Into An Agile Forest

The adventure of an iterativ game dev



Digtal Design & Udvikling

HTX Frederikshavn

10. november 2026

Omfang 59 normalsider

Vejleder:
Thomas Birk Abildgaard

Gruppemedlemmer:
The Manly Man
Blue Slime
Little Jimmy Two-shoes
Iguana Guy

Indhold

1 Indledning / Projektbeskrivelse **1**

2 Genre analyse (optional) **2**
2.1 Introduktion til genren . . . 2
2.2 Kendetegn ved genren . . . 2
2.3 Analyse af eksisterende spil . . . 2
2.4 Perspektivering til eget spil / egen problemstilling . . . 2

3 Metode & værktøjer **3**
3.1 The Layeres Tetrad . . . 3
3.2 Narrative Design (Courtney Milne) . . . 3
3.3 Projektstyring, Gantt & ScrumLite . . . 3
3.4 Versionsstyring & Git . . . 3
3.5 LibreSprite . . . 3
3.6 UI & UX . . . 3
3.7 Godot Game Engine . . . 4
3.8 Idegenerering metode . . . 4
3.9 Mood board / Storyboards . . . 4

4 Projektstyring & Udviklingsmetode **5**
4.1 ScrumLite . . . 5
4.2 Stand-up mødet . . . 5
4.3 Sprints . . . 5

5 Teori **6**
5.1 A.I In Games . . . 6
5.2 The Layered Tetrad . . . 6
5.3 UI & UX . . . 6

6 Idegenerering **7**
6.1 Brainstorm . . . 7
6.2 Ide-stafet . . . 7
6.3 1-2-Team . . . 7
6.4 Selektering af ideer . . . 7
6.5 Moodboard . . . 7
6.6 Farvepalette . . . 7
6.7 Opsummering forud for første sprint . . . 7

7 Sprint #0 **8**

8 Sprint # 1 **10**
8.1 Udvalgte features for sprint . . . 10
8.1.1 Feature / Story 1 . . . 10
8.1.2 Feature / Story 2 . . . 10
8.1.3 Feature / Story 3 . . . 10
8.1.4 Feature / Story 4 . . . 10
8.2 Test . . . 10

8.3 Sprint review 11
8.4 Sprint retrospective 12

9 Playtest **13**
9.1 Playtest strategi 13
9.2 Playtest resultater 13
9.3 Analyse af playtest 13

10 Sprint # 2 **14**

11 Sprint # 3 **15**

12 Konklusion **16**

13 Evaluering / Diskussion **17**

A Tidsplan **18**

B Programkode **19**

C Trello **20**

Resumé



Resume goes here.. Max ½ side. (kort og præcis) (Angiv evt. link til produktdemo, if any)

Kapitel 1

Indledning / Projektbeskrivelse

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Trefry 2017 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Shiffman 2012 Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. Fullerton 2018 Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Koster 2004 Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer. Dunn og Parberry 2011 Verth og Bishop 2008 Lockhart og Lang 2025 Isbister 2016 Skolnick 2014 J. Sutherland og J. J. Sutherland 2014 Doshi 2016 Sylvester 2014 Schell 2008 Eberly 2003 Lengyel 2011 McCabe 2016 Verth og Bishop 2008 Schreier 2018 Schreier 2021 Stahler 2004

Jeg gik mig over sø og land, der mødte jeg en gammel mand... æøå Trefry 2017

Kapitel 2

Genre analyse (optional)



I genreanalyse undersøger du systematisk og beskriver den spilgenre, dit projekt hører til, for at vise at du forstår genrens kendetegn og kan begrunde dine designvalg fagligt. Det er altså ikke bare en beskrivelse af et eller flere spil inden for genren – men en analyse af hvad der kendetegner genren, som fundament for hvordan du kan placere dit spil i denne genre.
I DDU handler det om at vise, at dine designvalg er:

- Begrundede
- Fagligt funderede
- Inspireret af eksisterende spil
- Ikke bare tilfældige

Det viser, at du:

- Forstår spilteori
- Kan analysere digitale produkter
- Kan arbejde metodisk

Du kan sammenligne genreanalysen med problemanalysen fra faget Teknologi.

- I Teknologi analyserer man et problem for at kunne designe en løsning.
- I DDU analyserer man en genre for at kunne designe et spil.

Typisk struktur:

2.1 Introduktion til genren

2.2 Kendetegn ved genren

2.3 Analyse af eksisterende spil

2.4 Perspektivering til eget spil / egen problemstilling

Kapitel 3

Metode & værktøjer

Lidt meta text goes here... (hvad skal dette afsnit handle om? Hvorfor skal læseren vide dette?)



Metodeafsnittet er vigtigt, fordi det giver læseren indsigt i, hvordan arbejdet er blevet udført, hvilke værktøjer og metoder der er blevet anvendt, og hvorfor disse valg er blevet truffet.
Hvor detaljeret skal metodeafsnittet være? Det afhænger lidt af rapportens omfang og formål, men generelt gælder det, at metodeafsnittet skal være tilstrækkeligt detaljeret til, at en anden kunne følge dine metoder og eventuelt replikere arbejdet. Det skal både beskrive de overordnede metoder og de specifikke værktøjer/software, der er blevet brugt. (metode afsnit er ikke en dybde gående tutorial i brug af fx. et stykke software) Hvis formål er f.eks at omhandle A.I ved FSM, skal der større og dybde på AI og tilhørende teori. *(i kan evt. beskrive det det generelt i metodeafsnit, og henvise til et dedikeret teoriafsnit om emnet der går mere i dybden)*

3.1 The Layeres Tetrad

Ganske kort info, og henvis til senere uddybning i Kapitel **??**

3.2 Narrative Design (Courtney Milne)

3.3 Projektstyring, Gantt & ScrumLite

3.4 Versionsstyring & Git

- - Hvad er versionsstyring? (hvorfor er det nødvendigt? hvorfor er det smart?)
- - Hvad er Git?
- - Hvad er jeres arbejdsprocess?
- - Information ift. jeres repository. (url) / (kan også placeres i sprint 0)

3.5 LibreSprite

3.6 UI & UX

I bør gøre jer tanker omkring jeres UI, og begrundes designvalg ud fra heuristicts om god UI skik. Her (og måske senere) må der gerne falde referencer til vores "ven af husen" Jacob Nielsen, og hans teorier, samt overvejelser ift. gestaltlovene.

3.7 Godot Game Engine

3.8 Idegenerering metode

I bør her være explicitet i hvilke metoder i anvender, og striks ift. at følge dem i praksis således i har et godt grundlag for at evaluere processen efterføglende. Sørg for stærk relation til teknologifaget tværfagligheden fremgår tydeligt. Evt. jeres teori ift. Interaktionsdesign (hvis i tager stilling til dette i projektet) Wireframes, Arrow, Ink

3.9 Mood board / Storyboards

Kapitel 4

Projektstyring & Udviklingsmetode

Projektstyringsværktøjer og generel strategi for udvikling.

- Hvad er SCRUM(lite)
- Standup-mødet
- Hvad er et Sprint ?
- Anvendelse af overordnet tidsplan
- Yderligere teori vedr. The Layered Tetrad (som i bør referere tilbage til i jeres sprints, hvor i finder det relevant)

4.1 ScrumLite

4.2 Stand-up mødet

4.3 Sprints

Kapitel 5

Teori



Her er det relevant at gå i dybden med de emner som er "omdrejnings punktet" for projektet, og hvis der er ny tekonologi/viden som indgår som fundament for jeres udvikling.
fx. ved Projekt "Spil Design & A.I", vil det her være på sin plads at skrive om AI rolle i spil design, og om hvordan i FSB og BT fungerer.

5.1 A.I In Games

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

5.2 The Layered Tetrad

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

5.3 UI & UX

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Kapitel 6

Idegenerering

Start med lidt Meta-tekst. Hvad er idegenerering. Hvilken metode har i anvendt og hvorfor?

6.1 Brainstorm

6.2 Ide-stafet

6.3 1-2-Team

6.4 Selektering af ideer

Den generelle retning på spille skal fremgå her på et relativt højt abstraktionsniveau.
Tegninger, storyboard og moodboards er velkomne.

6.5 Moodboard

6.6 Farvepalette

6.7 Opsummering forud for første sprint

Kapitel 7

Sprint #0



Sprint 0 er ikke et rigtigt udviklingssprint. Det fungerer som en **forberedelsesfase**, hvor man gør klar til at arbejde effektivt i Sprint 1. Man udvikler *ikke* funktioner endnu, men skaber **fundamentet** for projektet.

Nedenfor ses de vigtigste elementer:

Produktvision – hvad går projektet ud på?

Hvad: En kort, tydelig sætning om hvad produktet skal løse, og for hvem.
Hvorfor: Uden en fælles retning risikerer teamet at arbejde i forskellige retninger.
Eksempel: "Vi laver en app, der hjælper studerende med at huske deres afleveringer."

Mål for MVP – hvad skal første version kunne? (vigtigt!)

Hvad: Den mindst mulige version af produktet, der stadig giver værdi for en bruger.
Hvorfor: Teamet får fokus på, hvad der er *need to have* frem for *nice to have*.
Eksempel: Brugeren kan oprette én aflevering og få en notifikation.

Grov produkt-backlog – hvad skal laves først?

Hvad: En liste over de vigtigste opgaver (user stories), uden detaljer endnu.
Hvorfor: Teamet skal kende de første opgaver til Sprint 1.
Eksempler:

- Som bruger kan jeg oprette en aflevering.
- Som bruger kan jeg få en notifikation.

Teknisk opsætning – så alle kan arbejde med ens setup

Hvad: Opsætning af alt det praktiske:

- Git-repository
- Projekttavle (fx Jira, Trello, GitHub Projects)
- Udviklingsmiljø (IDE, frameworks, dependencies) samt specifikke versions numre.
- Fælles kode-regler

Hvorfor: Sprint 1 skal bruges på at levere værdi, ikke på at installere software.

Definition of Ready (DoR) og Definition of Done (DoD)

Hvad:

- **DoR:** Hvornår er en opgave klar til at blive udviklet?
- **DoD:** Hvornår er en opgave 100% færdig?

Hvorfor: Det fjerner misforståelser og skaber fælles forventninger.
Eksempel på DoD: "Koden virker, er testet, commit'et og vist i review."

Minimal arkitektur – det absolut nødvendige

Hvad: Kun de tekniske beslutninger, der er nødvendige for at komme i gang.
Hvorfor: Man skal undgå at låse hele projektet fast for tidligt, men teamet skal vide nok til at starte.
Eksempel: "Vi bruger MariaDB som database og laver et proof-of-concept for login."

En hurtig metafor

Sprint 0 = At gøre klar til at bage en kage

- Find opskriften (vision)
- Vælg hvilken kage du starter med (MVP)
- Læg ingredienserne klar (backlog)
- Tænd ovnen (teknisk setup)
- Aftal hvordan kagen skal se ud, når den er færdig (DoD)

Først i **Sprint 1** begynder man at "bage".

Kapitel 8

Sprint # 1



Det er her der arbejdes, og det er her i skal være skarpe på at demonstrere hvad i kan, og hvilken teori og metoder i anvender ift. at løse jeres udfordringer/opgaver.

8.1 Udvalgte features for sprint

- Feature / Story 1
- Feature / Story 2
- Feature / Story 3
- Feature / Story 4
- Feature / Story 5
- Feature ...

Husk at beskrive hvad disse features gør for spillet, gennem en linse af TLT modellen eller anden teori, fx. Tynan Sylvesters - Designing Games.

Formål: Hvad er formålet med dette sprint? (Formået med første sprint er typisk at nå til Minimal Viable Product (MVP). Dvs et minimum produkt der giver **værdi** og som kan **testes**.)

Features / stories: For hver sprint kan du have et afsnit, der beskriver hvad der blev arbejdet på i sprintet, Dvs hvilke features/designs bliver implementeret.

8.1.1 Feature / Story 1

8.1.2 Feature / Story 2

8.1.3 Feature / Story 3

8.1.4 Feature / Story 4

8.2 Test

Sprint skal testet i et eller andet omfang. (internt, eller endnu bedre en dedikeret playtest med frivillige udenfor projektet.

Der kan være forskellige typer af fejl som kan observeres gennem test fx.:

- **Teknisk fejl:** Spillet crasher, fryser eller reagerer ikke.
- **Funktionel fejl:** En funktion gør ikke det, som kravet siger.
- **Logisk fejl:** Spillets regler eller beregninger giver forkert adfærd.
- **Brugerinteraktionsfejl:** Spilleren forstår ikke, hvad der sker i spillet.
- **Oplevelses-/designfejl:** Spillet føles ikke som tiltænkt.
- **Teknisk fejl:** Spillet crasher, fryser eller reagerer ikke.
- **Funktionel fejl:** En funktion gør ikke det, som kravet siger.

- **Logisk fejl:** Spillets regler eller beregninger giver forkert adfærd.
- **Brugerinteraktionsfejl:** Spilleren forstår ikke, hvad der sker i spillet.
- **Oplevelses-/designfejl:** Spillet føles ikke som tiltænkt.

8.3 Sprint review



Formålet med Sprint Review er at inspicere **resultatet** af Sprint og bestemme fremtidige tilpasninger. Scrum Teamet præsenterer resultaterne af deres arbejde for nøgle interessenter, og fremskridt hen imod Product Goal diskuteres. (I studie scenarie, kan en enkelt elev tage kunde-kasketten på) evt. indrages vejleder i denne evaluering Her gennemgåes hvad der blev opnået i Sprintet, og hvad der har ændret sig i deres omgivelser

8.4 Sprint retrospective



Målet med Sprint Retrospective er kontinuerlig forbedring, hvor teamet lærer af hver Sprint og tilpasser deres **praksis/metode** for at blive mere effektive.
Selvom forbedringer kan implementeres når som helst, giver **Sprint Retrospective** en formel mulighed for at fokusere på inspektion og tilpasning. (ift. Scrum og Agil udvikling). I en Sprint Retrospective diskuterer teamet:

- Hvad der gik godt i dette sprint?
- Hvad kunne forbedres?
- Hvad vil i forpligte jer til at forbedre i næste sprint?

Kapitel 9

Playtest

9.1 Playtest strategi

- udformning af strategi. (vær meget konkret, så alle kan udføre testen)

9.2 Playtest resultater

- Husk at inkludere feedback fra bruger.
- Jeres egne observationer.
- Generelt alt data i samler op. ifm jeres playtest strategi.

9.3 Analyse af playtest

- Jeres analyse af hele playtesten
- Giver analyse anledning til nyt i backlog forud for næste sprint?

Kapitel 10

Sprint # 2

Kapitel 11

Sprint # 3

Kapitel 12

Konklusion



Kun fakta her ift. målsætning/krav. Dvs ingen holdninger, ingen udskyldninger og ingen forklaringer.

Kapitel 13

Evaluering / Diskussion



Hvordan vurderer i projektets arbejdsproces og metodiske tilgang, samt i hvilken grad den udviklede løsning har været hensigtsmæssig og velfungerende i forhold til problemformuleringen? I kan her inddrage følgende:

- Faglige og metodiske læringsudbytter i relation til projektets problemfelt
- Vurdering af centrale styrker ved arbejdsprocessen og den anvendte metode/løsning
- Identifikation af forbedringspotentiale i både metode, produkt og proces
- Analyse af væsentlige udfordringer, begrænsninger og fejlkilder
- Refleksion over projektforløb, herunder arbejdsorganisering, tidsstyring og samarbejdsform

Bilag A

Tidsplan



Her indsættes den indledende tidsplan. (Gantt), samt evt. justeret tidsplan (Gantt) Rotér gantt diagram 90 grader, og brug **hele siden**. Fyi: Der bruges gantt diagram til tidsplan.

Bilag B

Programkode



Har du mange (over 30) sider programkode angiver du her et link til dit Git repository i stedet, som nedenstående:

- Programkode Github: `https://github.com/yourname/repositoryname`

Nævn gerne, allerede i dit resume, at spil kan tilgåes ved link til Git i bilag X.

Bilag C

Trello

Bibliografi

Doshi, Hiren (nov. 2016). *Scrum Insights for Practitioners: The Scrum Guide Companion*. Practiceagile.com. ISBN: 978-0-69280717-0. URL: https://www.amazon.com/Scrum-Insights-Practitioners-Guide-Companion/dp/0692807179 (se s. 1).

Dunn, Fletcher og Ian Parberry (2011). *3D Math Primer for Graphics and Game Development*. 2nd. A K Peters/CRC Press. ISBN: 9781568817231 (se s. 1).

Eberly, David H. (2003). *Game Mathematics*. Elsevier. ISBN: 9781558605930 (se s. 1).

Fullerton, Tracy (2018). *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. Fourth. A K Peters/CRC Press. ISBN: 9781138098770 (se s. 1).

Isbister, Katherine (2016). *How Games Move Us: Emotion by Design*. Red. af Jesper Juul. Paperback edition also ISBN 9780262534451. MIT Press. ISBN: 9780262034265 (se s. 1).

Koster, Raph (2004). *A Theory of Fun for Game Design*. Paraglyph Press. ISBN: 9781932111972 (se s. 1).

Lengyel, Eric (2011). *Mathematics for 3D Game Programming and Computer Graphics*. 3rd. Cengage Learning. ISBN: 9781435458864 (se s. 1).

Lockhart, Bobby og Eric Lang (2025). *The Game Designer's Workbook: Hands-on Tools, Exercises, Walkthroughs, and Resources for New Game Designers*. 1st edition. Hoboken, NJ: John Wiley & Sons, Inc., s. 208. ISBN: 9781394273935 (se s. 1).

McCabe, Patrick (2016). *Create Computer Games*. Packt Publishing. ISBN: 9781786463357 (se s. 1).

Schell, Jess (2008). *The Art Of Game Design*. CRC Press. ISBN: 9780123694966 (se s. 1).

Schreier, Jason (2018). *Blood, Sweat, and Pixels: The Triumphant, Turbulent Stories Behind How Video Games Are Made*. Harper Paperbacks. ISBN: 9780062651235 (se s. 1).

— (2021). *Press Reset: Ruin and Recovery in the Video Game Industry*. Grove Press. ISBN: 9780802158023 (se s. 1).

Shiffman, Daniel (2012). *The Nature of Code*. Available online: https://natureofcode.com/. self-published (se s. 1).

Skolnick, Evan (2014). *Video Game Storytelling: What Every Developer Needs to Know about Narrative Techniques*. Watson-Guptill / Random House. ISBN: 9780385345828 (se s. 1).

Stahler, Wendy (2004). *Beginning Math and Physics for Game Programmers*. New Riders Publishing. ISBN: 9780735713904 (se s. 1).

Sutherland, Jeff og J. J. Sutherland (sep. 2014). *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Currency. ISBN: 978-0-38534645-0. URL: https://www.amazon.com/Scrum-Doing-Twice-Work-Half/dp/038534645X (se s. 1).

Sylvester, Tynan (2014). *Designing Games: A Guide to Engineering Experiences*. O'Reilly Media. ISBN: 9781449337933 (se s. 1).

Trefry, Jeremy (2017). *Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#*. 2nd. Addison-Wesley Professional. ISBN: 9780134659862 (se s. 1).

Verth, James M. Van og Lars M. Bishop (2008). *Essential Mathematics for Games and Interactive Applications*. 2nd. Morgan Kaufmann. ISBN: 9780123742971 (se s. 1).