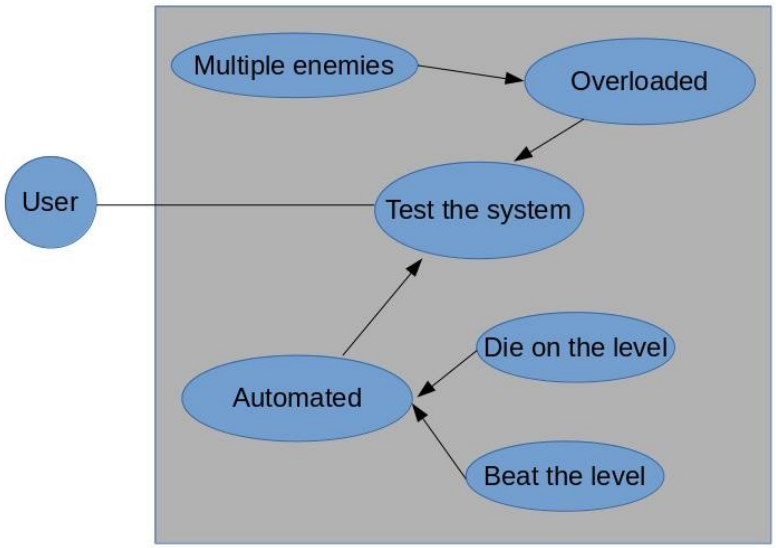
Name_____Dustin Pierce____ Mark _____/50

1. Brief Introduction __/3

   My feature is to test the game with automated code. This will include one automation that the player will go through and win every level and one time that the player dies on each level. In addition, I will be stress testing the system with multiple enemies, a variable number of coins collected, and so on.

2. Use case diagram with scenarios __/14

   Use Case Diagram:



Scenario:

Name: Test the system

Summary: The user connects to the system and either feeds it a script for each level or overloads the amount of enemies and plays it normally.

Actors: User who will test the game.

Preconditions: The game has been created and is ready to be tested.

Basic Sequence:

Step 1: The user writes a script to beat each level.

Step 2: The user writes a script to die on each level.

Step 3: The user writes game code so that the game can accept and deal with scripts.

Step 4: The user runs the game using both scripts for each level.

Step 5: The user overloads each level with a ton of spawning enemies.

Step 6: The user plays each level.

Exceptions:

Step 4: If the game crashes, glitches, or freezes then the game creator must edit the game code.

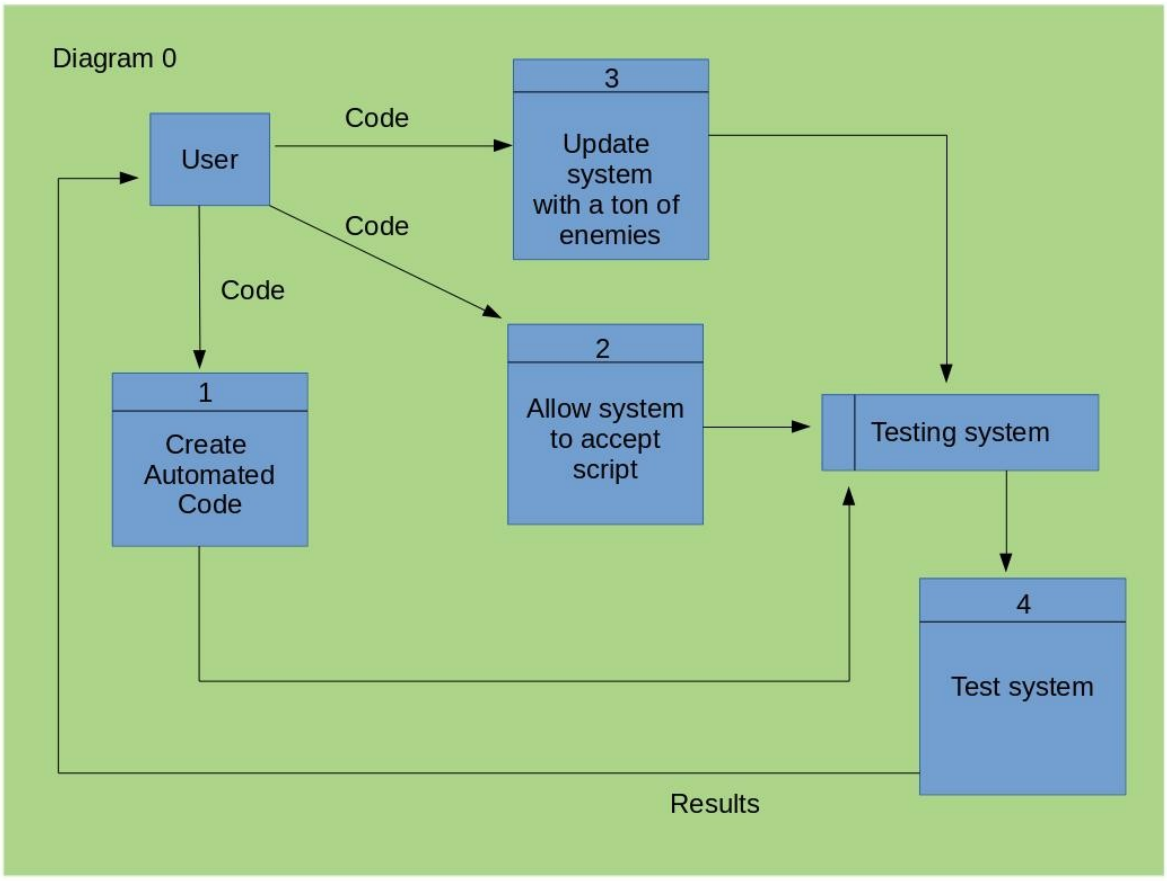
Step 6: If the game crashes, glitches, or freezes then the game creator must edit the game code.

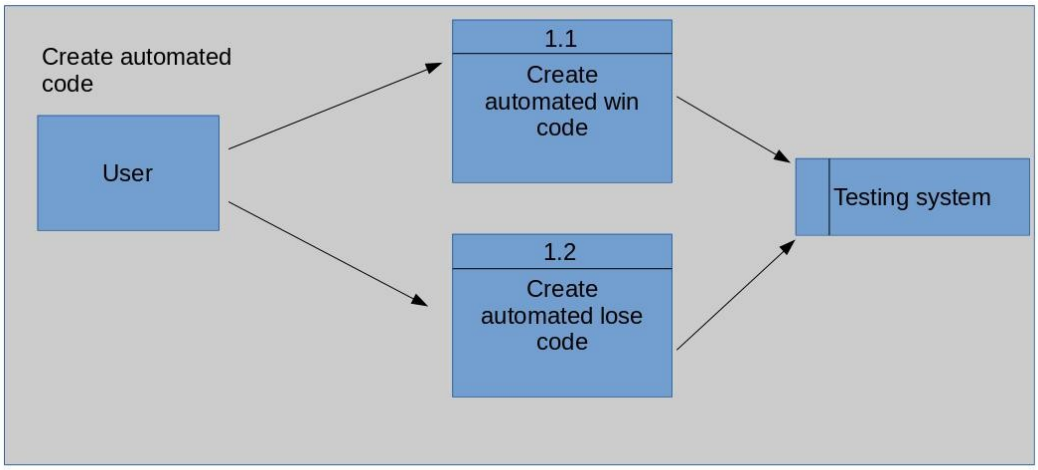
Post Conditions: The system is ready for release.

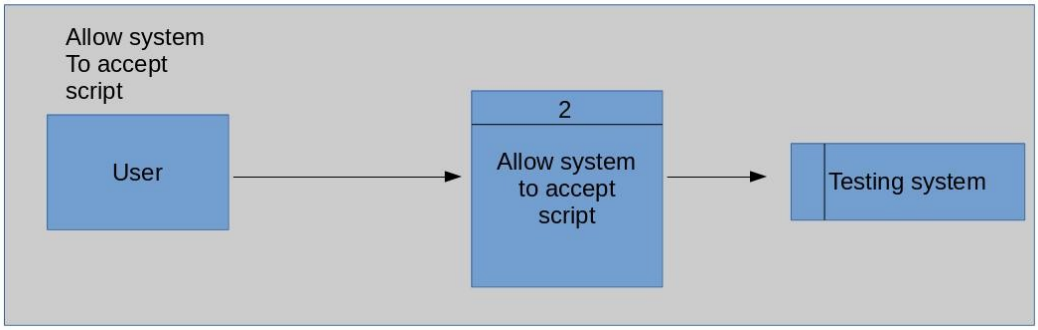
Priority: 2

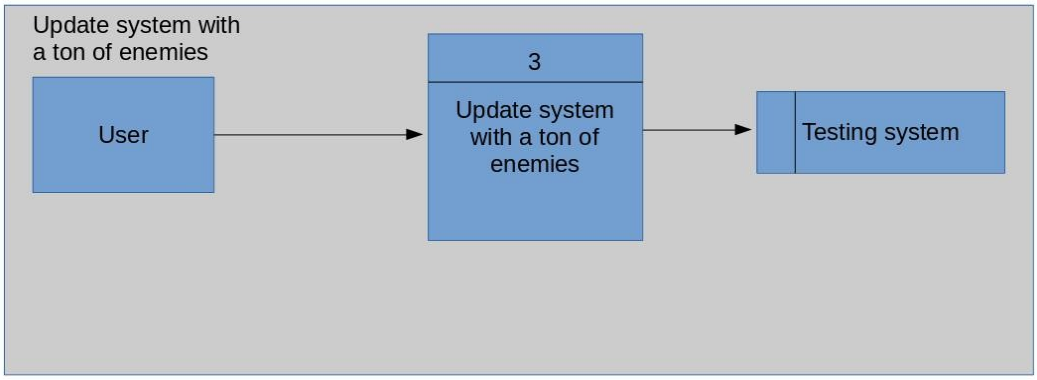
ID: TS1

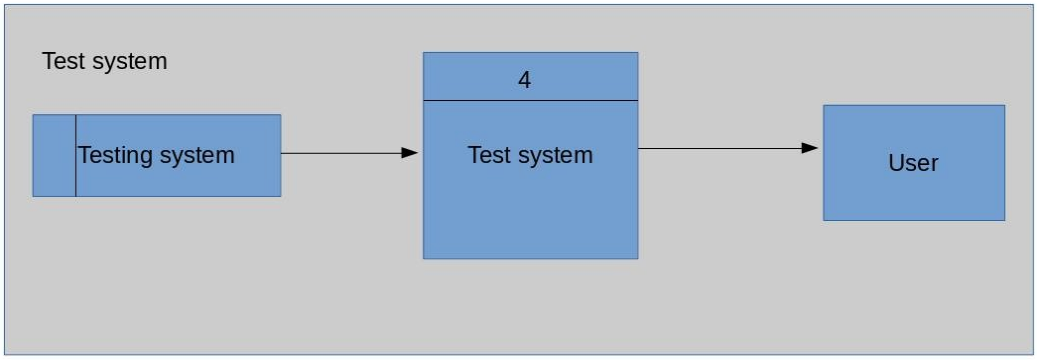
3. Data Flow diagram(s) from Level 0 to process description for your feature __/14




Create automated code
User
1.1
Create automated win code
1.2
Create automated lose code
Testing system


Allow system To accept script
User
2
Allow system to accept script
Testing system





Process Descriptions:

Create automated win code:

Create code to automate a win at each level. This code will allow the player to avoid or kill enemies, collect coins or other items and kill the boss. It will be written as a script of input keys that will finish the level in the best possible time.

Create automated lose code:

Create code to automate a 'death' on each level. This code will go through each level but at some point it will 'die'. This will happen by either falling off of an object or running into an enemy. This will be written as a sequence of commands in a script that will be passed to a running system.

Allow system to accept script:

The game system must be modified to accept a script for the automated testing. This should not be too hard!

Update system with a ton of enemies:

The system must be tested with an overload condition so

I will overload it with multiple enemies. This will be updated when they spawn and is an exaggerated real time scenario.

Test system:

The system will then be tested using the two automated scripts as well as the mass enemies condition. The system will be evaluated based on crashing, freezing, and glitching.

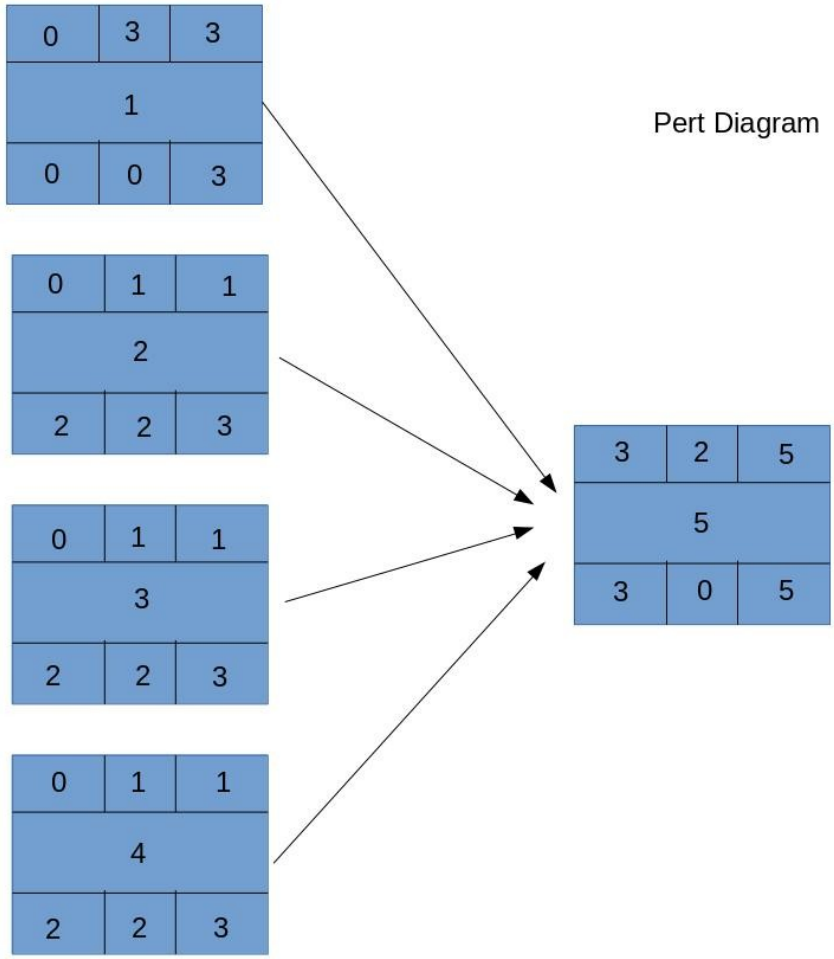
4. Acceptance Tests __/9

Each of the three tests will be run 10 times. Glitching, freezing or crashing will be recorded during these 30 tests. If any of these occur, the game creator will be notified and the game will be modified.

5. Timeline __/10

Work Items:

Task	Duration	Predecessor Task(s)
1. Allowing scripts	3 w	Game must be made
2. Creating win script	1 w	Game must be made
3. Create lose script	1 w	Game must be made
4. Update enemy count	1 w	Game must be made
5. Test system	2 w	1,2,3, and 4


Pert Diagram
0 3 3
1
0 0 3
0 1 1
2
2 2 3
0 1 1
3
2 2 3
0 1 1
4
2 2 3
3 2 5
5
3 0 5

Gnatt timeline