* Mechanics
  + Present a table outlining the mechanics of the game.

|  |  |  |  |
| --- | --- | --- | --- |
| Mechanic | Use (if explanation needed) | Trigger/Control (if needed) | Controller Trigger |
| Jump |  | Space bar | Xbox A button  PlayStation X button  Switch B button  (The controller’s bottom face button) |
| Movement |  | W, A, S, D, or arrow keys | All controller’s left joystick |
| Crouch |  | Left Shift key | Xbox B button  PlayStation O button  Switch A button  (The controller’s right-most face button) |
| Inventory/  Hotbar | Your storage for items you find. To tell what item you are currently holding to will be highlighted & you would see it in the character’s hand(s). | The mouse scroll wheel & number keys (Not the num pad) to change what you are holding | All controller’s bumper buttons |
| Use item | Use the current item that is in your hand | Left mouse click &or Q key | Either all controllers right trigger or  Xbox Y button  PlayStation r button  Switch X button  (The controller’s top face button) |
| Interact | Use to interact with an interactable | Right mouse click &or E key | Either all controller’s left trigger or  Xbox X button  PlayStation o button  Switch Y button  (The controller’s left-most face button) |
| Camera Control |  | Mouse movement | All controller’s right joystick |

* Code
  + Outline what coding will the game need.
    - The game will need:
      * Code to implement the movement controls (counting the jump & crouch controls).
      * Code to implement the interaction & use item controls.
      * Code to implement the Inventory/Hotbar.
      * Code to implement camera controls.
  + Specify with pseudo-code how the main mechanics will be implemented.
    - Movement:
      * Pressing the W, up arrow key, or moving the controller left joystick up makes the player character moves forward.
      * Pressing the A, left arrow key, or moving the controller left joystick left makes the player character moves left.
      * Pressing the S, down arrow key, or moving the controller left joystick down makes the player character moves backward.
      * Pressing the D, right arrow key, or moving the controller left joystick right makes the player character moves right.
    - Camera Control:
      * Moving the mouse or moving the controller right joystick moves the direction the camera is looking.
    - Jump:
      * Pressing the space bar or the controller’s bottom face button makes the player character jump.
    - Crouch:
      * Pressing the left shift key or the controller’s right-most face button toggles the crouch action.
    - Interact:
      * Pressing the right mouse click, the E key, the controller’s left trigger, or the controller’s left-most face button. This will only work if something is glowing yellow to mean it is interactable. Pressing the interact button on an interactable will make you pick it up if it is an item but if it is a piece of paper or a sign then it will trigger a text box.
    - Inventory/Hotbar:
      * The currently held item will be shown as a different colour & will be shown in the character’s hand(s). The hotbar will only show when the player has an item & it will show only the item that the player has. It will also show the number key that you can press to select the item without needing to scroll through your whole hotbar.
    - Use item:
      * Pressing the left mouse click, Q key, controller’s right trigger, or the controller’s top face button. By pressing the use button, you will use the current item that is highlighted in your hotbar.
  + Include examples of code that could be used in the game.
    - For movement
    - For Interaction
* Physics
  + Specify what physics the game will need.
    - It will need the rigid body physics for at least the gravity physics.
  + Explain how these physics will be implemented.
    - By applying the rigid body component to the player character.
* AI
  + Specify what AI the game will need.
    - This isn’t really needed.
  + Explain how the AI will be implemented.
* Sound engineering
  + Outline how the sound will be implemented into the game engine.
    - All the sounds are planned to be 3dimensional & have a depth of field effect.
    - There would be footstep sounds from the player character when they walk.
      * It would be implemented by making it play each time the player character steps.
    - There would be an eerie(s) sound that plays randomly & at a random depth of field each time.
      * It would be implemented by making it trigger at certain areas &or on a random timer controller program.
* Prototype and testing
  + Specify how the game will look in the following phases.
    - Prototype
      * It would be very minimalistic in design, but it would have most of the controls & mechanics.
    - Alpha
      * It would have at least some of the maze layouts done & have textures for the walls of the current maze layouts. It would also have some of the item’s models (the items that can be used for the current maze layouts that are in the game).
    - Beta
      * It would have almost all the maze layouts done & have all the wall textures in the game ready for use. It would also have most if not all of the models for the items & have all the items in the game.
  + Explain how the game will be tested throughout production.
    - The way that it would be tested is by starting the game at the point that I want to test (like if I need to test physics, an item pickup, or testing an item use).