Name Tristan Clawson   Mark ______________________/50

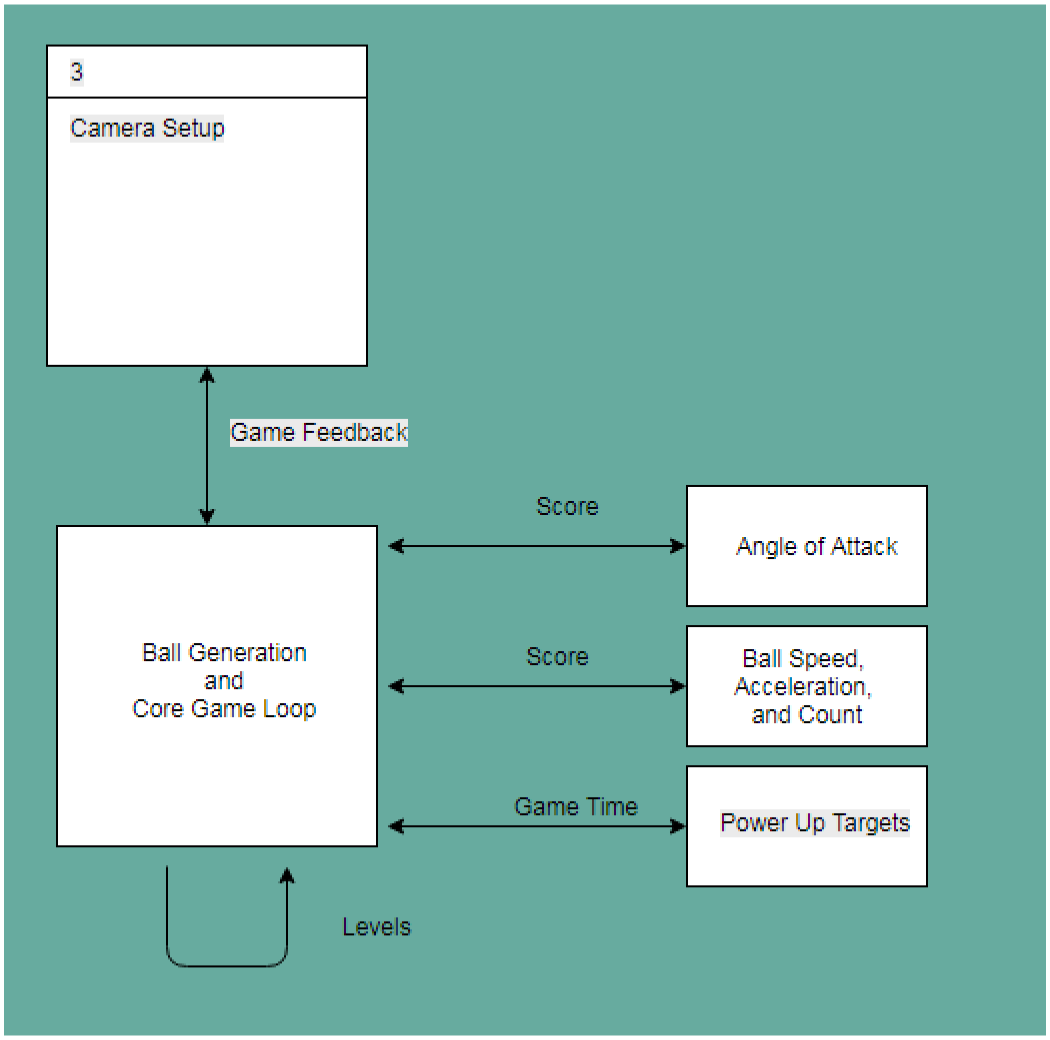
1. Brief introduction __/3

My code will generate balls and fire them at the player. This will vary the way the balls are fired based on variables. Later, this code will also include targets for the player to hit that give various bonuses.

2. Use case diagram with scenario __14

Example:

Use Case Diagrams



Scenarios

Ball Generation and Core Game Loop

Name: Add Numbers

Summary: The accountant uses the machine to calculate the sum of two numbers.

Actors: Player.

Preconditions: Game and skybox have been initialized.

Basic sequence:

Step 1: Angle of Attack is calculated.

Step 2: Ball speed, velocity, and count for the level are determined.

Step 3: Balls are generated and launched towards the player.

Step 4: Power ups spawn.

Step 4: Once the arbitrary score is reached, balls will stop being spawned. Balls will then de-spawn, and a new level will start with added difficulty.

Exceptions:

Step 1: Balls start going at such a velocity that the player can not keep up. The display may not even keep up depending on framerate.

Step 2: The player will lose life until the game ends.

Step 1: The score will go beyond the arbitrary limit.

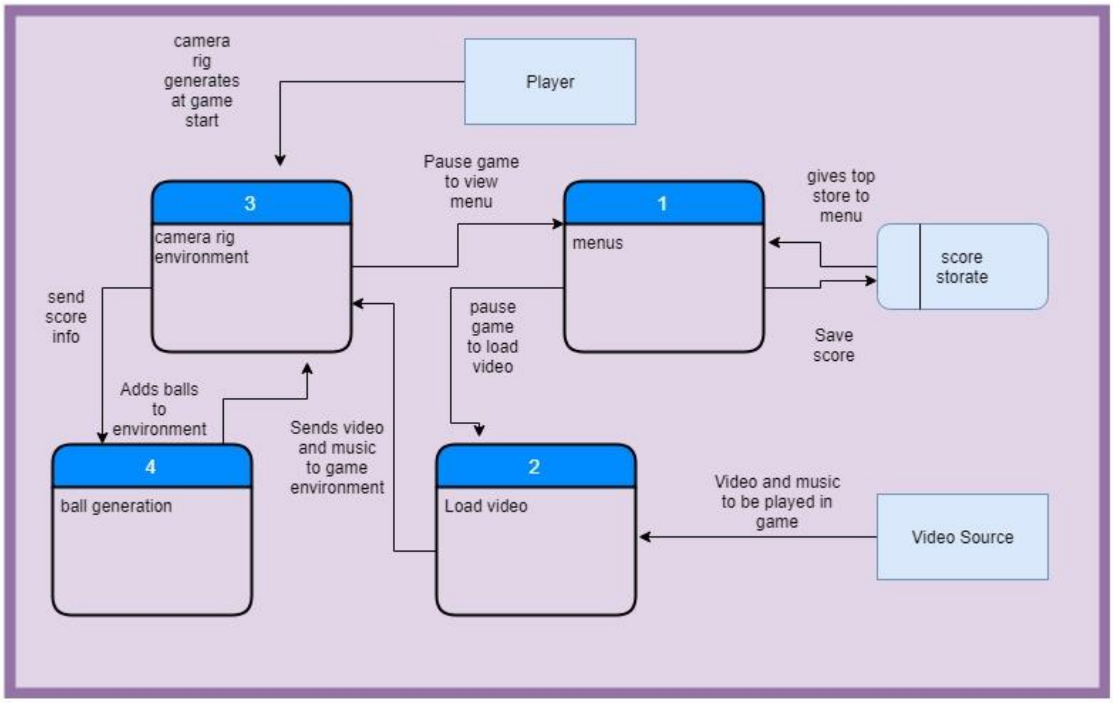
Step 2: The score will become unsigned and show negative.

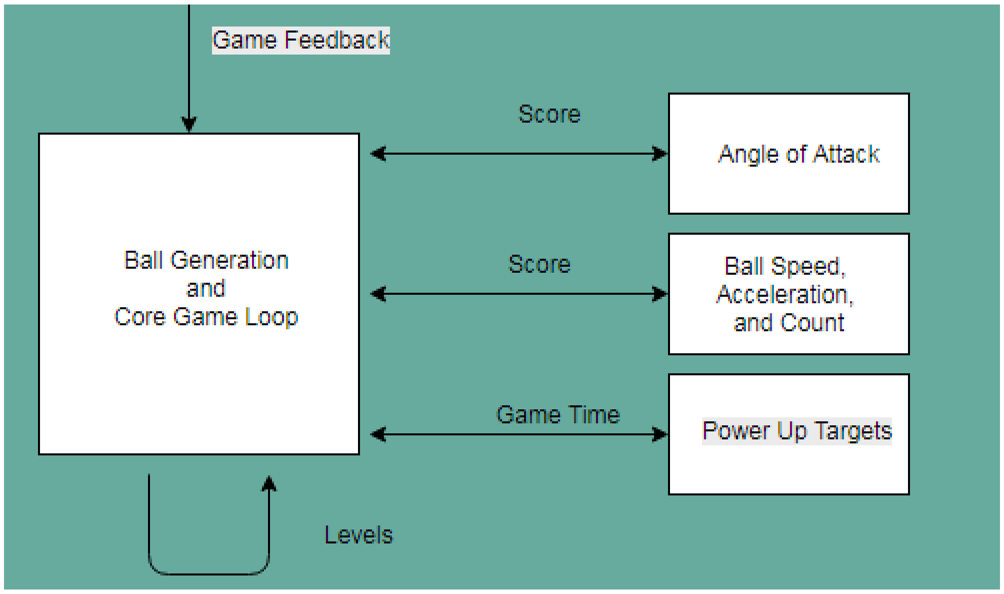
Post conditions: Game is saved and or exited through the menu.

3. Data Flow diagram(s) from Level 0 to process description for your feature ______14

First, the game will load all necessary assets for the skybox and menu. Then, from within the game, the menu will spawn. The player can choose to start a new game. (Note: the in game video functions are beyond the scope of this feature.) Once a new game starts, the balls will be generated and launched at the player for them to deflect. Once the arbitrary score has been reached, the level will reset by de-spawning all the balls and power ups. The angle of attack, velocity, and count will be recalculated. This loop will continue until the player saves the game and or exits it.

Data Flow Diagrams




Game Feedback
Score
Angle of Attack
Ball Generation and Core Game Loop
Score
Ball Speed, Acceleration, and Count
Game Time
Power Up Targets
Levels

Tidy up

Process Descriptions

Load skybox then menus

WHILE the game is active

```
    IF score is less that arbitrary
        Generate balls and launch them at the player
        Once deflected, send a message to the score keeper
        If a ball is not deflected, send a message to remove life
        If a ball hits a power up, activate the power up
    ELSE IF go to the next level
        De-spawn the balls and power ups
        Re calculate angle of attack, ball velocity, and ball count.
        Start this loop again.
END WHILE
```

4. Acceptance Tests _______9

Ball hits paddle and is reflected

Ball hits paddle at extreme angel and is deflected appropriately

Ball is reflected back into another ball, and both are affected appropriately

Ball reacts with (scores and de-spawns) player hit box accordingly

Balls and power ups spawn when called for

Balls and power ups de-spawn when needed

FEEL:

Skill level ramps appropriately, and players become gradually better over time

Players feel like it was their mistake, and not a game bug, that caused them to lose

Example for stats for level one

Use console out commands to print values to the screen or a file

The first level may have the following settings depending on user feedback:

- Angle of attack = 30
- Ball Speed = 20
- Total balls spawned = 25
- Delay between each ball = 2
- Power Ups count = 2
- Power Ups de-spawn all current balls when hit, and increase the score
- Ball deflection is counted

    - Life is set to 3, and decremented on each hit
    - When life = 0, then game over
- **Example of output:**

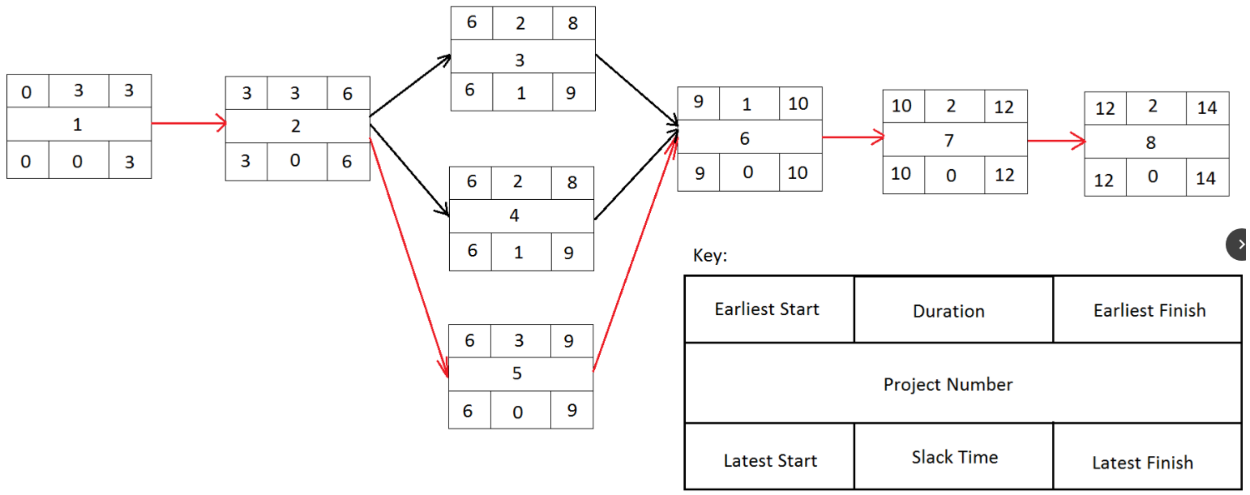
Angle of attack	30
Ball Speed	20
Total balls spawned	25
Ball Delay	2
Power Ups spawned	2
Power Ups used	1
Ball deflections (points)	24

5. Timeline _______/10

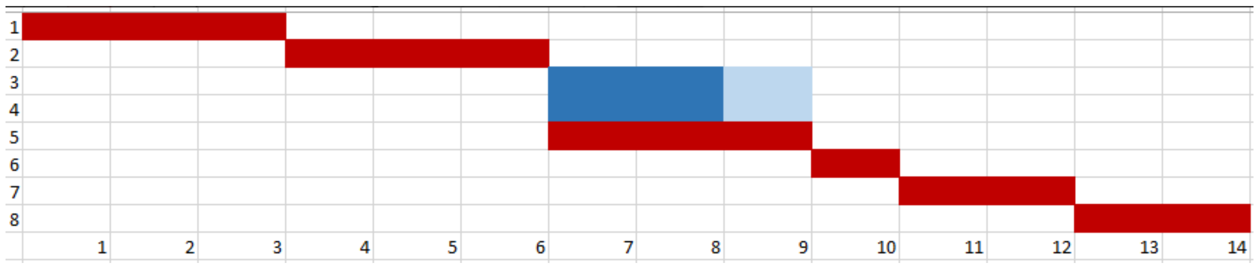
Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Outline of Game, and Brainstorming	3	-
2. Ball Generation	3	1
3. Angle of Attack, Speed Velocity, Acceleration, and Rate of Increasing Difficulty	3	2
4. Testing	1	3,4,5
5. Power Up Targets	2	6
6. Balancing	2	7

Pert diagram



Gantt timeline


1
2
3
4
5
6
7
8
1 2 3 4 5 6 7 8 9 10 11 12 13 14