**Bark of the 90’s Game Proposal**

Team: Team 5

Members: Kodiak Shepard, Matthew Satterfield, & Rebecca Tester

Development Platform: Unity

**High Concept Statement**

A skater needs to rescue their dog from criminal Ella McDogH’tor by skating through a 90’s-inspired city and shooting enemies.

**Gameplay**

**Gameplay Description**

The player will be able to move in all directions across the screen. The player will be moved forward through the level automatically. Obstacles will appear that the player needs to dodge. Enemies will spawn and shoot bullets that will harm the player if they come in contact with them. At the end of each level is a boss that the player must defeat to beat the level.

**Controls**

WASD to move. Space to dodge. Shift/Ctrl to perform a trick. Use the mouse to aim and left click to fire.

**Genre**

Our game combines three genres: shoot ‘em up, like Galaga, rail shooter, like Star Fox, and bullet hell, like the Touhou Project series.

**Gameplay Elements**

While on their journey the player can pick up Power Up’s and health items.

There is a point system. The player gains points whenever they defeat an enemy, pick up an item, or perform a trick.

**Length of Play**

There will be 3 levels. The player is encouraged to replay levels to improve their highscore.

**Game Flow**

Main Menu → Level Select → Level 1 → Level End Screen → Level 2 → Level End Screen → Level 3 → Level End Screen → Win Screen → Main Menu → Exit Game

**Plot**

The main character is a skater trying to get their dog back from a band of cartoonish criminals run by Ella McDogH’tor. The story takes place in the future during the 2090’s, where the 1990’s aesthetic has made a comeback.

**World Details**

The game takes place during the 2090’s, a time period with futuristic technology like flying vehicles, but also with a nostalgic 1990’s aesthetic. Skateparks are staples in every city, skating being a very popular sport. The protagonist’s hometown is a bustling city with a bit of a crime problem. Many of the buildings of the city are very old and covered in graffiti, juxtaposed with the high-tech flying cars on the streets.

**Backstory**

The protagonist of unspecified gender is a skater who owns a swaggering dog and enjoys innocently walking their dog through dark alleys.

Unfortunately, the protagonist’s city has a local band of criminals, run by Ella McDogH’tor, who stalk the alleys.

**Central Conflict**

Ella McDogH’tor has stolen the main character’s dog and has sent goons to prevent them from getting their dog back.

**Motivation**

The player needs to rescue their dog and must survive the criminals sent to stop them.

**Targets**

**Target Audience**

Our target audience is millennials, people who grew up in the 90’s and are familiar with the aesthetic. We expect men to be more inclined to play our game. Casual gamers and arcade fanatics will also be drawn to our game.

**Target Platform**

Our target platform is PC, because using a reticle to aim will require some precision with the mouse.

**Target Rating**

Our game will have an ESRB rating of E / E10 due to its cartoony style and minimal violence.

**Artistic Style**

The game will have pixel art sprites and backgrounds with a cartoonish style. The setting will be a city with bright signs and colorful graffiti. The menus will be simple and have an arcade game feel.

**Project Summary**

The designer for our game is Kodiak and the programmers are Rebecca and Matthew.

We are creating a bullet hell / rail shooter / shoot em’ up game about a skater rescuing their dog in a 90’s-inspired city, using Unity with the help of Github Desktop.

Our production will follow the week 3, 6, and 9 milestones.

**Development Platform**

Our game will be made using Unity due to its various tools and our good experiences with it, and we will use Github Desktop to help with version control.

**Project Timeline**

Rebecca will write the user stories by Thursday, Week Two.

Matthew will create object lists by Thursday, Week Two.

Kodiak will create the placeholder artwork by Friday, Week Two.

Rebecca will create the flowcharts by Friday, Week Two.

Matthew will have the player moving in the first prototype by Friday, Week Two.

Rebecca will expand the pitch document into a Game Design Document by Saturday, Week Two.

Kodiak will create the storyboard by Monday, Week Three.

Kodiak will create the tone documents by Tuesday, Week Three.

*Functional* Main Menu, Instructions, Level Select, Level End Screens, and Win Screens will be done by Friday, Week Three.

*Functional* prototype of Level 1 will be done by Saturday, Week Three.

*Functional* prototype of Level 2 will be done by Saturday, Week Three.

*Functional* prototype of Level 3 will be done by Saturday, Week Three.

*Final* version of Level 1 will be done by Saturday, Week Six.

*Final* version of Level 2 will be done by Wednesday, Week Eight.

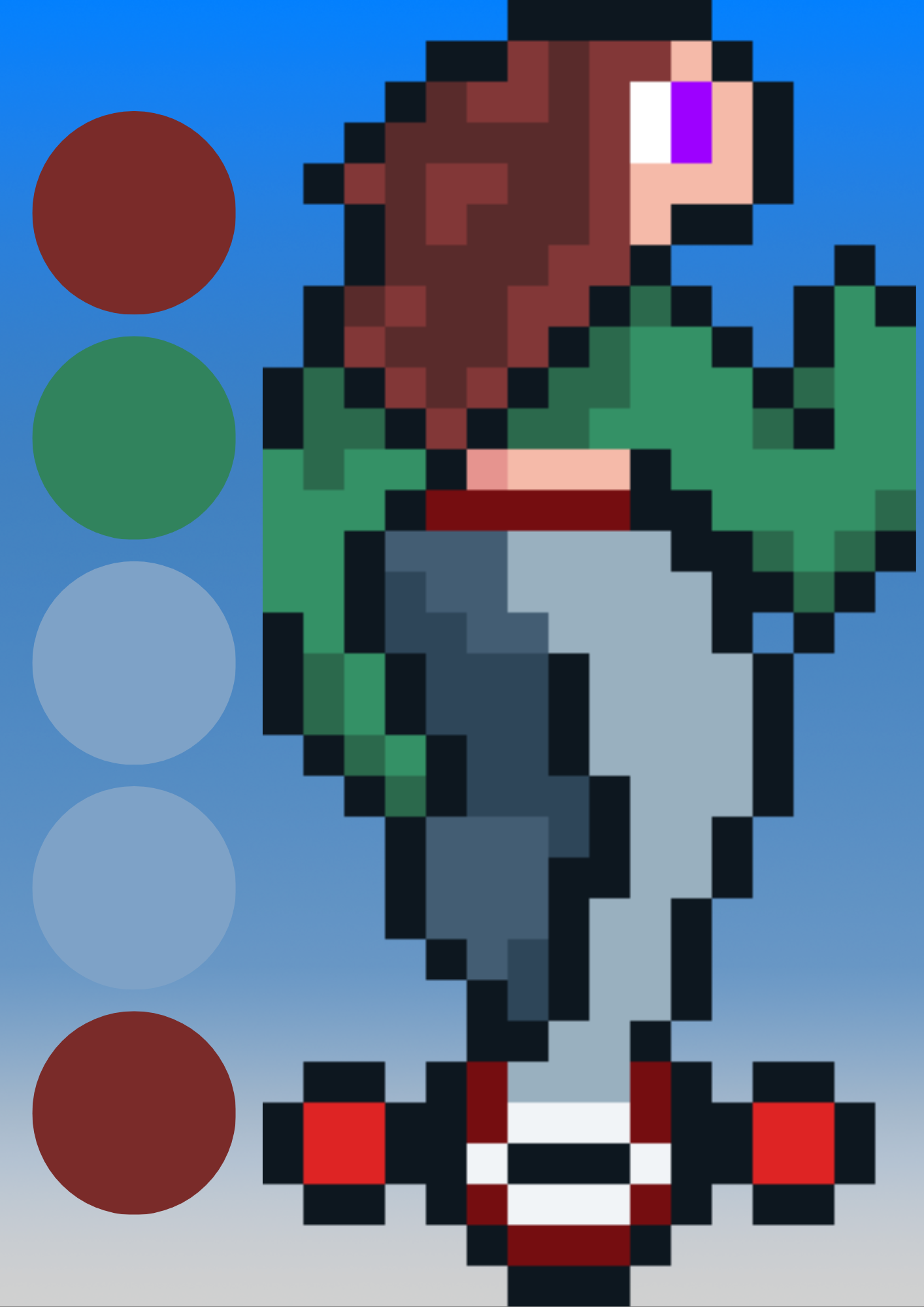
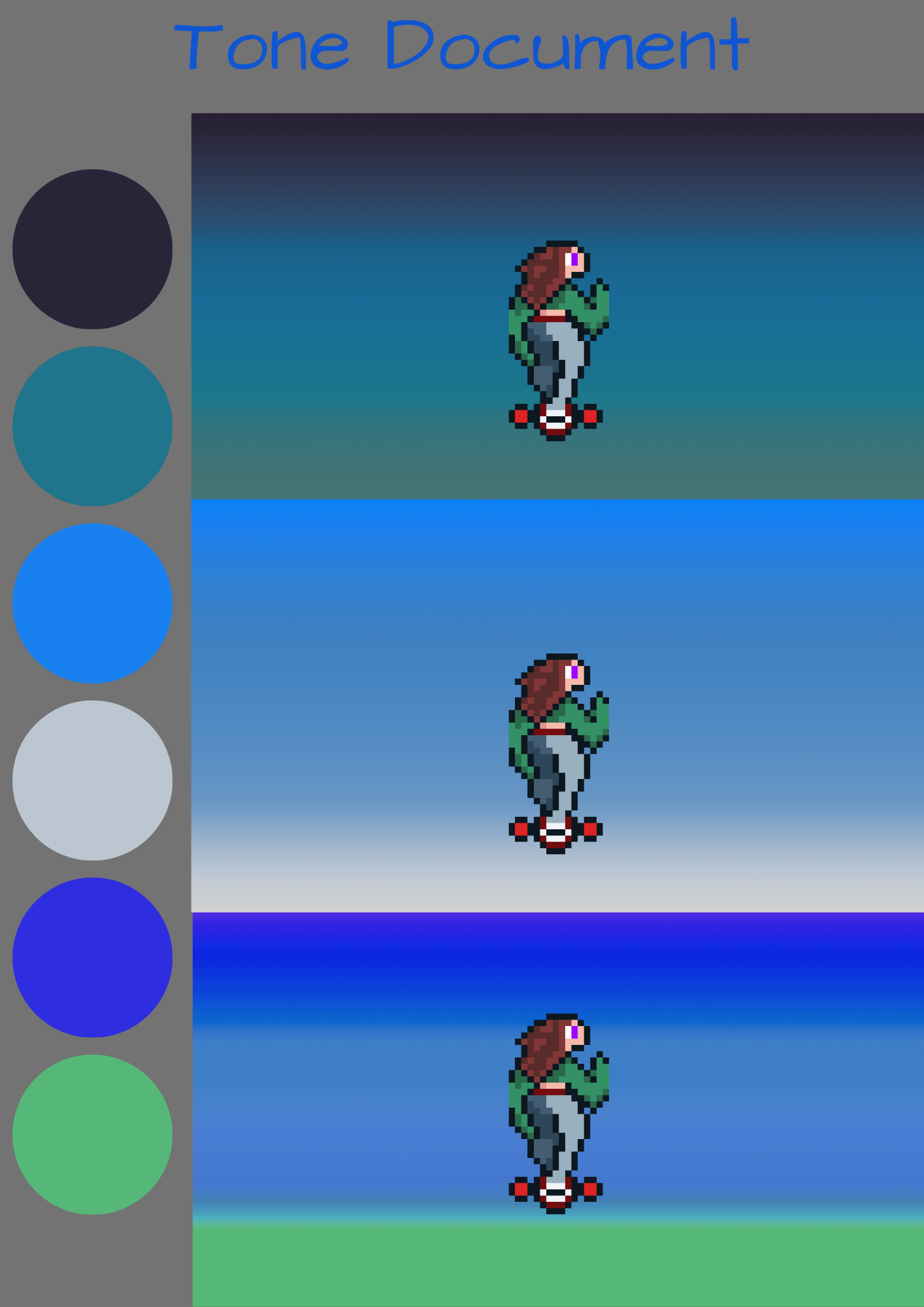
*Final* version of Level 3 will be done by Saturday, Week Nine.

*Final* Main Menu, Instructions, Level Select, Level End Screens, and Win Screens will be done by Saturday, Week Nine.

*Final* art assets will be implemented by Saturday, Week Nine.

**Tone Documents**

We are going to use mostly blues, but will incorporate some colors that are meant to class with the blue so the player stands out more.

.

**Sample Art**

For the background we will use a gradient, there will be 3 layers. A top, a middle, and a bottom. On the bottom and top parts we will add assets that move giving the appearance of a moving background on a 2d plane. As the player progresses the background gets more colorful and vibrant.

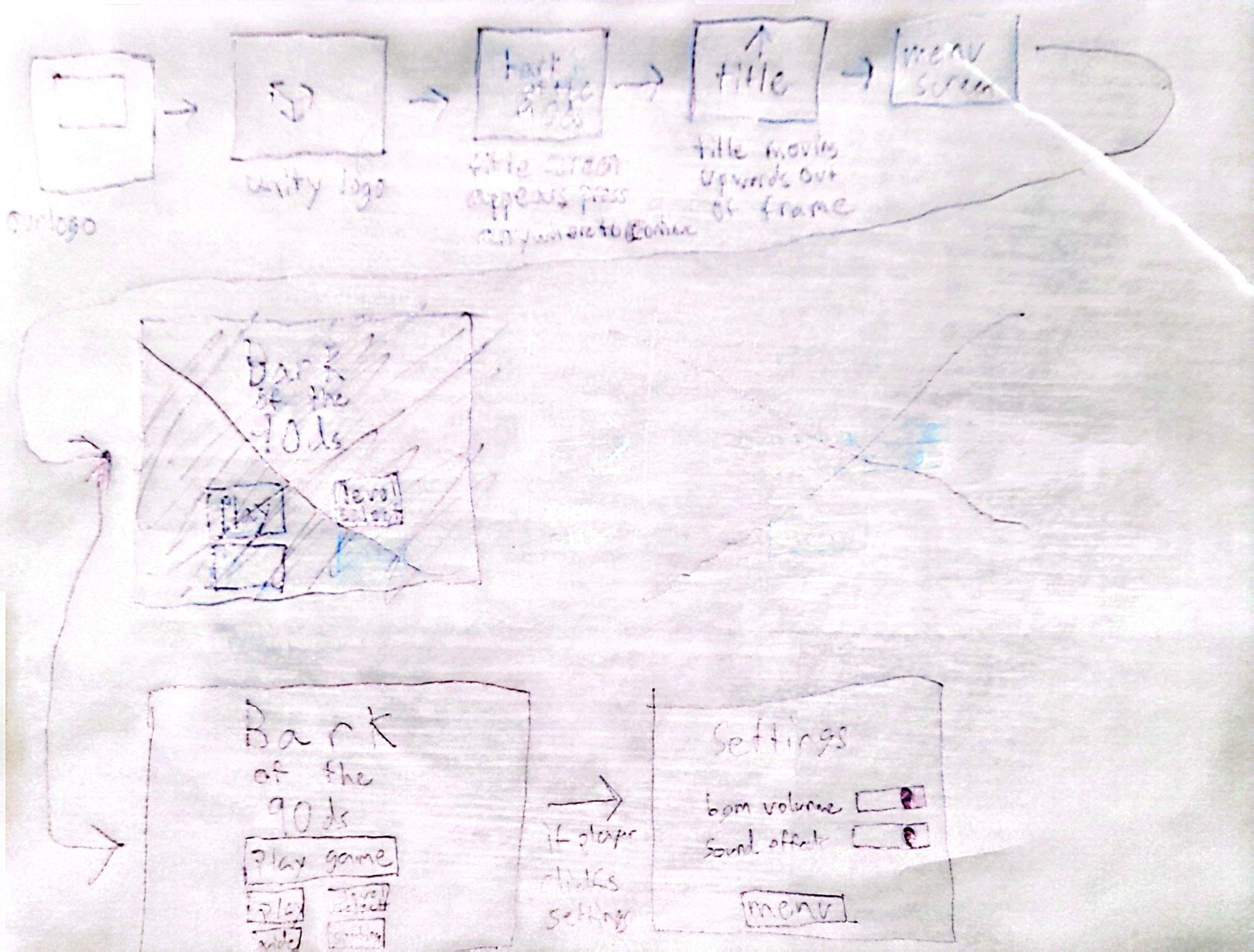
Sample art:

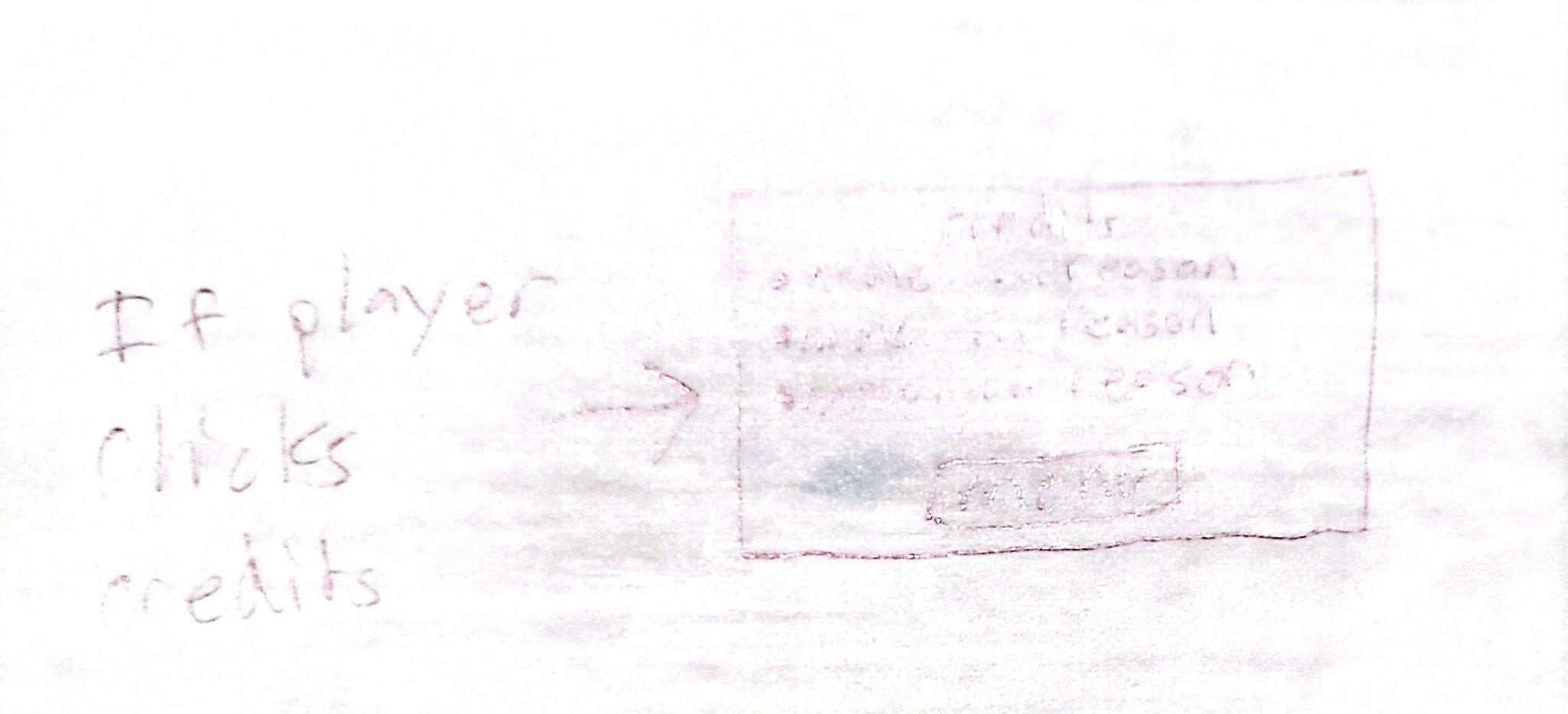


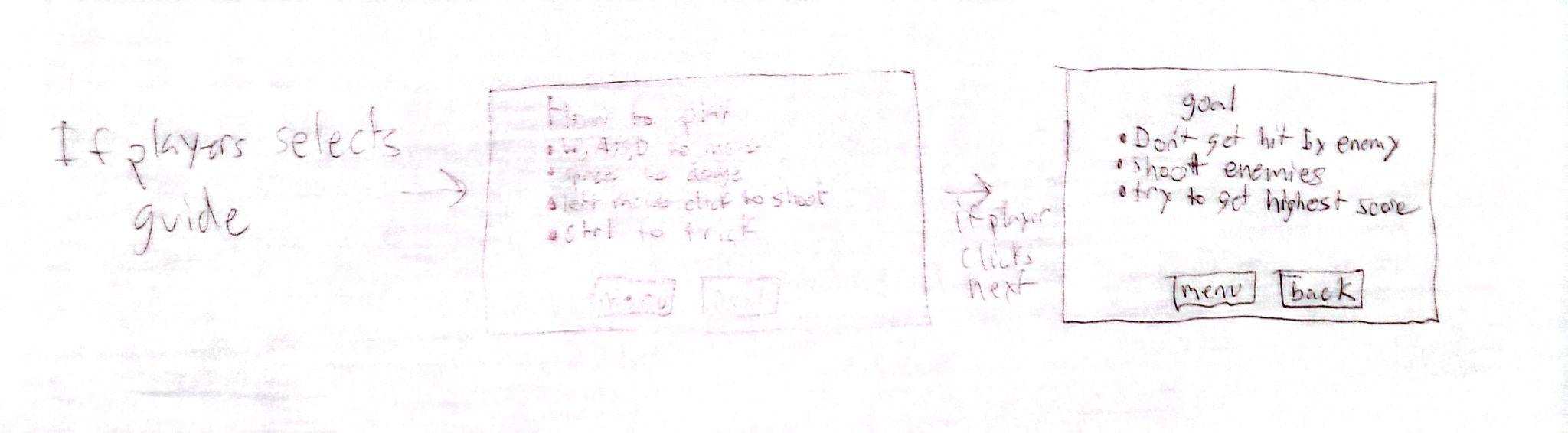
For the player we will use 16 by 32 pixel art

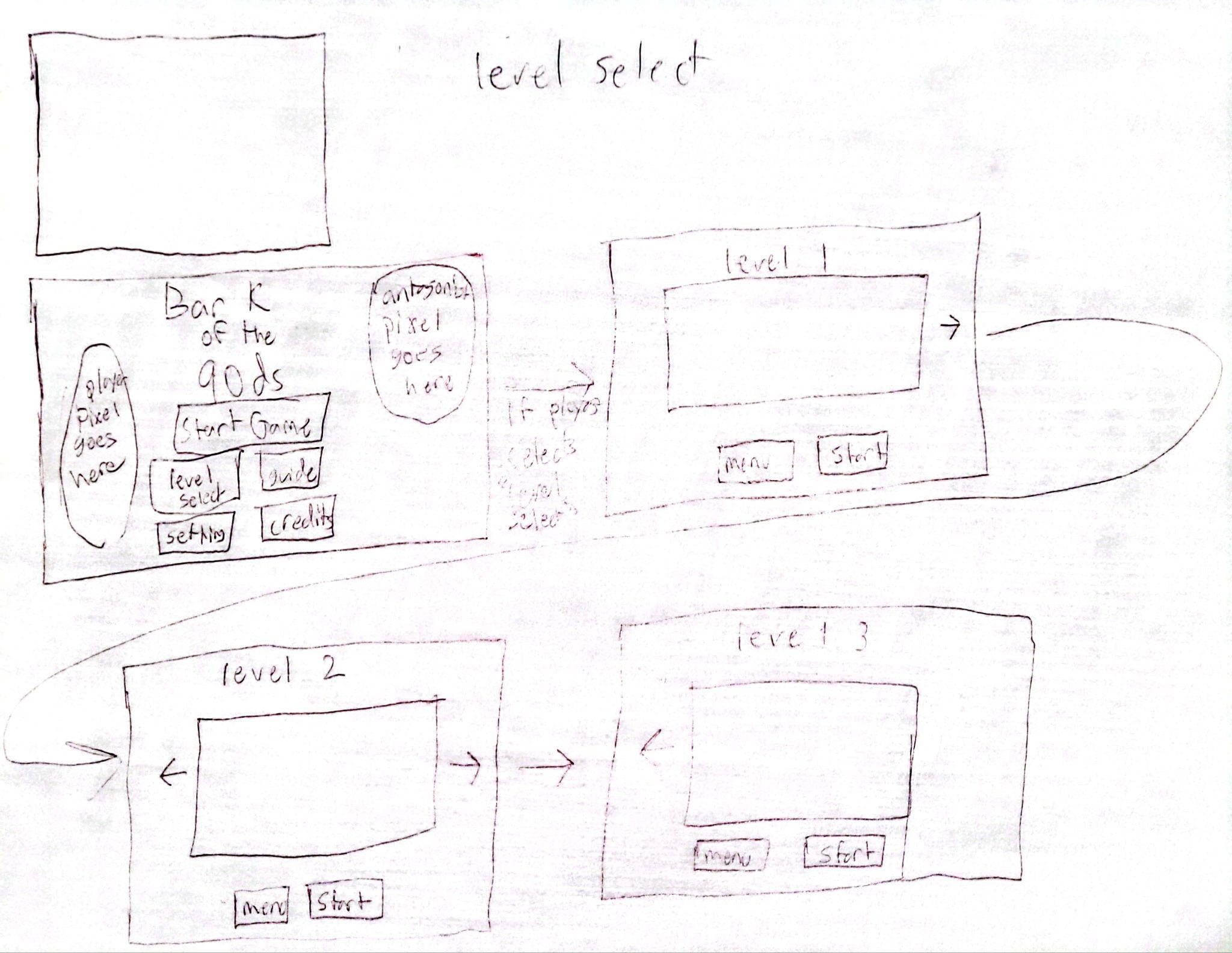


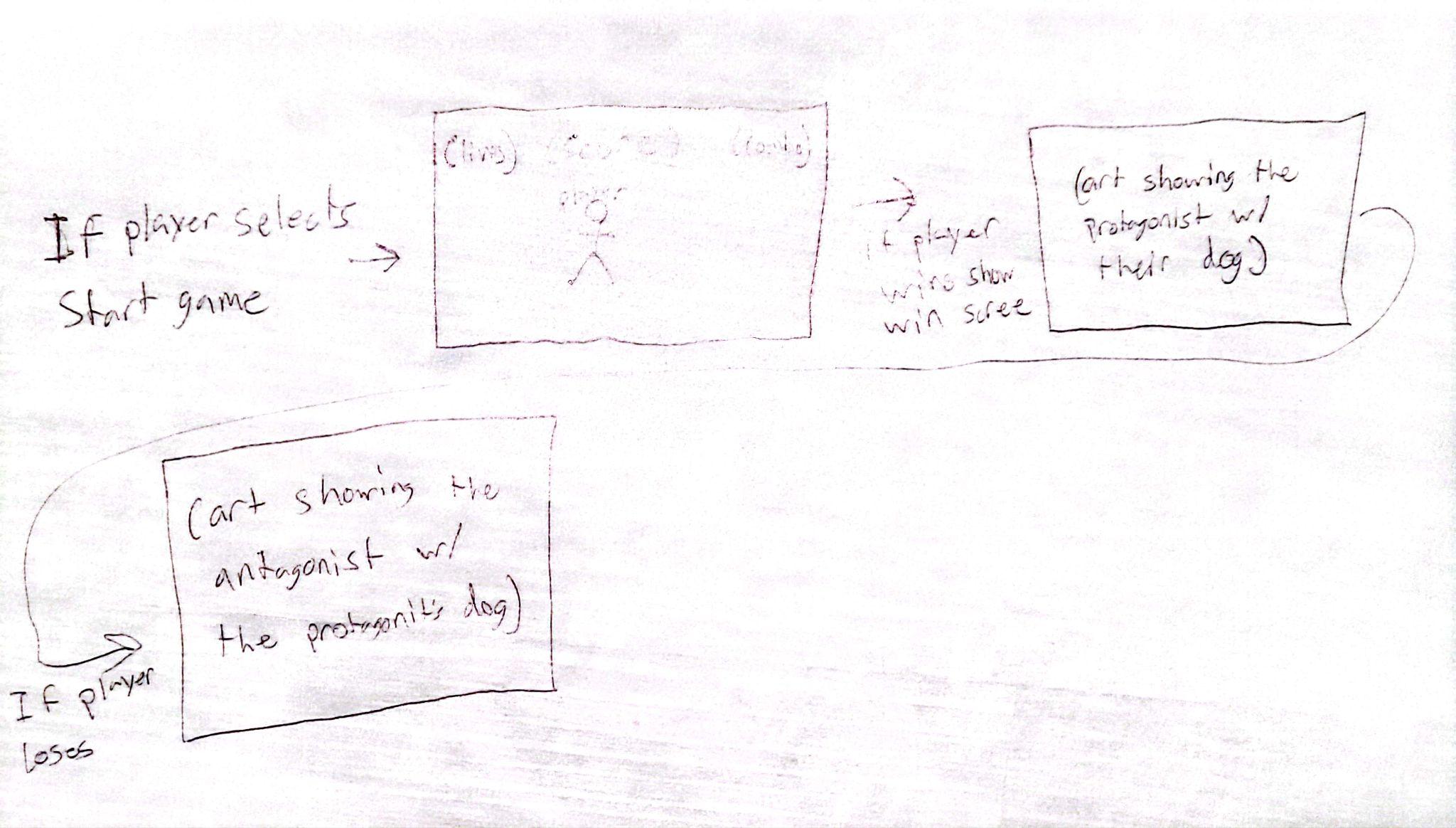
**Storyboard**

****

****

****

****

****

**User Stories**

As a player

I want to see my score go higher on the screen

So I can know that I’m doing something.

As a player

I want the enemies to flash when I attack

So I can tell that I’m doing damage.

As a player

I want a replay button

So I can replay levels easily.

As a player

I want an instructions/control screen

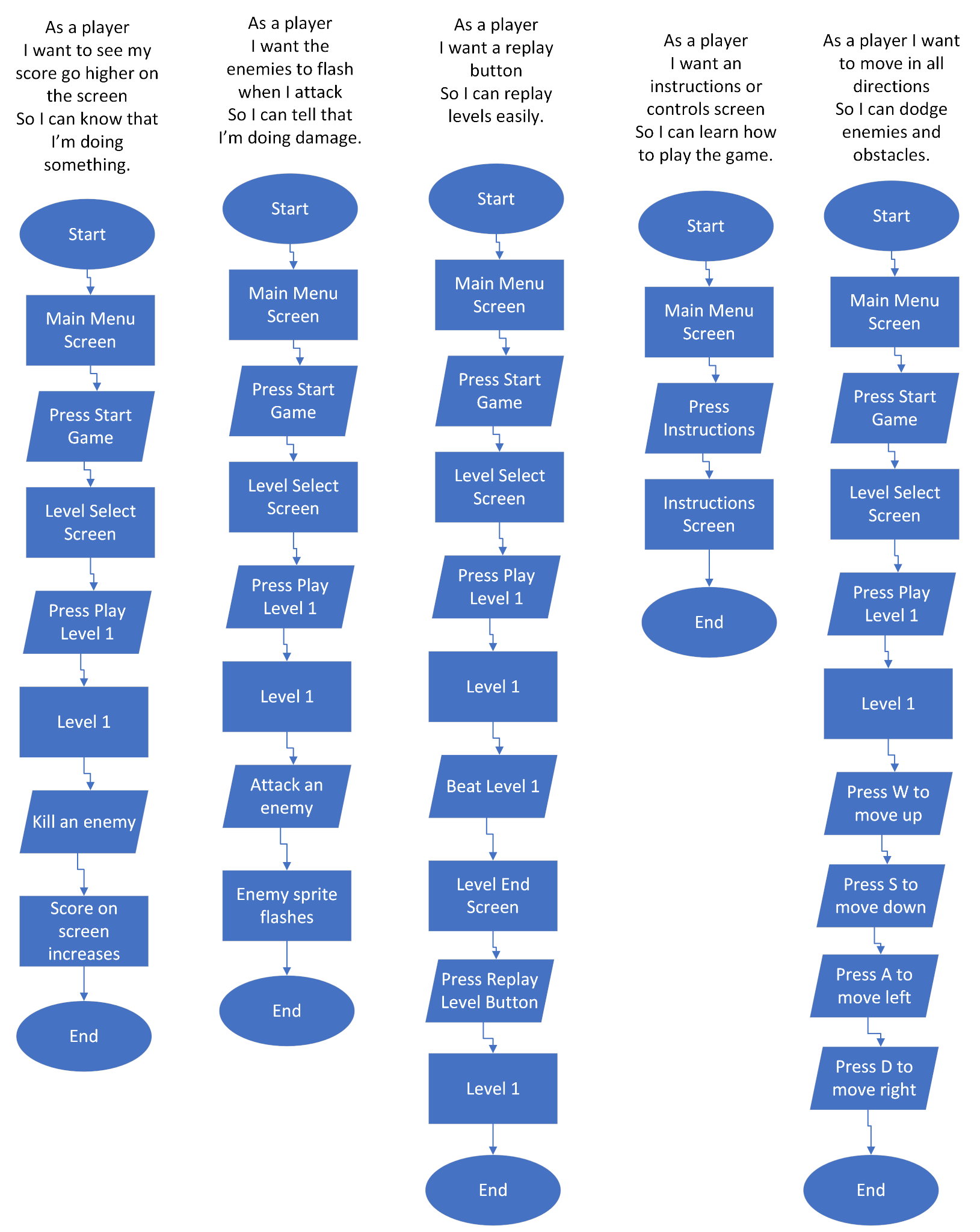
So I can learn how to play the game.

As a player

I want to move in all directions

So I can dodge enemies and obstacles.

**Flowcharts**

****

**Objects List**

| Object | Class | Methods | Attributes | Relationships |
| --- | --- | --- | --- | --- |
| Player | PlayerScript | -Movement  -Take damage  -Trick  -Dodge  -OnTriggerEnter2D | -Health  -isAlive (could be method)  -Speed  -Dodge speed / multiplier  -Dodge cooldown  -Trick animations  -Trick score value  -Sfx for movement / trick / dodge / taking damage  -Particles for movement, attacking, death | -Reference to GameManager for changing score  -OnTriggerEnter2D:  ~Health calls TakeDamage to increase health  ~Enemy projectile calls TakeDamage  ~Power up increases score |
| Reticle | ReticleScript | -MouseMove  -Normal attack  -Charge attack | -Size  -Color  -Fire rate  -Sfx for attacks  -Charged percent  -Normal dmg  -Charge dmg  -Charge rate | -Reference to player for spawning projectiles  -Damage amounts may be inherited by projectiles |
| Power up | PowerUpScript |  | -Score value  -Collect sfx  -Collect particle  -Speed modifier  -Attack modifier | -Collected by player |
| Health item | HealthItem |  | -Health amount  -Score value  -Collect sfx  -Collect particle | -Collected by player |
| Enemy projectile | ProjectileScript | -Movement | -Damage receiver (player tag/mask)  -Damage amount  -Speed  -Direction/Angle  -Size  -Spawn delay | -Hits player |
| Player projectile | ProjectileScript | -Movement | -Damage receiver (enemy tag/mask)  … | -Hits enemy |
| Enemy Type 1 | EnemyScript | -Movement  -Take damage  -Attack  -OnTriggerEnter2D | -Health  -Speed  -Target pos / move pattern  -Sfx for attacking / taking damage  -Score value  -Damage amount  -Attack pattern  -Death particle  -Attack particle | -Damage amount may be inherited by projectiles  -OnTriggerEnter2D:  ~Player projectile calls TakeDamage  -Reference to Player for potential aimed attacks |
| Enemy Type 2 | EnemyScript | … | … | … |
| Enemy Type 3 | EnemyScript | … | … | … |
| Boss 1 | BossScript / EnemyScript | -BossDie  … | … | -Triggers level end when defeated. (With Menu or Game Manager)  … |
| Boss 2 | BossScript / EnemyScript | … | … | … |
| Boss 3 | BossScript / EnemyScript | … | … | … |
| Game Manager | GameManager | -Change / Increase Score | -Level Score(s)  -Total Score  -High Score(s)  -Levels unlocked | -Is referenced by many other classes/objects |
| Menu Manager | MenuScript | -QuitGame  -ChangeScene | -Previous scene | -Reference to GameManager levels unlocked |
| Score Label | ScoreLabelScript | -Get score | -Dropdown (enum):  Level score, total score, and high score | -Reference to GameManager to get score |
| “Curtain” for scene transitions | GameManager |  | -Target position  -Speed / Rate | -Child of Game Manager |
| Pop-up / trick callouts |  |  | -Text  -Color  -Lifetime | -Instantiated by player performing a trick |
| Attack pattern prefabs |  |  |  |  |