Uitwerkingen aanvullende toets Gameprogrammeren (INFOB1GP)
Vrijdag 3 januari 2014, 8.30 - 10.30 uur

---

1. Deze opgave bestaat uit een aantal vragen. Houd het antwoord kort: één of twee zinnen per onderdeel kan al genoeg zijn.

   (a) *(2 punten)* Hoe kun je in een klasse een member maken die vanuit een andere klasse wel kan worden bekeken, maar niet worden veranderd?

   Antwoord: Declareer een private variabele, en een public property met alleen een get-deel, die de waarde van de private variabele oplevert.

   (b) *(2 punten)* Wat is de betekenis van de operator %? Geef een expressie die voor positieve getallen x en y dezelfde waarde heeft als x%y, zonder daarbij de operator % te gebruiken.

   Antwoord: De operator % geeft de rest bij deling. Een equivalente expressie voor x%y is x−(x/y)∗y.

   (c) *(2 punten)* **Antwoord:**

```
// aanpak 1: vooraf controleren
if (waarde<0)
    s = "onmogelijk";
else
    s = "De wortel is " + Math.Sqrt(waarde);

// aanpak 2: foutsituatie afvangen
try
{
    s = "De wortel is " + Math.Sqrt(waarde);
}
catch (Exception e)
{
    s = "onmogelijk";
}
```

   (d) *(4 punten)* Kruis aan welke van de volgende stellingen waar zijn:

   - ☐ We noemen een klasse abstract indien minstens een van de methoden of properties in die klasse een lege body heeft.
   - X **base** mag niet worden gebruikt in een statische methode.
   - X Bij arrays moet bij de initialisatie aangegeven worden uit hoeveel elementen de array bestaat; dit hoeft niet met objecten die als type een Collection (sub)klasse hebben.
   - X Indien de game loop in variable timestep modus in plaats van in fixed timestep modus wordt uitgevoerd, dan zijn game time en real time hetzelfde.

   (e) *(2 punten)* Wat is polymorfisme? Geef een voorbeeld van hoe polymorfisme gebruikt kan worden in een game programma.

   Antwoord: Polymorfisme betekent dat je gelijkwaardige objecten kunt beschrijven met een zelfde interface. In games kun je dit gebruiken voor een inventory. De inventory bestaat uit een lijst van Item objecten met methoden (zoals Use). Specifieke items (zoals een sleutel of een potion) worden gemodelleerd door subklassen van de Item klasse. Als we de Use methode op objecten van het type Item aanroepen, dan wordt door dynamic binding automatisch de juiste versie van die methode aangeroepen.

   (f) *(3 punten)*

   Antwoord:

   - GameObject bevat een abstracte methode maar is zelf geen abstracte klasse;
   - de SpriteGameObject klasse implementeert de abstracte Update methode niet;

- de SpriteGameObject constructormethode roept de constructor van de superklasse niet aan (**base**(id)).

2. (a) *(6 punten)* De methode LoadLevel is verantwoordelijk voor het laden van een level uit een tekstbestand. Dit level wordt opgeslagen in een 2D array van **bool** waarden, waarbij de waarde **true** inhoudt dat de betreffende tile een obstructie is. Normaalgesproken bevat een tekstbestand alleen regels met hetzelfde aantal karakters. Als dat niet het geval is, dan moet de methode LoadLevel een exceptie van het type IOException werpen. Schrijf de body van de loadLevel methode uit.

Antwoord:

```
List<string> lines = new List<string>();
StreamReader fileReader = new StreamReader(path);
string line = fileReader.ReadLine();
int width = line.Length;
while (line != null)
{
    lines.Add(line);
    if (line.Length != width)
        throw new IOException("The length of the lines is different.");
    line = fileReader.ReadLine();
}
this.tiles = new bool[width, lines.Count];
for (int y = 0; y < lines.Count; ++y)
    for (int x = 0; x < width; ++x)
        this.tiles[x, y] = lines[y][x] == 'X';
```

(b) *(4 punten)* In de Level klasse willen we ook een property RandomFreePosition toevoegen. Deze property moet een willekeurige positie in het level teruggeven, uitgedrukt in indices van de tile-array, niet in pixel-coördinaten. De conditie is wel dat die positie geldig is, oftewel: de positie valt binnen de schermbreedte en de positie is niet al bezet door een obstructie. Voor het gebruik van deze property, zie bijvoorbeeld de constructormethode van de Level klasse. Werk de property RandomFreePosition uit.

Antwoord:

```
public Vector2 RandomFreePosition
{
    get
    {
        Vector2 pos;
        do
        {
            pos = new Vector2(random.Next(Width), random.Next(Height));
        } while (!IsAvailable(pos));
        return pos;
    }
}
```

(c) *(5 punten)*

Antwoord:

```
Vector2 newdirection = direction;
if (inputHelper.KeyPressed(Keys.Left))
    newdirection = new Vector2(−1, 0);
else if (inputHelper.KeyPressed(Keys.Right))
    newdirection = new Vector2(1, 0);
else if (inputHelper.KeyPressed(Keys.Up))
    newdirection = new Vector2(0, −1);
else if (inputHelper.KeyPressed(Keys.Down))
    newdirection = new Vector2(0, 1);
if (newdirection.X == −direction.X || newdirection.Y == −direction.Y)
    return;
direction = newdirection;
```

(d) *(7 punten)*

Antwoord:

```
time += gameTime.ElapsedGameTime.TotalSeconds;
if (time > 0.2)
{
    time −= 0.2;
    Vector2 newpos = snakePositions[snakePositions.Count − 1] + direction;

    if (!level.IsAvailable(newpos) || snakePositions.Contains(newpos))
        level.GameOver = true;

    snakePositions.Add(newpos);
    snakePositions.RemoveAt(0);

    if (level.Food == newpos)
    {
        snakePositions.Insert(0,snakePositions[0]);
        while (snakePositions.Contains(level.Food))
            level.Food = level.RandomFreePosition;
    }
}
```

(e) *(3 punten)* In de Draw methode van de Snake klasse moet de slang op het scherm getekend worden (in het blauw). Schrijf de body van de Draw-methode uit.

Antwoord:

```
foreach (Vector2 pos in snakePositions)
{
    Vector2 finalpos = pos * new Vector2(blockSprite.Width, blockSprite.Height);
    spriteBatch.Draw(blockSprite, finalpos, Color.Blue);
}
```