Our process included frequent meetings, either online or face to face. As discussed in Phase 1, we have started to use text messaging to better coordinate meetings. Meetings are still quite frequent, around two or three times a week.

A lot of time was spent brainstorming, laying out the structure of the project and researching useful libraries before we got into the actual coding of the project itself. This has allowed us to implement what we want at a rapid pace since many decisions have already been made and questions that may have popped up in coding have previously been answered.

Initial Planning

Our definition of scrum master is the head of communications for our group. He contacts everyone to find the best time for meetings. We discuss the tasks that need to be completed and break most of the larger tasks down into smaller tasks to make it easier for everyone to contribute. We denote a task a small, medium, or large based on our estimate of the amount of time required to properly complete it and the amount of code required for implementation. For example, a one class task would be considered small, but an interface with several classes would be medium, and an entire package would be large.

Sprint Backlog

During this phase, we wanted to build the basic functionality of our game, we plan to do the details of graphics and different game levels during a later phase. Thus, this phase involved creating the basic game display window, question database and random question generator, input interpreter, score calculation, and basic game graphics (moving monsters).

Sprint backlog items completed during phase 2:

- Create Game Class - to run game
- Create Window Class - 3 panel game window, display and respond input
- Create Questions database - xml and dtd files
- Create Question interface and associated classes
- Create Input interpreting class - parse input, check if an answer to question is correct and adjust the score accordingly
- Create game area (ie graphics of game area) class and associated monster/enemies classes
- Have score board in game area

Division of Responsibilities:

Will was responsible for creating the Window class and interpreting input. Laith was responsible for creating the Questions database, interface and associated classes. Ken was responsible for creating the game area and enemies. Laura was responsible for creating the score related tasks. Will, Laith, Ken, Kaeto, and Laura were all responsible for populating the questions database as well as debugging and testing the team's code.

Update Meetings

Monday Feb 22nd Meeting Notes:
https://github.com/willischarted/project-team10/blob/master/meetingFeb22.pdf

Friday Feb 26th Meeting Notes:
https://github.com/willischarted/project-team10/blob/master/meetingFeb26.pdf

Sunday Feb 28th Notes:
https://github.com/willischarted/project-team10/blob/master/meetingFeb28.pdf

Monday Feb 29th Meeting Notes:
https://github.com/csc301-winter-2016/project-team10/blob/master/301.project.meeting.Feb.29.2016.pdf

Burndown Chart

PDF with charts

project-team10/doc/phase2/Burn Down Charts.pdf

Review and Retrospective

Our original idea was to create a desktop adventure game that introduces children to the basic concepts behind programming. We were originally going to have a simple story line involving a main character who completes various programming concept based puzzles throughout the adventure story. As we began programming, we changed our simple story line to a more action-based game, similar to space invaders. In our game, monsters crawl down the game screen and the player must correctly answer programming questions to defeat these monsters.

Although our design idea changed, our basic plan for implementing it remained the same. We divided up tasks and discussed design implementation during our regular meetings. We provided updates on tasks as we completed them through Facebook chat between meetings. We also shared any questions and/or technical difficulties we were having through Facebook chat between meetings. We also exchanged phone numbers to enable easier contact for arranging meetings and easier contact in case of coding emergencies (for example, when our game went from working to crashing with errors). Having regular meetings and brainstorming sessions has been very helpful to keeping our design on track; it also helped us to quickly code up our implementation as we had a very concrete idea to work with. We may need to improve our documentation of meetings and ideas.

How did your plan evolve?

The tasks that we left unfinished weren't completed because we focused mainly upon the barebones implementation of our game. We want the simple components to work before we link them all together. For example: we have our game window set up with an enemy that moves down and questions that spawn and scroll; for our next phase we will have the spawning of enemies linked with the spawning of questions. In our next phase we will have completed putting together these unfinished mechanics.

We finished all the tasks we wanted to complete for this phase. We still have issues open for what we want to have completed by the next phase.

The questions database was populated by various members of the group. Additionally, the main game screen was split up and contributed to as people finished their components for it.