

Final Presentation Guidelines (total time is 15 min per team):

1. Introduction, Motivation and Objectives: Why you did this project and what you hope to achieve through this project. (1 min)

Introduction:

- Screen with Project Name
- Team member names and IDs
- Z-Dork is a watered-down version of the arcade mode in the Street Fighter game series, based on the C language. It focuses on a turn-based fight between two human players, following the logic of Rock, Paper, Scissors, meaning each player must make his move without prior knowledge of what their opponent's move will be. In theory, this leaves the results of each interaction up to chance, but in reality, it forces the player to try and figure out what the opponent is thinking.

Motivation:

- The project was initially proposed as a text-based adventure game, but imagination took hold and the final product is neither text-based, nor an adventure.
- The implementation of graphics pushed the game towards the Street Fighter style, while the underlying groundwork of turn based fighting, kept the algorithm simple to understand and implement.

Objectives:

- The objective was to make a game that would attract users and leave them amazed with what can be done with a programming language that is often seen as out dated and ill-fitted for making games.
- Additionally, the game had to be something that was easy to improve upon, is open to new ideas and could be updated with new features. The game has huge potential and can easily be updated into a market-worthy game, given the proper time and effort.

2. Project Details: How did you convert the dream into reality, meaning what is the project about and what it does and how it works (not technical details). What are the features this project contains? How it can help fulfill/meet the Objectives you mentioned earlier? (2 min)

- The game begins with a main menu, with options for beginning a new game and displaying high scores.
- The high scores screen displays previous game scores along with their date and time from an external file that uses Caesar's Cipher as a form of encryption to make it difficult to manually edit scores.
- The game itself begins with a screen displaying the current score, images for the two players, their health bars, their healing potions, and instructions for each player.
- Player 1 is first given the choice to either attack Player 2 or heal themselves. Healing while health is already full or when heal potions have run out shows an error, and the player is allowed to choose again. If health has decreased due to damage, healing will refill health by 10 points and will decrease the number of heal potions.
- Attacking is done in two ways, either with a sword which acts as a high attack, or with a dagger which acts as a low attack. When Player 1 has made a choice, Player 2 is asked to attempt to dodge the attack, by either ducking, which would avoid the sword attack, or by jumping, which would avoid the dagger attack. Once the choices have been made by both players, damage is done depending on whether Player 2 was able to dodge the attack or not. The tables are then turned, and Player 2 is presented with the choice to either heal themselves, or attack. This turn based attack system is continued until one of the players runs out of health. At this point, the scores are updated and the players are given the choice to play another round.
- If the players decide to not play another round, they are taken back to the main menu and their scores are saved.

- Every attack and dodge choice is accompanied by a series of images that work as an animation. This, along with the console-less interface, draws in the users and gives the feel of a well-designed game.

3. Timeline: How did you plan to finish the project? Now you know how you actually did it or when you did which part of the work, so show that in the timeline. What problems did you face and how you had to change the timeline to adjust it to your current feature list or objectives? (2 min)

The project timeline has been updated from the original timeline to reflect the changes in objectives. The project has undergone some major changes due to which some work that was completed had to be discarded.

- 15.07.2019 – 22.07.2019 (Week 1)
 - Work out storyline and gameplay. (Later changed to turn based gameplay.)
 - Setup basic console interface and characters. (Later changed to fully graphical interface.)
- 22.07.2019 – 28.07.2019 (Week 2)
 - Setup functions for player actions.
- 29.07.2019 – 04.08.2019 (Week 3)
 - Work out enemy logic and actions. (AI enemy was discarded.)
- 05.08.2019 – 11.08.2019 (Week 4)
 - Add scoring system. (This brought the need to reset all statistics and images, which brought up some minor problems and took some time to fix.)
- 12.08.2019 – 18.08.2019 (Week 5)
 - Make changes to switch to 2-player turn based game. (This was a significant change, so some debugging had to be done before we could continue.)
- 19.08.2019 – 25.08.2019 (Week 6)
 - Test current version after changes for bugs.
- 26.08.2019 – 01.09.2019 (Week 7)
 - Research graphics in C.

- 02.09.2019 – 08.09.2019 (Week 8)
 - Begin work on basic graphics (This was mostly some text and the health bars. We were new to using graphics in C at the time and it took quite a while to learn how to change the health bar cells in the middle of the game depending on the current player health.)
- 09.09.2019 – 15.09.2019 (Week 9)
 - Increase scope of game to include multiple rounds.
- 16.09.2019 – 22.09.2019 (Week 10)
 - Add option to save game. (There were a few problems here. Storing the scores was simple enough but printing onto the graphical window directly from an external file in an orderly manner was a new challenge that had to be overcome. The encryption used, albeit simple, was new to us and required some debugging and trial and error.)
- 23.09.2019 – 29.09.2019 (Week 11)
 - Improve gameplay algorithm (Used to be purely chance based. Added player choices to make game more engrossing.)
 - Improve interface (Switch to fully graphical interface. This part was particularly difficult since use of user input in the graphics window had to be learnt. A significant number of challenges were faced in order to make the interface look good while keeping it user friendly. A major example of just how difficult this was is the amount of time and effort it took to create an input field for the player names that would display the name as the user typed and would be editable in real time, a process that would have been extremely simple with the use of a console. Several errors occurred and had to be overcome due to this feature, since a lot of the code depended on being able to differentiate between the two players, which was done largely with their names.)

- 20.09.2019 - 06.10.2019 (Week 12)
 - Integrate graphics. (Added final character images and tested for bugs. This took up a huge amount of time. The problems faced include linking player actions to the images associated with them, finding character sprites that go well with the game, figuring out the proper dimensions required for each image and making them play somewhat smoothly in a series to replicate an animation.)
- 07.10.2019 - 13.10.2019 (Week 13)
 - Final testing and debugging.

Other than this, throughout the project we had to face some major challenges. This was the largest amount of code any of us have ever worked with and keeping things clean and organized was essentially. Additionally, working on different parts separately presented problems of integration and communication that had to be worked through. Since we were also essentially allowed to create whatever we could possibly imagine, we also had to prioritize our ideas and keep them in check with what we were capable of doing within the time period.

4. Future Work: If you had more time, what would you have done to improve the work? (1 min)

- Add several character choices with different statistics and allow the user to choose their characters.
- Increase the scope of the game further, making algorithm changes that would cause attack and damage statistics to vary depending on the character chosen.
- Scale all graphical elements so that game runs smoothly on any computer. Currently, computers with lower resolutions will face problems.
- Add ability to save in the middle of a game and load from it later.
- Improve encrypting algorithm for saving high score files.
- Remove turn-based system and give users the ability to control the characters' movement, attack and dodging in real time. (This is the only improvement that we have no clue how to implement. With all the other points we have at least some idea on how we could have implemented them.)

5. Project Demo: In brief explain which part of the program corresponds to which feature. Run the program and show how it runs or performs a specific function. (2 min)

6. Questions and Answers: Respond to the questions by the teacher, answer specific to the point and only answer if you are asked question (do not answer to someone else's question unless you are asked to). (5 min)

N.B: The presentation should have the same title as the name of the project and specify who is who when introducing team members. You will be given at max 2 min for initial setup and you **MUST** bring a working version of your program and show it live. You will give presentation according to the serial of your Team ID or Team Serial (both are same). Check google sheet link in previous posts to check your Team ID.