

بسم الله الرحمن الرحيم

هذه الوثيقة التقنية للعبة حرب القواعد . وسيتم ذكر الجانب التقني منها ، حيث سيتم عرض بعض التفاصيل الدقيقة التي تهم المبرمجين والمصممين ، و هذه الوثيقة تحتوي على ( ( 8 صفحات ) ) ، وهي منقحة وقابلة للطرح بين أيدي متبني المشاريع ، ومن يحاولون السير على طريق صحيح لطرح فكرة لعبة معينة .

### المقدمة :

في هذه الوثيقة سيتم ذكر تفاصيل تقنية ، كنوع الأجهزة التي ستطرح اللعبة عليها ، وبعض الخوارزميات والرسوم التخطيطية للكود ، وتفاصيل على كيفية تصميم المجسمات ثلاثية البعد وغيرها ، ففي الوثيقة السابقة ( الوثيقة الوظيفية ) تم ذكر ماهي العناصر التي سنقوم بعملها ، أما في هذه الوثيقة سيتم ذكر كيفية العمل .

### - ميكانيكية اللعبة :

سنقوم بطرح بعض الشروح التفصيلية عن ميكانيكية اللعبة التي تتكون من :

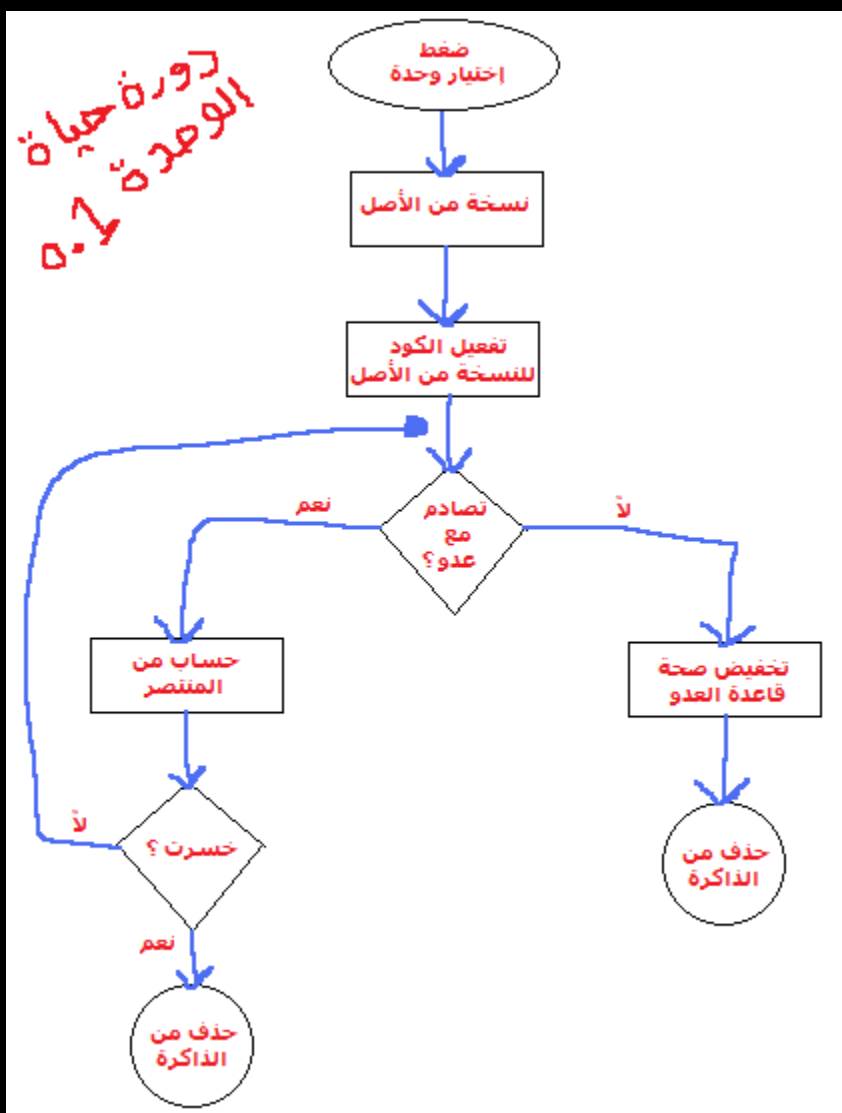
المنصة ونظام التشغيل : سيتم المشروع على جهاز الكمبيوتر الشخصي PC Personal Computer ، بحيث تكون المواصفات الدنيا لتشغيل اللعبة ( كلما زادت المواصفات كان ذلك أفضل ) :

Pentium III or later, Windows 98, 2000 or XP

512 Mb RAM, 64 Mb video card

وسيتم إستخراج نسخة Standalone Application والتي تنتهي بلاحقة Exe مختصرة من Executable Files وهي الملفات التنفيذية ، حيث بعد الإنتهاء من الكود سيتم إرفاق ال Distributed Files الخاصة بمحرك الألعاب وتنفيذ عملية ال Packaging بحيث تخرج اللعبة على صيغة ملفات تنصيب Install Files يتم تنصيبها على جهاز اللاعب . وستكون النسخة الأخيرة من اللعبة متوفرة على موقع الإنترنت للتحميل المباشر .

خوارزميات الكود : حيث أن هناك العديد من الأكواد المستخدمة في اللعبة ، فسيتم وضعها هنا في مراحلها الأولية ويتم تفصيلها إلى مرحلتين أو ثلاثة مراحل إن إقتضت الحاجة .



كود إرسال وحدة وحساب دورة حياتها :

يمكن تلخيص الأمور الأساسية بالخطوات التالية :

- ضغط اللاعب على رمز الوحدة في أعلى الشاشة .

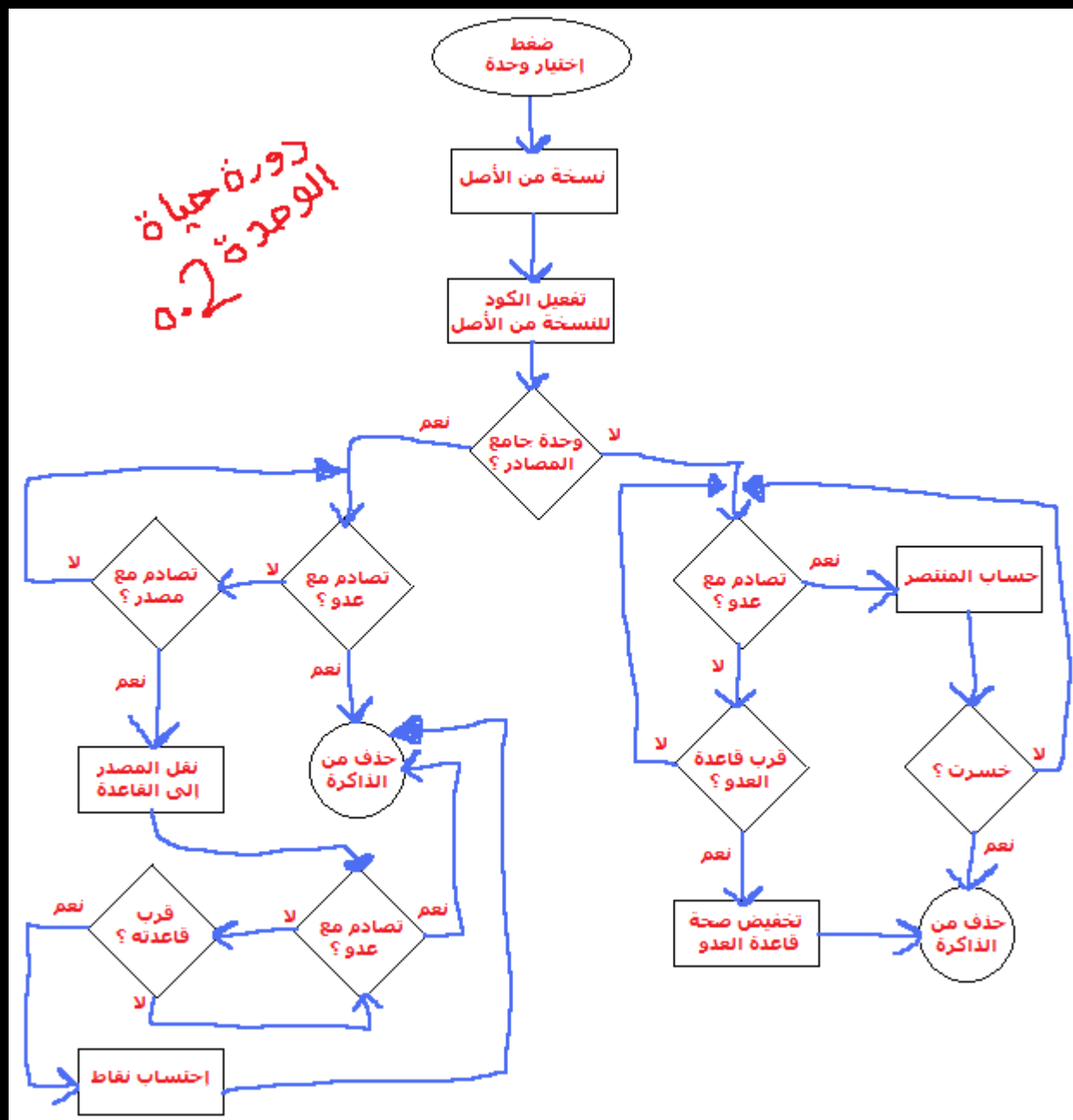
- عمل نسخة من الأصل ( مخزنة مسبقاً ) كمجسم والكود الخاص به .

- تفعيل الكود وبدأ حركته .

- التحقق من حدوث تصادم أو لا

- في حال حصول تصادم مع وحدة العدو سيتم التحقق من الوحدة الناجية ، فإن نجت فستعيد الكرة مع وحدة العدو أخرى .

- في حال عدم وجود تصادم سيتم التحقق من وصول الوحدة إلى قاعدة العدو بسلام ومن ثم تخفيض صحتها وإنهاء الوحدة وحذفها من ذاكرة الكمبيوتر .



ممکن تلخیص الإضافات في :

- تم فصل الوحدات والتأكد من أن الوحدة هي وحدة مقاتلة أم جامعة للمصدر .
  - التحقق من أن وحدة جمع المصدر قامت بالحصول على المصدر ومن ثم إرجاعه إلى القاعدة الأساسية .
  - التحقق من أن الوحدة المقاتلة قامت بالوصول الناجح لقاعدة العدو .
- من الشكل السابق يمكننا التقسيم أكثر والحصول على النسخة الثالثة من الخوارزمية ، ولكننا سنفصلها أكثر في الكود اللاحق .

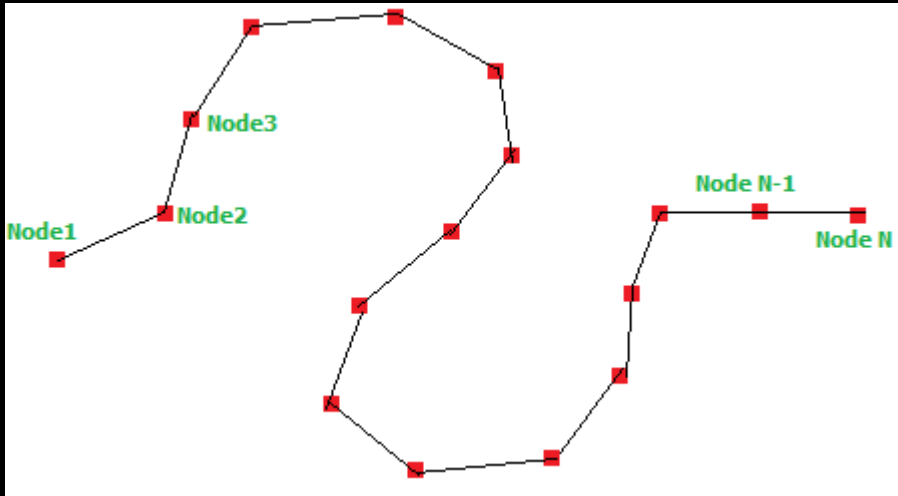
الكود التجريدي pseudo code : بناءً على الخوارزميات الأولية السابقة ، فإنه يمكن الإستنتاج والإستنباط منها العمليات الأساسية التي ستحدث خلال اللعبة ، وسيتم عرضها بطريقة الكود التجريدي كما التالي :

1 - كود حياة الوحدات :

Start Code :

```
//General Variables
Define ArrayOfUnits as array of objects [ ID , Mesh , Health , AttackPoint] // to store my units
Define ArrayOfUnitsEnemy as array of objects [ID , Mesh , Health , AttackPoint ] // to store the enemy units
{
  if ( click on UnitLabel ) Then
  {
    Define NewUnit = new UnitClicked(); // its define as 3d mesh
    Add To ArrayOfUnits = [ ID=ArrayOfUnits.Length +1, Mesh=MeshType , Health=15 , AttackPoint =6] ; //example
    NewUnit.Activate(); // start his code , movement
    if ( NewUnit.Type == ResourceCollector ) Then
    {
      LABEL STARTCOL
      While (ArrayOfUnits(NewUnit).Health > 0 ) Do
      {
        For ( int i = 0 ; i < ArrayOfUnitsEnemy.Length ; i++)
        {
          if ( Get Distance ( NewUnit.Pos, ArrayOfUnitsEnemy[i].Pos) < 5 ) Then // that mean will Collision by enemy
          {
            Remove NewUnit from ArrayOfUnits ; // because its restrained by current NewUnit
            while ( NewUnit.Health > 0 AND ArrayOfUnitsEnemy[i].Health > 0 ) DO
            {
              if ( CanAttackThisType ) Then NewUnit.Health = NewUnit.Health - ArrayOfUnitsEnemy[i].AttackPoint ;
              if ( CanAttackThisType ) Then ArrayOfUnitsEnemy[i].Health = NewUnit.AttackPoint ;
            }
            If ( NewUnit.Health > 0 ) Then // the unit is win
            {
              Add To ArrayOfUnits ; // to start new life with collision detection
              Remove ArrayOfUnitsEnemy[i] from ArrayOfUnitsEnemy ; // end enemy unit life
              Go To label STARTCOL ;
            }
            else // That mean the unit is lost .
            {
              Remove To ArrayOfUnits ; // end unit life
              Add ArrayOfUnitsEnemy[i] from ArrayOfUnitsEnemy ; // to start new life with collision detection
              End Code ;
            }
          }
        }
      }
    }
  }
}
```

## 2 - كود حركة الوحدات :



```

Start Code :
//General Variables
Define CurrentNode , NextNode
Send Collector;
For ( int i = 0 ; i < NumberOfNodes ; i( ++
    CurrentNode = i;
    NextNode = i+1;
    Go to NextNode Position;
}      If Find Resource then Break the For loop and Go to next Loop{
Next i

//When find the Resource will start this loop
For ( int n = CurrentNode ; n>0 ; n( --
    CurrentNode = n;
    NextNode = n-1;
    Go to NextNode Position;
}      If Collision by his Fort .. Then Calculate point{
Next n

```

## - واجهات اللعبة :

كما ذكرنا في الوثيقة الوظيفية ، فإننا سنقوم ببناء واجهات بخلفية ثلاثية الأبعاد ومتحركة ، ويمكن إعطاء صورة كمثال عن السابق :



حيث سيتم وضع موديلات ثلاثية البعد وبسيطة تدل على طبيعة الجزء وطبيعة اللعبة ، مثل شخص يقوم بحمل خاروف .



أما بالنسبة للصورة داخل اللعبة فستكون شبيهة بالصورة التالية :

بحيث أن اللعبة في وقت التنفيذ ويمكن للاعب رؤية العناصر والوحدات القتالية في الشريط العلوي .

**- التصاميم والرسومات و السكيتشات ومحركات الألعاب :**

التصاميم ثنائية البعد : سيتم استخدام برنامج الفوتوشوب Photoshop لرسم السكيتشات الأولية Concept Art ومن ثم الانتقال إلى الصور المرجعية لبناء الموديلات ثلاثية البعد Blueprints وبعدها سيتم الانتقال إلى الواجهات الأساسية في اللعبة ، وبناء الأيقونات للعبة مثلاً علامة يوجد صوت ، وعلامة أن الصوت مكتوم وتكون بإضافة حرف X على الصوت .

التصاميم ثلاثية البعد : سيتم استخدام برنامج الثري دي ماكس 3D Studio Max لعمل التصاميم الثلاثية الأبعاد بناءً على الصور المرجعية Blueprints التي تم رسمها لكل وحدة .

تم ذكر الستايل العام للتصاميم في الوثيقة الوظيفية ، وسنذكر هنا أهم الخطوات التي يجب على مصمم الثري دي مراعاتها :

- أخذ الصور المرجعية Blueprints من مصمم الفوتوشوب .

- بدء العمل على البرنامج مع مراعاة مبدأ ال Low Poly في تصميم المجسمات .

- بعد الإنتهاء من الموديل الأساسي يقوم بإضافة ألوان للمجسم على شكل متيريال ، ويمنع استخدام أي صورة Texture من الصور المعروفة إلا للضرورة ويتم استخدام تقنية Sub Surface Scattering SSS الموجودة في مقبس الفيراى للحصول على اللون المطلوب .

- إجراء عملية إستخراج الصورة النهائية للموديل ( Complete Map ) Backed Material ، وتطبيقها على الجسم .

- التأكد من أسماء المجسمات بحيث تبقى على صلة بإسم وطبيعة المرحلة مثل : Level2\_Ground .

- بدء الإستخراج والتصدير لمحرك الألعاب مع التأكد من القيام بخطوات التصدير المرفقة مع المحرك .

محركات الألعاب : كوننا سنقوم بعمل نسختين مختلفتين من اللعبة الأصلية ، فسيتم استخدام محركين مختلفين لتحديد الأفضل من ناحية الجودة Performance ، وهما :

- حرب الخرفان : وسيتم استخدام محرك الفيرتولز Virtools ، ويتميز بسهولة الإستخدام وتطبيق مبدء الدوال الجاهزة Building Blocks BB ، التي تؤدي الوظائف بطريقة مختصرة .

- حرب التكنولوجيا : وسيتم استخدام محرك XNA المعروف بإستخدامه لغة السي شارب C# والمعروف أيضاً بكفاءته وسرعته من حيث ال Performance .

سيكون هناك بعض الوظائف التي نفترض وجودها في المحركين السابقين وهي : رسم المجسمات ثلاثية البعد ، وجعلها قابلة للحركة ، سهولة التنقل بين الواجهات المختلفة ، حسابات التصادم Collision Detection ، تشغيل الأصوات والموسيقى وقبول ال 3D Sound ، إستخدام مؤثرات على الأجسام Shader Material ، وإستخدام مؤثرات على الشاشة Post Processing Image ، تحويل اللعبة في صيغتها النهائية إلى ملف تنفيذي Standalone Application قابل للعمل على بيئة الويندوز Executable File EXE .

