Project 3: Technical Document / Risk Assessment

EGD220: Project Team 2

(Cody Winchester, Kristin Darling, Shain Strother, Micaela Jawor, Paul Kubik, Evan Schipellite)

Prepared by: Evan Schipellite

Technical Synopsis

The purpose of this document will be to provide a detailed break-down of the various tasks for each sprint, thus assisting in the division of roles for the programmers and conveying the scope and potential risks for the various systems and classes that will need to be created as the project progresses. Therefore, it will be crucial that each Sprint begins with the process of taking into account the specific goals required by the end of the cycle, thus allowing the programmers to only focus on building and polishing the application according to each week's incentives, opposed to writing out the entirety of the project from the beginning and following set goals each milestone that were set down from the start of the project. As the design and game material may change over the duration of the project, it may be useful that systems in place early on in the project are flexible and easily alternated to implement various features for controls and the display of gameplay. Therefore, the early sprint cycles will focus primarily on creating these simply and malleable systems, thus allowing for the direction of the game to gradually become much more defined before specific components are further examined and integrated into the code.

This document will therefore list out a number of features for each sprint cycle, beginning with the sprint's description concerning the week's main focus. For each week, the programmers will be tasked with developing a certain feature of the game as required by the rest of the team, and therefore the polishing and completion of these features will take priority over working as extra additions, as many of those extra additions will likely have their own sprint cycles dedicated to them as the project continues onward. Each Sprint component will feature tasks conveying the requirements of the week, the division of roles between programmers, and the risk assessment for those various tasks depending on the comments and concerns of the programmers. At the end of each sprint, a brief postmortem concerning the progression of the previous cycle will be created to assist in showing how the development of features is progression. This write-up will also assist in directing the next milestone, as it will determine the focus of the next sprint depending on what features were deemed complete during the previous week.

Programming Requirements / Further Introduction

The given amount of time for this project will likely be about six weeks, although the actual progression of the project may allow for some time to test and allow for art polishing toward the end of the project. Two important aspects of this project relate to the fact that there are two programmers and also the necessity for the programmers to work with mobile development, a skill that neither programmer has had experience working on. Having noted that, there will likely be some sense of benefit from having multiple programmers, as it will be possible to further divide the roles during each sprint, although, in these cases the programmers will still have to work with slight synchronization in order to ensure that features fit together in the proper format. Also, with regards to mobile development, although neither programmer had ported games to mobile devices beforehand, once the basis is set this process should not be too difficult. With both programmers able to learn the process for porting within the first sprint cycle, the only other concern would be the smoothness of the gameplay as content continues to develop.

This project will be developed utilizing FlashDevelop, and the game will be created for the Asus Tablet, ee Pad Transformer. Developing for the mobiles devices becomes a slight issues where working with various drivers and devices would limit the amount of time developing the game, so at the current moment the project will be limited to one type of device, although it is expected that the process of porting to other tablets would be relatively simple. Android development allows for a lot more freedom for new developers looking to work with mobile devices. It should be noted that the current game concept, that being a side-scrolling game, is predicted to be relatively simply to program, but will likely include many features according to the project specifications. For instance, while the game does not seem too ambitious, working on mobile features, gameplay smoothness, level editors, gameplay systems, and art integration within the six weeks will be more than enough for the programmers to cover together. Even if the goals are overestimated, there will still be time for further polishing to make the game much more enjoyable to play on the mobile device.

Therefore, the purpose of this document will be to convey the requirements decided upon each week during the starting meetings, while also further dividing the roles between the two programmers. At the end of each cycle the programmers will review the progress, and then comment on the various aspects that either went well or require more consideration for future demonstrations. Developing for the mobile device, while keeping the intended mobile audience in mind, may require a decent amount of reworking, even at the programming level. Therefore, with all the unknown variables relating to design, art, and programming, the programmers will work on building up the core game system will also allowing for additional room for error and alterations to account for the unknowns relating to mobile development.

Sprint 1: Risk Expectations

For the first milestone, while the game concept has been chosen and discussed, it will be best for the programmers to avoid working on any specifics of the game to allow for the actual design to become well-founded during the next sprint cycle. However, it is known that the game will feature a play character that is constantly moving to the right from a side view, and the world will be tile-based. With these features in mind, there are more than enough tasks that can be tackled during the initial week to create a playable prototype while also testing the device. Therefore, aside from setting up the TRA and testing the device, this sprint cycle will primarily concern itself with creating a level editor and developing a playable prototype. It is likely that these roles will be divided between the two programmers, and it is also expected that both aspects may not be fully complete by the end of the sprint, as the level editor may convey more features as the project progresses and the prototype will eventually be expanded to include the actual developing game.

Required Components

- The Technical Risk Assessment Document (Evan)
  - The TRA should be created with the technical synopsis, introduction, and details relating to the first sprint.
- Mobile Research (Evan and Paul)
  - The process of porting to the mobile device should be started, and mostly completed.
  - This will allow for the processing of testing a flash project on the device, while also integrating some mobile features, mainly touch.
- Level Editor (Paul)
  - A basic level editor should be started that will allow for the placement of tiles loaded from a sprite sheet.
  - The level editor may eventually allow for the integration of active components, such as enemies, but for now it should simply output a series of numbers utilized for creating a level.
- Playable Prototype (Evan)
  - A playable prototype, likely conveying a simple system showing a player and basic movement should be created. The game will simply convey the player moving in a tile-based world with some simple obstacles to hop over.

Optional Components

- If any art is available, it may be possible to begin adding art tiles or player pieces into the game to further convey the visuals during presentation.
- Utilizing the level editor, it may also be possible to combine the prototype and the level editor to create a tutorial example to show the functionality of the game’s core.

Risk Assessments

- **Low:** Creating the TRA should not present an issue, but perhaps working from an Agile Scrum perspective may cause some slight setbacks in terms of understanding how the process actually works.
- **Medium:** Creating the level editor could take more time than originally expected, due to the fact that the system will also need to be designed to eventually allow for object placement throughout the level. While at this time to focus will largely be based on the creation of a level with tiles extracted from a spritesheet, eventually it should allow for the placement and creation of various objects in the game world.
- **Medium:** The playable prototype should not be too difficult, as most of its features should be kept as a minimum, but creating the level manager, player character, and physics system will require some time and thought.
- **Mobile Research:** A necessary part of the sprint, but still worth noting. Due to the fact that neither programmer knows much about mobile development, researching the porting process will take some amount of time to effectively get to work, as not only will a project for a mobile device need to be created, but also the programmers will have to account for the complications that will occur when placing the game on the device.

Sprint 1: Review

Relating to this sprint, the most important features were tackled and completed. The main objective was to create a simple prototype to begin harnessing the mechanic of the game, and through the simple prototype devised, this gameplay was effectively built and displayed. The prototype itself still requires much adjustment to allow it to be more flexible later on, but it does convey the movement, jumping, and tile-based obstacle work that the player will encounter. Notably, there are some issues relating to tile interaction that will need to be sorted out in the next sprint as the actual gameplay is taken into account and the system will be perfected. Furthermore, the actual process of reducing lag on the devise will need to be noted. This is because, although device interaction isn't scheduled for the next week, early tests show cases of lag due to the tile interactions. However, an algorithm has been discussed and implementation will occur in the following week.

Mobile research was successful in terms of getting the programmers used to porting and managing the application on the device, and due to the fact that it was intended to be the focus of the second week, it is likely that the only issue required will be research into the mobile devices additional features. However, with much of the content already set, more focus can be placed on improving the gameplay and reducing lag during the second sprint. The TRA was successfully written and created for further adjustments, and the level editor was at least started and should be fully functional by the end of the second sprint. The prototype itself has all of the core features requested, including the ability to read in text files for levels, a player class, and the start of a platformer system for the player. Therefore, the goal of the second sprint will largely focus on the development of this system as well as the game's performance on the mobile device.

Sprint 2: Risk Expectations

Now that the prototype has been successfully created, it's now possible to focus on fully working with the system to build it up to be a 'flexible' platformer. While many of the elements of the game could be subject to change through design and art, it is likely that at this point the basic set-up for controls and gameplay will likely remain about the same. Therefore, since the prototype was mainly built to convey the environment, the focus for this week should be to not only effectively port the game to the device, but also to ensure that the system fully works so that future development will focus on the specifics of the gameplay, mainly the implementation of items and features. The prototype, while meaningful in its intentions, only laid down the framework for the system, and therefore there are still bugs relating to the player's movement, jumping, and the reality that collision is not detected for objects above the player. Furthermore, the collisions were previously done through graphic comparing, and each individual tile was rendered to the screen on its own. While unnoticeable on a computer, this had severe consequences when played on the device due to the amount of processing power being demanded by the game. Therefore, while the goal of porting to the device has technically already been met, the actual process of researching methods to make the code more efficient has not been implemented. Furthermore, the level editor will need to be 90% done by the end of this week, largely so that the designer and programmers could easily build test levels for various features. Therefore, the focus of this week will be based on the finalizing of the game system, reduction of lag on the device through code efficiency, and the completion of the level editor.

Required Components

- Updated Technical Risk Assessment Document (Evan)
  - This should just include a basic update, and merely requires time to write after aspects and roles have been sorted out.
- Game System (Evan)
  - Although largely built in the prototype, the game system is still far from perfection.
  - The system should allow for a bug-free environment where the player can move and interact with tiles in any location without producing any anomalies.
  - This will likely call for a large alteration from air-ground collision to x-y collision.
- Lag Reduction / Code Efficiency (Evan)
  - Due to the game's poor performance on the device, multiple areas will need to be adjusted to improve the game's playability.
  - The collision system can be changed to a grid-based system, allowing for the four points surrounding the player's graphic to be transferred into grid coordinates and then matched against the grid to test for collidable tiles.

        - The tiles being rendered can be transferred into a tilemap, meaning that instead of drawing tons of individual tiles, the game will simply be creating its own internal graphic and managing fewer graphics.
            - Due to the size of the one bitmap created, it is likely that this will be broken down to a few bitmaps depending on the level size.
        - General performance alterations can also be researched, such as minor properties settings that could improve performance.
    - Level Editor (Paul)
        - The Level Editor should mostly be complete by the end of the sprint.
        - It should allow for the creation of a level through a tile spritesheet and drawing to the screen.
        - The Level Editor should allow for the specification of various heights / widths for the level, and allow for onboard saving / loading of data.

Optional Components

- Mobile Feature (Evan)
    - While technically not initially a part of the week, there may be time to begin working with one of the main gameplay features. This feature will allow the player to activate and use items by shaking the device, and therefore it may be possible to test the utilization of this feature.
- Egg Ability Feature (Evan)
    - If the testing for the mobile feature is successful, then it will be possible to begin development of the first item. This will likely involve some sort of collectible manager and egg collectible.
    - Upon obtaining the egg, it should be useable after jumping by shaking the device, causing a parachute to reduce the player’s fall speed.

Risk Assessments

- **Low:** The TRA update should simply be a matter of time, and finishing the review is just a matter of making sure that it is complete a day in advance.
- **Medium:** Perfecting the Game System will likely take a large amount of time due to the precise nature of creating a platformer system. Making sure the game works through jumping, falling, and colliding will take some time. Due to the addition of later features, there is a chance that the system will need to be monitored.
- **Medium:** Cutting out the lag will require a decent amount of research, as assistance in working with the Grid and Tilemap features may not be simple. There are classes and algorithms available, it’s just a matter of finding them for Flash and then managing them in a way that allows for the conversion of previous detection / rendering methods into more efficient methods.
- **Medium:** Completing the Level Editor may still take some time due to the accessibility nature that will have to be tackled. It should have an onboard interface for loading / saving information, setting up the base level, and drawing to the screen.
- **Medium:** While the mobile feature itself may not be difficult, setting up a flexible collectible manager may take up the most of the extra time during this sprint.

Sprint 2: Review

Even before the end of this Sprint, all the aspects that were discussed in the 'requirements' section were effectively resolved or created. The Game System was further improved by translating collisions to a Grid system and then building the Tiles into several Tilemaps. This reduces almost all cases of lag on the device, thus ensuring that further additions to the game will also be simple enough to solve in terms of further reducing lag. The player can now collide with tiles from all directions, allowing the level design to be extremely flexible. The Level Editor is still progressing, and should likely be a main focus for the third Sprint by both programmers. If tackled by both programmers, it should be complete by the end of the third Sprint in addition to whatever other aspects are going to be built. Now with a flexible collision system and a method to cut out lag utilizing bitmaps and Grids, it should now be possible to focus on the process of creating the actual game mechanics, mainly the items and features, and further test those items in the game environment.

In addition to the required components, the aspect of testing the mobile feature and adding the initial aspects for the 'egg' feature were also implemented since time was available. Each level now has a text file providing information for a given collectible, and upon loading a level, a collectible manager will also add collectibles according to the information within the text file. Upon initializing the collectible manager, the items will become present throughout the level. Although the 'egg' is the only current item available, the manager will be flexible enough to accept more variants as the Sprint cycles progress. Therefore, at the current moment, the game can check to see if the Player has a given item, and if so, it will test for the specifications for its utilization, mainly the shaking of the device. Upon registering the proper information, that is falling and hold the current item while the device is shaken, the parachute will activate and cause the player to glide downwards instead of simply falling. Furthermore, if shaken again, the player's parachute can be toggled on and off, thus allowing them to fall at any given time.

Sprint 3: Risk Expectations

With the game system mostly in place, and the level editor in a working stage, the goal and expectations for this Sprint should largely revolve around adding many of the essential features to the game. The game system will allow the player to interact with many forms of collision objects through an efficient grid system, thus making the implementation of threats and items very simple. Therefore, one of the major tasks will be to implement the 'butter' and 'pepper' abilities into the game for testing purposes. The butter effect will speed the player up for a small duration on activation, and the pepper ability will allow the player to throw an item behind to slow down the chasing mob. This Sprint will also revolve around creating the checkpoint feature, the level manager, the prototype version of the mob, the ascending / descending areas for the levels, and the audio system. It is likely that art will be implemented toward the end in terms of tiles, background, and character animations. Furthermore, some audio may become available, but the focus of this milestone should mostly be based in just creating the overall manager to 'loan' and monitor the various pieces of audio throughout the game. Therefore, this Sprint is going to mostly be dedicated to creating the mechanics and forming a playable game to allow the designer to begin creating levels and testing for the 'fun' aspects in the game. If time allows, after creating all of these features, further polishes and menu aspects can also be tackled.

Required Components

- Updated Technical Risk Assessment Document (Evan)
    - This version serves as an important update due to the division of roles amongst the programmers.
- Level Editor (Paul)
    - Basic adjustments to further integrate with the content of the game.
- Butter Ability (Evan)
    - This is a basic ability that allows the player a one-time use to provide them with an increase of speed for a limited duration.
    - Further HUD components may be required later to convey the length of this.
- Pepper Ability (Evan)
    - This is an ability that allows the player to throw a pepper backwards at the crowd, slowing them down for a short duration.
    - The purpose of this item is to allow the player to gain distance from the mob.
- Mob (Evan)
    - Essentially the Mob will be designed to be a 640 x 768 block that will gradually be following the player at a slower speed. If it collides with the player, the player will reach an end state.
    - This should simply move up and down when the player also moves to the second area.

- Level Manager (Evan)
  - The Level Manager should allow for Levels to be created at the start of the game, thus running the proper level depending on the current game state.
  - The Levels will likely hold all information for the levels, including the tilemap information, checkpoints, and collectible information.
- Audio System (Paul)
  - This should simply be a prototype manager to allow for audio to be set to variables and accessible by the other classes within the game.
- Checkpoints (Evan)
  - The checkpoints will be read from textfiles and will dictate the locations of various points in a level that will be marked by food characters.
  - At these locations, upon reaching an end state past the checkpoint, the player will be able to restart at the nearest checkpoint passed previously.
- 'Threats' (Evan)
  - Various obstacles that can be placed throughout the level should be created. Essentially these are just blocks that cause an end state upon collision.

Optional Components

- Menu System
  - An initial Beta Menu System to allow entry into the game could be created to control the various game states.
- Art Implementation
  - If available, art should be implemented whenever possible according to the standards for spritesheets and backgrounds previously noted.

Risk Assessments

- **Low:** Both the pepper and butter abilities should be easy to implement with the current system, although the pepper ability will require the mob aspect for full functionality and testing.
- **Medium:** The Mob aspect may require a lot of working to ensure that it doesn't cause lag on the device. This is simply because it is expected to be a large animating image that will constantly be partially on the screen.
- **Medium:** Saving the information all into the Level Manager in various Level classes will require a reworking of the system which is designed to essentially load a new level from a text file each round. Now, it should load and save all the information at the beginning instead.
- **Medium:** Having done little work with audio for a while, the system may require various amounts of testing to ensure that all of the eventlisteners work properly.
- **Low:** Checkpoints should be simple, as they're just a matter of resetting the game information and placing the player and mob at a particular location.
- **Low:** The 'Threat' blocks should be simply the process of colliding with tiles on the grid that cause a reset in the current game.

Sprint 3: Review

Sprint 3 was a bit of a roller-coaster in terms of programming, but overall the majority of the expected were completed in preparation for the game systems main mechanics. Still, one thing that should be recognized now, since it'll likely impact later sprints, is the simple reality that the programmers may not be able to fully dedicate themselves to the project as a result of other closing projects in various areas. Between Game Architecture, and final papers for labs and COR classes, a decent amount of time will be extracted, mostly on the weekends, and therefore it may not be possible to continue as much productivity. However, this reality isn't meant to be as much of an excuse as it is a simple note of caution when continuing with new and ambitious features. Due to the fact that the game's main system has largely been created, the remainder of the sprints can likely be directed toward polishing, level design, menu systems, and audio components on the mobile device. Therefore, it should not be as programming intensive, as the focus will now be based mostly on implementing art and giving the game its levels and challenges.

Up until now, the focus of the sprints has mostly been to create the game system, and therefore it should be noted that aside from some basic test levels to display the items, the game itself does not yet harness the type of scenarios that will be present in the construction of levels in the upcoming sprint. Furthermore, allow the items work properly, it is likely that additions in terms of time limits / uses will be further added depending on balancing components. With regards to the current expected goals for this sprint, all features aside from the audio system were completed. The level editor and all of the items and level features should now be available in the game, and starting next sprint it will be possible to relate the two aspects when creating / loading levels relating to the game. It is likely that the audio system will continue construction during the next sprint, which will allow for more time for music research to be conducted. Any available art was implemented and tested, so now artists should have a better understanding in terms of the limits of animations and the amount of details that are viewable on the images over the duration of the game. Overall, the majority of expected areas were covered during this sprint, but due to the stress of time, it might be wise to tone down the extent of objectives to allow for the time required for external projects.

Sprint 4: Risk Expectations

As of last Sprint, the entire game system should largely be set in place for further development, and the Level Editor should be near completion, thus allowing the designer to begin creating and exporting the levels after a brief tutorial. While the game system is essentially, there's still a necessity to start up various other external systems in order to prepare the game for its final stages. These mostly include the audio system, the scrolling background, and the HUD components. For the most part, these features did not hold priority in the previous milestones, and assets were not readily available to implement, so the demand for these features were low. However, by implementing the features and substituting programmer assets, it's likely that the purpose of the features will become more apparent and audio and art will be set by the fifth and sixth Sprints in order to implement into the game. Therefore, the focus of this Sprint will involve building, polishing, and creating levels. The animations and items will first need to be balanced and combined with the game. Then, some of the art details will need to be polished, as some of the new animations will be available to test within the game world. Next, it should be possible to cooperate with the designer in order to begin creating and testing various levels, as once the designer is given the initial understanding of how to export the levels, it will be possible to begin designing levels according to difficulty and intentions. Finally, in between level design, the audio, background, and HUD systems will be built in order to prepare for the final stages in week sixth. It should also be noted that menus and tutorial components should be part of the focus for the upcoming Sprints.

Required Components

- Updated Technical Risk Assessment Document (Evan)
    - This update should include the various system designs and polishing aspects that will be the source of focus for this week.
- Level Creation (Evan)
    - By cooperating with the designer, time should be set aside to begin include and exporting levels now that the game system is mostly prepared. This should occur after the balancing changes, but it still serves as important due to the limit of time between Sprints.
- HUD System (Evan)
    - A basic HUD system to conveys the player's current item and other scores features (Attempts, World, etc.). This HUD system would simply display the currently held item, and provide status details relating to the current level.
- Scrolling Background (Evan)
    - A feature to enable scrolling backgrounds, randomly being generated from a vector of set background, should be enabled for each level.
- Balancing / Item Design (Evan)

        - Now that the mechanics are available, the items will need to be further designed according to user feedback and designer intentions.
        - Items such as the butter and the egg should allow for the player to only utilize the ability for a set amount of time.
        - The pepper should slow the crowd down for a longer duration the closer they are, thus encouraging the player to save it for when they need it most.
    - Animation / Art Implementations (Evan)
        - Now that a number of animations have become available since the end of the last sprint, it should now be possible to add the idle animations to the various items and further improve the bacon animations.
    - Audio System (Evan)
        - The audio system should simply act as a library from which certain clips can be pulled, thus allowing any class easy access to certain sounds.
        - Otherwise, classes could simply hold onto and manage their own audio.

Optional Components

- Menu System
    - It would be useful if a simple start screen was implemented. This shouldn't be a difficult task, but it may require concept art for a full effect.
- Tutorial
    - Since the beginning levels will require tutorial features, aspects on the first level to halt the gameplay, remove the mob, and teach the player about the items would be good to at least start thinking about.

Risk Assessments

- **Medium:** Level Creation won't be difficult, as it has already been tested, but installing the proper software on the designer's computer and merging exported levels may become a problem on SVN, this should simply be a topic of discussion for a later meeting.
- **Low:** While the HUD system shouldn't be too difficult, it will take some testing and reworking to get properly working. Furthermore, the amount of detail might vary, as a level distance (showing the player's current percentage through the level) might be added in upcoming Sprints.
- **Low:** Scrolling background should be simple, and mostly a matter of acknowledge the layer and presence of certain graphics.
- **Low:** Balancing will likely be just a matter of spending time adding the timer additions and properly conveying them in the HUD, otherwise, testing should be done on the down time to further grasp the difficulty and fun aspects of the game.
- **Low:** Designing the audio system may take some reworking, and in the end, it may just be a matter of allowing objects to own their own audio portions.

Sprint 4: Review

Sprint 4 went mostly well, although there were a few issues in terms of arriving at a few obstacles that caused design and art changes. The scrolling background system took a while to finish due to some logic bugs, but the real issue surfaced after it was created, since during testing on the mobile device, large amounts of performance hits were located. After much further research, it was eventually concluded that, although the scrolling background addition works fine on computer systems, allowing for it to occur on the mobile device will cause too much performance drops for it to be an acceptable addition to the game. Basically, it’s not so much the scrolling backgrounds, as the simple reality that the game is already graphic heavy with its tilemaps, collectibles, and HUD elements. Large graphics being rendered on the screen are the source of the highlighted performance issues, and both the tilemaps forming the visuals for the level and the scrolling background are classified as large graphics. Therefore, adding the three extra graphics required for the scrolling background ends up being too much for the mobile device to handle, and therefore it was eventually decided that the scrolling background system would have to be left out. Although background art was in the process, since it wasn’t fully complete, and only simple prototype versions were available, this was likely not a major problem for the artists, and since there are still many areas in the game that require animations and visuals, there’s more than enough room for alternative art progression if the artists decide to begin working in those areas. It may be possible to continue fiddling with the performance at a later time, but for now a simple blue background will be sufficient for the game, and it may actually be beneficial for the artists to simply focus on the more prominent areas, since the backgrounds would have likely required 9-15 detailed 1280 x 1536 images.

Other than that issue, the rest of the Sprint 4 went according the expectations, and even though there was pressure from many external projects in other courses, it still was possible to complete all necessary aspects for this sprint. The items were given limitations, balancing, and the HUD system was added to show their life span and activity. Additional tile sets were added and level design has begun moving toward additions to the actual game. The audio system was added to the game, and sound effects were created to further convey item utilization. Credited music was researched and can be eventually added to further improve the aesthetics of the game. All art features that became available by the last day before the presentation were also added to the game, allowing for the artists to review their progress and notice any blips or problems in their artwork as viewed through the gameplay perspective. Many other internal issues relating to performance, debugging, and polishing were also tackled this sprint to further improve the game. It is likely that during the upcoming week that the focus will be more toward the menu system, world / levels finalization, art implementation, and other polishing aspects. Also, similar to the last sprint, caution should be taken when creating these tasks, since the external projects are still present, thus hindering the amount of work possible each week.

Sprint 5: Risk Expectations

The focus of this Sprint should largely be to create the final systems relating to the level system and HUD elements, and then proceed on to add the levels into the game. Ideally, the game should be in a fully working state by the end of this Sprint, thus allowing for final art assets and design changes to be implemented during the sixth and final Sprint. In the case of any additional and necessary additions to the game's structure, the sixth Sprint should be open and available for extra architecture implementations, although this should be avoided if possible. Due to the game's current state and the approaching end point in the development process, it is likely that many of the design concepts should be reanalyzed in order to allow the designer and artists to better divide their time. For example, due to the extent of animations, screens, crowd, and goal art pieces that will be required by the end of this milestone, it is unlikely that they will have time for additional items, animations, or collectible additions to the game. Furthermore, due to the amount of iterations of design documents and the necessity for balanced and fun levels, it is important that the designer concentrates on the creation of levels for the game. Therefore, it will largely fall on the programmer to assist in scoping the remainder of the project according to the expected productivity, and from that outlook it will be possible to narrow down additional features in order to allow for better polishing. This essentially means that this Sprint will concentrate on the menu and HUD system, the art and music implementations, and the addition of level's to the game's structure.

Required Components

- Updated Technical Risk Assessment Document (Evan)
  - This update should convey the necessity for project scoping for the remainder of the development process.
- HUD Improvement (Evan)
  - Since some of the art is available, it should be possible to create a bar on the top of the screen that conveys the player's progress through the level. It should show an icon indicating where the player currently is with respect to the end goal. Further art assets could improve what is shown.
  - It might be interesting, from a design perspective, to experiment with adding a cooldown component to show how much of a certain item is available to the player. This could be done through a scaling component, but it would be best if art assets could be added to convey length.
- Menu System (Evan)
  - Further specifications may need to be inquired about, but the basic menu should have buttons that can be tapped, thus loading the game or the current location within the game. It will likely be simple for now.
  - Due to the challenge of implementing saved data, the game will likely reset completed when the program is closed out.
- Level Integration (Evan)

- Levels created by the designer should be added to further improve the world’s structure, additional levels may need to be created by the programmer, and it could be possible to test and adjust these levels.
- Art / Music (Evan)
    - When available, all final art assets should be added to the game.
    - Music may require further examination, due to the fact that a previous mp3 couldn’t be read by Flash due to its properties.

Optional Components

- Tutorial
    - At this point in development, the discussed tutorial may need to be left out in favor of visual / experimentation levels, but it would be a nice addition given the time.

Risk Assessments

- **Medium:** The bar system shouldn’t be too difficult, although it is still an unknown and could present unknown errors upon testing.
- **Medium:** The Menu System shouldn’t be too difficult, but generating the proper states may take some reworking of the architecture. Furthermore, additional requirements to the option components could cause further complications.
- **Low:** The addition of levels should be a relatively simple process, but if additional levels are created, and if testing is required, it could become a tedious process.
- **Low:** Most of the art, aside from animations, is largely going to replace placeholders. The music is the only real issue, as some of the mp3 files are not being properly built by Flash.

Sprint 5: Review

Sprint 5 went according to expectations, and although there was some free time to work on additional features, the time wound up being devoted largely to level creation, polishing, and addition. As noted, the final systems for the game were completed, mostly revolving around the screen and HUD system. The HUD should now properly convey the progression bar showing the player's percentage through the level, as well as the following crowd. Depending on the item, it should also show the item's ability power left through a scaling. In this perspective, item on the plate icon indicates the amount of remaining ability power left through its scaling. The menu system is relatively simple, but allows the player to navigate through several menu screens through button presses, and at the end of the game a final screen allows the player to return to the main menu. The game now runs via game states, thus creating the potential for a pause function to easily be added next milestone. Ideally, research will be conducted at the same time to make sure that moving from the game also initializes the pause function, thus stopping the music and gameplay whenever the player navigates away from the game. Aside from the HUD and Screen system, multiple other smaller details were tackled in order to bring the game to a more finalized and balanced.

Many of these smaller details represent instances were unpredicted bugs or changes in game design caused alterations to the code, and therefore it might be difficult to note every single one of them. Of aspects to note, the Level Editor apparently was not fully functional, and therefore some work was conducted to bring it into a useable state, and this unfortunately may have halted productivity in terms of level implementation. At another time, memory consumption on the device caused the graphics to go blank, and although this was fixed by ensuring the Tilemaps were deleted, there is a potential for a similar issue to occur later if more levels are continuously added, but that is currently unlikely due to the small size of the memory currently being handled now that Tilemaps are only added each level, and are not kept for all the levels. The game seems to lag now when the Tilemaps are created, but further improvements may be conducted next Sprint, but there's a chance that nothing can be done about this particular issue, as it merely represents the loading of the next level. Overall, at this point in the project, it's likely best to avoid further gameplay additions, since the memory and graphic limit available on the mobile device seems to be approaching a limit, and therefore more emphasis should be placed on getting the levels and art added to the game. All art available this week was added to the game, but there's still a number of components that are required for the final week that relate to the goal and the threats for the game. Other aspects would be useful too, but it might be too much for the artists to complete at this point. Aside from implementing the final art pieces and ensuring the game runs smoothly, and adjust the game to pause when the user moves to window elsewhere, it will also be crucial that levels are created and tested, and the programmer will likely be heavily involved in this process.

Sprint 6: Risk Expectations

The bulk of this final Sprint will mostly be focused on wrapping up various pieces of the project, while also ensuring that the entirety of the project meets the expectations relating to the deliverable guidelines. In this sense, some of the additional mobile-specific features will need to be tackled, most notably, the aspect of window swapping on the android device. When the mobile user moves from the game to another aspect of the mobile device, the game should pause and music should mute, and it should return to the game once the user decides to continue playing. To help deal with this situation, it'll be important to create a basic pause function that stops all updates within the game, thus allowing the game to mute and cease all movement whenever the player decides to move away from the screen. However, the gameplay implementation of the pause feature may only be accessible when the player moves away from the game, since there are no external buttons to activate the pause screen. Although advanced gestures or shaking could be utilized for pausing the game, the average user will likely hit the home button to stop playing the game, and therefore a universal pause aspect that deals with this state is the most important. In other words, no pause screen will need to be implemented, but rather the game itself should simply freeze until the player returns to the game. Aside from the pause feature, the majority of this week will be focused on adding final art implementations, adding and balancing levels, adding final crowd animations, and making sure that the game runs smoothly on the device and does not experience any game-breaking bugs upon full play tests.

Required Components

- Updated Technical Risk Assessment (Evan)
    - Similar to previous Sprints, this is just a basic update of all of the required weekly tasks, and it the review should be added toward the end of the Sprint.
- Pause Aspect
    - A pause aspect should be added that mutes and stops updates in the game whenever the user moves away from the window focus of the game. Upon returning the focus, the game should continue to run.
- Level Additions / Balance
    - Basic work should be done on current levels to assist the designer in balancing and adding the available levels.
    - With the designer, additional levels should be added and tested to ensure that the difficulty curve is set properly throughout the game.
- Art Implementations
    - Final art implementations should be added to the game. This will include crowd animations, obstacles art, goal art, and final screen art. Further polishing will be left up to the artists.
- Testing

        - Basic testing should be done to ensure that there are no blatant problems with the game, and although improvements could be made, the focus of testing will mainly be to search for any problematic bugs.
    - Team / Individual Post Mortem
        - If required, a basic set-up for the Post Mortems should be written to convey the progress of the project throughout all of the Sprints.

Risk Assessments

- **Medium:** The Pause Aspect shouldn't be too difficult, but it will require some additional research and testing to properly set-up, and further test cases will be required to make sure that all sounds are muted whenever the user moves away from the game state.
- **Low:** The creation of the individual / team post mortems shouldn't be too difficult, but it will require some time to schedule and set-up during the final crunch to make sure that all of the game's features are implemented properly.
- **Low:** Level balancing and additions are mainly a time aspect, and therefore a few meetings with the designer should allow for the implementation of all of the levels.
- **Low:** Art implementations shouldn't be too much of an issue as long as the extra polishes mainly relate to visuals and not actual additions of frames in the animations. In those cases, time should be allotted to test and implement changes in animations.

Sprint 6: Review

Overall, this final sprint went according to all expectations, mostly due to the fact that the tasks set out were simple and designed to wrap up the project. In that manner, the goals of this sprint did not include any optional features to implement, as the final goal was to move the game into a working and final state without pushing the boundaries in terms of adding additional features or altering the game system in terms of code and gameplay. At this point in the stage, due to the limit of time, it was best to make sure that all of the final art pieces and animations were integrated into the game properly and added and balance all of the levels required for the game. This meant that, even though many ideas and features did surface that would have been great to implement, many of them were avoided, just in case the implementation might cause some game-breaking situations. On a larger scale, this meant that the additional feature of the 'tofu' ability, a collectible that allows the player to increase their jump height, was avoided due to the fact that the current levels were not designed with it in mind, and the fact that its relations with the current game system could potentially cause unpredictable bugs. Furthermore, this would require additional art and sound, and given the amount of art that was already required for this Sprint in terms of the collectibles, crowd, and screens, it was better for the artists to spend more time on the current tasks to wrap up the project. There were other aspects similar to the 'tofu' ability, and most of them were left out in favor of spending time on all of the currently included aspects of the game.

The pause feature, final bug checks, and level implementations were all successfully done without any noticeable issues, although one small bug did remain relating to audio that could be solved in future development, as it was merely factored as an extended looping sound effect that was never fully located, but eventually patched with a few other methods to monitor and control the active sound effects in the game. During this Sprint, a considerable amount of time was spent working with the designer on levels, and during that time the designer was able to adjust to the system and test the levels with ease, resulting in a large addition of balanced levels during the Sprint. The artists were also easily able to get most of the final pieces in during the early stages of the Sprint, thus ensuring that the art would appear as intended in game without leaving the visual checks for too late in the Sprint development process. While not fully required, the post mortems were worked on in order to ensure that the team would have a document to reflect the project's overall progress throughout the Sprints. At the conclusion of this milestone, it is definitely safe to say that all of the project goals have been covered and the final Sprint ended in accordance to the expectations set out throughout all of the previous Sprints.