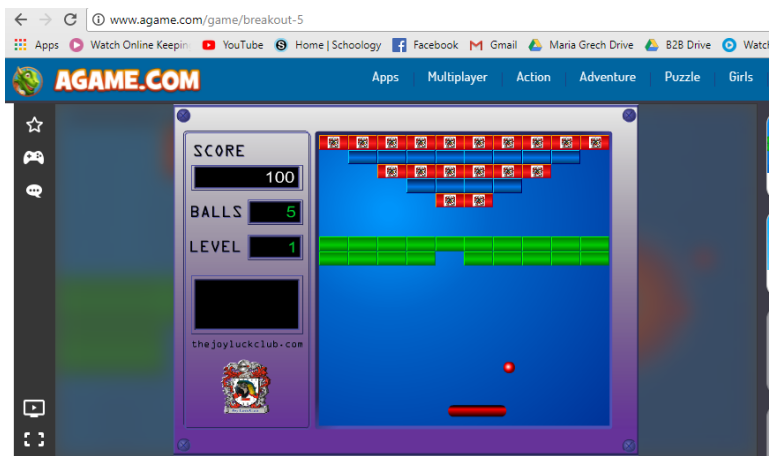


1.Explore the use of Game Engines



Welcome to C#, this unit we are learning about game development. I was new to all this as usual and I found it hard at first to grab onto the subject. I started researching a bit regarding game development and I looked into an easy to do game. I decided to duplicate a brick breaker game. There were many variations of the game but I chose a simple classic one.

When it comes to doing projects I generally work on finishing the project completely before putting the product out. After researching the several models that are usually implemented during game production I thought the one that looked the most similar to what I usually use was the Iterative Model. This model I thought worked perfectly with my game as this game had to be finished as a whole level or else it wouldn't be able to work. This process includes the repetition of prototypes, playtests and revaluation of the game before releasing it. Compared to the waterfall model where the steps are followed one by one after each other, the iterative model focuses more on a cyclical development where certain steps are repeated numerous times. When using the iterative model there are 5 steps which need to be followed:

- 1. Planning and Requirements:** Before starting to code and finishing a project one must first plan how the work will be conducted and what will be produced first. During this step the game developer must check what is required for the game to be complete at the end, which at this stage consists more of brain storming. One mustn't either forget what software is required for the final product.
- 2. Analysis and Design:** After finishing the requirements and planning required for the game we would then move on to analysing the appropriate business side of the game and how the game data would be stored. During this step one would also think about the design and layout of the game, example the size of the screen or the background they would be using.
- 3. Implementation:** After finishing the analysis needed and the design of the game we would then move on to the actual game where it gets coded in this phase. Here the designer would start seeing the game fall into place step by step.
- 4. Testing:** Once all the coding is finished and implemented we would then move on to testing it. This step is crucial as from this step one could determine any possible problems and bugs in the game so as to be able to fix them before publishing the game.
- 5. Evaluation:** After the testing is done we would move on to evaluation where a thorough search and testing is done to check what needs to be changed and fixed. During this process both the coders and the targeted audience are tested to see what they think and what could be improved.

After all these steps are finished they are repeated numerous times until the game is close to perfect. The iterative model has its ups and downs as all things do. Some of the pros are:

1. It is more flexible to make changes and also the least costly.
2. It is easier to test and find the problems that occur.
3. Less risky situations as they could be identified while testing.
4. Easier to program and see it working immediately.

On the other hand the disadvantages of an Iterative Model are:

1. Problems with the device might come out later in the production of the game.
2. The project's succession depends on user's feedback
3. Takes time to develop the changes as they have to go through all the steps.

To produce my game I will first start doing some research on what the games require and how I could code them. I will check all the required material which I would need and read about what code I might make use of. I will then continue to the design of the game where I will either use my own design or search for a design to my liking. After I have gathered all the necessary information and items I need I would then continue on to start to code the game each time I make a new change I play it to test it out. After I do a lot of testing and finish the game I would then continue to presenting my game and checking that it is in working order.

Some of the components that I will be using are necessary in the game and the game wouldn't be able to work normally without them.

1. **Screen Size:** For screen size I will be using Quads so as to mke a border around the game. These will have a Rigidbody 2D which means that they will be staying in place and not move or have any reaction when the ball bounces with them. They would also be set to Kinematic and it's material would be bouncy so the ball would bounce off of it instead of going through it. Around it, it would also have a Box Collider so as to be able to keep the ball from passing through the screen.
2. **Bricks:** The Bricks are the same as the quads, they will be having a Rigidbody2D so they would stay in place. It would also have a BoxCollider 2D which would help to keep score every time the ball touches the bricks. For the bricks I would use prefabs as it would make it easier for me to duplicate the bricks.
3. **Paddle:** The game would also be needing a paddle and it would have a BoxCollider 2D on it so as to be able to interact with the ball and it would bounce back when they touch. I would then, from the Rigidbody 2D freeze the y-axis and also the ability for it to rotate so it would only move from left to right.
4. **Ball:** The best thing is to set the paddle as the parent of the ball for them to be able to bounce with each other. For the ball I added a ball-collider as the ball is round and easier to find out if the ball hit the brick correctly. This will also be given the material of bouncy for it to bounce off the walls and the paddle.

5. **Scoring System:** The game will also feature a scoring system where they would get a point for every block they break.
6. **Life:** It will also feature a life system where the player has 3 tries to finish the game otherwise they would loose and would have to restart the game from the start.

References:

Powell-Morse, A. (2018). *Iterative Model: What Is It And When Should You Use It?*. [online] Airbrake Blog. Available at: <https://airbrake.io/blog/sdlc/iterative-model>. [Accessed 12 Apr. 2018].

Testing Excellence. (2018). *Iterative Model in Software Development and Testing*. [online] Available at: <https://www.testingexcellence.com/iterative-model/>. [Accessed 12 Apr. 2018].

Techopedia.com. (2018). *What is Iterative Game Design? - Definition from Techopedia*. [online] Available at: <https://www.techopedia.com/definition/27045/iterative-game-design>. [Accessed 12 Apr. 2018].