

### Project Features list

- Document (as a bullet point list) the major FEATURES of your software application.
- Each features should have a TITLE followed by a brief DESCRIPTION of that feature.

### Requirements

- Document the key FUNCTIONAL and NON-FUNCTIONAL requirements for a minimum of SIX features.
- Follow one of the formats/templates provided in class.
- Your application will likely have many more features, and you should create requirements documents for all the features.
- But for this milestone, you need only turn in SIX requirements documents.

### Project Plan

- Create a Project Plan in your chosen Project Management tool. Maybe a GANTT chart, Kanban board, or other suitable format.
- The Project Plan should include each Feature that is to be developed.
- The Project Plan should indicate the SEQUENCE of Sprints. That is, the plan should show in what order the features will be developed (designed, coded, tested, integrated.)
- The Project Plan should show for each feature WHEN (by calendar date) the feature will be worked on.
- The Project Plan should show for each feature WHO will be working on the development of the feature.
- The Project Plan should result in a completed, working application that is ready to present by April 22, 2020

# Project Plan

## Major Features:

- Basic Movement
  - Move left and right and be able to jump.
- Cyclic ability mechanic
  - Abilities will be cycled through and only one can be equipped at a time
  - Abilities may include:
    - Double jump
    - Air dash
    - Teleport to last point on the ground
    - Teleport to 2nd Player
    - Dash through walls
    - Wall climb
- Local Co-op
  - Up to two players can work together to solve puzzles more easily.
- Puzzle Platformer
  - Must use the changing ability mechanic in the correct order and flow to pass each level.
- Hidden secondary objectives in each level
  - A hidden object that can be collected in every level for extra points
- Custom sprites and backgrounds
- Slide when there is obstacles
- Online Scoreboard

## Functional Requirements:

	<u>Functional</u>		<u>Non-Functional</u>
<b>Use Case Name</b>	Communicate	<b>Data Item</b>	<ul style="list-style-type: none"><li>• Resources</li><li>• Positional</li><li>• Score</li><li>• Content/ Category</li></ul>
<b>Title</b>	UX	<b>Type</b>	<ul style="list-style-type: none"><li>• Pointers</li><li>• Integers</li><li>• Integer</li><li>• Text</li></ul>
<b>Description</b>	The user uses the controller to interact with the game.	<b>Description</b>	<ul style="list-style-type: none"><li>• Active ability time</li><li>• Location of character in level</li><li>• Number of points User has accumulated</li><li>• Descriptions of abilities and game flow.</li></ul>
<b>Actors &amp; Interfaces</b>	Controller, Monitor, keyboard	<b>Comment</b>	Encompasses the Users experience.
<b>Initial Status and Preconditions</b>	The controller is connected to the video game and the user has access to it.		
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1) The user picks up the controller</li><li>2) User learns the button mapping</li><li>3) The user plays the game.</li><li>4) The user enters tag for scoreboard</li><li>5) User puts down the</li></ol>		

	controller		
<b>Alternative Flows</b>	<p>If the User does not have a controller:</p> <ol style="list-style-type: none"> <li>1. Use the Keyboard of the computer</li> <li>2. Learn the key bindings</li> <li>3. Continue from Step 2 of Basic Flow.</li> </ol>		
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• Game has ended.</li> <li>• Record score into scoreboard.</li> </ul>		
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>1) The ability to pause the game or quit at any time.</li> </ol>		
<b>Other</b>			

Overview		
<b>Title</b>	Cyclic Ability	
<b>Description</b>	A pop-up menu that the user can choose a specific mechanic to navigate portions of the level's puzzle	
<b>Actors &amp; Interfaces</b>	Around 3 mechanic choices, the user's selection, puzzle hints	
<b>Initial Status and Preconditions</b>	Inside the game loop	
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. The user holds down a button and a pop-up will</li> </ol>	

	<p>appear</p> <ol style="list-style-type: none"> <li>2. User selects a direction of which ability they want</li> <li>3. The user releases the button and the pop-up disappears, empowering the user with the selected ability.</li> </ol>	
<b>Alternative Flows</b>	<p>If ability is disabled</p> <ol style="list-style-type: none"> <li>1. Continue from 2</li> <li>2. If the chosen direction's ability is grayed out, then no ability is selected and the audio will buzz.</li> </ol> <p>Selecting player difficulty</p> <p>For <i>easier</i> difficulties:</p> <ol style="list-style-type: none"> <li>1. Continue after step 1</li> <li>2. Time/Movement slows down</li> <li>3. Continue at step 2</li> <li>4. After step 3, Time resumes at its previous pace.</li> </ol>	
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>• User's selection of ability is enabled and decays after a while.</li> <li>• Game is resume its normal mode</li> </ul>	
<b>Exception Paths</b>	<p>User may cancel ability selection by letting go of the button without choosing an ability</p>	

<b>Use Case Name</b>	Puzzle Platformers	
<b>Title</b>	Gameplay Loop	
<b>Description</b>	Each level will have a problem and solution, where the user has to use the cyclic mechanic to navigate the solution.	
<b>Actors &amp; Interfaces</b>	<p>The character interacting with the platforms and environment</p> <ul style="list-style-type: none"> <li>Platform interfaces get more challenging as the game proceeds. tel</li> </ul>	
<b>Initial Status and Preconditions</b>		
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. The player spawns into the level</li> <li>2. The player approaches the obstacle</li> <li>3. The player selects a mechanic to pass the obstacle</li> <li>4. The approaches the next obstacle</li> <li>5. Repeat steps 3 and 4 until the level ends</li> </ol>	
<b>Alternative Flows</b>	<p>If the player does not follow the flow</p> <ol style="list-style-type: none"> <li>1. The character dies</li> <li>2. The level restarts</li> </ol>	
<b>Post Condition</b>	The level is finished and the player may move on to the next level	
<b>Exception Paths</b>	Users have a choice to use the teleportation ability to revive and resume playing.	

