SCHEDULE IS AT BOTTOM!

NOTE FROM EDITOR:

Initially we had four different types of Ore, but after hours of deeply titillating rumination, we decided that we should instead just use one ore because using 4 would be a real headache. So if you see diamond, gold, emerald, and iron ore in the below, please treat all those as just one.

Initial notes:

I think Henry might have made a little oopsies. After talking with him I have confirmed this oopsie. He said that the character should have a coinbar and also have different resources. The resources are used to purchase different upgrades and the coins are never used so there is no reason for the coins to exist. So I will not use coins and instead of doing a bar, I will just have a counter for each resource which is displayed on the screen. There are also quite a bit of other pieces in which some degree of specificity was left out so I will take creative liberty there. Most, if not all, of these ideas have been approved by the game designer himself - the man, the myth, the legend - Henry Sanderson. Also after talking to him we decided to change the hierarchy of the Ores to (from lowest to highest) iron, gold, emerald, and diamond.

So here are what I think the main classes of the game should be and some notes for their design:

Item Name	Requirements (and suggestions)	Difficulties	Notes	Instance Variables	Methods
Block (main Character)	- Can wear different armour - Has a specific health potential (armour) with a visual health-bar - Has a specific mining speed (pick) - Has a specific attack level (bow) - Can shoot enemies by using mouse pad - Can mine resources by collision - Can move in four directions - Should start off as a green square - Can gain health by colliding with hearts - Block cannot shoot and mine at the same time	- This is probably going to be the hardest object to code just because it has to be able to interact with almost every other part of the game		- ironCount (int) - diamondCount (int) - goldCount (int) - emeraldCount (int) - healthMax (int) - attackSpeed (int) - attackPower (int) - healthCount (int) - isAlive (boolean) - isShooting (boolean) - isMining (boolean) - direction (int)	- moveUp() - moveDown() - moveLeft() - moveRight() - shoot() - takeDamage() - gainHeart() - gainGold() - gainIron() - gainDiamond() - gainEmerald() - mine() - getters and setters

Game	- You have to be able to advance to the next level and if you die restart that level - You also have to be able to toggle between multiplayer and single player mode - It would also be nice if you were able to pause the game and restart - Somehow in between levels there will have to be some sort of shop that appears that will allow you to upgrade the block	- I am having difficulties of figuring out where to keep all the instruction for the different levels (like how much ore / enemies spawn etc.) - I think its going to be very hard to have the shop appear between levels	- I think this would be class that would end up using all the other classes	- levelCount (int) - multiPlayer (boolean)	- restartLevel() - restartGame() - switchGameMode() - pauseGame() - startLevel() - upgradeHealth() - upgradeMiningSpeed() - upgradeAttack()
Level #n	- Each level will have a different number of different ores, enemies, and mobs appear	- One thing henry wants to do that seems VERY had is have a different proportion of different types of blocks appear for different letter (I will expand upon this in my feasibility analysis I have a solution.)		- MAXIRON (int) - MAXGOLD (int) - MAXEMERALD (int) - MAXDIAMOND (int) - ironMined (int) - goldMined (int) - emeraldMined (int) - diamondMined (int) - length()	
Health Bar	- Must accurately and visually display how much health the block has - Must change as the health of the block changes	- This should be pretty easy just because it can just	- (I'm visualizing just a bar that is filled with green	- Doesn't really need any (it can just grab the healthCount from the	- refill() - looseHealth() - gainHealth()

	- Resets at full health at the beginning of each game (learned after talking to the designer)	grab healthCount from Block and it will looseHealth() whenever Block does takeDamage() and it will gainHealth()	and as it loses health the bar falls and when it below half health it turns yellow and when it is low health it turns red)	Block)	
Hearts	- Note: this was not in the original plan but after talking to the designer this has now been added - Must boost the health of the block - Must appear randomly in the game - Must disappear after a certain number of seconds have passed		- The same amount of hearts will appear each level so they don't get easier and the health boost will always be the same	- HEALTHBOOST (int)	- appear() - disappear() - getEaten()
Iron Ore	- These have to appear in the level and then be minable by the block. They are mined by having the block touch the block for a certain about of time.		- The ore will have a maxMana and then as long as the Block is touching the ore, the ore will lose mana. When it's mana is zero then it will be mined and disappear	- MAXMANA(int) - manaCounter(int)	- appear() - disappear()
Diamond Ore	- Same as above		- Same as Emerald Ore but it will have a higher maxmana	- MAXMANA(int) - manaCounter(int)	- appear() - disappear()

Gold Ore	- Same as above		- Same as Iron Ore but it will have a higher maxmana	- MAXMANA(int) - manaCounter(int)	- appear() - disappear()
Emerald Ore	- Same as above		- Same as Gold Ore but it will have a higher maxmana	- MAXMANA(int) - manaCounter(int)	- appear() - disappear()
Skeletons	- They have to shoot the block and do damage to him	- Henry said that he wants these to have different strengths for different levels, that would be hard. The instance variables would have to be different depending on the level		- attackDamage (int) - direction (int) - health (int)	- spawn() - shoot() - die()
Zombies	- They have to track the block and do damage to him whenever they collide	- Same as above		- attackDamage (int) - direction (int) - health (int)	- spawn() - die()

Now I am going to take a look at the variables in a little greater detail. Also you should note that I am going to have all of these be local variables and then just use getters and setters. In reality a lot of the variables in the Block class are going to function like global variables because they will be changed from lots of other classes, but the reason I'm going to keep them local is for two player games you can have both have their own variables. The only thing that will be global are the variables from the Game item (like levelCount and stuff):

Item	Variable	Description / notes

Block	ironCount (int)	This is just going to keep track of how many iron ores the block has mined. It will decrement when the ore is used to purchase upgrades.
	diamondCount (int)	See above
	goldCount (int)	See above
	emeraldCount (int)	See avoe
	healthMax (int)	This is how much health the block has if it has taken no damage. This will only change if the health is upgraded.
	healthCount (int)	This keeps track of how much health the block has. It will decrement when the block takes damage from enemies. If this goes to zero the level restarts. This is reset to healthMax at the beginning of each level.
	isAlive (boolean)	This just keeps track of if the healthCount is greater than zero. If this is false then the level will reset.
	isShooting (boolean)	If the block is shooting then this boolean will be true. Otherwise it will be false. This is key information because the block cannot shoot and mine at the same time.
	isMining (boolean)	If the block is mining then this boolean will be true. Otherwise it will be false. This is key information because the block cannot shoot and mine at the same time. It's also important because when this is true we will know to decrement the mana of the block that is being mined.
	direction (int)	This is the direction that the mouse is pointing the aim of the block if it were to shoot. This will help us determine where to shoot projectiles.
	attackSpeed (int)	A rate of fire of your shots. Can be increased by buying upgrades.

	attackDamage (int)	How much damage each one of your shots does.
Game	levelCount (int)	This will just keep track of what level the player is on.
	multiPlayer (boolean)	This will keep track of whether we are in multiplayer or single player mode.
Level #n	MAXIRON (int)	This will keep track of how many irons will appear though a given level.
	MAXGOLD (int)	See above
	MAXEMERALD (int)	See above
	MAXDIAMOND (int)	See above
	ironMined (int)	This will keep track of how many irons have been mined in a given level. When this equals the MAXIRON then the level has been won.
	goldMined (int)	See above
	emeraldMined (int)	See above
	diamondMined (int)	See above
	LENGTH (int)	This is going to be how long time-wise a level will be. Henry wanted to have differing spawn rates for each level and different proportions of different types of ore. I think this creates a round-about way of accomplishing that. By dividing MAXGOLD (or any ore) by the length we get the spawn rate. This way to change the different MAXores and LENGTH for the levels to have different spawn rates for different ores. Once the length is over, if there are still left over ores the game will keep going in that enemies will keep spawning but no more ores will spawn.

	scelPerSpawn	
	zombiesPerSpawn	
Hearts	HEALTHBOOST (int)	This is just how much health this will give to the block.
Iron Ore (and all of the ores)	MAXMANA(int)	This is going to allow for a way for us to allow ore to be mined. More valuable ore will have higher MAXMANA so that they will take longer to mine.
	manaCounter(int)	This is going to start at 0 for each ore and increase at some rate as the block is touching the ore. When manaCounter = MAXMANA, the ore has been mined.
Skeletons	attackDamage (int)	How much damage they do to you when they hit you with a shot.
	direction (int)	Where they are aiming their shooting. It will be in the direction of the block. This is going to allow them to shoot at the block.
	health (int)	How much damage they can take before they can die.
Zombies	attackDamage (int)	How much damage they do to you when they collide with you.
	direction (int)	What direction they are moving in. This is going to be based off the blocks position so that they are following him
	health (int)	How much damage they can take before they can die.

Now I am going to take a look at the methods in a little greater detail.:

Item	Method	Description / notes

Block	moveUp()	Up arrow will make it move up. Multiple arrows can be pressed at a time so that block can move diagonally as well.
	moveDown()	WoLoG see above
	moveLeft()	See above
	moveRight()	See above
	shoot()	Will shoot little projectiles at a certain rate that will do damage to enemies. This will be triggered by clicking the mouse pad. The direction will be determined by the variable direction. While this method is being called isShooting will be true. isMining must be false.
	takeDamage()	This will lower the health of the block. Will also call loose health for the health bar.
	gainHeart()	This will raise health. Will also call gain health for the health bar.
	gainGold()	Will gain gold increasing the counter.
	gainIron()	See above
	gainDiamond()	See above
	gainEmerald()	See above
	mine()	This will be triggerd by the block touching an ore. It will make isMining true. It will decrement the mana of whatever ore it is mining
	getters and setters	These should be there for all the items. It just makes it easier to have a method in one item affect another object.

Game	restartLevel()	What it says. Gets called when you die or your health is zero.
	restartGame()	What says.
	switchGameMode()	Switches to multiplayer.
	pauseGame()	Just pauses game.
	startLevel()	Between levels the game should automatically pause. You will have to click start to begin the next level.
	upgradeHealth()	Increases healthMax. Decrements your ore count. Amount increases depends on what type or ore you use to purchase.
	upgradeMiningSpeed()	Increases attackSpeed. See above.
	upgradeAttack()	Increases attackDamage. See above.
Health bar	refill()	Bar goes all the way up
	looseHealth()	Goes down
	gainHealth()	Bar goes higher (if not full).
Hearts	appear()	Appearing at a random spot in the screen.
	disappear()	After 10 seconds of being if not eaten it will disappear.
	getEaten()	If the block collides with it then it disappears but the healthCounter increases

Iron Ore	appear()	Appearing at a random spot in the screen.
	disappear()	Once it is mined it disappears.
Diamond Ore	appear()	Appearing at a random spot in the screen.
	disappear()	Once it is mined it disappears.
Gold Ore	appear()	Appearing at a random spot in the screen.
	disappear()	Once it is mined it disappears.
Emerald Ore	appear()	Appearing at a random spot in the screen.
	disappear()	Once it is mined it disappears.
Skeletons	die()	Disappears
	spawn()	Appears
	shoot()	Shoots in the direction of block. If the bullets hit block then healthCount is decreased.
Zombies	die()	disappear
	spawn()	appear

Feasibility Analysis

To be completely honest I do not think that this is completely feasible. There are just lots of aspects going on here. One thing that I think should go is the upgrade store that happens between levels. I think this store is going to be extremely hard to make because you will have to create some space outside of the realm of just levels which seems to be a lot of extra work and then you have to be able to enter back into the realms of levels. It seems easiest to just have a smooth linear game that transitions straight from level to level. I also think another very difficult part of this game is the different types and number of ores and the increasing power of enemies as levels move on. Its just very hard to do this because you have to change the attributes of the objects each time you go to the next level. I don't think the gameplay would be lessened if instead of this, higher difficulty of levels was merely accomplished by increasing the volume of ores and enemies which appear (with only one form of ore). I also think the multi-player doesn't seem super easy or fun. It would also be complicated because then how would the upgrades work? If only one is mining then only one gets ore to use to buy upgrades. Overall though I really like the concept of the game but it sure does feel a lot like minecraft, and I can't really see anything desire satisfied in this game that is not already met or exceeded in minecraft. I'm not trying to bash the game or anything I just want it to live up to its potential.

After talking to the designer he said that he wanted to have some way to upgrade so instead of getting rid of the upgrade store we will just have three icons in the top left corner of the screen (armor, bow, pick) which when you have enough ores you can click on one of the icons to upgrade that aspect. The cost of an upgrade will start at 1 ore and every time you upgrade that specific characteristic the upgrade cost for that characteristic will increase by one. We will also only have 1 ore to make everything easier. That means in order to make the game harder as the levels increase we will just have more ore spawn with higher mana. Also I need to make some new limitations because Henry wants to have it so you can't step over ores so it must stop movement. Also we decided we would only code zombies and then make sceletons if we have time. Also we decided to have no multi-player mode.

So…. currently it is late and I didn't know there was a rubric for this thing so I think I missed a lot of stuff. I just saw the rubric right now so I'll try to patch up some things I didn't cover.

Data structures:

For data structures I can't think of too many places I will need them but maybe for levels I could create an array of levels. For anything that spawns (enemies, ores, and hearts) I want to store it in a type of data structure which allows me to add stuff so let's just say I'll use an arrayList.

Performance impact:

I don't know what this means.

Prioritize each feature:

1. Block: clearly the most important because everything is going to revolve around the block. All the interactions are going to change variables in the block class so nothing means anything without a solid code for the block.
2. Enemies: Maybe just make one enemy, but without any bad guys there is literally nothing in the game that is gonna kill you so it will be pointless.
3. Ore: You do not need to make 4 different types of ore but one is essential. Without ore there is no way to win.
4. Levels: Boring without multiple levels but a lot of work.
5. Game: If you have multiple levels you need some way to keep track of that.
6. Health bar: It's way more fun if you can see that you are about to die.
7. Hearts: These just don't add too much.
8. The store to buy stuff in between levels: This is just going to be super hard and not too crucial.
9. Multi-player: may mean some redesigning of the Block class and everything else. Super hard.

Functional code relational map:

In my description of the different methods I described which functions would call other functions.

What libraries should be included:

I don't know what libraries are.

Coding schedule:

Day 1: Just create the block and get him to move around
Day 2: Create ores and get them to spawn
Day 3: Get the block to be able to mine ore
Day 4: Create an ore counter in the corner of the screen (bottom right) which updates when ore is mined
Day 5: Create zombies and have them attack block
Day 6: Make the block able to shoot the zombies
Day 7: Create a health counter for the block and create the visual health bar which decreases when zombies hurt him
Day 8: Create the appearing hearts which give health to the block
Day 9: Create upgrade buttons in the corner of the screen (upper left) which allow the block to upgrade it's attack, mine speed, and health
Day 10: Make it so the block can win or lose (if he loses health he dies, and limit the number of ores that spawn such that if he mines all them he wins)
Day 11: Make more levels with different spawn rates
Day 12: Create convenient gameplay features like pause game or play earlier level or pause screen between levels

EXTRA DAYS FOR THE CHASES OF THE WORLD

Day 13: Introduce a skeleton villain alongside the zombie (he shoots at the block rather than touches him)

Day 14: Introduce different sprites for upgrades. Ex/ when you upgrade armor three times the block is wearing different armor

Day 15: Introduce all three different types or ore and figure out some way to do conversion for buying upgrades with different ores

Day 16: Introduce a multiplayer mode