**Instructor:** John K. Bennett  
Professor of Computer Science  
Associate Vice Chancellor for Innovation Initiatives  
Inworks Director  
303-315-0044  
jkb@ucdenver.edu

**Office Hours:** Tuesday, 3:00 PM - 4:00 PM Inworks Office  
Others by appointment.

**Note:** My travel schedule will on occasion cause me to miss office hours. If this happens when you really need to see me, send me mail.

**Meeting Times:**  
Lecture: 12:30 PM - 1:45 PM, T-Th (Inworks 1300)  
Laboratory: Both in class and unscheduled (Inworks 1300)

**Prerequisites:** A level of programming experience consistent with a CSCI 1410/1411, or IWKS 2300. If you are in doubt about your abilities, talk to me in the first week of class.

**Texts:** There are no required texts, however; several recommendations will be discussed in class. If you are new to C# (even if you are an experienced programmer), I strongly encourage you to buy *The C# Player’s Guide (3rd Edition): Using C# 7.0 and Visual Studio 2017*, RB Whitaker, Starbound Press. ($20.00-$30.00 on Amazon: [https://www.amazon.com/C-Players-Guide-3rd/dp/0985580135](https://www.amazon.com/C-Players-Guide-3rd/dp/0985580135)). This book is an excellent introduction for beginners, but also covers advanced material that will benefit those with more experience. You should be able to find everything else you need in class notes and labs, and online. Other suggested C# and MonoGame references are listed below.

**Piazza Group “IWKS-3400”:** I will use Piazza to communicate important class information. Piazza is a private course management tool that provides support for you to ask questions and to respond to questions raised by other students. HW0 will provide instructions for how to enroll (You will be invited to join this private group on Piazza during the first week of class.)

**Canvas:** This course does not use Canvas.

**COURSE ENROLLMENT MAY BE LIMITED** – This course will be enrollment-limited due to lab constraints. Information on enrollment will be discussed at the first class meeting.

**Course Description**  
Game development involves a rich set of interrelated fields including computer science, graphics design, physics, music, and narrative. This course seeks to introduce the fundamental principles of computer game development. Students will use C# and MonoGame to develop 2D and 3D games. This course involves extensive programming, but requires only introductory programming experience (IWKS 2300, or CSCI 1410/11). Enrollment may limited.
This is a serious course. It will involve substantial software development in the form of a series of laboratory assignments of increasing complexity. The course will culminate with a final project consisting of a team-developed complete game. If you want to take this course, be prepared for a semester-long immersion into game development. Also, this course is relatively new to CU Denver, making you pioneers. Please be patient when the wheels occasionally fall off of the wagon.

**A note about the choice of course software:** There are many platforms one might choose to teach game development. I chose MonoGame because it is powerful, exhibits reasonable performance, and easier to learn (at least in my view) than other platforms. Learning MonoGame and C# will help prepare you for more advanced game development, regardless of the platform you use. In addition, MonoGame is open-source, showing considerable promise, has no legal limitations or cost (unlike many other game engines), and is likely to be around for some time. We will talk about other platforms such as Unity and Unreal, just so you can see how what you learn here will give you a strong foundation for other game development platforms.

**Guilds:** The class will be divided into guilds of 2-4 people each. Guild membership is assigned by the Game Master (the instructor), with input from you. In any class, some students, depending upon their major, gaming experience, artistic talent, etc. may exhibit more proficiency than others on certain aspects of the assigned course work. We will use the data from your course information form to “balance” the abilities of students in the guild (this process will be further explained in class). You will also have the opportunity to request to be in the same guild (or a different guild) than up to two other members of the class. Several aspects of the course utilize guild membership, including your grade, as described below. Each guild should come up with their own name (let’s keep these PG-Rated please). Guilds may select a leader, or not, at the discretion of the majority of guild members. If selected, the Guild Leader is responsible for helping to coordinate the Guild’s activities.

Guild members are strongly encouraged to work together on labs and homework assignments (in some cases this is required). One way I encourage such activity is to award a substantial bonus to guilds with 100% completion of an assignment (even individual assignments). There are several other ways in which guild participation can benefit every guild member, as described below.

All members of the guild begin as a “Guild Initiate.” Each guild member “levels up” to more advanced designations by completing various course assignments that allow them to earn reputation points. These reputation points determine your grade in the class.

The various designations of Guild membership, and the course grade associated with that designation (based upon Reputation Points), are as follows:

- **Guild Initiate**  F  (0-99 Reputation Points)
- **Guild Crafter**  D  (100-129 Reputation Points)
- **Raid Leader**  C  (130-169 Reputation Points)
- **Guild Mage**  B  (170-229 Reputation Points)
- **Guild Master**  A  (>229 Reputation Points)

**Tests:** There are no tests, unless poor attendance causes me to rethink this decision.

**Participation in the Research Project Associated with the Course:**
As you may know, how best to teach interdisciplinary courses like game development is a subject of considerable debate and research. Some of the techniques that will be used to teach IWKS 3400 are experimental, and throughout the course we will use various measures to help us evaluate their effectiveness. We will try to make these activities fun and engaging. Some activities will be online, and some we will do in class. Some of these activities are intended to help you better master the course material, and some are intended to help us make the
Electronic Submission of Course Work: In an effort to kill fewer trees, I have endeavored to make this a nearly paperless class. All homework and lab assignments will be turned in electronically, as described on the pages associated with those topics.

Homework: Homework is assigned on a fairly regular basis. Homework assignments will typically be designed to help you master certain material. My goal will be to make these assignments short and focused. If you do the reading and come to class, weekly homework should never take you more than an hour to complete, often much less.

Labs: Laboratory assignments will be assigned more or less weekly. At first, these will be individual assignments. Then, as things get more complex, assignments will be designed to be completed by a guild. The reason for this is that most game software is in fact developed by teams, and working together often improves learning. The potential downside of team-based work is that some team member might not pull their weight. This issue is addressed by having anonymous peer evaluations. At the end of the semester, each guild member will evaluate the contributions of all guild members (including themselves).

Game Play Experience: Game development has evolved considerably over the past few decades, largely in response to exponential increases in processor speed, and later GPU speed, as well as considerable advancement in graphical algorithms associated with lighting. That said, good game design is good game design, and many games created over thirty years ago still offer enjoyment and challenge. Classic older games also continue to serve as inspiration for new games. As part of this course, you will play a number of these classic games, and will be asked to write a short response paper that allows you to reflect on what these games have to teach us about good game design.

Game Evaluations: Game design is an inherently creative process, one which benefits from constructive criticism. We also learn from the critical evaluation of existing games. You will be required to evaluate some of your colleague’s game designs, as well as several commercial games. You will have the opportunity to earn extra credit for evaluating more than the required number.

Turning in Lab Assignments: In lieu of printing out pages of code, you will upload your completed working projects as a zipped archive, as described on the lab web page.

Note Taking: In order to help everyone assimilate new material, and to ensure that everyone has an accurate set of notes, we will distribute the responsibility for taking notes. Each guild will be responsible for creating and editing a set of notes for a few class periods. If needed, I will assign someone as note taker before the first class, thereafter we will rotate this responsibility among the guilds. Class notes are to be prepared using Microsoft Word or PowerPoint (this is so I can edit them), and should be complete, neat, and thoughtfully prepared. Any figures should be prepared using the Word or PowerPoint drawing tools. Your prepared notes should be emailed to me (jkb@ucdenver.edu) NO LATER THAN the next class period. After review, I will post these notes for everyone.

Class Work: Work in this class will consist of both individual and guild activities.

Individual:
1. reading the assigned materials, attending class, and participating in class discussion
2. completing your assigned note-taking responsibilities in a timely fashion
3. completing all homework assignments
4. completing all individual lab assignments
5. completing all game play assignments
6. evaluating other individual and guild game designs
7. completing peer evaluations for the members of your guild at semester’s end

**Guild:**
1. completing all guild lab assignments
2. assisting all members of your guild in mastering course material (this is where the completion bonus comes into play)
3. completing your assigned note-taking responsibilities in a timely fashion
4. completing all homework assignments (guild bonus for 100% completion)
5. completing all individual lab assignments (guild bonus for 100% completion)

**Optional:**
1. completing a final project (individual or guild)
2. evaluating extra individual and guild game designs

**Grading:** Each course assignment is worth up to a particular number of reputation points, but in general, the relative values of these assignments are as follows:

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>20% + up to 5% bonus</td>
</tr>
<tr>
<td>Lab Assignments</td>
<td>45% + up to 10% bonus</td>
</tr>
<tr>
<td>Game Play Assignments</td>
<td>20% + up to 5% bonus</td>
</tr>
<tr>
<td>Class Attendance and Participation</td>
<td>10% + up to 5% bonus</td>
</tr>
<tr>
<td>(includes note taking)</td>
<td></td>
</tr>
<tr>
<td>Game Evaluations</td>
<td>5% + up to 5% bonus</td>
</tr>
<tr>
<td>Final Project (optional)</td>
<td>Up to 20% bonus</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100% + bonus</strong></td>
</tr>
</tbody>
</table>

As described above, the reputation points that you earn individually and in your guild will determine your final grade.

**Lab and Project Information:**
The goal of the labs and the optional final project is for you to gain practical experience designing and building game software. We will begin with simple labs that get you comfortable with C# and the MonoGame development environment, usually following a scripted process to reach a known result. After you gain this basic experience, you, either alone or with your guild, will be asked to design and implement modest projects of your own design.

**Lab Hardware and Software:**
You may use any computer system available to you to complete work for this course. The software necessary to complete the labs can be downloaded at no cost from the links in the labs tab. Machines with these (and other) software tools installed are available in Inworks.

Some lab assignments may benefit from the use of Xbox game controllers. A limited number of these will be available in lab. If you want your own, they are available for purchase from a number of sources, including Amazon ([https://www.amazon.com/dp/B07F2Z5377/](https://www.amazon.com/dp/B07F2Z5377/)). You will want a wired (USB) controller, as these will work on a Laptop PC (Or Mac running Windows). These will cost you about $15-$20.
Use of the Inworks Computers – For use in lab, Inworks provides laptop computers preloaded with all required software. In order to make these computers fully functional, we have not locked them down fully. As a result, shared use of these computers requires strict adherence to the following rules for use:

- **You may not install software of any kind.**
- Personal files may only be place on the desktop, and these files must be removed at the end of your session (this is just like the Inworks Prototyping space – we expect you to clean up after yourself). You can save and restore your work from a USB drive, from DropBox (your personal account), or via webmail.
- You may not attempt to recover deleted files from another user.
- **Inworks computers may not be removed from Inworks.**

Class Mechanics:

**Attendance** - Students in IWKS 3400 are required to attend class. Much of the material presented in class is complementary to the material learned in lab, and for that reason, class attendance is required. In addition, since there is no official text for the course, material necessary to complete many assignments will only be covered in class. If you aren’t there, you are missing important material, and may be letting your teammates down. **You may miss up to two classes without penalty; after that your course grade will be negatively impacted.** Your course grade will benefit from not missing any class. If you must miss class, please tell me beforehand. Those with extraordinary circumstances such as serious illness should contact me as soon as possible.

**Workload** - Keeping up with the assigned work is crucial. **If you get behind, you will never catch up.** You should expect to spend at least two to six hours per week on this course, besides time in class, on average (some weeks will be lighter, others heavier). You can spend more time if you want, of course, but let me know if you find that it is taking you more time than that just to keep up.

**A Word to the Advanced Student** - One of the problems in any course is that people come in with different amounts of background, so defining activities at the right level for everyone is hard. I expect that some of you will already know quite a bit about C# or MonoGame. In designing the activities for the class I have tried to choose activities that will be valuable to people at all levels, though in different ways. But please let me know if you feel that I’m asking you to do something that does not seem worthwhile, or if some modification would make it more useful for you. I also hope that you’ll share your experiences and skills with the rest of the class.

**Use of the Inworks Facilities** – This course takes place in Inworks. The Inworks prototyping facilities offer a wide variety of tools and equipment that allow you to design and fabricate working prototypes of your innovations. Many of these tools and equipment are potentially dangerous to you or to others if operated in an unsafe manner. In addition, the operation of Inworks prototyping equipment has inherent potential risks, including rotating machinery, moving tools and blades, flying objects, hot objects, sharp objects and tripping hazards. Safe use of our facilities requires strict adherence to principles of general workshop safety and to the specific safety requirements for each machine or tool. In order to use the Inworks prototyping facilities you must:

1) Receive training on the safe use of this equipment
2) Demonstrate proficiency using the equipment to an Inworks staff member (your instructor is an Inworks staff member)
3) Sign the “Inworks Terms of Use” agreement indicating your promise to comply with those terms of use.

**Academic Integrity** - There may be on-line (or course file) solutions for some of the assignments. Do not use them, or those of your colleagues (except when you are supposed to be working together). It is fine (and in fact encouraged) to work together on course assignments, even individual assignments (where this makes sense), but each student is responsible for mastering the material and turning in the documentation indicated that
demonstrates their personal completion of the assignment. Representing someone else’s work as your own is serious academic misconduct, a gross breach of academic integrity, and will likely result in you failing this course. Any questions regarding this policy, as it relates to any work required for this course, should be referred to me.

**Honor Code** - All students of the University of Colorado Denver are responsible for knowing and adhering to the academic integrity policy of this institution. Violations of this policy may include: cheating, plagiarism, aid of academic dishonesty, fabrication, lying, bribery, and threatening behavior. Incidents of academic misconduct will be reported to the appropriate academic unit. Students who are found to be in violation of the academic integrity policy will be subject to both academic sanctions from the faculty member and non-academic sanctions (including but not limited to university probation, suspension, or expulsion).

**Student Code of Conduct** - Students are expected to know, understand, and comply with the ethical standards of the university, including rules against plagiarism, cheating, fabrication and falsification, multiple submissions, misuse of academic materials, and complicity in academic dishonesty. Please see the [Student Code of Conduct](#) and the [Academic Honor Code](#) at for suggestions on ways to avoid academic dishonesty.

**Accommodation of Disabilities** – The University of Colorado Denver is committed to ensuring the full participation of all students in its programs, including students with disabilities. If you have a disability or think you have a disability and need accommodations to succeed in this course, I encourage you to contact Disability Resources and Services (DRS) and/or speak with me as soon as you can. All such discussions will remain confidential. I am committed to providing equal access as required by federal law and would like to help develop strategies for your success in this course. If you have a temporary medical condition or injury, please likewise discuss your needs with me.

If you qualify for accommodations because of a disability, please give me a copy of your letter from Disability Services in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities.

DRS is located in the Student Commons Building, Suite 2116.
E-mail: disabilityresources@ucdenver.edu
Phone: (303) 315-3510

**Mental Health and Well-Being**
I recognize that sometimes confronting difficult life circumstances or personal problems can make it impossible to focus the way you would like to on your studies. Please know that your well-being comes first—and please take advantage of campus resources to provide support in difficult times. The [Student and Community Counseling Center](#) serving the downtown campus is located in room 454 of the Tivoli, providing weekly drop-in hours. You can also reach them at (303) 315-7270 or for emergency after-hours support at (303) 615-9911. In terms of the class, please talk with me as early as possible so we can devise a strategy or modified timeline for your work in the class. I do not need to know details, and any such discussion will remain confidential.

The Campus Assessment, Response, and Evaluation (CARE) team was created to promote a safe and productive environment for learning, living, and working by addressing the health and safety needs of students and the campus community. If you or someone you know is in need of help, please reach out at (303) 315-7306 or at shareaconcern@ucdenver.edu so that the CARE team can provide assistance. More information can be found at [http://www.ucdenver.edu/life/services/CARE/Pages/default.aspx](http://www.ucdenver.edu/life/services/CARE/Pages/default.aspx).

**Non-Discrimination** - The University of Colorado Denver is committed to maintaining a positive learning, working, and living environment. The University of Colorado does not discriminate on the basis of race, color,
national origin, sex, age, disability, creed, religion, sexual orientation, or veteran status in admission and access to, and treatment and employment in, its educational programs and activities. (Regent Law, Article 10, amended 11/8/2001). CU Denver will not tolerate acts of discrimination or harassment based upon Protected Classes or related retaliation against or by any employee or student. For purposes of this policy, “Protected Classes” refers to race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, or veteran status.

Additional Resources:
If you wish to report discrimination or need additional information, contact the EO/AA Compliance Officer at Human.Resources@ucdenver.edu (303-315-2700) or send correspondence to PO Box 173362, Campus Box 130, Denver, CO 80217-3362.

The Ombuds Office is a resource available to all members of the University community. It is an independent source that will provide informal, confidential and neutral services to members of the university community in resolving conflicts, complaints, and disputes. You can reach the Ombuds Office downtown at (303) 315-0046 or in the Lawrence Street Center, Suite 1003; and at the Anschutz Medical Campus at (303) 724-2950 or in Building 500, Room 7005C. CU Denver’s Nondiscrimination Policy can be found at: CU Denver Nondiscrimination Policy.

Religious Observances – Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, we will make every reasonable effort to accommodate you in this regard. Please let me (and your teammates) know as soon as you know that you will be absent.

Classroom Behavior – Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with differences of race, color, culture, religion, creed, politics, veteran’s status, sexual orientation, gender, gender identity and gender expression, age, disability, and nationalities. Academic freedom and diverse viewpoints are also highly valued at the University of Colorado Denver and in this classroom. As specified in the Law of Regents 5.D, “All members of the academic community have a responsibility to protect the university as a forum for the free expression of ideas.” Any behavior to the contrary will not be tolerated. Our commitment is to create a climate for learning characterized by respect for each other and the contributions each person makes to class. We ask that you make a similar commitment.

Classroom Identity - Class rosters are provided to the instructor with your legal name. I will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the semester so that I may make appropriate changes to my records.

Electronic Cigarettes (e-cigarettes) - The use of e-cigarettes is distracting in the classroom environment not only to the instructor, but to other students. The use of e-cigarettes on the Inworks premises is prohibited at all times. Any student who does not comply with this rule will be asked to leave the area immediately so as to not disrupt the learning environment. Pursuant to the Auraria Campus Smoking Policy, the use of e-cigarettes indoors and within twenty-five (25) feet of any entrance is strictly prohibited. See http://www.ucdenver.edu/faculty_staff/employees/policies/Policies%20Library/Admin/Smoke-Free.pdf for additional information.

Use of Laptops, PDAs and Cell Phones in Class:
Using an electronic device during class distracts other students, and is likely to adversely affect your grade (a recent study at CU found that students who routinely use their laptop, etc. in class performed no differently on
average from students who did not attend class at all.) I also recognize that some of you may have a desperate need to check email, update your Facebook page, or tweet. So here is my policy: For the first 5 minutes of class, you may use whatever silent device you wish, for whatever purpose you wish. After that, you may not. If you cannot abide by this policy, find another class. Caveats: Your use of an electronic device cannot be disruptive or offensive to others. Also, I usually make announcements and answer questions during this time; you are responsible for this information.

There are three exceptions to this rule: 1) if we are all using computers in class, 2) if I am demonstrating how to use course software using my computer, you are welcome to follow along using your computer (but not to check your email, update your Facebook status, etc.), and 3) if you are a designated note taker for that class.

**Recommended Reading**

Note: Most students will be able to learn all of the necessary material by attending class and using on-line resources. However, some students (and your instructor) benefit from seeing material in print. The following list points to print and on-line resources with which your instructor is familiar.

In this class we will use Microsoft Visual Studio C# 2017 and MonoGame, which is an open-source implementation of Microsoft’s XNA 4.0. There are some important differences between these tools and their earlier incarnations, which you should particularly keep in mind if you seek to reuse XNA 3.x code in your projects.

**C# References - There are dozens; here are four:** (Recommendation: Buy Whitaker’s book (#1 on the list) unless you are already conversant with C# 7.0, in which case the online resources may be adequate)

2. *Learn C# in one day and Learn It Well*, Jamie Chan, Lean Coding Fast
3. Microsoft also has a host of free online resources, including the following [C# Tutorials](#)
4. *Beginning C# Game Programming*, Ron Penton, Thomson Course Technology. (An accessible introduction for those with less programming experience, especially with C#. Some of the computer science is a little sketchy, but not so much that it’s a concern. Only read Part 1, Chapters 1-5), and note errata [here](#).

**MonoGame References**

Completeness of documentation is not a strength of MonoGame, but since it is open-source, we can always go look to see how things work. There is some documentation, however.

See [http://www.monogame.net/documentation](http://www.monogame.net/documentation)

**XNA 4.0 - There are dozens; here are some:**

MonoGame is an almost-faithful implementation of XNA. There are a few notable differences, but, in general, what you learn about XNA 4.0 will translate easily to MonoGame.

*(Recommendation: If you are going to buy one of these, buy #1 or #2, depending upon your programming ability; buy #3 if you want to get serious about writing shader (HLSL) code)*
1. XNA 4.0 Game Development by Example: Beginner's Guide, Kurt Jaegers (2D only, but good coverage. This is an excellent book for the non-computer scientist).
2. XNA Game Studio 4.0 Programming, Miller and Johnson, (Addison Wesley), (This is an excellent book for the more advanced student, written by two XNA insiders).
3. 3D Graphics with XNA Game Studio 4.0, Sean James (This is the best book I have found for low-level details (lighting, shaders, etc.; It is very well written).
4. Learning XNA 4.0, Aaron Reed, (O’Reilly), (This a quick but well-written update of the XNA 3.0 book of the same title. Several good examples; primary focus on 3D).
6. The Complete XNA Game Studio 4.0, Jonathan S. Harbour
7. Microsoft XNA Game Studio 4.0: Learn Programming Now!, Rob Miles (XNA 4.0 update)
8. XNA Game Studio 4.0 for Xbox 360 Developers, Jonathan S. Harbour
9. Again, Microsoft has free online resources and tutorials, including the following: Introduction to XNA Game Studio 4.0.

This site provides helpful prerequisites for installing and using Microsoft XNA Game Studio 4.0. XNA 4.0 is now almost a decade out of date. It can be made to work on modern platforms, but this is probably not worth the effort. Use MonoGame instead.

**DirectX:**

1. Microsoft has several of free online resources, among them:
   - DirectX Graphics and Gaming (Windows)
   - Programming Guide for Direct3D 11 (Windows)
   - Direct3D Tutorial Win32 Sample
2. A set of excellent tutorials from RasterTek
3. Practical Rendering and Computation with Direct3D 11
5. https://www.3dgep.com/learning-directx12-1/
6. 3D Game Programming with DirectX 11, Frank Luna, Mercury Learning

**Game Design:** There are zillions; here are several (Recommendation: If you are going to buy one of these, buy #9)

1. Programming Game AI by Example, Buckland, Worldware. (Not XNA-specific, but an accessible treatment of this topic).
2. Mathematics and Physics for Programmers, Kodicek, Charles River Media (probably the most accessible of these books that is also thorough)
3. 3D math Primer for Graphics and Game Development, Dunn and Parberry, Worldware Publishing (fairly accessible; develops the needed linear algebra along the way)
4. Physics Modeling for Game Programmers, Conger, Thomson Course Technology (a thorough but reasonably accessible treatment)
5. Mathematics for 3D Game Programming and Computer Graphics, Second Edition, Lengyel, Charles River Media (fairly advanced treatment of this subject; presumes a solid math background, e.g., a year of calculus)
6. The Ultimate Guide to Video Game Writing and Design, Dille and Platten, Skip Press (a nice treatment of the many non-technical considerations involved in game development)
7. Designing Virtual Worlds, Bartle, New Riders (a comprehensive look at the entire space of creating virtual environments in which players interact)
8. Game Engine Architecture, Jason Gregory
9. The Art of Game Design: A book of Lenses, Jesse Schell (This is the best overall game design book I have seen).
10. On the Way to Fun: An Emotion-Based Approach to Successful Game Design, Roberto Dillon
11. A Theory of Fun for Game Design, Raph Koster
13. Game Design Workshop, Second Edition: A Playcentric Approach to Creating Innovative Games (Gama Network Series), Tracy Fullerton
14. Challenges for Game Designers, Brenda Brathwaite
16. Level Design: Concept, Theory, and Practice, Rudolf Kremers
17. The Art Of Computer Game Design: Reflections Of A Master Game Designer, Chris Crawford
18. Homo Ludens: A Study of the Play-Element in Culture, Johan Huizinga
19. Game Development Essentials, Jeannie Novak
21. The Game Maker's Apprentice: Game Development for Beginners, Jacob Habgood, Mark Overmars and Phil Wilson
22. Level Up!: The Guide to Great Video Game Design, Scott Rogers
23. Game Coding Complete, Third Edition, Mike McShaffry
24. Game Development Essentials: Online Game Development, Rick Hall and Jeannie Novak
25. The Ultimate Guide to Video Game Writing and Design. Flint Dille and John Zuur Platten
26. Game Development Essentials: Game Project Management, John Hight and Jeannie Novak
27. Beginning C++ Through Game Programming, Michael Dawson
28. Game Development Essentials: Gameplay Mechanics, Troy Dunniway and Jeannie Novak
29. Game Development with Unity, Michelle Menard
30. Game Engine Architecture, Jason Gregory, Jeff Lander and Matt Whiting
31. Advanced 2D Game Development, Jonathan S. Harbour
32. Game Development Essentials: Game Level Design, Jeannie Novak and Travis Castillo

Last updated 1/15/19