INVASORESS(Phaser-JS)

Group XIV:-

Daniel Sam Paul

Gautam Balamurali

Harikrishnan Menon

Problem Statement

- When we look at the various types of games around us, we can notice that there is only a linear type of process.

- This tends users to have less interest in such games.

- So to make a change in this we have used a system which when used in a different way can make it a little more challenging .

RELATED WORKS

GAMES USING GENETIC ALGORITHM

• Genetic algorithm is a search algorithm that mimics operation of evolution where we have an initial population of solutions and we find the best solutions through the process of breeding.

GAMES USING A * SEARCH ALGORITHM

• A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.We want to reach the target cell (if possible) from the starting cell as quickly as possible

GAMES USING REINFORCEMENT ALGORITHM

• Reinforcement learning is a learning paradigm concerned with learning to control a system. Its aim is to maximize performance measure that expresses a long-term objective.

Proposed System- Rule Based System

- In computer science, rule-based systems are used as a way to store and manipulate knowledge to interpret information in a useful way.

- Normally, the term 'rule-based system' is applied to systems involving human-crafted or curated rule sets.

- Rule-based systems constructed using automatic rule inference, such as rule-based machine learning, are normally excluded from this system type .

- Rule-based systems can be used to interpret computer programs.

Input Data

- For our game we have 2 types of input using rule based system;

1)Movement

a)Mouse Cursor
We use the mouse to move the user left or right.

b)Keyboard Cursors
We use the keys of the keyboard to move left or right

2)Firing Bullets

a)Space-bar
The space-bar key is used to fire bullets at the enemies.

Algorithm

We have three algorithms in aspect to the movement and path of bullets.

ALGORITHM-1(Urule)
It uses the sequential covering approach to extract rules from the data. The algorithm extracts the rules one class at a time for a data set.

```
begin
1:RuleSet$= \emptyset$; $//initialsetofruleslearnedisempty$
$2 : forEachClassci \epsilon C; do$
$3 : newRuleSet = uLearnOneRule(D, ci);$
$4 : RemovetuplescoveredbynewRuleSetfromDataSetD;$
$5 : RuleSet+ = newRuleSet;$
```

```
6:end for;
7:return RuleSet;
end
```

ALGORITHM - uLearnOneRule(Dataset D,Class ci)

The uLearnOneRule() procedure, shown here, is the key function of the uRule algorithm. It generates the best rule for the current class

```
begin
1: stop = false;
2: RuleSet = $\emptyset$;
3 : $repeat$
4 : $SplitDintogrowDataandpruneData$;
5 : $Rule = uGrow(growData)$;
6 : $PruneRulesbasedonpruneData$;
7 : $AddRulestoRuleSet$;
```

```
8: Remove data covered by Rule from D;
9: until Stop Condition is true
10: return(RuleSet);
end
```

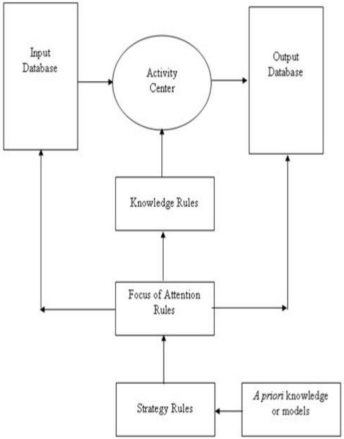
ALGORITHM - uGrow(Instances growData)

The algorithm selects the attribute and split point which has the highest probabilistic information gain and add them as an antecedent of the rule.

```
begin
1: coverData = $\emptyset$;
2 : $while(growData.size() \geq 0)^\wedge(numUnusedAttributes \geq 0)do$
3 : $FindtheattributeAiandthesplitpointsp,$
$whichhasthehighestprobabilisticinformationgame.$
```

```
4: Antecedent += RuleAntecedent(Ai, sp);
5: for (each instance Ij) do
6: if (covers(Ij)) then
7: coverData += inst;
8: end if;
10: end for;
9: growData -= coverData;
10: end while;
end
```

Flowchart

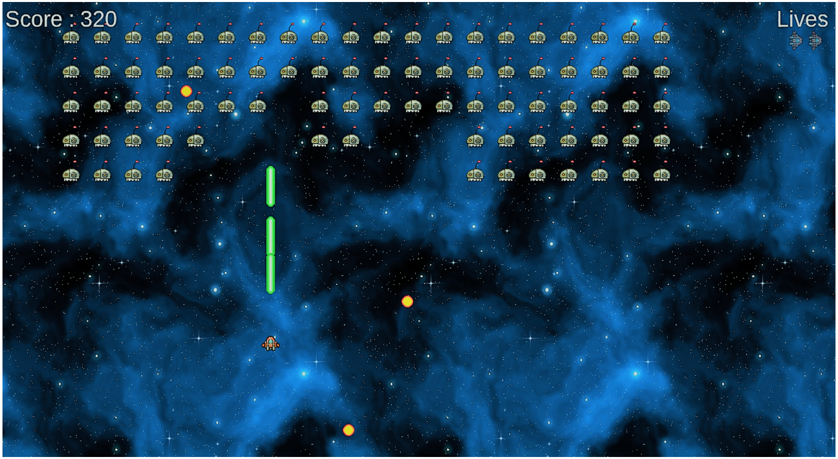

Input Database
Activity Center
Output Database
Knowledge Rules
Focus of Attention Rules
Strategy Rules
A priori knowledge or models

Testing

We have used LINUX for the below phases of testing.

- For Phase 1 of our game we had a fixed number of enemies.
- There was no movements for them as well.
- The bullets shot by them was slow.
- All this made the game a little easy.

PHASE 1



PHASE 2

- For Phase 2 of our game we have different types of enemies.

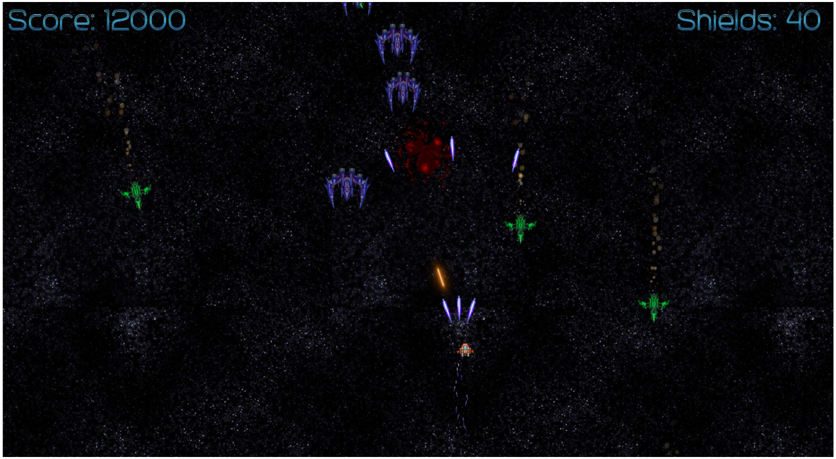
- The movements of the enemies is based on the cursor movements for the user.

- The bullets shot by them is fast paced.

- Removed the fixed number of lives for user and introduced a shield percentage

- The game is a little harder than before.

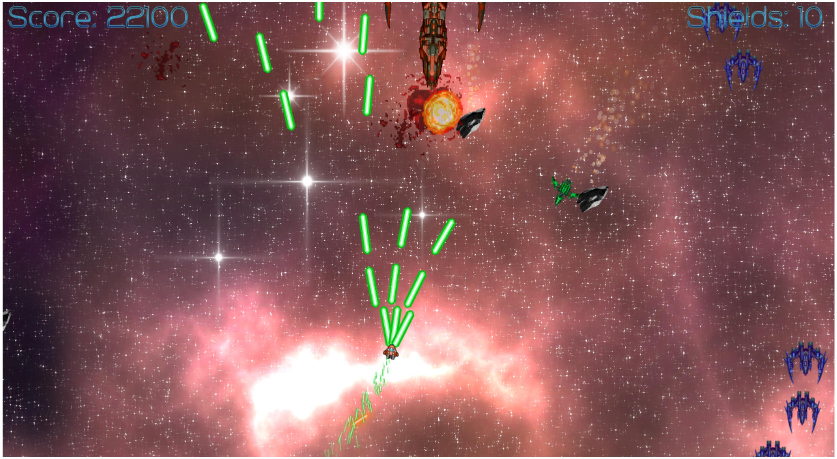
PHASE 2



PHASE 3 - OUTPUT

- For Phase 3 of our game we have introduced a boss enemy.
- The movements of the boss is also based on the cursor movements for the user.
- The shield strength reduces more with bullets by the boss.
- The game is a much more harder than before.

PHASE 3



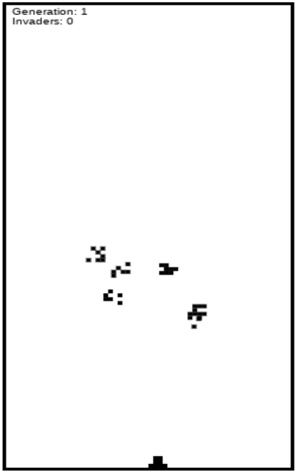
Future Work

So things we can do from this point in the future;

- Make more modifications in the game-play aspect.
- Implement AI and make game-play more complex.
- Can create new products in this domain.

Future Work

An existing game similar to our concept using genetic algorithm



Conclusion

- So as we have seen, Rule based system or knowledge based system are specialized to encapsulate "Human Intelligence" like knowledge and hereby make intelligent decisions quickly and in repeatable form.

- It attempts to derive execution set from an initial set of rules.

- Hence by following this method we get our assets to make their movements

References

•https://www.wired.com/2017/05/want-glimpse-power-ai-play-games

•https://en.wikipedia.org/wiki/Rule-based$_s$*ystem*

•www.wikipedia.org/genetic-learning

•https://www.eclipse.org/articles/Article-Rule200Modeling20With20EMF/article.html

•https://www.geeksforgeeks.org/a-search-algorithm

•https://www.webopedia.com/TERM/R/rule$_{b}ased_{s}ystem.html$

•http://shodhganga.inflibnet.ac.in/bitstream/10603/5651/9/09$_{c}hapter2$

•https://www.slideshare.net/sureshsambandam/rule-based-system-presentation

•https://sites.ualberta.ca/ szepesva/papers/RLAlgsInMDPs.pdf

•https://www.quora.com/

•http://www.mnemstudio.org/path-finding-q-learning.htm

•http://perfectlogic.com/articles/AI/ExpertSystems/ExpertSystems.html