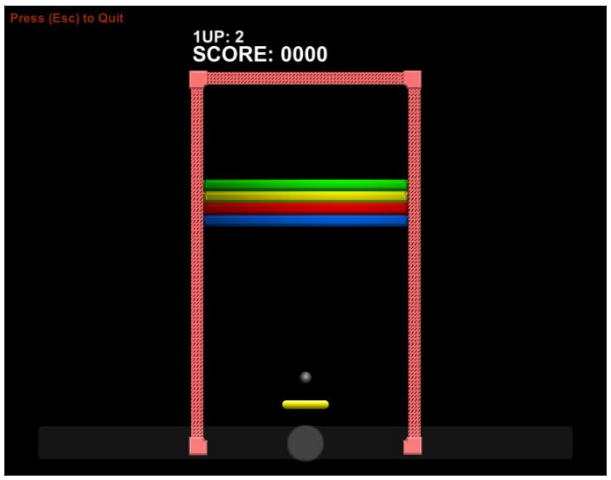
First assignment



Title:

Block Breaker

Weight:

20% of the final grade

instructions

Complete the game Block Breaker using the User Stories provided by the teacher and send me a link (Google Drive, OneDrive, Dropbox, iCloud…) to the project, executable of the final product and a presentation document/user manual.

Please compress your files before uploading them (.ZIP, .RAR, 7Z)

Details

Assignment (the game)

You need to add as many of the essential missing features for the game Block Breaker in order to fulfill most player expectations. Use the list of User Stories to identify the features with the most value and incorporate them in your game. **You don't have to do them all**, but the stories you choose must met the MVP (Minimum viable product) requirements, to get the approval of the teacher (Product Owner).

Assignment (the presentation document)

Your assignment also requires a written document. You need to include the following:

1) A list of all the user stories and/or tasks included in the game.
2) A list of all the external resources/references and credits for libraries, assets or fonts you haven't created by yourself.
3) A user manual, with the default controls layout and supported input devices (keyboard, mouse, gamepad, touchscreen…)

User Stories document

The User Stories document contain 14 User Stories, already sorted from the most valuable to the least.

Theses User Stories are divided into recommended tasks, so you can evaluate the amount of job ahead. ***You can change theses tasks*** if you think you have a different understanding of the workload.

The most valuable story isn't necessary the longest and the least valuable isn't necessary the shortest to do. You might choose to work on a short story, before a longer one, if it better suits your preferences or schedule.

All the Acceptance Criteria for the MVP must be met. If they are incomplete or broken, penalties will be added. ***Remove incomplete features.***

Homework submission

You need to provide:

1) Your entire project: (with the required folders: ***Assets***, ***ProjectSettings***, ***Packages*** and ***Library***)
2) Your game compiled for **Windows x86_x64,** with **data folder** and **related files**.
3) A presentation document/user manual. Format: **docx** or **pdf**

Details		***Performance criteria***	***%***
All files are present (folders: Assets, Project Settings, Packages and library)		00SR-3.2, 4.4 00SW-1.2, 2.3, 3.2, 4.3, 4.7	/15
The code is written efficiently and bug free		00SR-4.1,.2,.3,.4,.6 00SW-2.3, 4.1,.2,.3,.4,.5,.6,.7	/15
Good uses of the engine features		00SR-3.1,.2,.3, 4.2 00SW-2.1,.2, 3.1,.2,.3, 4.7	/15
User stories points		00SR-3.1,.2,.3 00SW-1.1,.2, 3.2, 4.1,.2,.3,.4,.5,.6,.7	/35
The compiled game is functional and reflect the Unity Project		00SW-1.2, 2.1, 3.3, 4.7	/5
The documentation is present and complete		00SW-1.2	/5
All references are mentioned		00SW-1.2	/5
The User Manual is clear and comprehensive		00SW-1.2	/5

Plagiarism

Plagiarism, attempted plagiarism or complicity in plagiarism during an assignment or any evaluated task contravenes the rules. This includes (but is not limited to):

1) The use any code that you didn't wrote yourself!
2) Sharing your code or project (even the code written in class) with anybody without teacher's approvals!

(If you missed a class, or didn't do the exercise, you are on your own to figure out how to do these things).

3) The use any project from another student
4) The use of an assignment done for another course or a project already submitted in the past, which is passed off as an original work.
5) Failing to cite a source (including help from other students).
6) Mentioning sources that doesn't correspond to what you have in your game.
7) Credit me for sources without ensuring that I am the author of that said source.

Failing to do comply to any of these points, may result in a grade of (0%) and any applicable sanctions mentioned in the IPEL.

Elements and performance criteria

Competency: Develop native applications without a database - 00SR ***General ministerial and institutional performance criteria:*** – Critical thinking – Creative spirit – Autonomy – Team work	
Elements of the competency	***Performance criteria specific to each element***
3. Generate or program the graphical interface.	3.1 Appropriate choice and use of graphic elements for display and input 3.2 Proper integration of images 3.3 Adaptation of the interface based on the display format and resolution
4. Program the application logic.	4.1 Proper programming of interactions between the graphical user interface and the user 4.2 Proper programming of communications between the peripheral devices and the software functions of the target platform 4.3 Effective use of execution threads 4.4 Proper integration of sounds and videos 4.5 Proper application of internationalization techniques 4.6 Precise application of secure coding techniques

Competency: Develop gaming or simulation applications - 00SW ***General ministerial and institutional performance criteria:*** – Critical thinking – Creative spirit – Autonomy – Team work	
Elements of the competency	***Performance criteria specific to each element***
1. Analyze the application development project.	1.1 Accurate analysis of design documents 1.2 Proper identification of the tasks to be carried out
2. Prepare the computer development environment.	2.1 Proper installation of software and libraries 2.2 Appropriate configuration of the version control system 2.3 Proper importing of the source code
3. Generate real or virtual world representations.	3.1 Appropriate choice and use of graphic elements for display and input 3.2 Proper integration of 2D and 3D images

	3.3 Adaptation of the interface based on the display format and resolution
4. Program the game or simulation logic.	4.1 Proper programming of behaviours of graphic elements and peripheral devices 4.2 Proper programming of visual effects 4.3 Accurate integration of sounds 4.4 Proper programming of interactions 4.5 Proper application of internationalization techniques 4.6 Precise application of secure programming techniques 4.7 Appropriate use of game or simulation engines