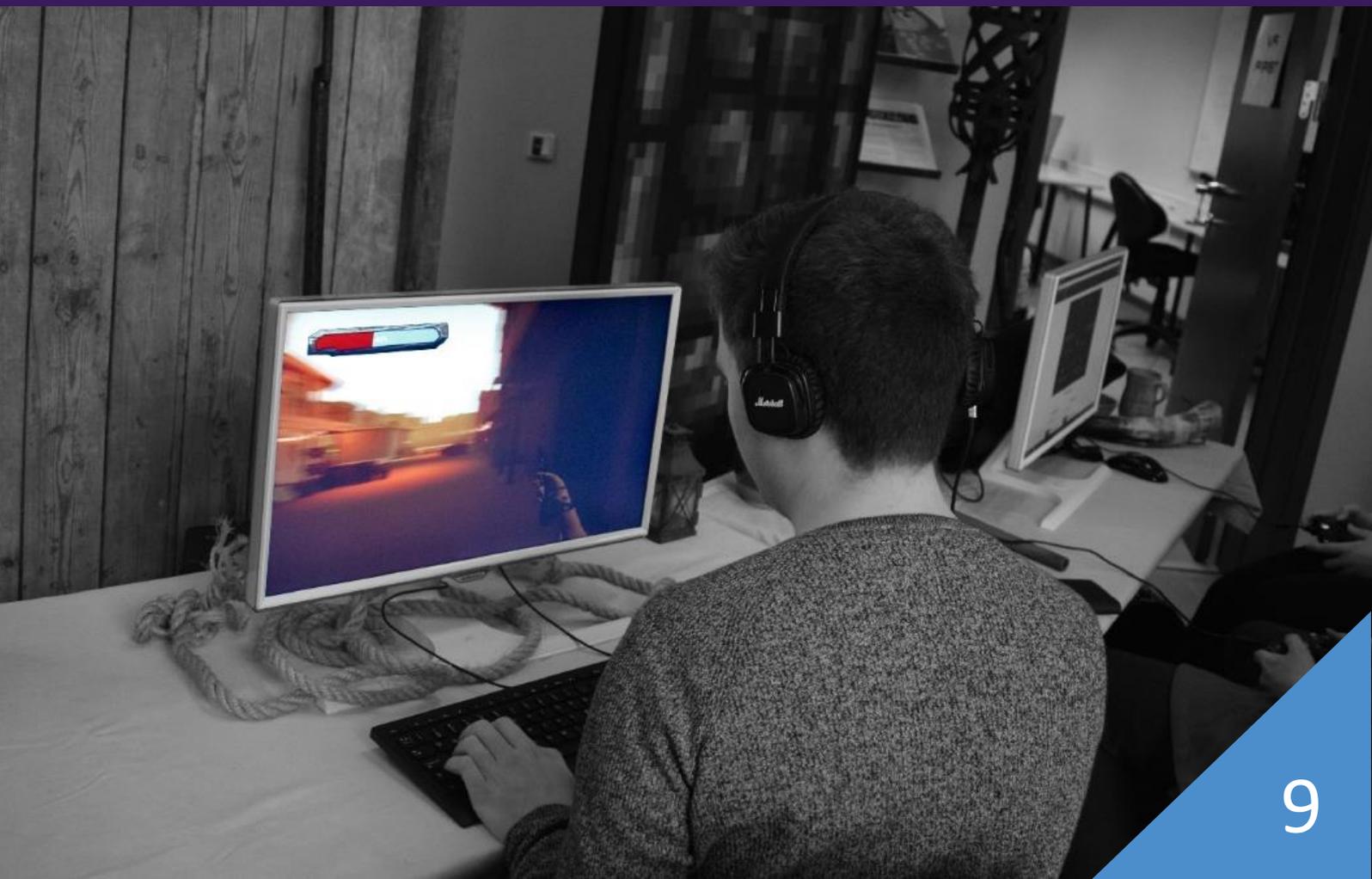
Not initially a part of the Erasmus+ project but as discussions about teaching persisted throughout the SARS-CoV-2 pandemic a lot of discussions surrounding this topic were held and a few findings were crystallized.

- Try and find reasons to break students into smaller groups, a lot of our students who are normally less vocal in a physical classroom become even more invisible when learning is held online.
- Try and pivot to a learning method with more assignments and individual practice tasks. We've found that digital lectures are harder to stay alert for when they run past the 90-minute mark.



- Take stock of and realize how digital and physical learning are different and try to create a digital learning space that the tools allow, instead of trying to fit the idea of a physical classroom into a digital space it was never designed for.

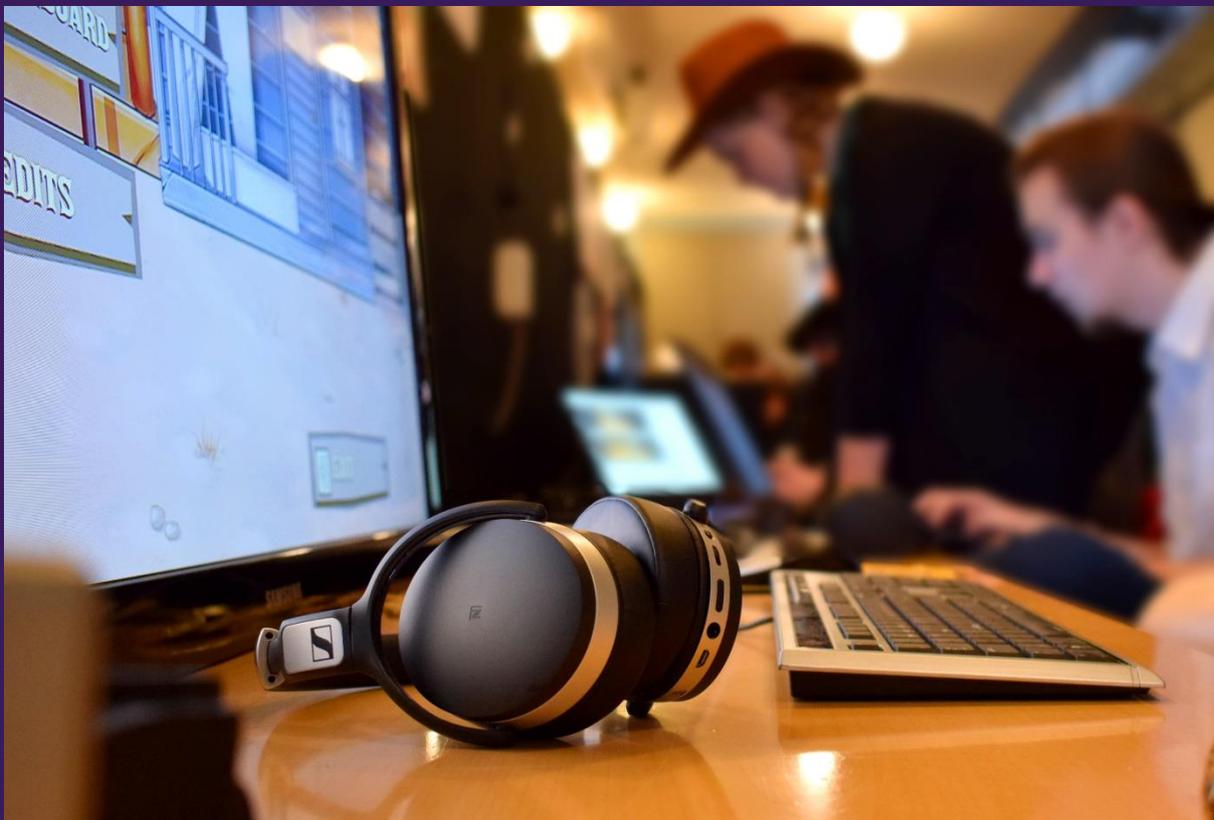
- Before digital learning starts with a new class, set very clear ground rules regarding the use of, for example, web cameras, as we've found that in the digital space a lot of students become more and more reluctant with these sorts of morale-boosting decisions as to the month's progress. It's important to remind them that even though the learning is digital the need for social contact remains the same.
- With lectures held online the ability to have international lecturers is greatly increased, this can help schools with representation and finding lecturers on niche topics.
- When digital and physical meetings are mixed, use the face-to-face time not for lectures but rather for mentoring, tutoring, group work, and following up with students. In Keilir this is the standard way of teaching as they use the [Flipped-learning methodology](#). The flipped-learning system was introduced and incorporated in certain aspects of Yrgo's distance learning during the pandemic.



Common tools, methods, and approaches

We noted a lot of overlap between what areas we wanted our students to know better. Our discussions centered around tools and methods that any person studying game development would do well to familiarize themselves with.

- Each student should have participated in one or more projects, preferably in a multidisciplinary setting within the games industry.
- Each student should have at minimum a basic understanding of the common ways to organize group work and how to manage a project. They should learn about agile methods, SCRUM, and lean.



- Version Control, an obvious part of a programmer's curriculum but this needs to be extended to each discipline as being comfortable in tools like Git will speed up teamwork and production, avoid confusion and strengthen the employability of our students.

- Knowledge about the history, origin, and future of games will help our students understand why their tools have developed in the way they have – and in what ways the industry is trying to innovate.
- Regardless of their specialization, each student should receive a basic introduction to game design principles. Knowing why you do something and what the effects of it may be, are crucial components to any game development discipline.
- All students need to gain a broad understanding of where games are situated today, what titles have leapfrogged the industry and how the landscape of games has changed over time.
- The flexibility of the game engine Unity combined with its presence and influence across the industry makes it the clear choice for teaching game development in a scalable fashion.
- Find time to talk about, or even better, introduce tools that are used in the industry but to a lesser extent than the industry titans. So students can broaden their perspectives and feel comfortable applying for a larger set of job listings where competition is less abundant.



Teaching Programming

Aside from the tools and methods outlined in the previous section, below are a list of guidelines for running a game programming course or program.

- Focus on [C#](#) as a primary language alongside the Unity engine.
- Spend a lot of time teaching students about [Object-Oriented](#) principles like e.g. [Inheritance](#) and [Polymorphism](#) as fundamental building blocks.
- Separate the learning of a [syntax](#) vocabulary for coding from the skills of problem-solving and problem reasoning. We've found that strengthening the latter is the fastest way to improve on skills when starting out with programming.
- Train students in verbalizing their thought processes and utilizing a professional vocabulary.
- Have students work in teams of 4+ people to quickly teach them the dynamics of working as a unit.
- Introduce [version control](#) as early as possible.
- Make [code review](#) and discussion a part of the learning process and find time for students to discuss their approaches with each other.
- Ensure that students are equipped with a large set of tools for [debugging](#) their applications and that they understand the [control flow](#) of a program.
- Utilize several short game projects to familiarize students with the full cycle of developing a game.

Teaching Game Art

Aside from the tools and methods outlined in the previous section, below are a list of guidelines for running a game art course or program.

- Inform students that there are a lot more jobs in 3D than 2D
- Focus on 3D during the majority of the students' curriculums.
- Ensure that students get a good understanding of light, form, and perspective as early as possible.



- Teach students about the technical aspects of the profession, not just the visual. The craft is one part creative and the other part highly technical when it comes to producing artifacts optimized for real-time applications.
- Let students know early on that the artist profession is very demanding, competitive, and takes a lot of time to get both quick and proficient in.
- Have students focus on training the way they observe their surroundings.

- Encourage students to move away from wanting to create their own approaches and stylistic choices and have them focus on well-established methods.
- Teach students about narrative structures, tropes, and myths to enrich the storytelling aspect of their work.
- Make students develop a portfolio of their best work as soon as possible. This is to help students get over the fear of showcasing themselves and to better help them take stock of their strengths and weaknesses. This can be accompanied by a private school portfolio where students can publish all of their work without outside actors being able to see it.



Teaching Game Design

Aside from the tools and methods outlined in the previous section, below are a list of guidelines for running a game design course or program.

- Create lists of games that students should play as part of their studies. These games should be the breeding ground for discussion as well as educate the students about games as a whole.
- Make sure that students can work on live projects to put their theoretical knowledge into practice.



- Use board game development and paper prototyping as a major tool.
- Give the design students (in particular) a firm understanding of the other professional roles and have them learn the basics in these to ensure they are aware of the limitations of their peers and their tools.
- Ensure that reading, analyzing, and writing are tools that the student feels comfortable with.
- Give the student lectures and workshops in Game Theory (from applied mathematics) as well as player behavior and psychology.

Miscellaneous

These guidelines don't fit neatly into one of the above categories and are instead collected below.

- Have the students, on multiple occasions, describe where they are (in their professional journey), where they want to go, and outline a series of actions that will help them get there. This exercise doesn't have to take more than 20 minutes - but can drastically help center a student as they move through the curriculum. By formulating where they want to go, their choice of thesis work, group projects, and individual focus will be made a lot clearer to them.
- Speak with students about difficult industry topics like [crunch time](#) and unpaid overtime. We must prepare our students for the industry they are joining so they can arm themselves against these practices and not be taken advantage of.



International Educations Game Jam Invitation

If you're a school teaching Game Development and your students are at least 16 years old then this is a formal invitation to join in the yearly game jam event, the International Educations Game Jam. The event is held in the spring each year and is a four-day-long vent where students from different countries come together to make short games based on a theme. The event is hosted on [Itch.io](https://itch.io) and held on the instant messaging platform [Discord](https://discord.com).

The first time the event was held there were over 200 students participating and this is scheduled to increase to almost 300 by the next event in 2022. The results of the event can be found [here](#).

Each submitted game is judged by an international jury of industry professionals and the winners are announced in an award ceremony where both the jury's favorites and the games voted on by the students themselves are announced.

To learn more about the event and get in touch with us please contact:

max.friberg@educ.goteborg.se or ahaakmat@glu.nl



game
educ